University of Colorado **Boulder**

Department of Computer Science

CSCI 5622: Machine Learning

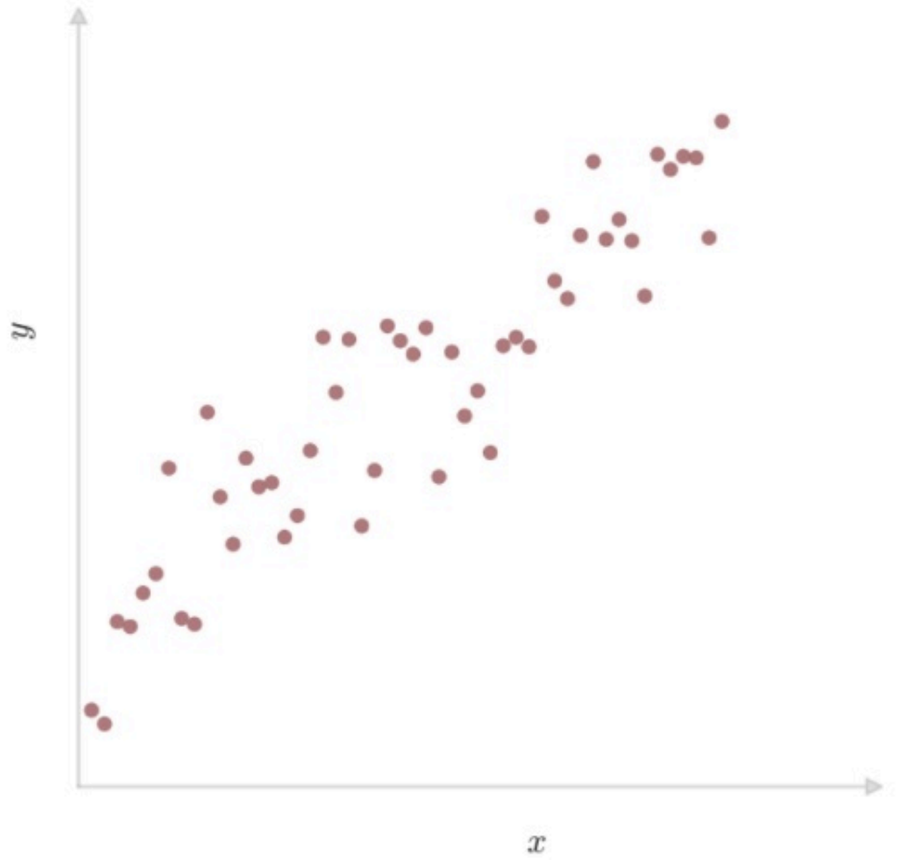
Chris Ketelsen

Lecture 08:
Linear Regression and the Bias-Variance Trade-Off

Linear Regression

Data are continuous inputs and outputs $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^m, \mathbf{x}_i \in \mathbb{R}^D$

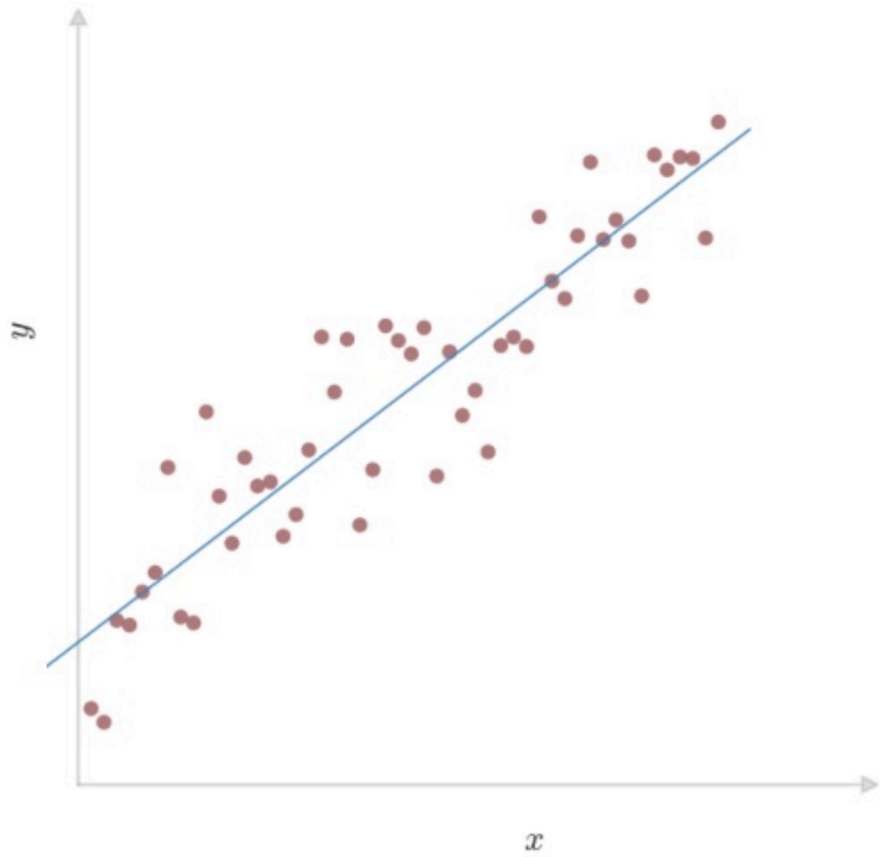
Want to predict y from $\mathbf{x}$ using a linear model: $y = f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$



Linear Regression

Data are continuous inputs and outputs $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^m, \mathbf{x}_i \in \mathbb{R}^D$

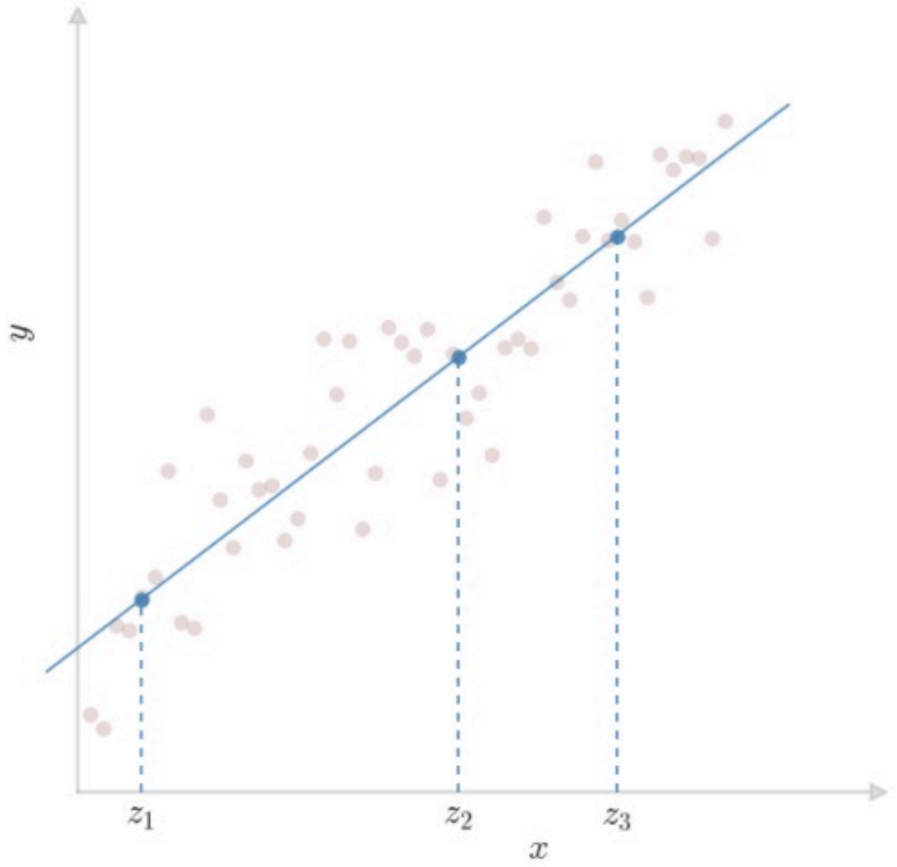
Want to predict y from $\mathbf{x}$ using a linear model: $y = f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$



Linear Regression

Want to predict y from $\mathbf{x}$ using a linear model: $y = f(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$

Prediction is $\hat{y} = f(\mathbf{z}) = \mathbf{w}^T \mathbf{z}$



Linear Regression Examples

- Given a person's age and gender, predict their height
- Given the square footage and number of bathrooms in a house, predict its sale price
- Given unemployment, inflation, number of wars, and economic growth predict the president's approval rating
- Given a user's browsing history, predict how long he will stay on product page
- Given the advertising budget expenditures in various media markets, predict the number of products sold

Linear Regression Assumptions

- Relationship between features and response defined by

$$Y = \mathbf{w}^T \mathbf{X} + \epsilon$$

- Each realization of data satisfies

$$y_i = \mathbf{w}^T \mathbf{x}_i + \epsilon_i$$

- The random noise ϵ follows a distribution $N(0, \sigma^2)$

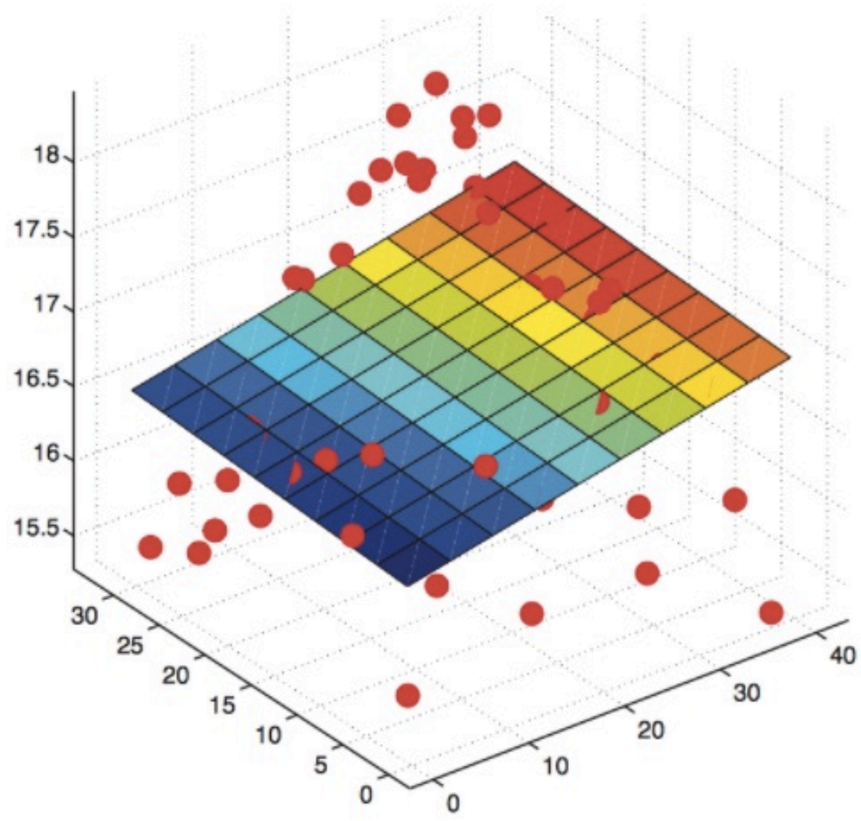
- Each of the ϵ_i's are independent

Multiple Features

Suppose the input data is 2D, so we have $\mathbf{x} = [x_1, x_2]^T$

Multi-linear model is $f(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 = \mathbf{w}^T \mathbf{x}$

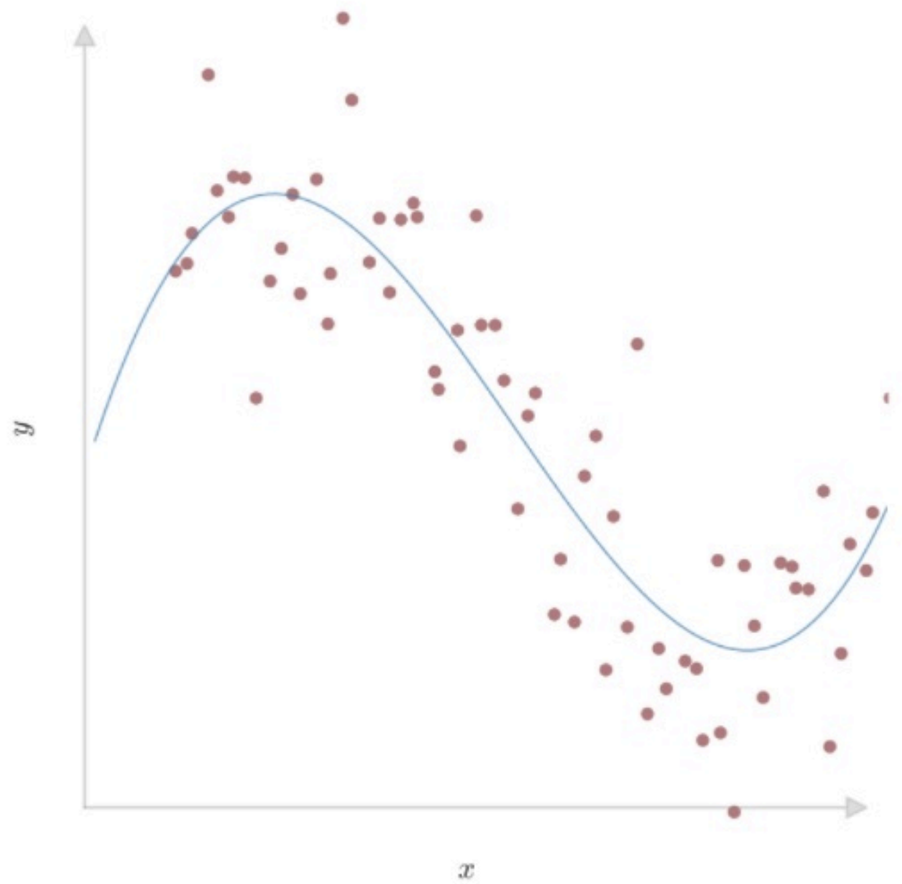
Here we've again prepended vector $\mathbf{x}$ with a 1 in first position



Derived Features

Example: For single feature x, can fit polynomials in x

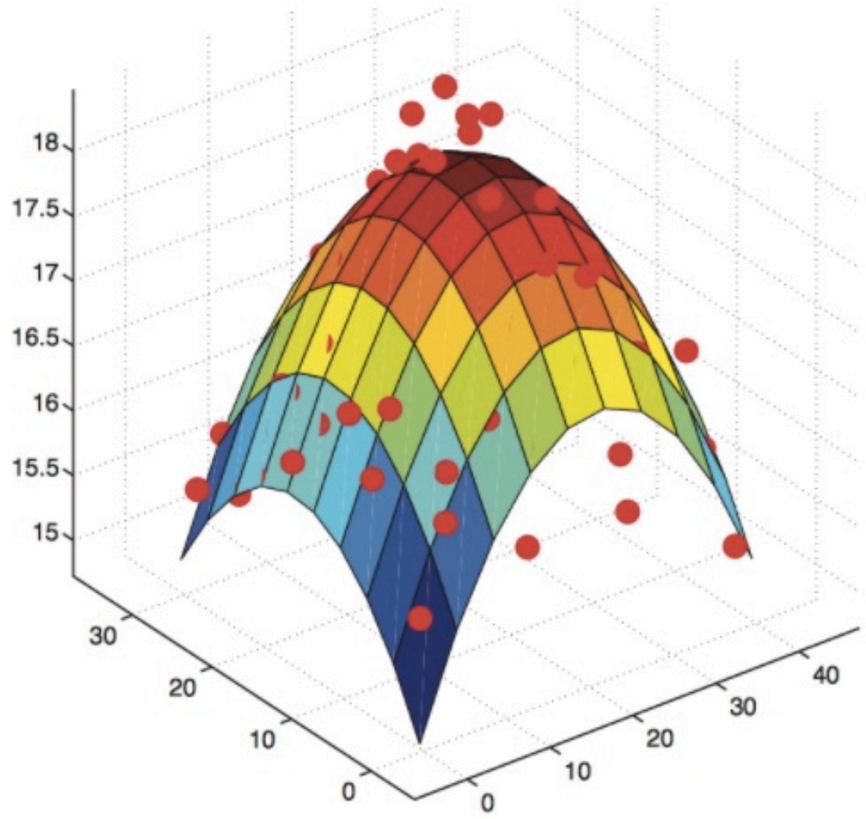
$$f(x) = \mathbf{w}^T \phi(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3$$



Derived Features

Example: For 2 features x_1 , and x_2 fit

$$f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2$$



Estimating the Parameters

We'll start by formulating the problem using linear algebra

It would be great if each example in our training set satisfied the equation $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$. Then we'd have

$$\begin{aligned} \mathbf{w}^T\mathbf{x}_1 &= y_1 \\ \mathbf{w}^T\mathbf{x}_2 &= y_2 \\ &\vdots \\ \mathbf{w}^T\mathbf{x}_m &= y_m \end{aligned}$$

Estimating the Parameters

Swapping the vectors in the dot products gives

$$\begin{aligned} \mathbf{x}_1^T \mathbf{w} &= y_1 \\ \mathbf{x}_2^T \mathbf{w} &= y_2 \\ &\vdots \\ \mathbf{x}_m^T \mathbf{w} &= y_m \end{aligned}$$

which is equivalent to the matrix equation

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

where the i^{th} row in the data matrix $\mathbf{X}$ is the feature vector $\mathbf{x}_i$ corresponding to the i^{th} training example

Estimating the Parameters

It would be nice if this equation had an exact solution, but that would mean that all of the data fit the model exactly

Instead we pick $\mathbf{w}$ to make the equation as *true as possible*

One way to do this is to choose $\mathbf{w}$ so that the **residual vector** $\mathbf{y} - \mathbf{X}\mathbf{w}$ is as small as possible in the ℓ_2-norm

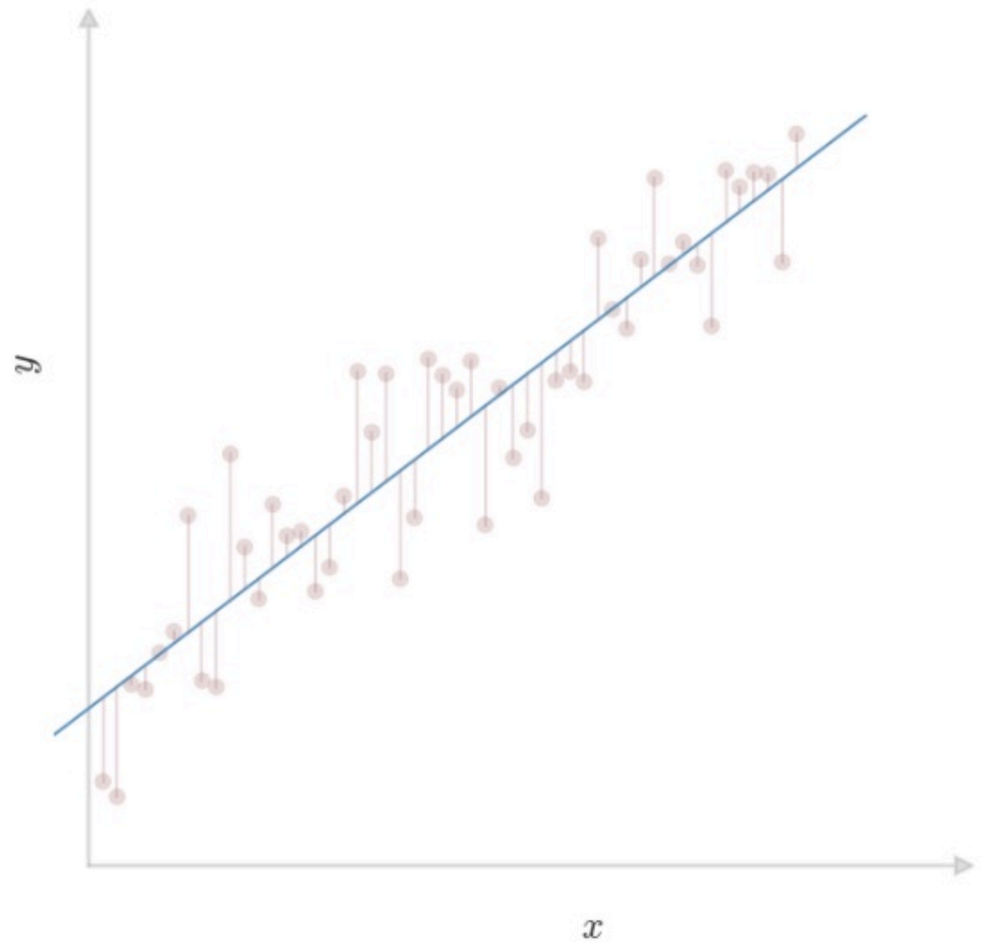
$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2$$

The objective function is called the **residual sum of squares**:

$$\text{RSS}(\mathbf{w}) = \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2 = \sum_{i=1}^{m} (y_i - \mathbf{w}^T \mathbf{x}_i)^2$$

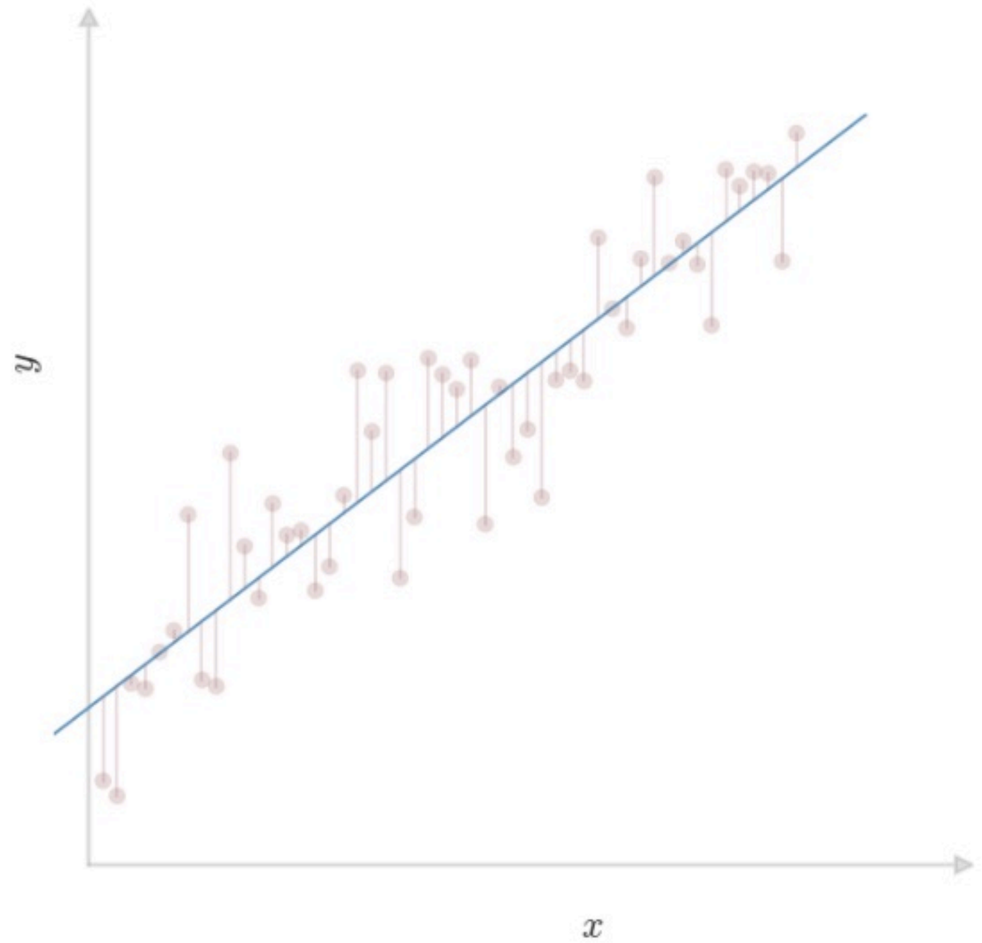
Estimating the Parameters

Notice that each term in $\mathrm{RSS}(\mathbf{w})$ is the square of the difference between the true value of y_i and the model prediction for $\mathbf{x}_i$



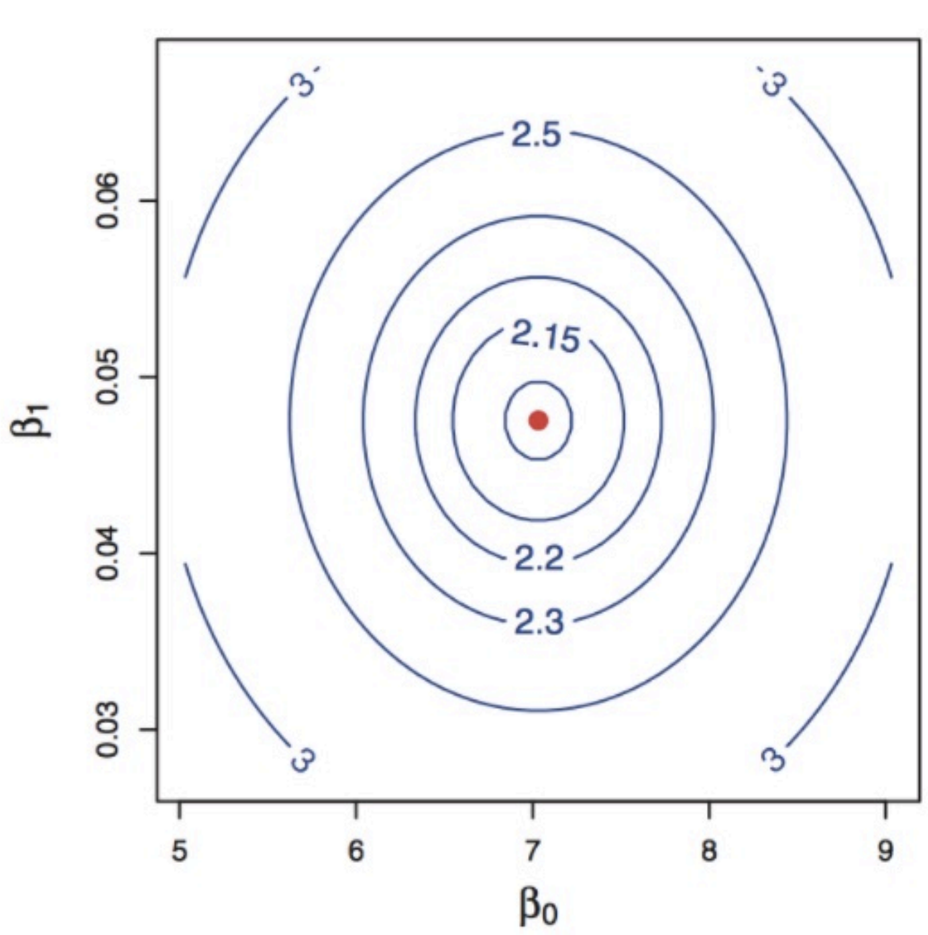
Estimating the Parameters

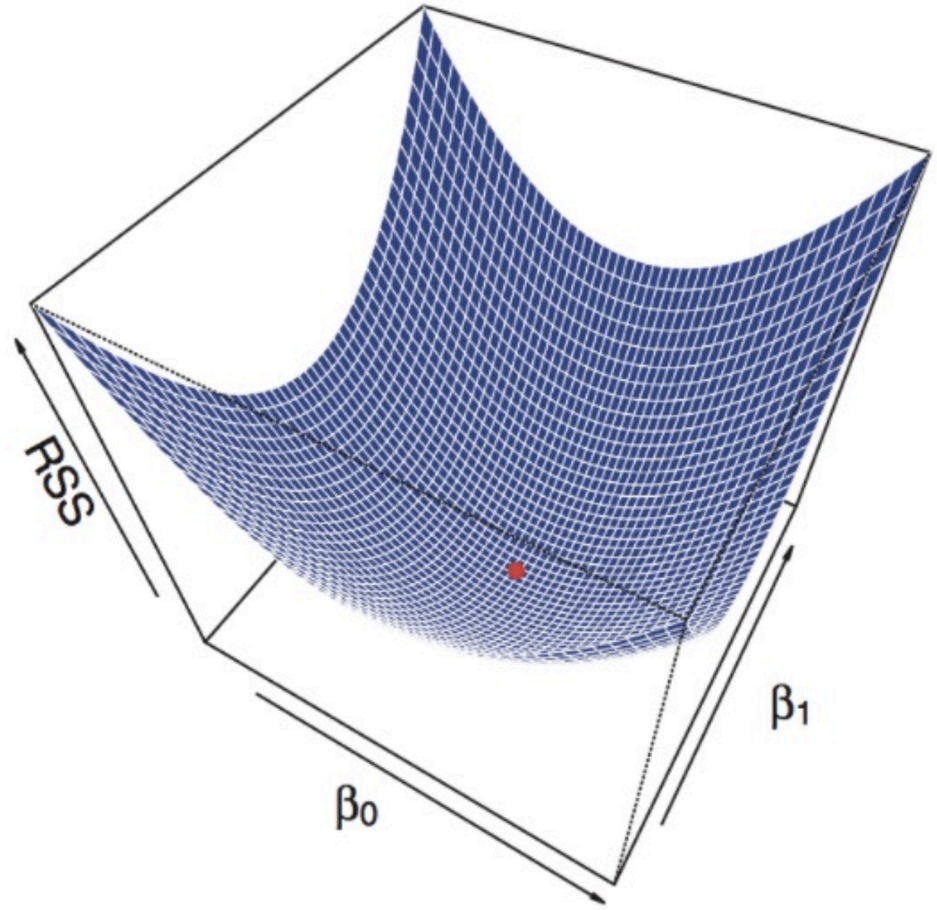
For this reason, the linear regression problem is often referred to as the **least-squares problem**.



Estimating the Parameters

The $RSS(\mathbf{w})$ function is convex, and so has a unique minimum





Estimating the Parameters

The minimizer is the unique point such that $\nabla_{\mathbf{w}} \mathrm{RSS}(\mathbf{w}) = 0$

Taking the gradient of the RSS with respect to $\mathbf{w}$

$$\nabla_{\mathbf{w}} \mathrm{RSS}(\mathbf{w}) = \nabla_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2 = 0$$

gives

$$\nabla_{\mathbf{w}} \mathrm{RSS}(\mathbf{w}) = \mathbf{X}^T\mathbf{X}\mathbf{w} - \mathbf{X}^T\mathbf{y} = \mathbf{0}$$

The optimal parameters $\hat{\mathbf{w}}$ are thus the solution to the linear system

$$\mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{y} \quad \Leftrightarrow \quad \hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

These equations are often referred to as the **Normal Equations**

Estimating the Parameters

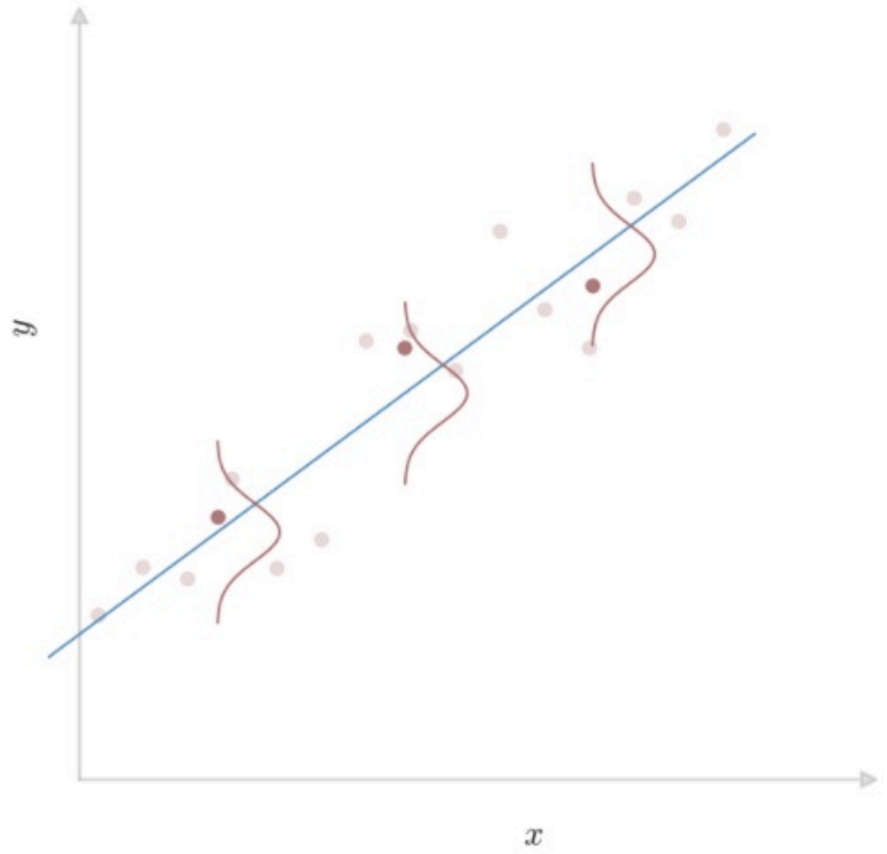
For practical applications you should never estimate the parameters using the Normal Equations:

- Just forming $\mathbf{X}^T\mathbf{X}$ is an $\mathcal{O}(n^3)$ operation
- If $\mathbf{X}$ has nearly dependent columns then $\mathbf{X}^T\mathbf{X}$ will be poorly conditioned and your answer could be garbage
- For data that fits into memory a solution technique based on **QR Factorization** or **Singular Value Decomposition** of $\mathbf{X}$ is better
- For large data do something akin to Gradient Descent of Stochastic Gradient Descent

Probabilistic Interpretation

Regression is a discriminative model that assumes the response is Gaussian with mean $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$

$$p(y \mid \mathbf{x}, \mathbf{w}, \sigma) = N(\, y \mid \mathbf{w}^T \mathbf{x}, \sigma^2) \quad \text{or} \quad N(\, y \mid \mathbf{w}^T \phi(\mathbf{x}), \sigma^2)$$



Probabilistic Interpretation

Regression is a discriminative model that assumes the response is Gaussian with mean $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$

$$p(y \mid \mathbf{x}, \mathbf{w}, \sigma) = N(\, y \mid \mathbf{w}^T \mathbf{x}, \sigma^2) \quad \text{or} \quad N(\, y \mid \mathbf{w}^T \phi(\mathbf{x}), \sigma^2)$$

Assuming **i.i.d.** samples, we can write the likelihood of the data as

$$\begin{aligned} L(\mathbf{w}) &= \prod_{i=1}^{m} p(y_i \mid \mathbf{x}_i, \mathbf{w}) \\ &= \prod_{i=1}^{m} \left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2} (y_i - \mathbf{w}^T \mathbf{x}_i)^2 \right) \right] \end{aligned}$$

Want to find $\mathbf{w}$ that maximizes this likelihood

Probabilistic Interpretation

Regression is a discriminative model that assumes the response is Gaussian with mean $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$

$$p(y \mid \mathbf{x}, \mathbf{w}, \sigma) = \mathcal{N}(y \mid \mathbf{w}^T\mathbf{x}, \sigma^2) \quad \text{or} \quad \mathcal{N}(y \mid \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}), \sigma^2)$$

Or, equivalently, minimizes the negative log-likelihood

$$\begin{aligned} NLL(\mathbf{w}) &= -\ln L(\mathbf{w}) \\ &= -\sum_{i=1}^{m} \ln\left[\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2\sigma^2}(y_i - \mathbf{w}^T\mathbf{x}_i)^2\right)\right] \\ &= \frac{m}{2}\ln(2\pi\sigma^2) + \frac{1}{2\sigma^2}\sum_{i=1}^{m}(y_i - \mathbf{w}^T\mathbf{x}_i)^2 \\ &= \frac{m}{2}\ln(2\pi\sigma^2) + \frac{1}{2\sigma^2}\mathrm{RSS}(\mathbf{w}) \end{aligned}$$

Probabilistic Interpretation

$$NLL(\mathbf{w}) = -\ln L(\mathbf{w}) = \frac{m}{2}\ln(2\pi\sigma^2) + \frac{1}{2\sigma^2}\text{RSS}(\mathbf{w})$$

Assume σ and m are fixed by the data

Minimizing $NLL(\mathbf{w})$ is equivalent to minimizing $RSS(\mathbf{w})$

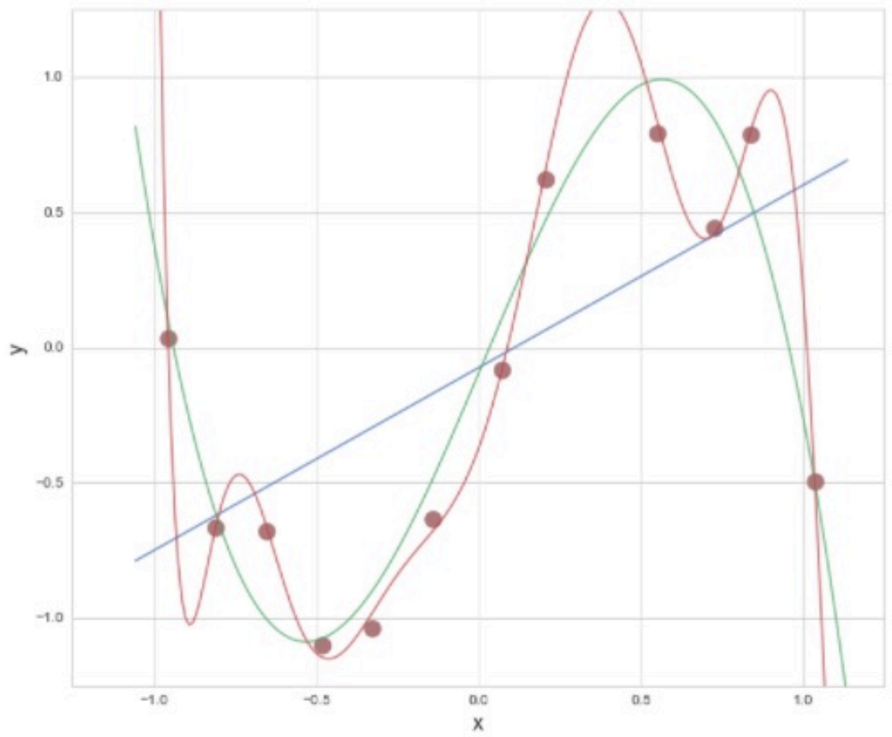
And thus equivalent to the linear algebra view derived previously

Next time we'll show how standard Ridge and Laso Regularization can be viewed as putting different priors on the model parameters

The Bias-Variance Trade-Off

Consider the case of fitting linear regression with derived polynomial features to a set of training data $\{x_i, y_i\}_{i=1}^{m}$

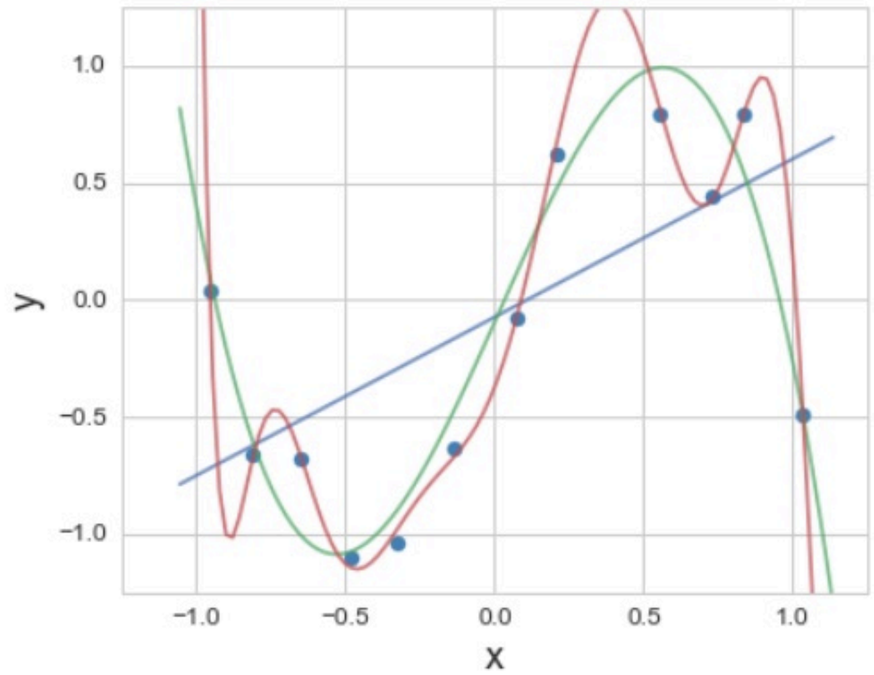
In general, want to simplest model that explains the training data and can still generalize to unseen test data

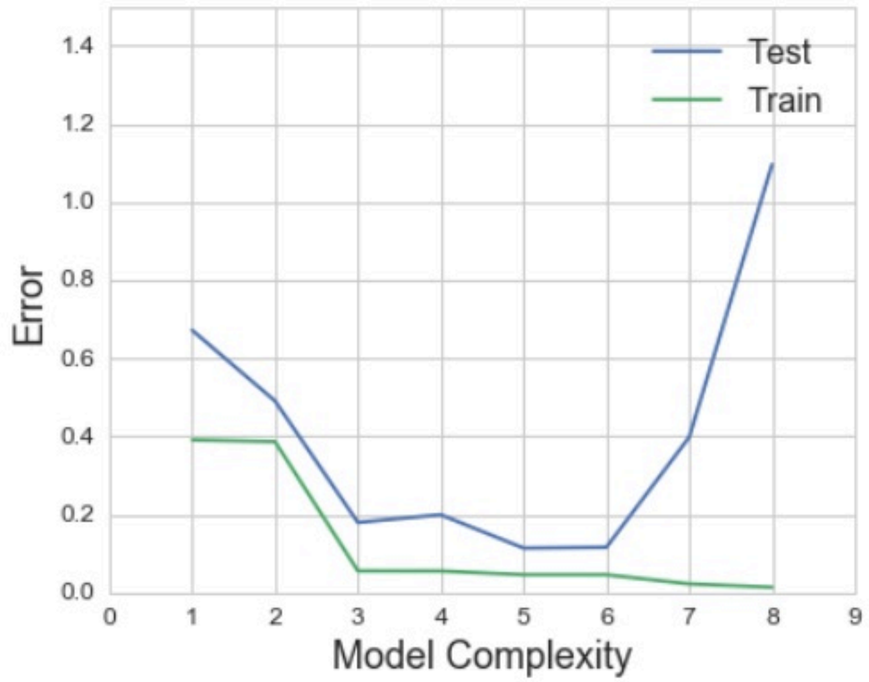


The Bias-Variance Trade-Off

Plot error on training and test sets for increasing polynomial degree

Test error initially decreases, then increases when overfitting occurs





The Bias-Variance Trade-Off

On opposite ends of the spectrum, we see poor test error when the model is too simple (the linear case) and when the model is too complicated (the degree 10 polynomial case)

The linear model is an example of a **high bias** / **low variance** model

The degree 10 polynomial is an example of a **low bias** / **high variance** model

Bias: We say a model has high bias when it can't represent the underlying truth behind the data well (usually b/c it's too simple)

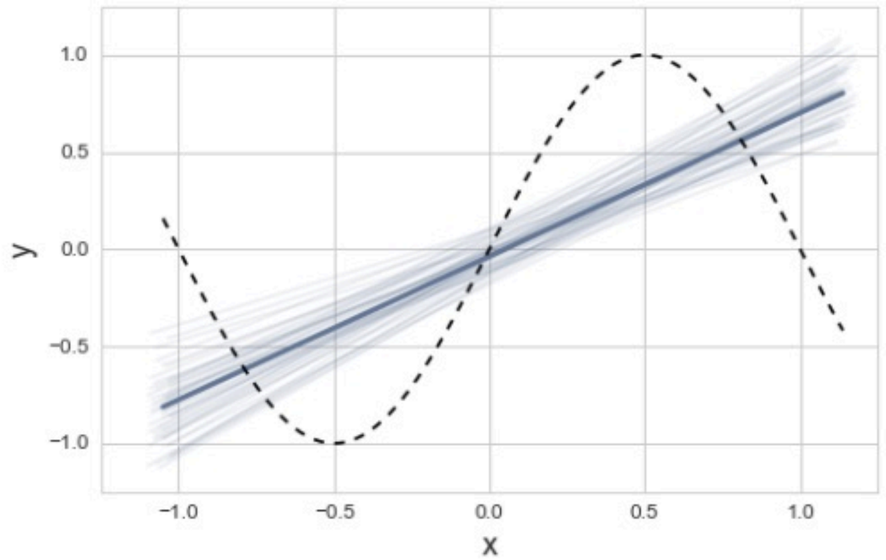
Variance: We say a model has high variance when it is highly sensitive on the training data (usually because it's too complicated)

The Bias-Variance Trade-Off Intuition

Using the same simulated data as before

$$y = f(x) + \text{Noise} = \sin(\pi x) + \text{Noise}$$

Fit model to many training sets. Then take mean of models

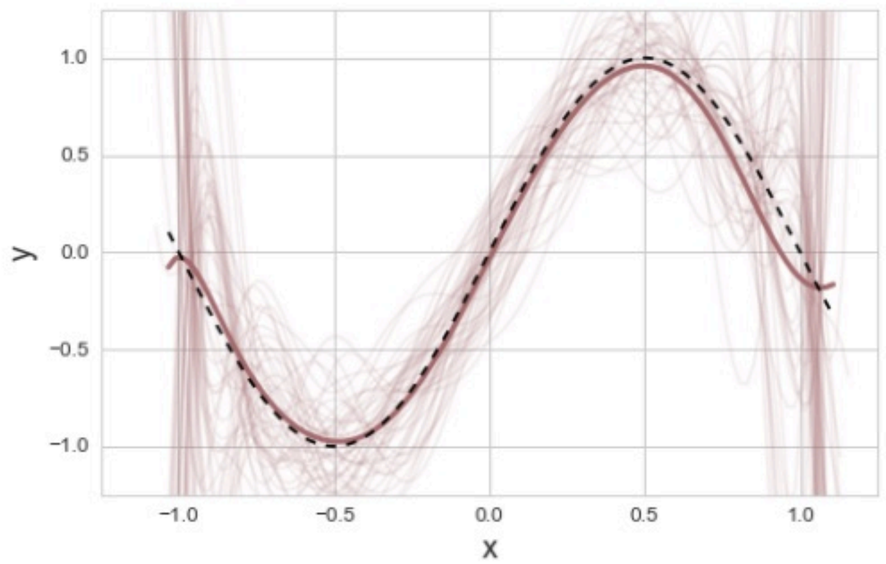


The Bias-Variance Trade-Off Intuition

Using the same simulated data as before

$$y = f(x) + \text{Noise} = \sin(\pi x) + \text{Noise}$$

Fit model to many training sets. Then take mean of models

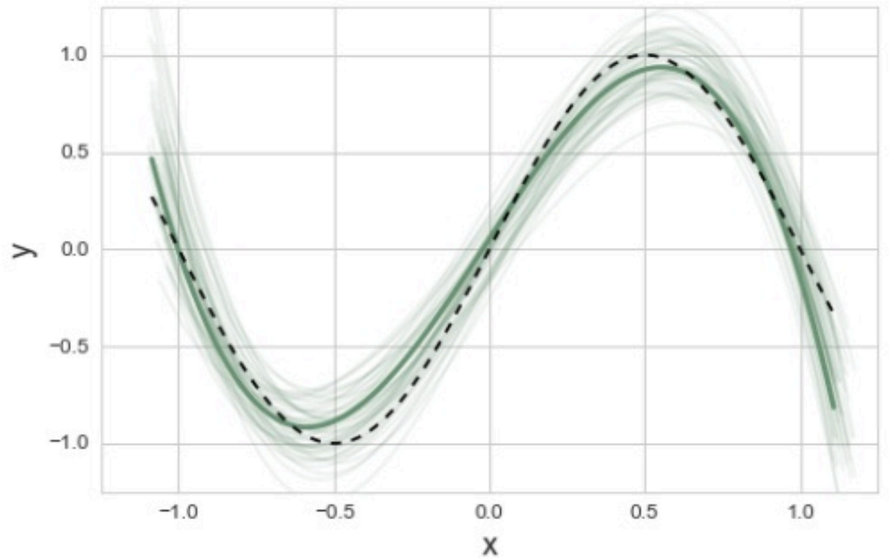


The Bias-Variance Trade-Off Intuition

Using the same simulated data as before

$$y = f(x) + \text{Noise} = \sin(\pi x) + \text{Noise}$$

Fit model to many training sets. Then take mean of models



The Bias-Variance Trade-Off

The Punchline:

The best model is obtained when the model has low enough bias to capture the underlying truth and low enough variance to generalize to unseen data.

The Math Behind the Bias-Variance Trade-Off

Assume that there is a true relationship between response y and data $\mathbf{X} = (X_1, X_2, \ldots, X_D)$ of the form

$$Y = f(\mathbf{X}) + \epsilon$$

Here f is a fixed (true) but unknown function and ϵ is a random error term independent of $\mathbf{X}$ with mean zero

Function f represents systematic information that $\mathbf{X}$ provides about Y

The error ϵ can come from different sources

- Features that influence Y not included in our feature set
- Measurements in data are not perfect

The Math Behind the Bias-Variance Trade-Off

Our goal is to approximate the unknown function f by a known function $\hat{f}$

Then, given some new test data, we can approximate

$$\hat{Y} = \hat{f}(\mathbf{X})$$

Think of $\hat{f}$ as a random variable itself. The actual form depends on training it on some data

Error in prediction can be measured by expected value of squared difference between Y and $\hat{Y}$

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

The Math Behind the Bias-Variance Trade-Off

We have

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

Last step follows from $E[\epsilon]=0$ and independence from $\mathbf{X}$

Decomposes error into *reducible* and *irreducible*

The Math Behind the Bias-Variance Trade-Off

Going further

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

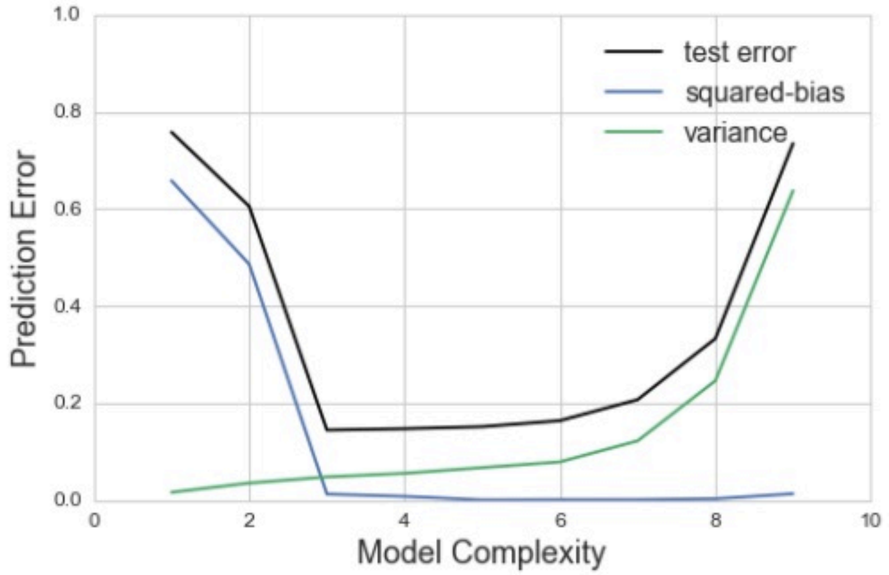
$$= E[\left\{(f(X) - E[\hat{f}(X)]) + (E[\hat{f}(X)] - \hat{f}(X))\right\}^2] + V[\epsilon]$$

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

$$= \text{Bias}^2 + \text{Variance}^2 + V[\epsilon]$$

The Math Behind the Bias-Variance Trade-Off

Here is the bias-variance decomposition for the simulated data



Notice that sum of blue and green curves not quite the black curve

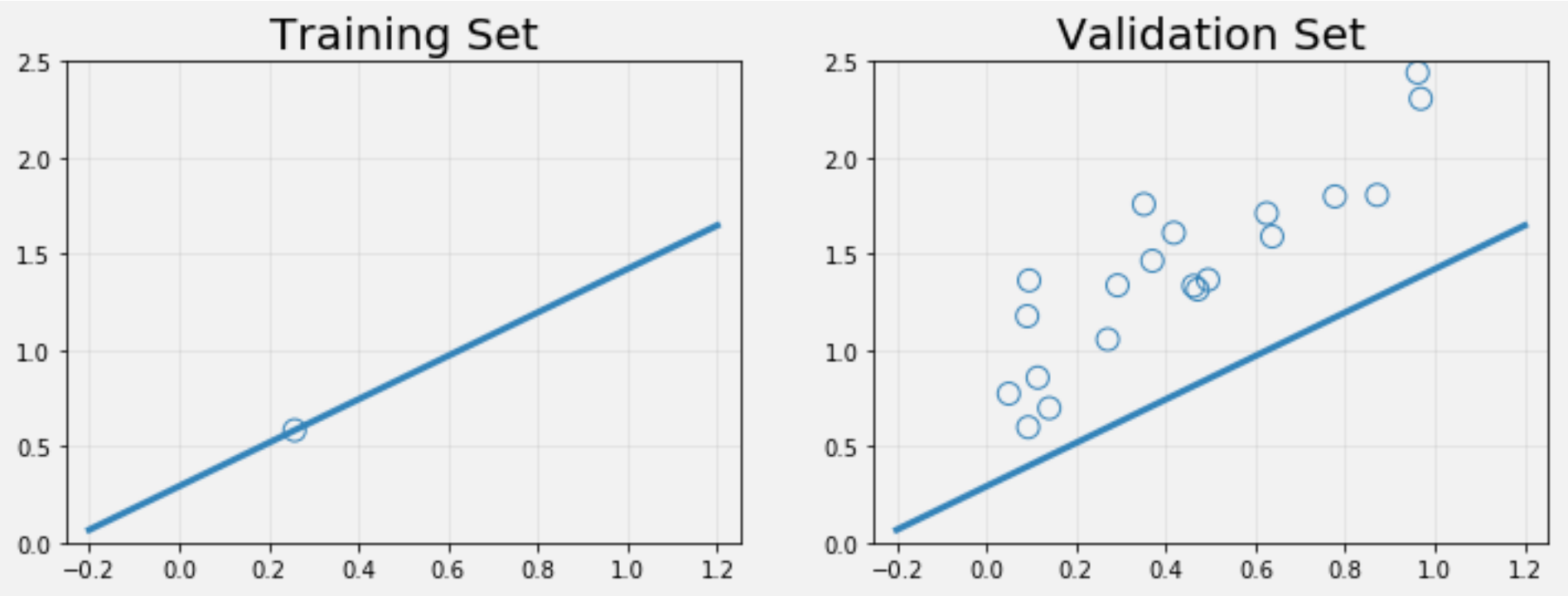
This is the irreducible error $\mathrm{Var}[\epsilon]$

We'll look more in-depth at the Bias-Variance Trade-Off next week when we talk about Learning Theory

Learning Curves and What They Tell Us

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

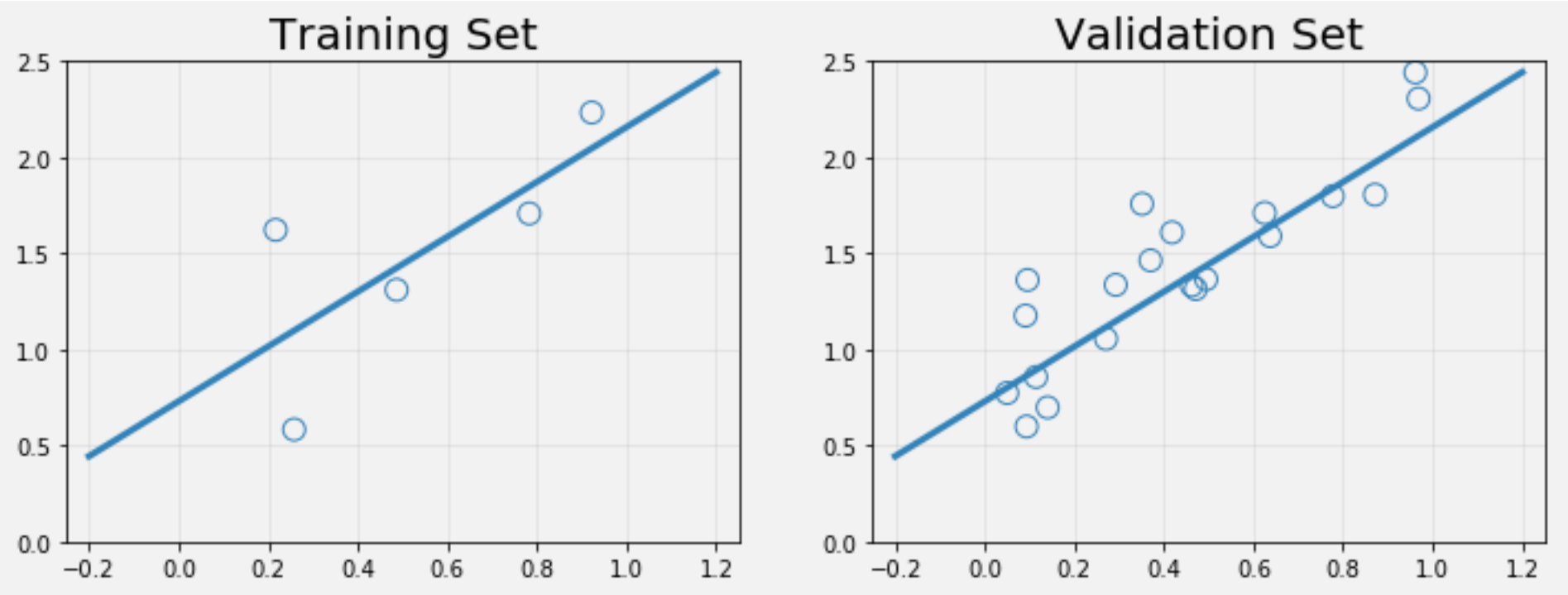
Evaluate your training and test error for increasing training set sizes



Learning Curves and What They Tell Us

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

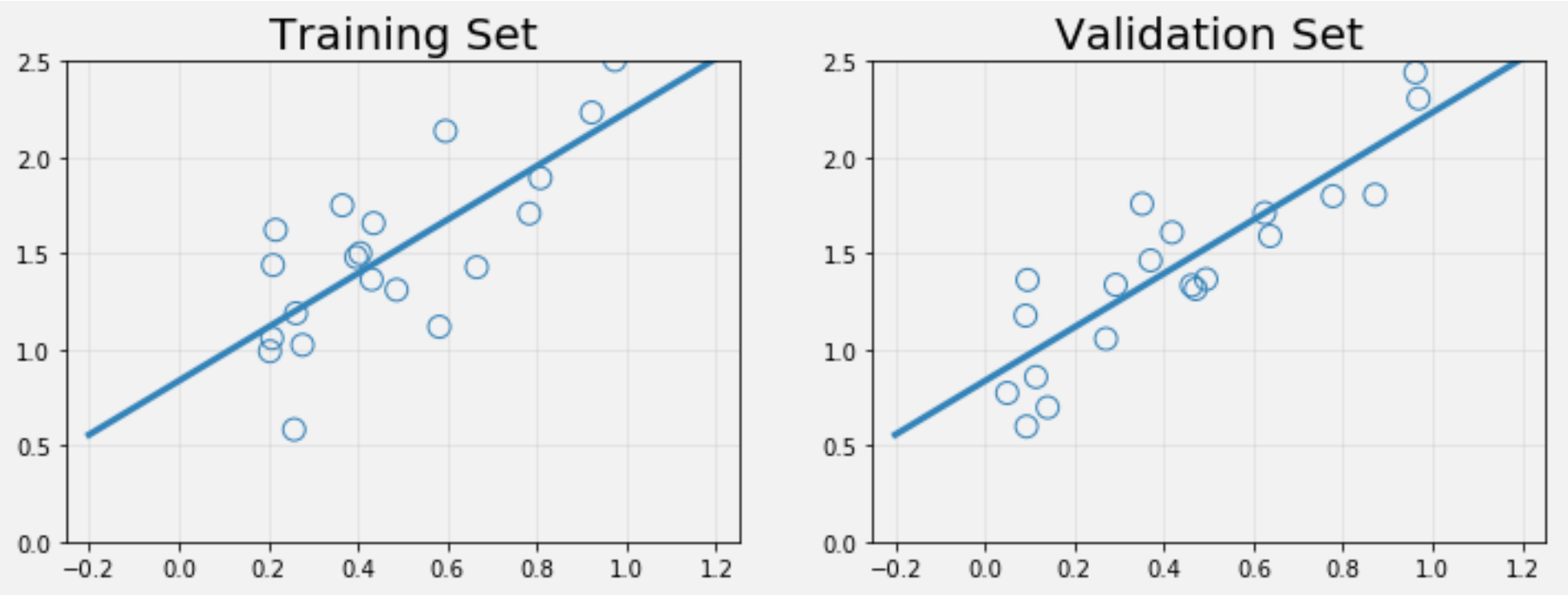
Evaluate your training and test error for increasing training set sizes



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

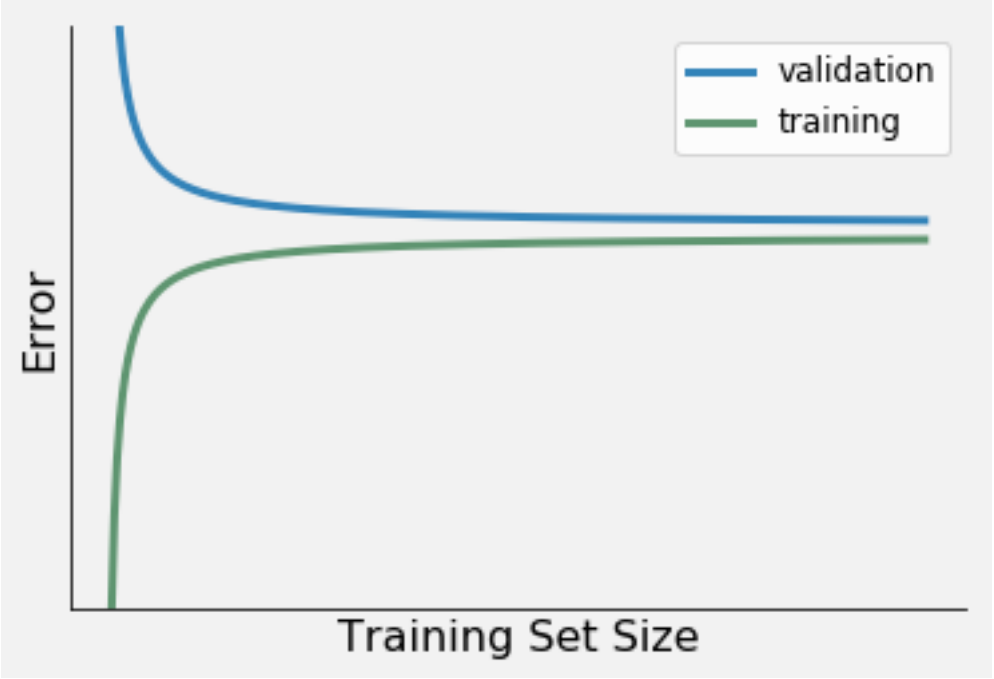


Learning Curves and What They Tell Us

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

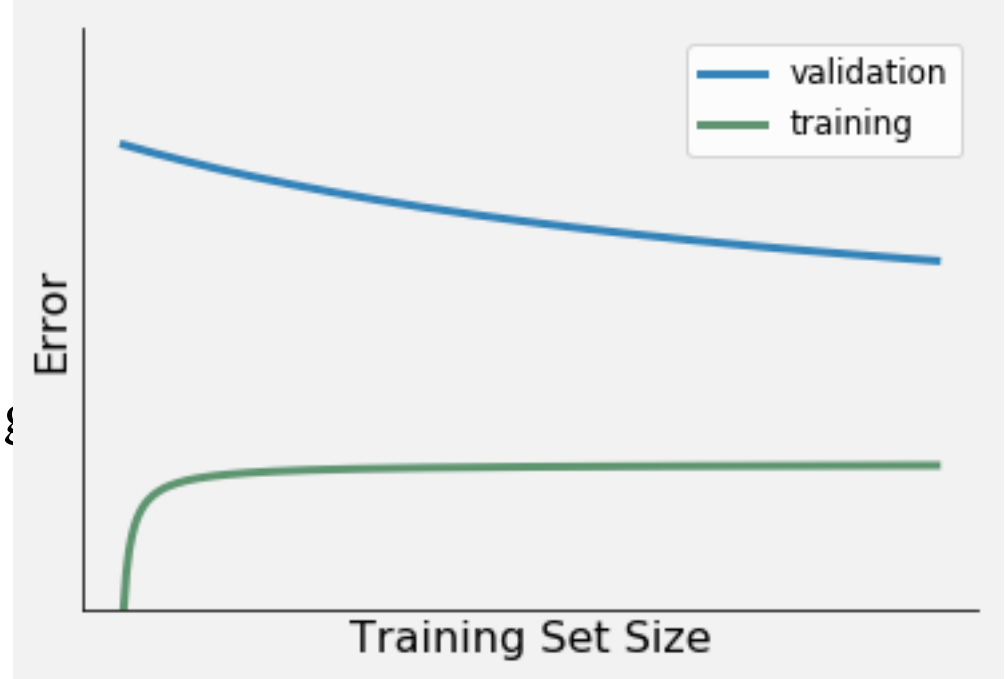


Learning Curves and What They Tell Us

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
  improving if we can g

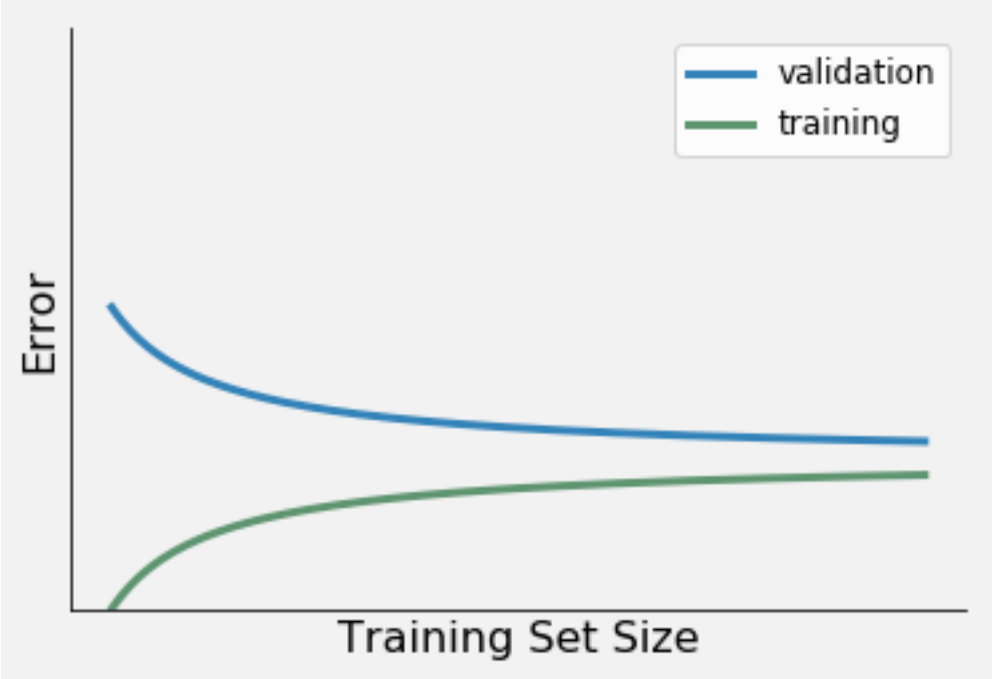


Learning Curves and What They Tell Us

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

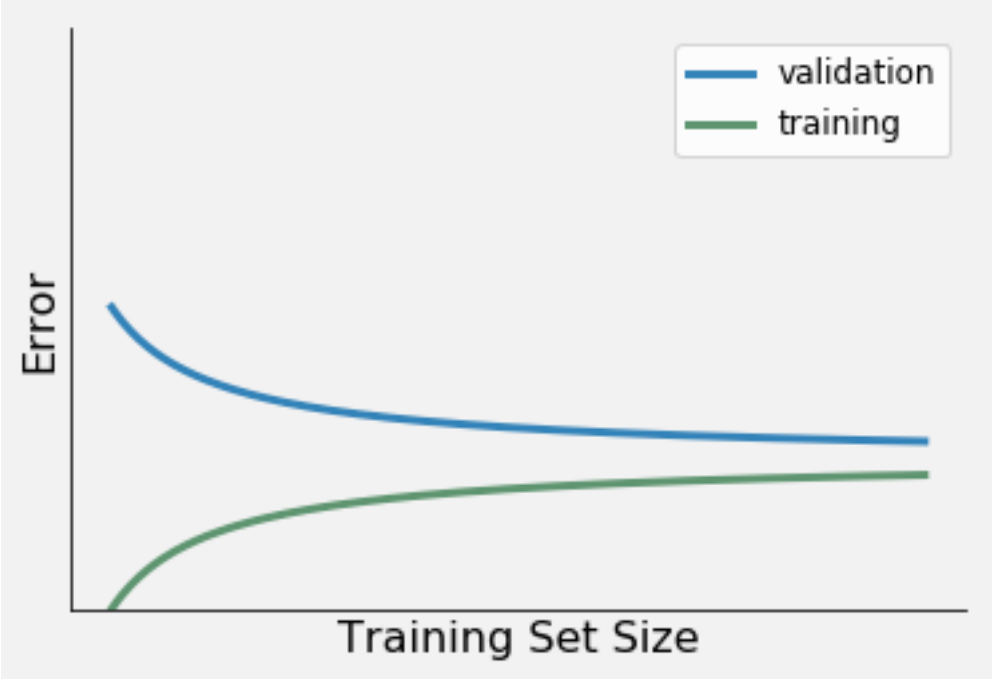


Learning Curves and What They Tell Us

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 more data



Next Time

- Regularization
- Better solution methods

Acknowledgements

Many of the slides in this presentation were adopted from Jordan Boyd-Graber and Lauren Hannah

Some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani

Scratch

Scratch

Scratch

Scratch

Scratch

Scratch

Scratch