University of Colorado **Boulder**

Department of Computer Science

CSCI 2820: Linear Algebra with CS Apps

Chris Ketelsen

Least-Squares, Linear Regression, and Machine Learning

The Story So Far

Given: $m \times n$ matrix A, and length-m vector $\mathbf{b}$

Goal: "Solve" $A\mathbf{x} = \mathbf{b}$

The Story So Far

Given: $m \times n$ matrix A, and length-m vector $\mathbf{b}$

Goal: "Solve" $A\mathbf{x} = \mathbf{b}$

- No exact solution unless $\mathbf{b}$ is in $\mathcal{R}(A)$ (probably not)
- Look for *best* solution in some sense

The Story So Far

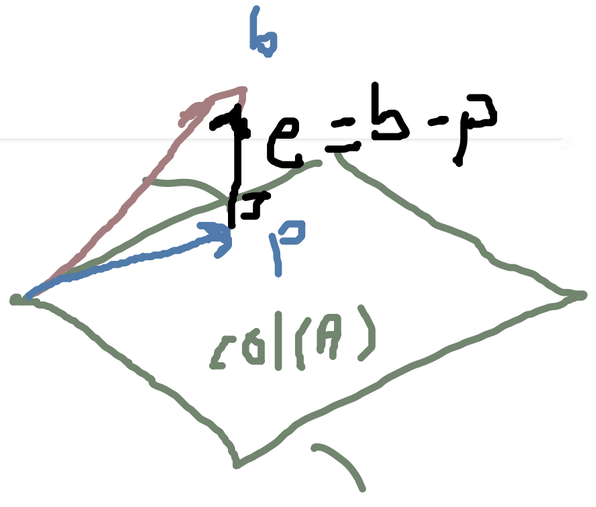
Given: $m \times n$ matrix A, and length-m vector $\mathbf{b}$

Goal: "Solve" $A\mathbf{x} = \mathbf{b}$

- No exact solution unless $\mathbf{b}$ is in $\mathcal{R}(A)$ (probably not)
- Look for *best* solution in some sense

$$\hat{\mathbf{x}} = \text{argmin}_{\mathbf{x}} \|\mathbf{b} - A\mathbf{x}\|_2$$

The Story So Far



Given: $m \times n$ matrix A, and length-m vector $\mathbf{b}$

Goal: "Solve" $A\mathbf{x} = \mathbf{b}$

- No exact solution unless $\mathbf{b}$ is in $\mathcal{R}(A)$ (probably not)
- Look for *best* solution in some sense

$$\hat{\mathbf{x}} = \text{argmin}_{\mathbf{x}} \|\mathbf{b} - A\mathbf{x}\|_2$$

- Best $\hat{\mathbf{x}}$ such that vector $\mathbf{p} = A\hat{\mathbf{x}}$ is the projection of $\mathbf{b}$ onto the range of A.

The Story So Far

Given: $m \times n$ matrix A, and length-m vector $\mathbf{b}$

Goal: "Solve" $A\mathbf{x} = \mathbf{b}$

- No exact solution unless $\mathbf{b}$ is in $\mathcal{R}(A)$ (probably not)
- Look for *best* solution in some sense

$$\hat{\mathbf{x}} = \text{argmin}_{\mathbf{x}} \|\mathbf{b} - A\mathbf{x}\|_2$$

- Best $\hat{\mathbf{x}}$ such that vector $\mathbf{p} = A\hat{\mathbf{x}}$ is the projection of $\mathbf{b}$ onto the range of A.
- Several ways to solve this problem

The Story So Far

Normal Equations: Solve $A^T A\mathbf{x} = A^T\mathbf{b}$

- $A^T A$ is $n \times n$
- $A^T A$ is nonsingular if cols of A are linearly independent.

The Story So Far

Normal Equations: Solve $A^T A\mathbf{x} = A^T\mathbf{b}$

- $A^T A$ is $n \times n$
- $A^T A$ is nonsingular if cols of A are linearly independent.

QR Factorization: Find Q and R such that $A = QR$

- Q is $m \times n$ with orthonormal columns
- R is $n \times n$ upper triangular
- Solution is $\hat{\mathbf{x}} = R^{-1}Q^T\mathbf{b}$

The Story So Far

Normal Equations: Solve $A^T A\mathbf{x} = A^T\mathbf{b}$

$2mn^2$

- $A^T A$ is $n \times n$
- $A^T A$ is nonsingular if cols of A are linearly independent.

QR Factorization: Find Q and R such that $A = QR$

- Q is $m \times n$ with orthonormal columns
- R is $n \times n$ upper triangular
- Solution is $\hat{\mathbf{x}} = R^{-1}Q^T\mathbf{b}$

Pop Quiz: Which one of these is better?

Refresher Example

Compute the least-squares solution to $A\mathbf{x} = \mathbf{b}$ where

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} 4 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

Refresher Example

Compute the least-squares solution to $A\mathbf{x} = \mathbf{b}$ where

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} 4 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

$$A^T A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} 4 & 6 \\ 6 & 14 \end{bmatrix}$$

Refresher Example

Compute the least-squares solution to $A\mathbf{x} = \mathbf{b}$ where

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} 4 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

$$A^T\mathbf{b} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 4 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 6 \\ 4 \end{bmatrix}$$

Refresher Example

Compute the least-squares solution to $A\mathbf{x} = \mathbf{b}$ where

$$A^T A = \begin{bmatrix} 4 & 6 \\ 6 & 14 \end{bmatrix} \quad \text{and} \quad A^T \mathbf{b} = \begin{bmatrix} 6 \\ 4 \end{bmatrix}$$

$$\left[\begin{array}{cc|c} 4 & 6 & 6 \\ 6 & 14 & 4 \end{array}\right] \sim \left[\begin{array}{cc|c} 4 & 6 & 6 \\ 0 & 5 & -5 \end{array}\right]$$

$$\Rightarrow \quad \hat{\mathbf{x}} = \begin{bmatrix} (6 - 6(-1))/4 = 3 \\ -5/5 = -1 \end{bmatrix} \Rightarrow \quad \hat{\mathbf{x}} = \begin{bmatrix} 3 \\ -1 \end{bmatrix}$$

Refresher Example

- What is the optimal projection of $\mathbf{b}$ onto $\mathcal{R}(A)$?
- What is the error vector and least-squares error?

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} 4 \\ 1 \\ 0 \\ 1 \end{bmatrix} \quad \text{and} \quad \hat{\mathbf{x}} = \begin{bmatrix} 3 \\ -1 \end{bmatrix}$$

$$\mathbf{p} = A\hat{\mathbf{x}} = \begin{bmatrix} 3 \\ 2 \\ 1 \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{e} = \mathbf{b} - \mathbf{p} = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix} \quad \text{and} \quad E = \|\mathbf{e}\|_2^2 = 1^2 + (-1)^2 + (-1)^2 + 1^2 = 4$$

Refresher Example

- What is the optimal projection of $\mathbf{b}$ onto $\mathcal{R}(A)$?
- What is the error vector and least-squares error?

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} 4 \\ 1 \\ 0 \\ 1 \end{bmatrix} \quad \text{and} \quad \hat{\mathbf{x}} = \begin{bmatrix} 3 \\ -1 \end{bmatrix}$$

$$\mathbf{p} = A\hat{\mathbf{x}} = \begin{bmatrix} 3 \\ 2 \\ 1 \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{e} = \mathbf{b} - \mathbf{p} = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix} \quad \text{and} \quad E = \|\mathbf{e}\|_2^2 = 1^2 + (-1)^2 + (-1)^2 + 1^2 = 4$$

Pop Quiz: How could we check that we didn't screw anything up?

Refresher Example

Error vector $\mathbf{e}$ should be orthogonal to $\mathcal{R}(A)$

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \quad \text{and} \quad \mathbf{e} = \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix}$$

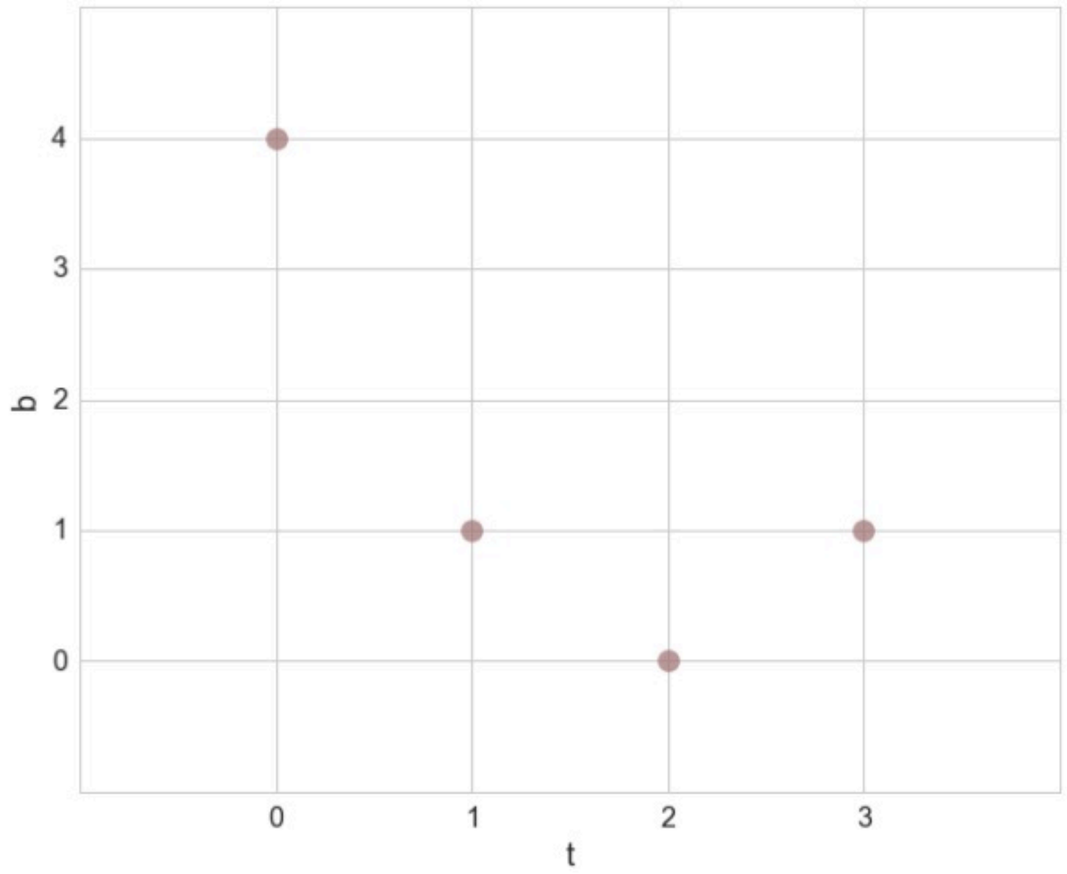
Check:

$$A^T\mathbf{e} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1-1-1+1 \\ 0-1-2+3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \checkmark$$

Data Fitting

Example: Suppose you have data with inputs (t_i) and outputs (b_i):

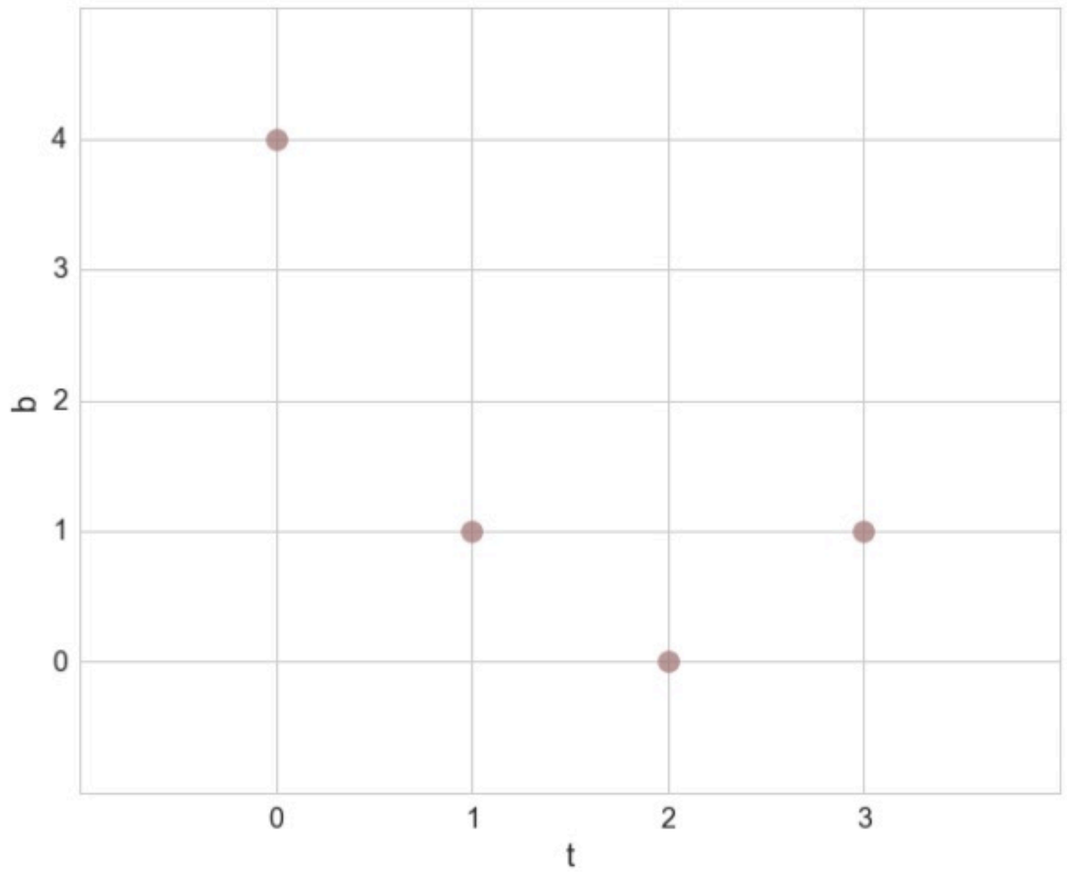
t	0	1	2	3
b	4	1	0	1



Data Fitting

Example: Find linear function, $p(t) = C + Dt$, that *best* fits data

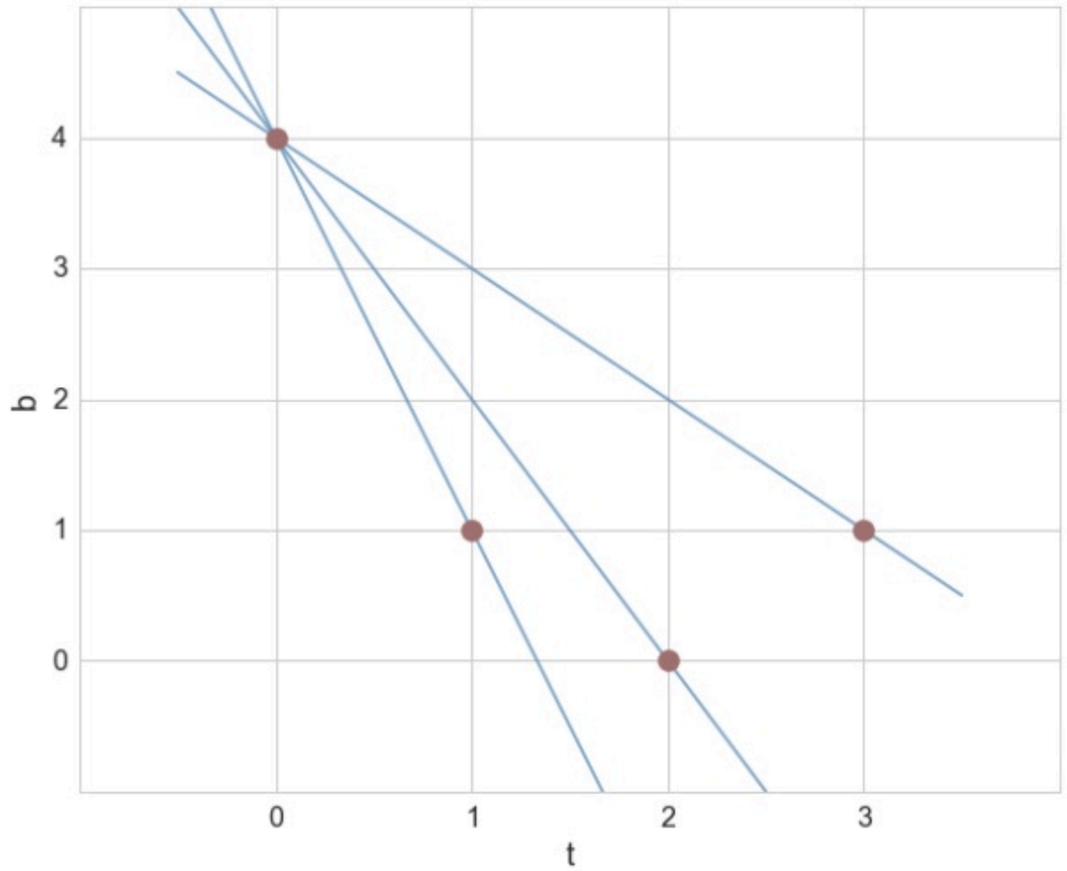
t	0	1	2	3
b	4	1	0	1



Data Fitting

Example: Find linear function, $p(t) = C + Dt$, that *best* fits data

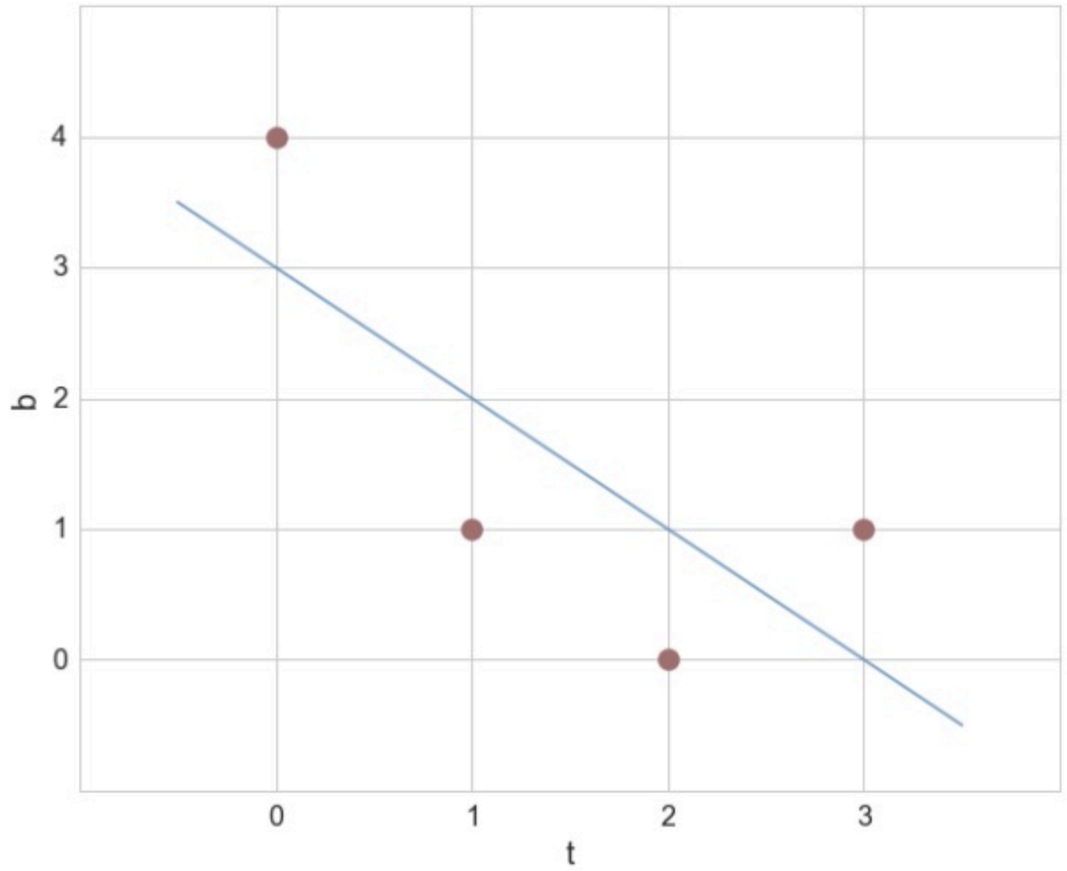
t	0	1	2	3
b	4	1	0	1



Data Fitting

Example: Find linear function, $p(t) = C + Dt$, that *best* fits data

t	0	1	2	3
b	4	1	0	1



Data Fitting

$P(0) = 4$

Example: Find linear function, $p(t) = C + Dt$, that *best* fits data

t	0	1	2	3
b	4	1	0	1

Need to determine coefficients C and D. How could we do this?

Data Fitting

Example: Find linear function, $p(t) = C + Dt$, that *best* fits data

t	0	1	2	3
b	4	1	0	1

Need to determine coefficients C and D. How could we do this?

$$
\begin{aligned}
t_1 = 0 \quad & C + D \cdot 0 = 4 \\
t_2 = 1 \quad & C + D \cdot 1 = 1 \\
t_3 = 2 \quad & C + D \cdot 2 = 0 \\
t_4 = 3 \quad & C + D \cdot 3 = 1
\end{aligned}
$$

$Ax = b$

$$
x = \begin{bmatrix} C \\ D \end{bmatrix} \qquad A = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}
$$

Data Fitting

Example: Find linear function, $p(t) = C + Dt$, that *best* fits data

t	0	1	2	3
b	4	1	0	1

Need to determine coefficients C and D. How could we do this?

$$
\begin{aligned}
t_1 = 0 \qquad & C + D \cdot 0 = 4 \\
t_2 = 1 \qquad & C + D \cdot 1 = 1 \\
t_3 = 2 \qquad & C + D \cdot 2 = 0 \\
t_4 = 3 \qquad & C + D \cdot 3 = 1
\end{aligned}
$$

Kinda already know we have no hope of solving exactly

Data Fitting

Example: Find linear function, $p(t) = C + Dt$, that *best* fits data

t	0	1	2	3
b	4	1	0	1

Need to determine coefficients C and D. How could we do this?

Notice that we can write it as a linear system

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} C \\ D \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \\ 0 \\ 1 \end{bmatrix} \quad \Leftrightarrow \quad A\mathbf{x} = \mathbf{b}$$

Data Fitting

Example: In fact it's the **same** system from last example!

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} C \\ D \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \\ 0 \\ 1 \end{bmatrix} \quad \Leftrightarrow \quad A\mathbf{x} = \mathbf{b}$$

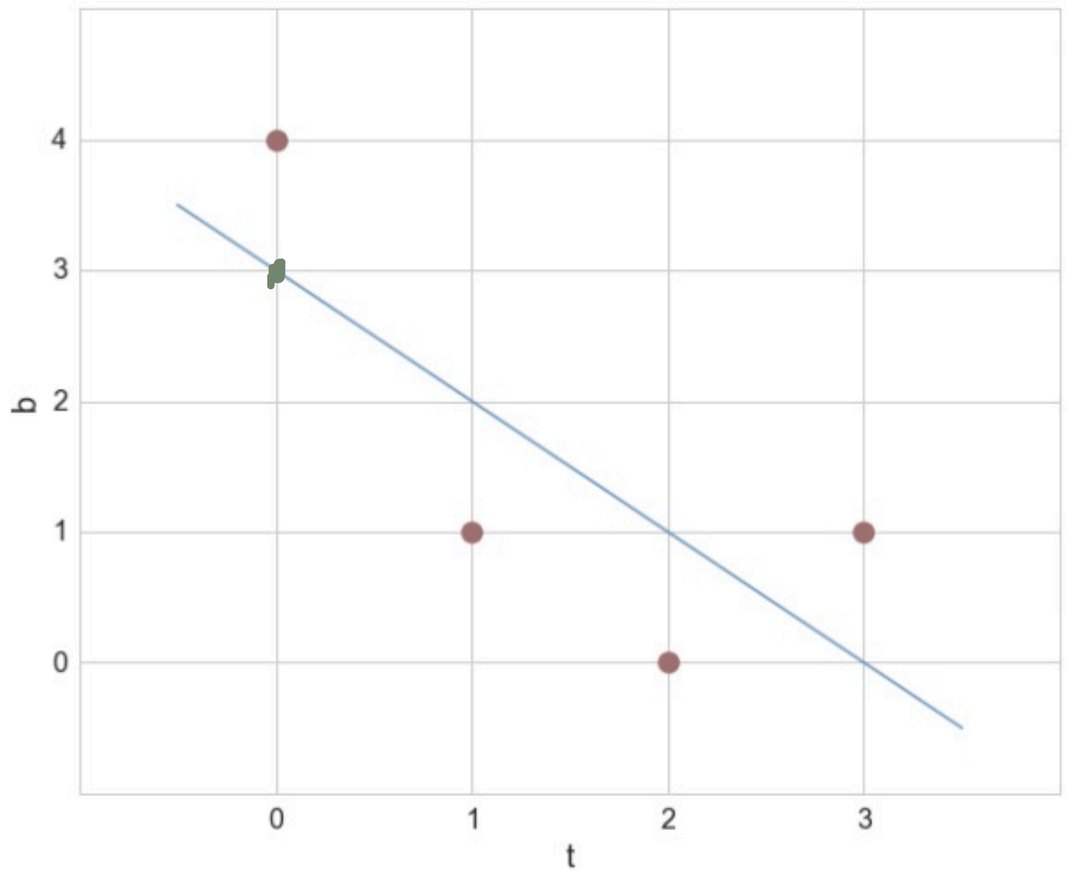
By solving the normal equations, we found

$$\hat{\mathbf{x}} = \begin{bmatrix} C \\ D \end{bmatrix} = \begin{bmatrix} 3 \\ -1 \end{bmatrix}$$

Best linear polynomial is $p(t) = 3 - t$

Data Fitting

Example: Best linear polynomial is $p(t) = 3 - t$

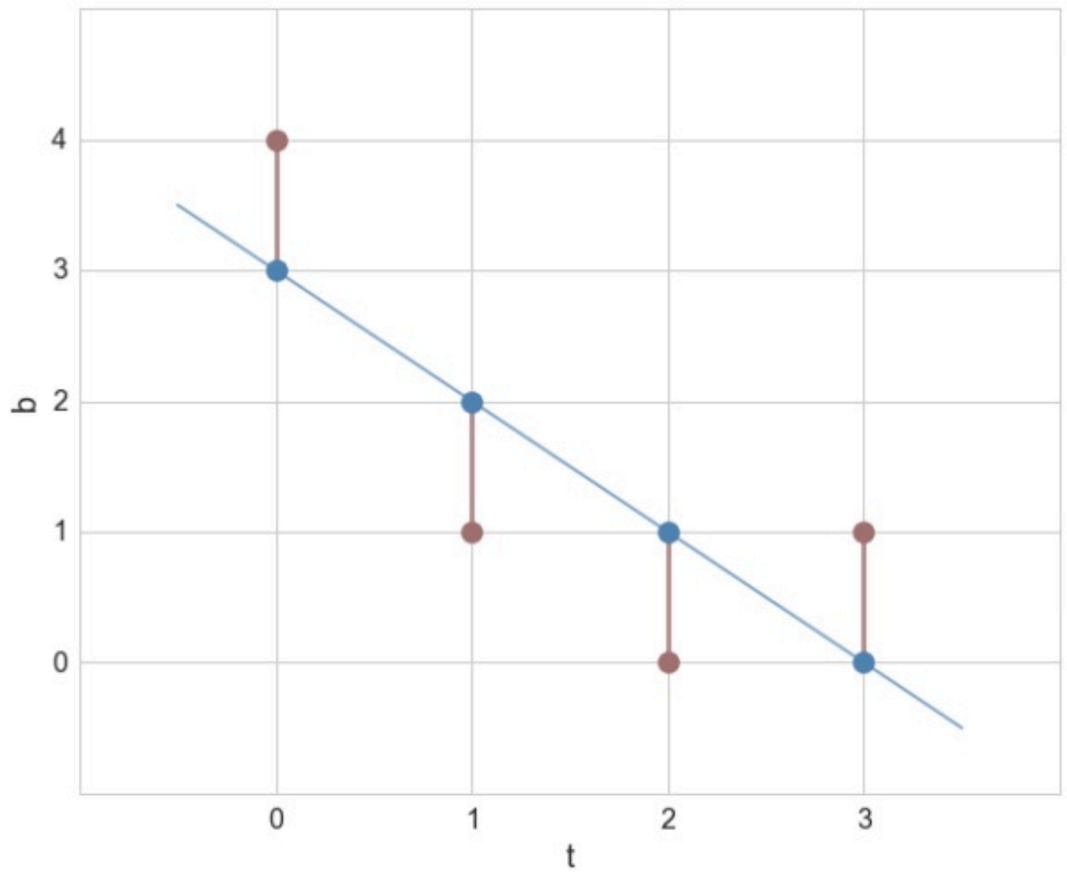


Polynomial Coefficients $\Leftrightarrow$ LS-solution vector $\hat{\mathbf{x}}$

Data Fitting

0, p(0) = 3 - 0 = 3
1, p(1) = 3 - 1·1 = 2
2, 1
3, 0

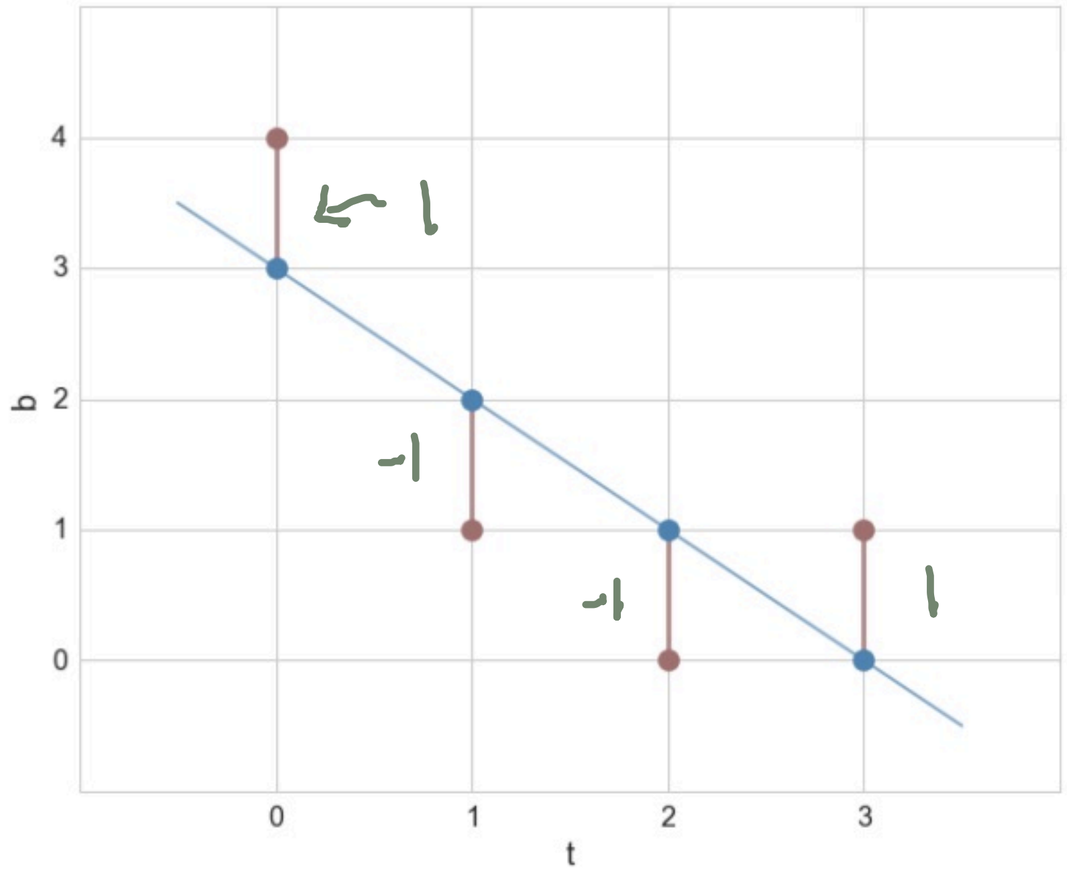
Example: Recall $\mathbf{p} = (3, 2, 1, 0)$. Plug t-values back into $p(t)$



Points on Line $\Leftrightarrow$ Projection of $\mathbf{b}$ onto $\mathcal{R}(A)$

Data Fitting

Example: Recall $E = \|\mathbf{e}\|^2 = 4$. Sum up squares of errors.



LS-Error $\|\mathbf{e}\|_2^2 \Leftrightarrow$ sum of squared errors

Data Fitting

General: Suppose you have points $\{(t_i, b_i)\}_{i=1}^{m}$.

Finding best linear function $p(t) = x_1 + x_2 t$ equiv. to $A\mathbf{x} = \mathbf{b}$ where

$$A = \begin{bmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_m \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Data Fitting

General: Suppose you have points $\{(t_i, b_i)\}_{i=1}^{m}$.

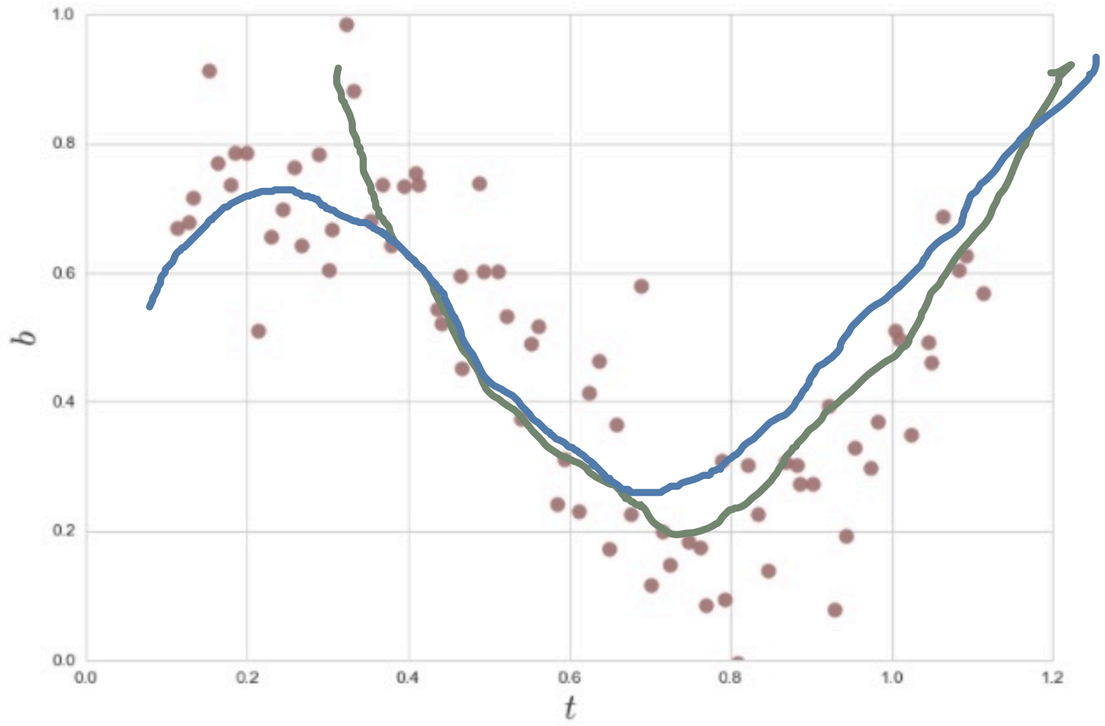
Finding best linear function $p(t) = x_1 + x_2 t$ equiv. to $A\mathbf{x} = \mathbf{b}$ where

$$A = \begin{bmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_m \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Could we take it further than this though?

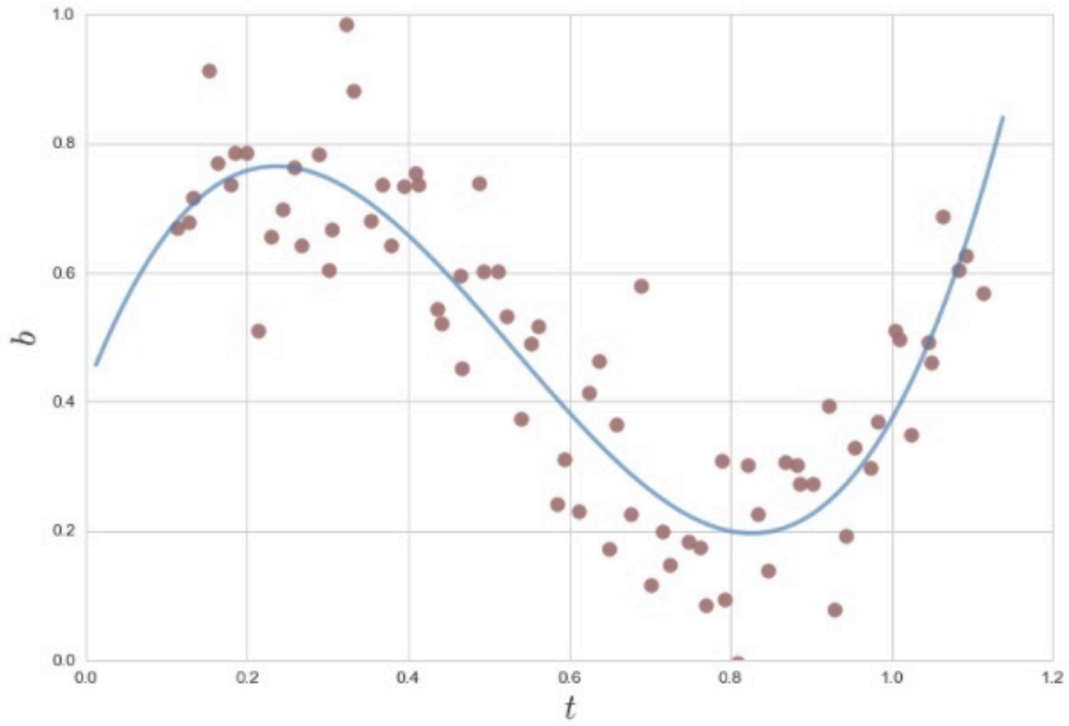
Data Fitting

What if the data looks nothing-like linear?



Data Fitting

Fit a higher degree polynomial!



In this case: $p(t) = x_1 + x_2 t + x_3 t^2 + x_4 t^3$

Data Fitting

What would A look like for $p(t) = x_1 + x_2 t + x_3 t^2 + x_4 t^3$?

Data Fitting

What would A look like for $p(t) = x_1 + x_2 t + x_3 t^2 + x_4 t^3$?

$$A = \begin{bmatrix} 1 & t_1 & t_1^2 & t_1^3 \\ 1 & t_2 & t_2^2 & t_2^3 \\ 1 & t_3 & t_3^2 & t_3^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & t_m & t_m^2 & t_m^3 \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Coefficients are again LS-solution to $A\mathbf{x} = \mathbf{b}$

$p(t) = x_1 + x_2 t + x_3 t^2$

Machine Learning

OK, so what? What does fitted polynomial $p(t)$ do for us?

Machine Learning

OK, so what? What does fitted polynomial $p(t)$ do for us?

- Make predictions on new data t by plugging into $p(t)$

Machine Learning

OK, so what? What does fitted polynomial $p(t)$ do for us?

- Make predictions on new data t by plugging into $p(t)$
- Make inferences about the data

Machine Learning

OK, so what? What does fitted polynomial $p(t)$ do for us?

- Make predictions on new data t by plugging into $p(t)$
- Make inferences about the data

In Machine Learning and Statistics, taking real inputs and predicting real outputs is called **Regression**.

Machine Learning

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 a product page or predict probability that they'll click on an ad
- Given the advertising budget expenditures in various media markets, predict the number of products sold

Machine Learning

General Framework:

1. Collect some *training data* $\{\mathbf{x}_i, y_i\}_{i=1}^{m}$
2. Fit a regression model: $y = p(x)$
3. Use model to make predictions about new $\mathbf{x}'s$.

This is an example of **supervised learning**

Some Questions:

Machine Learning

General Framework:

1. Collect some *training data* $\{\mathbf{x}_i, y_i\}_{i=1}^m$
2. Fit a regression model: $y = p(x)$
3. Use model to make predictions about new $\mathbf{x}'s$.

This is an example of **supervised learning**

Some Questions:

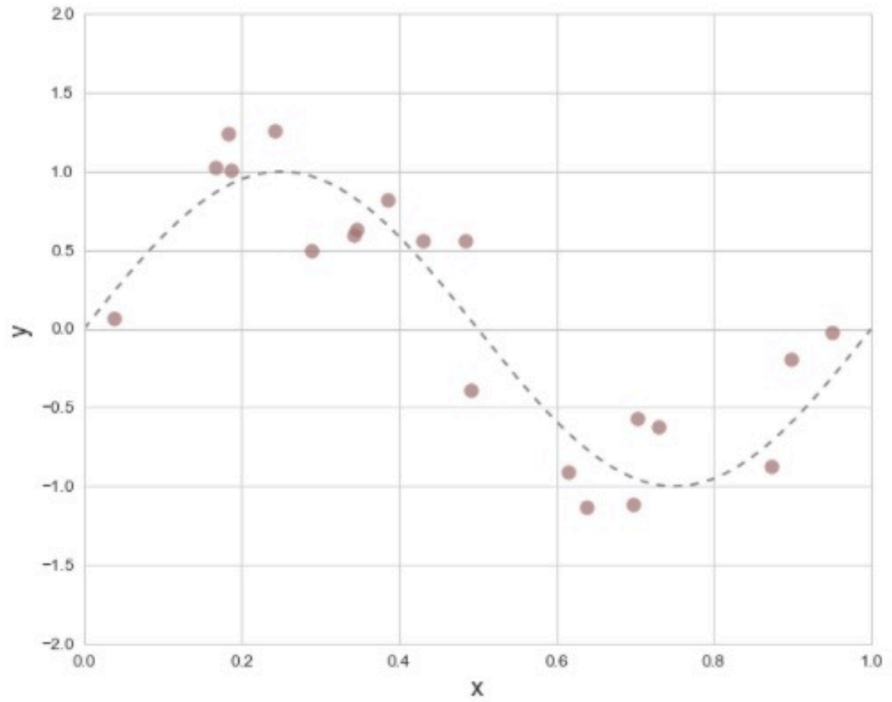
- How complicated of a model should I use?
- How well do I expect the model to do on unseen test data?

Machine Learning

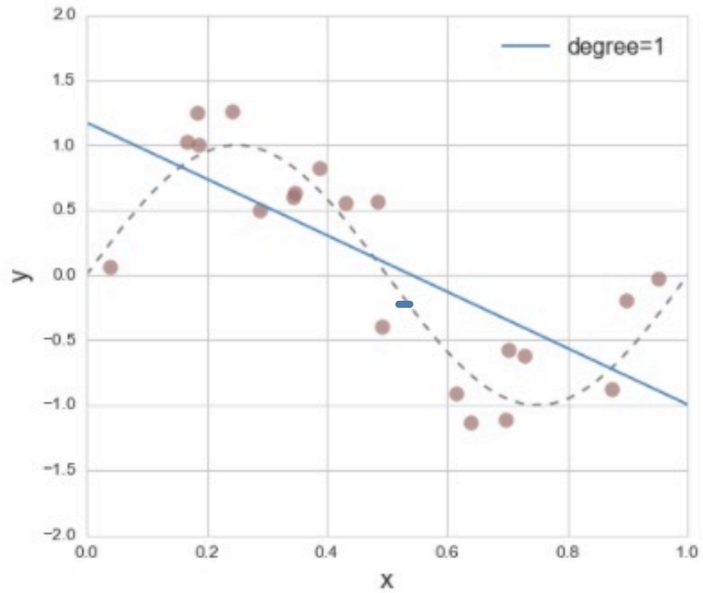
Some Questions:

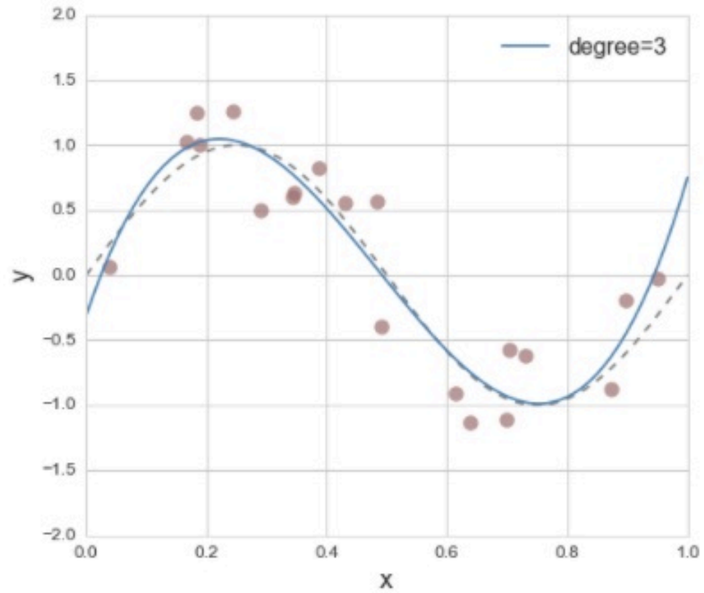
- How complicated of a model should I use?

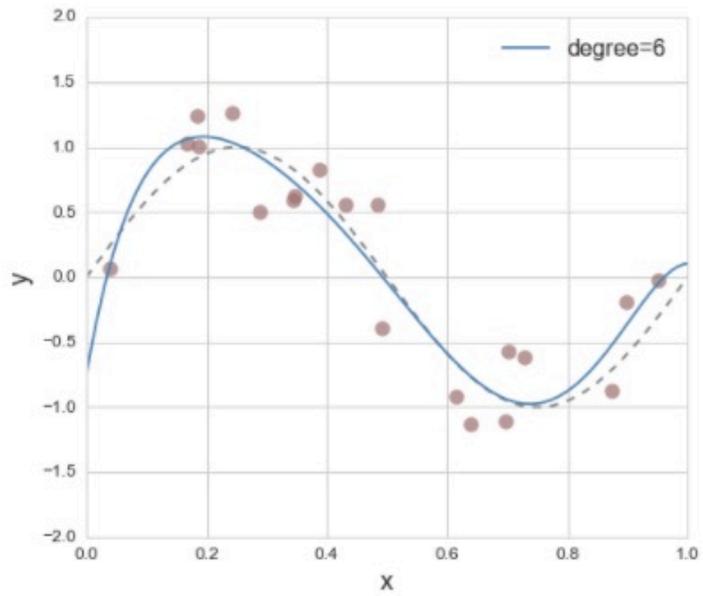
Example: Suppose data comes from $\sin(2\pi x) + \text{Noise}$

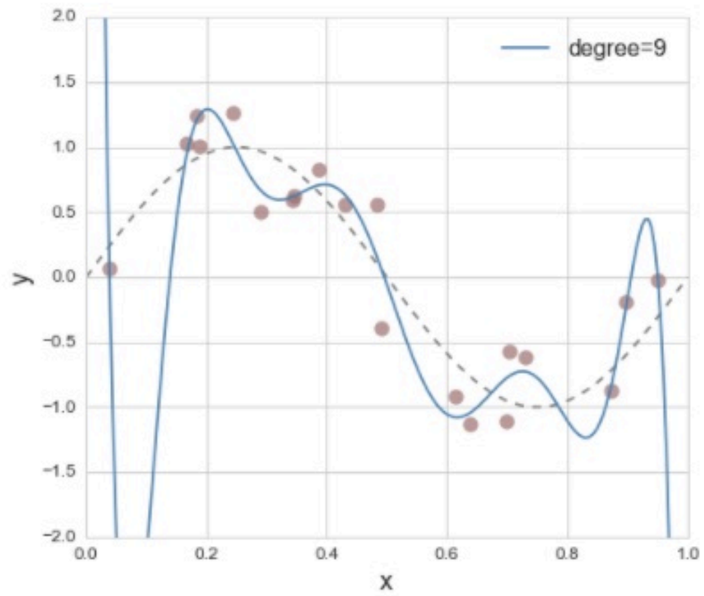


Machine Learning









Machine Learning

What do we want in a model?

- Complicated enough that it doesn't **underfit** the data
- Not too complicated that it **overfits** the data

But how do I know?

- For high-dimensional data you can't really make a graph and decide. Need a more mathematical/numerical way to evaluate performance

Machine Learning

Model Evaluation:

80% 20%

- Split labeled data into a **training set** and a **validation set**
- Fit model to training data
- Check goodness of fit on training data
- Check goodness of fit on validation data

How do we check goodness of fit?

Machine Learning

Model Evaluation:

- Split labeled data into a **training set** and a **validation set**
- Fit model to training data
- Check goodness of fit on training data
- Check goodness of fit on validation data

How do we check goodness of fit?

Use least-squares error!

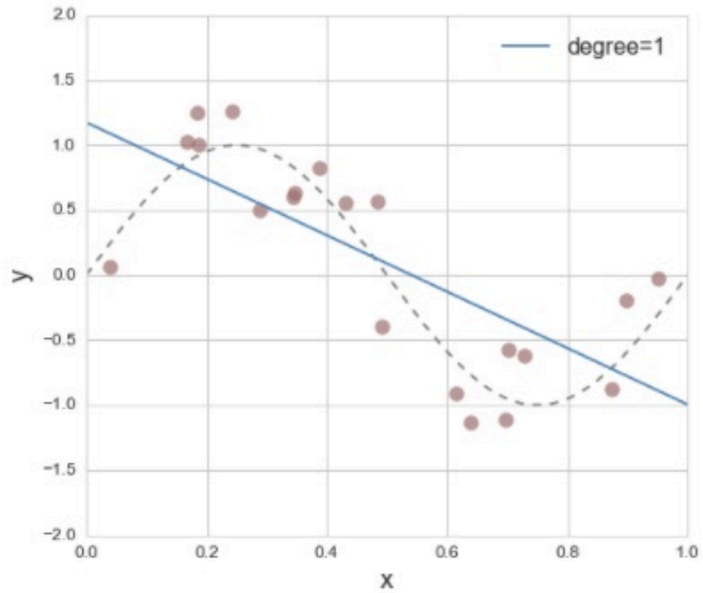
$$E = \|\mathbf{e}\|_2^2 = \frac{1}{m}\sum_{i=1}^{m}(p(x_i) - y_i)^2$$

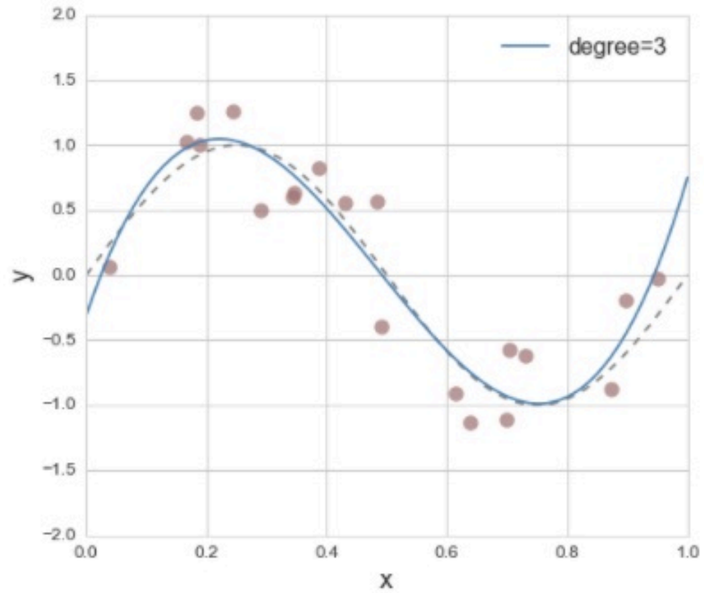
Machine Learning

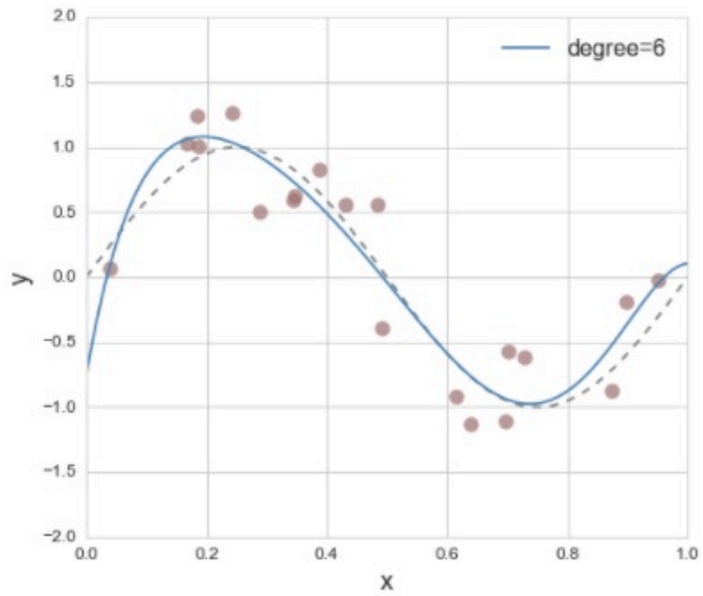
Model Evaluation:

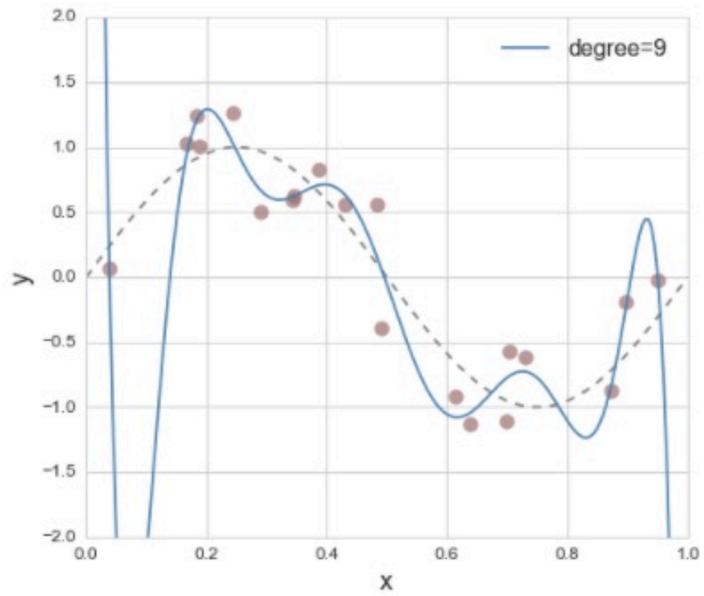
- Try increasingly complex models, plot LS-errors
- What do you expect to happen to the training error as complexity increases?

Machine Learning







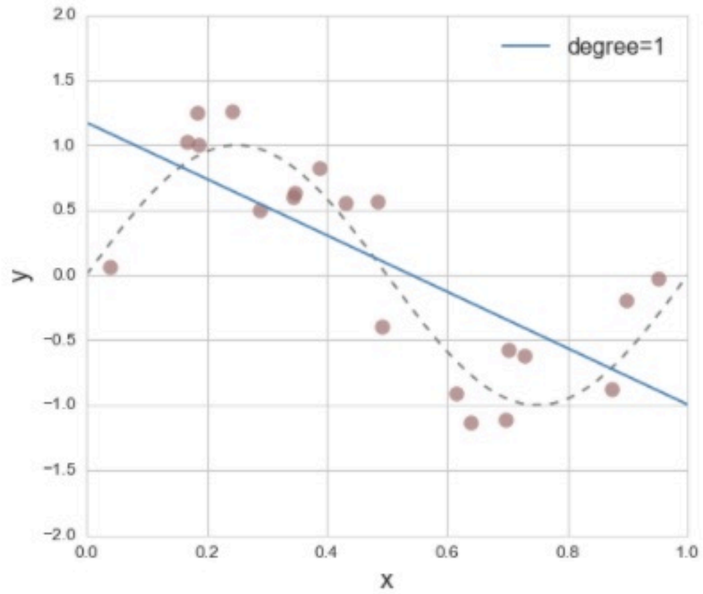


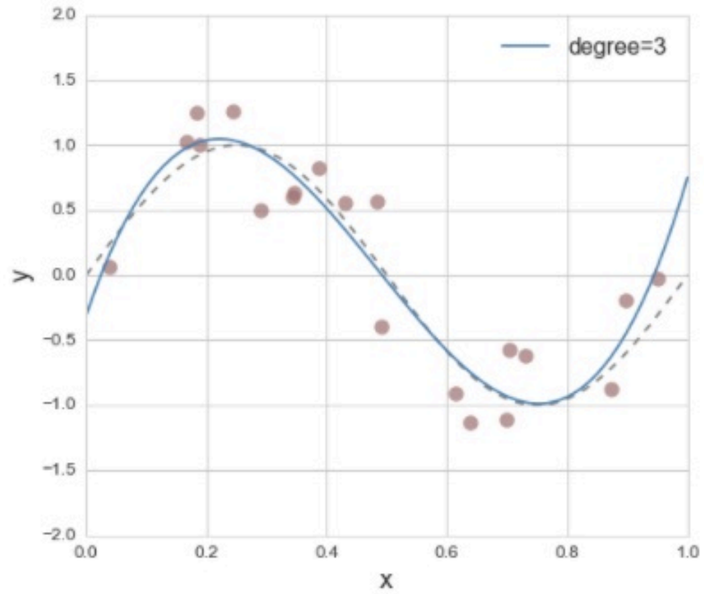
Machine Learning

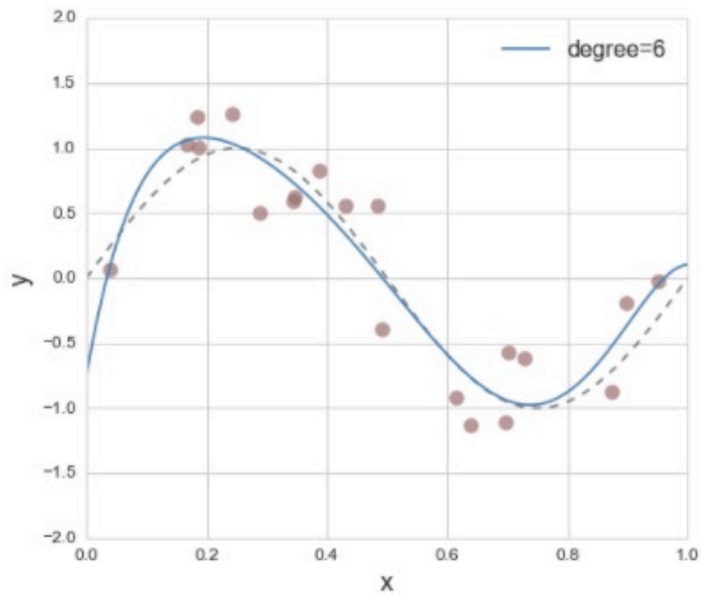
Model Evaluation:

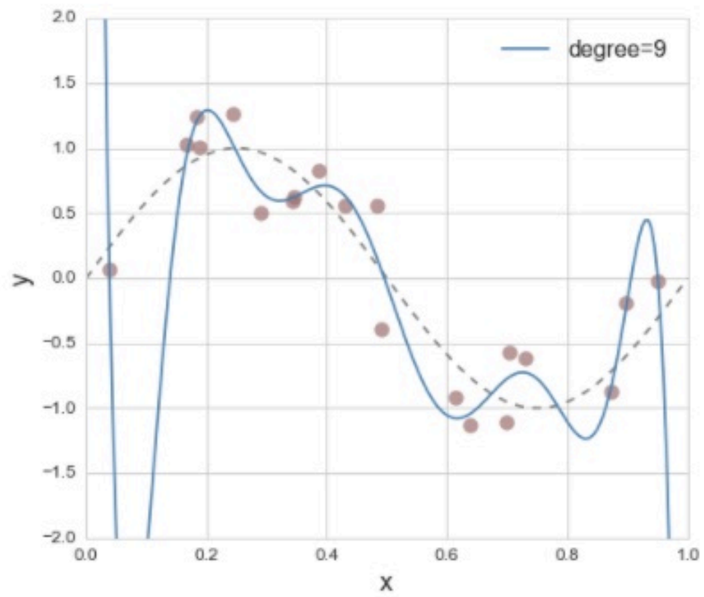
- Try increasingly complex models, plot LS-errors
- What do you expect to happen to the training error as complexity increases?
- What do you expect to happen to the test/validation error as complexity increases?

Machine Learning

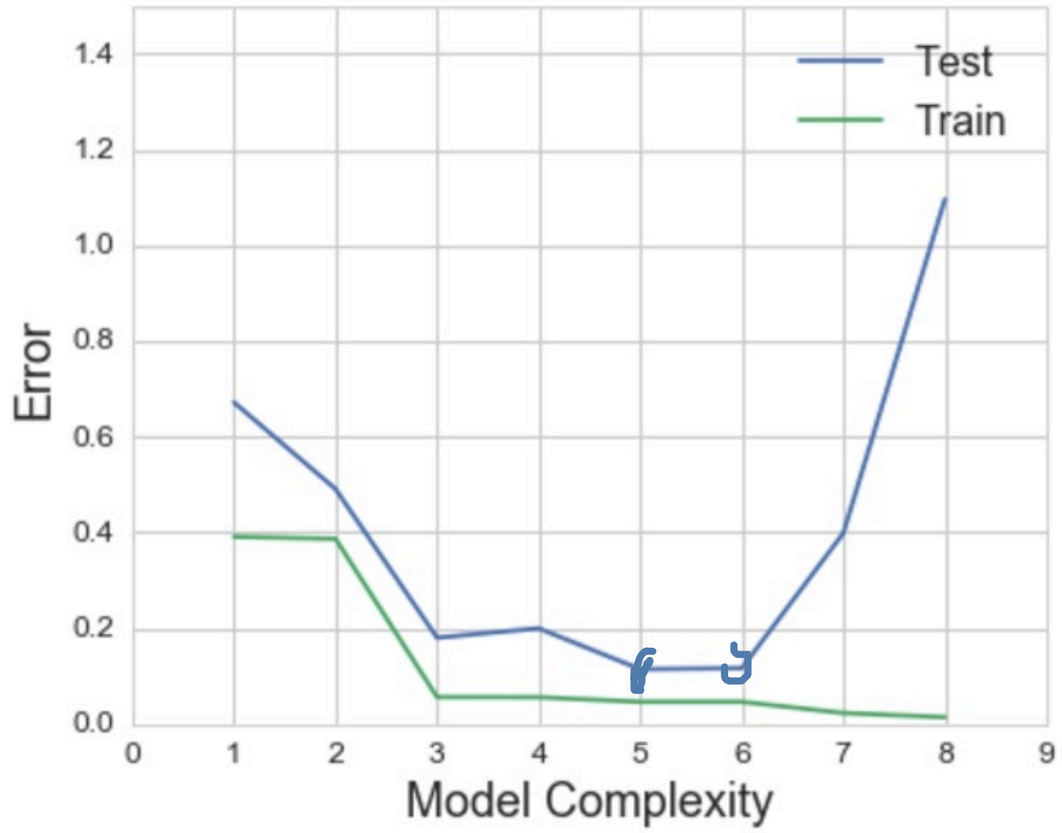








Machine Learning



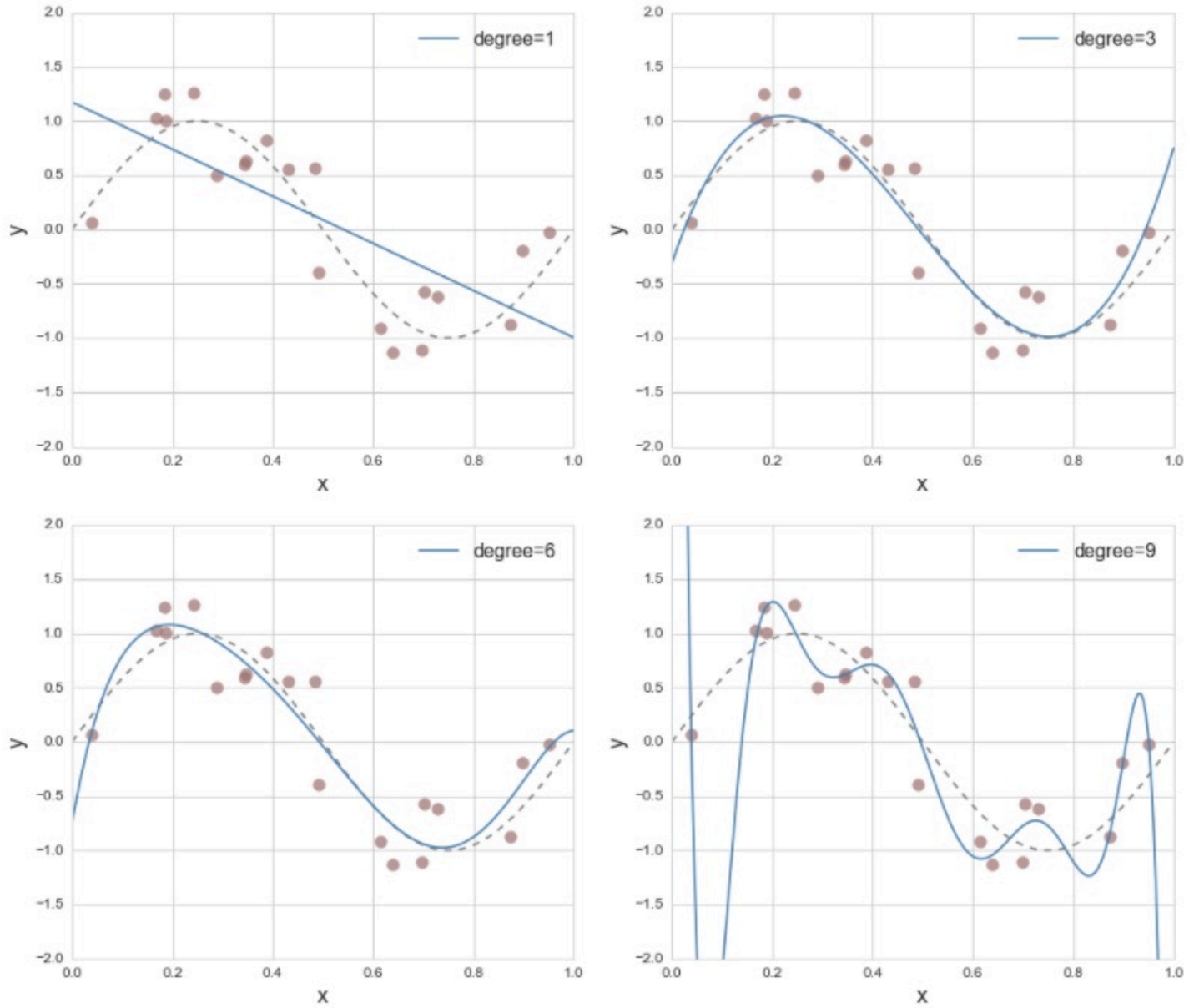
Optimal-ish model complexity is where validation error starts increasing again.

Machine Learning

It's instructive to think a bit harder about the nature of **underfitting** and **overfitting**

There's this thing called the **Bias-Variance Trade-Off**

Machine Learning



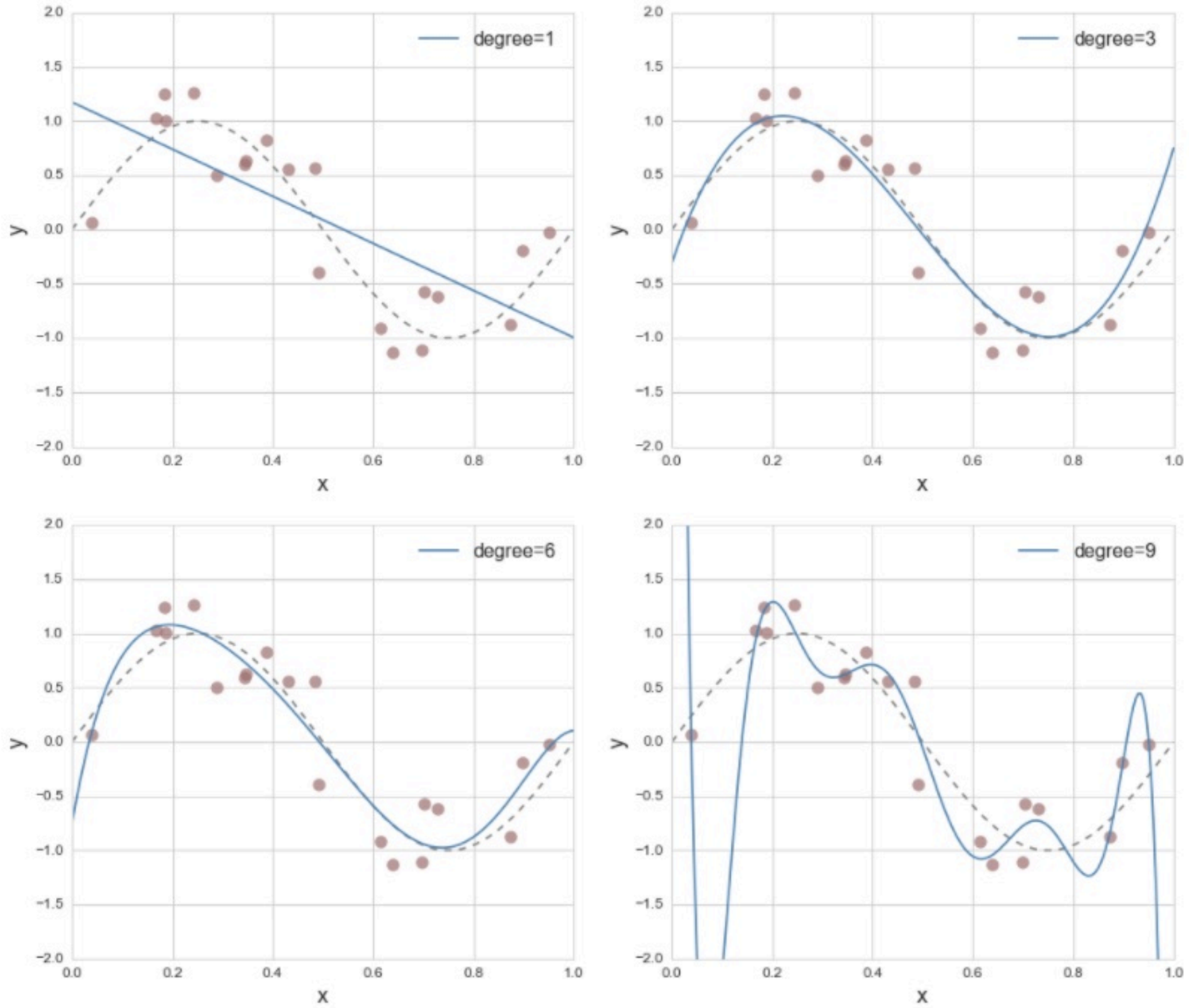
Machine Learning

It's instructive to think a bit harder about the nature of **underfitting** and **overfitting**

There's this thing called the **Bias-Variance Trade-Off**

High Bias methods are very *insensitive* to changes in the training set

Machine Learning



Machine Learning

It's instructive to think a bit harder about the nature of **underfitting** and **overfitting**

There's this thing called the **Bias-Variance Trade-Off**

High Bias methods are very *insensitive* to changes in the training set

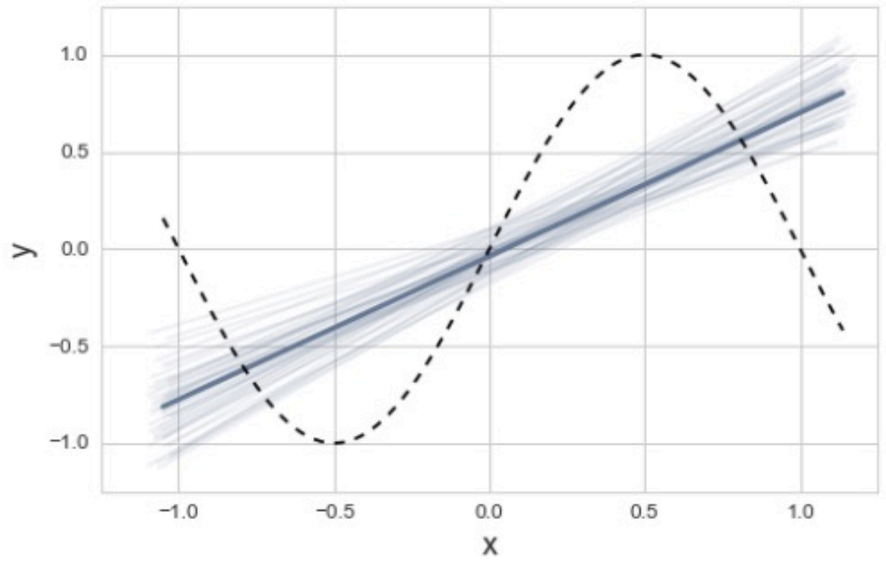
High Variance methods are very *sensitive* to changes in the training set

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

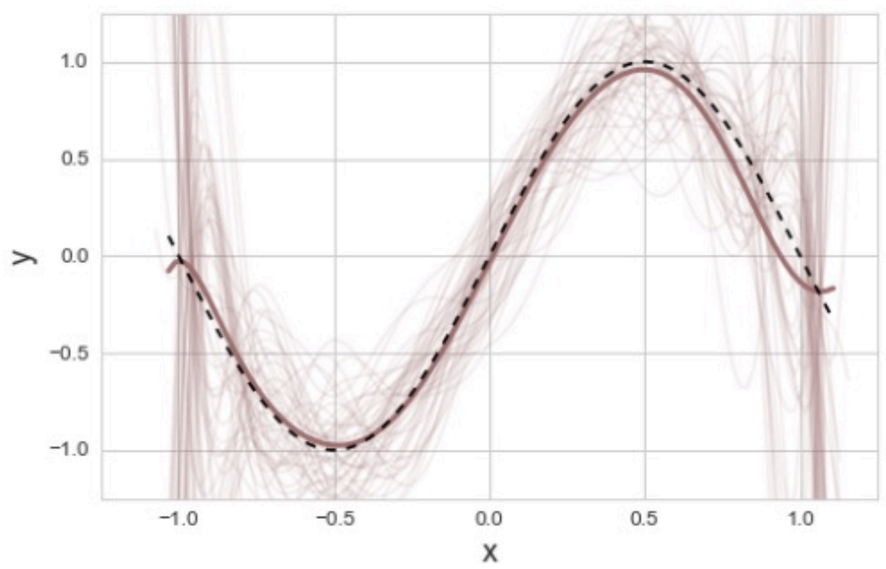


The Bias-Variance Trade-Off Intuition

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

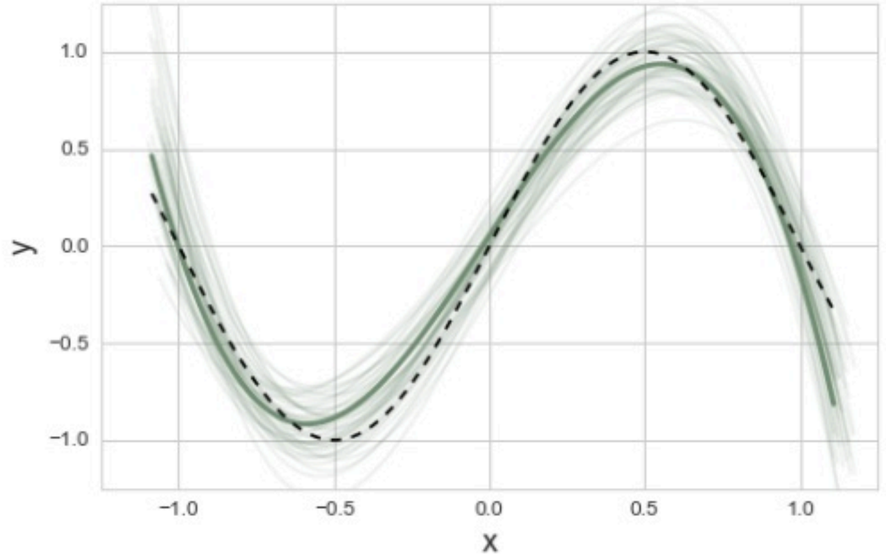


The Bias-Variance Trade-Off Intuition

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



Machine Learning

Want to Know More??:

- CSCI 3022 - Introduction to Data Science
    - Next Fall
- CSCI XXXX - Undergraduate Machine Learning
    - Next Year (Hopefully)

In Class

In Class

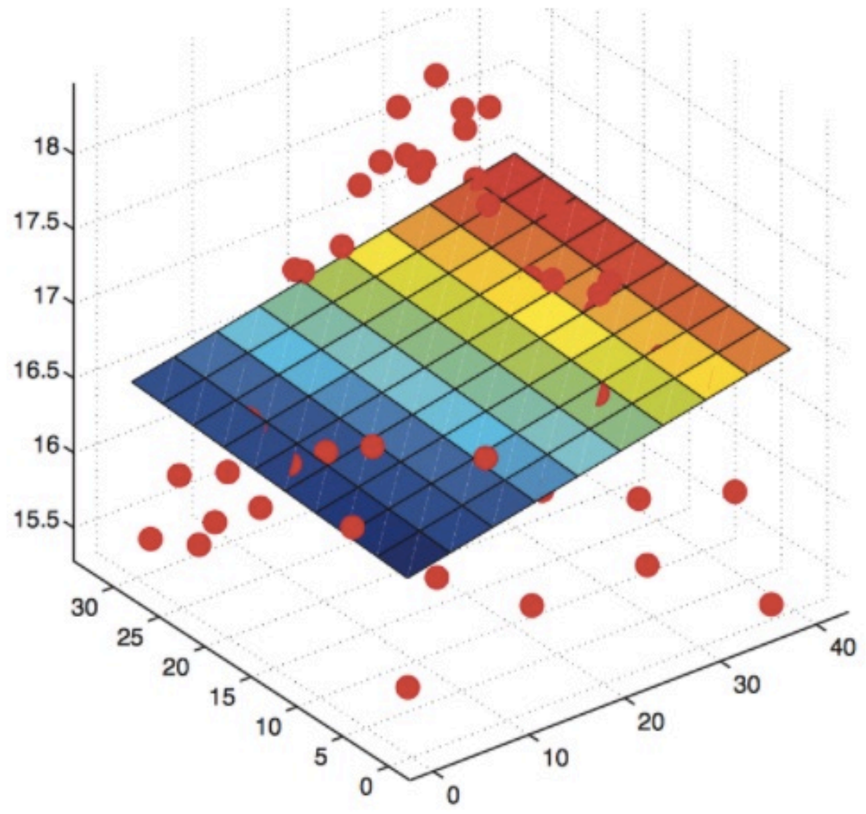
In Class

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 2 features x_1 , and x_2 fit

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