



Data X

About Me:

## Underfitting / Overfitting, Polynomial Regression & Regularization

Alexander Fred Ojala

Alexander Fred Ojala  
Visiting Scholar, IEOR UC Berkeley  
afo@berkeley.edu

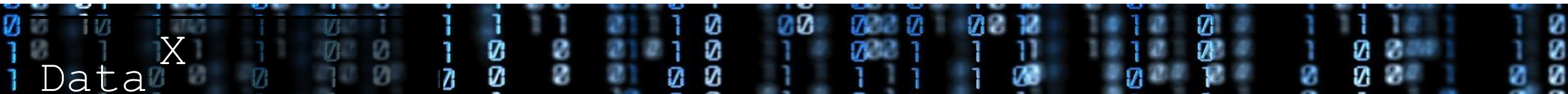
Underfitting



# Underfitting / High bias

## Characteristics of underfitting:

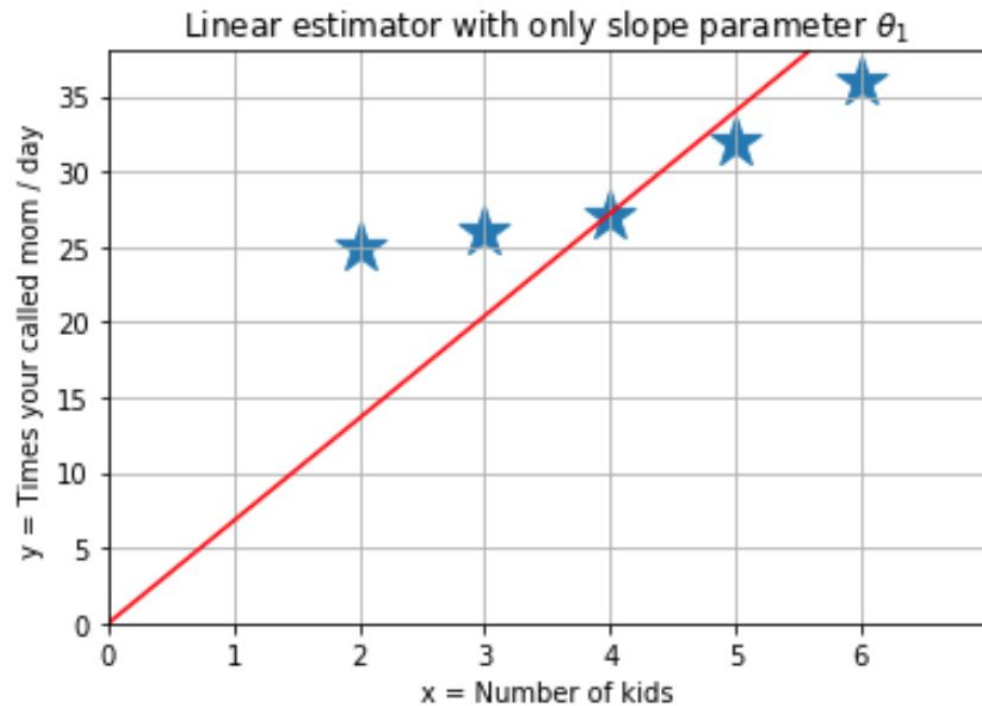
- Our model is too simple (few degrees of freedom)
- Low variance, but high bias
- Strong preconception about model parameters



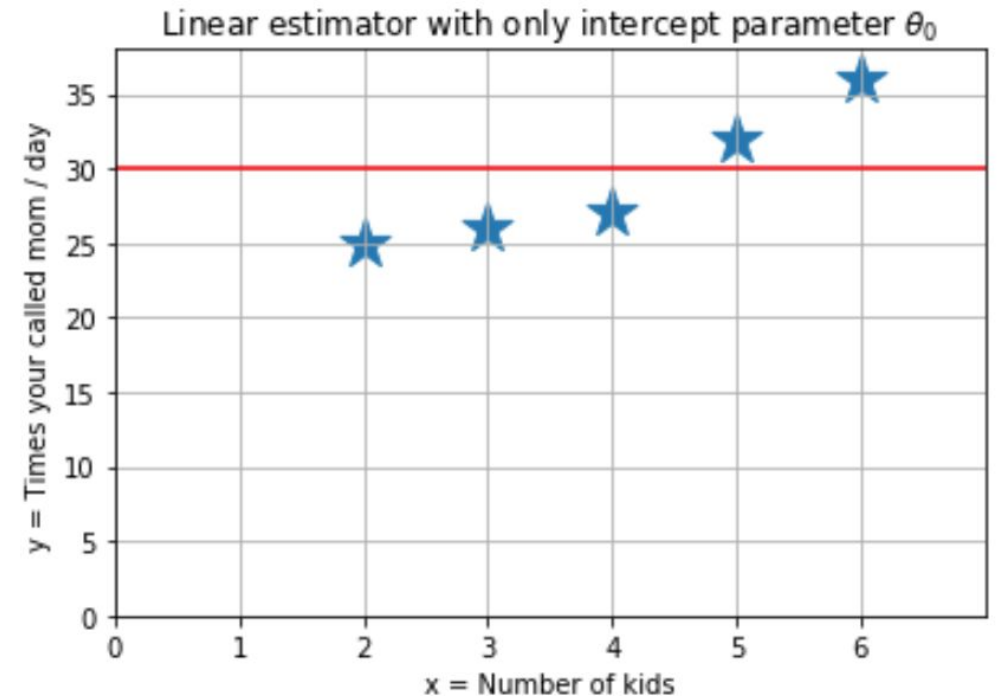


# Underfitting / High bias

**Examples of underfitting:** One degree of freedom is not enough



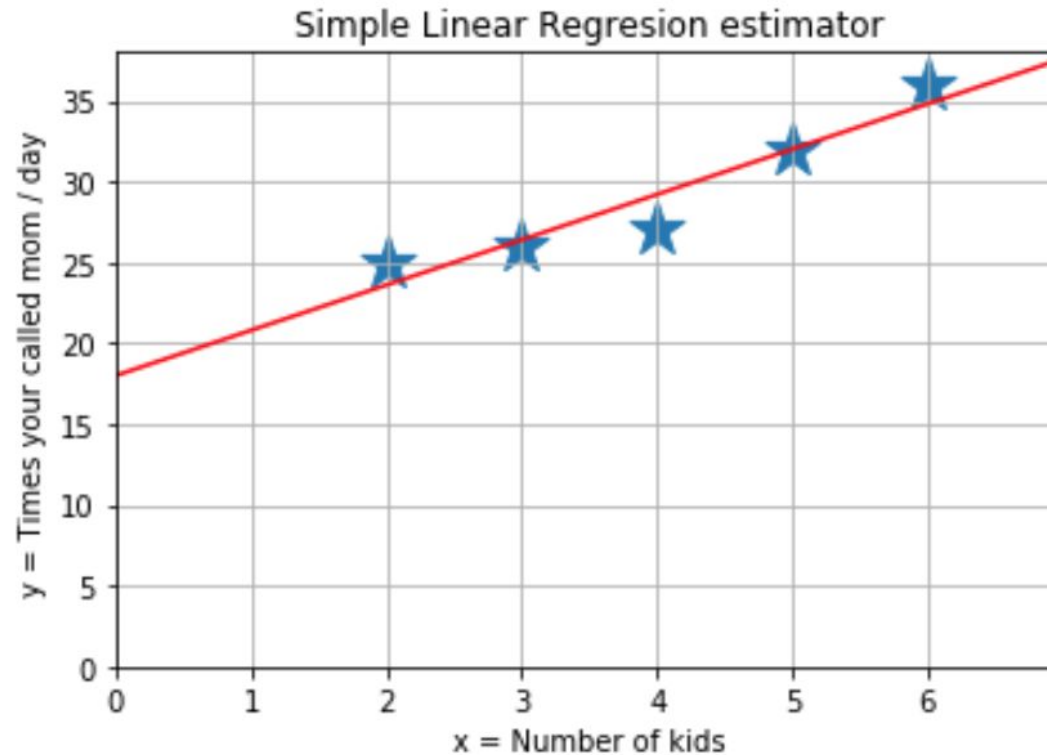
$$\hat{y} = h_{\theta}(x) = \theta_1 x$$



$$\hat{y} = h_{\theta}(x) = \theta_0$$



# Good model approximation for our data



$$\hat{y} = h_{\theta}(x) = \theta_0 + \theta_1 x$$



# Underfitting data / High bias

## How to prevent underfitting:

- Add more features to the model
- Transform features to increase complexity of the model
- Construct features from existing ones



# Polynomial Regression



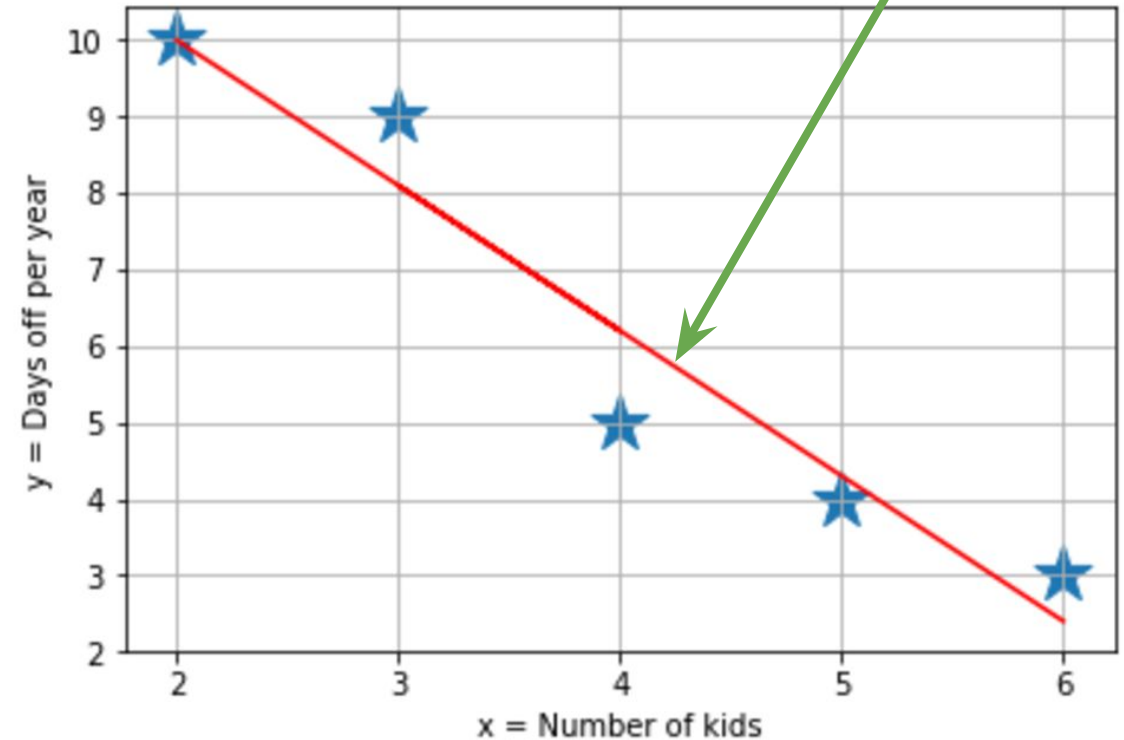
# (Simple) Linear Regression

Works when have a linear relation between dependent and independent variables

$$\hat{y} = f(x, \theta) = h_{\theta}(x) = \theta^T x = \theta_0 + \theta_1 x_1$$

$h_{\theta}(x)$

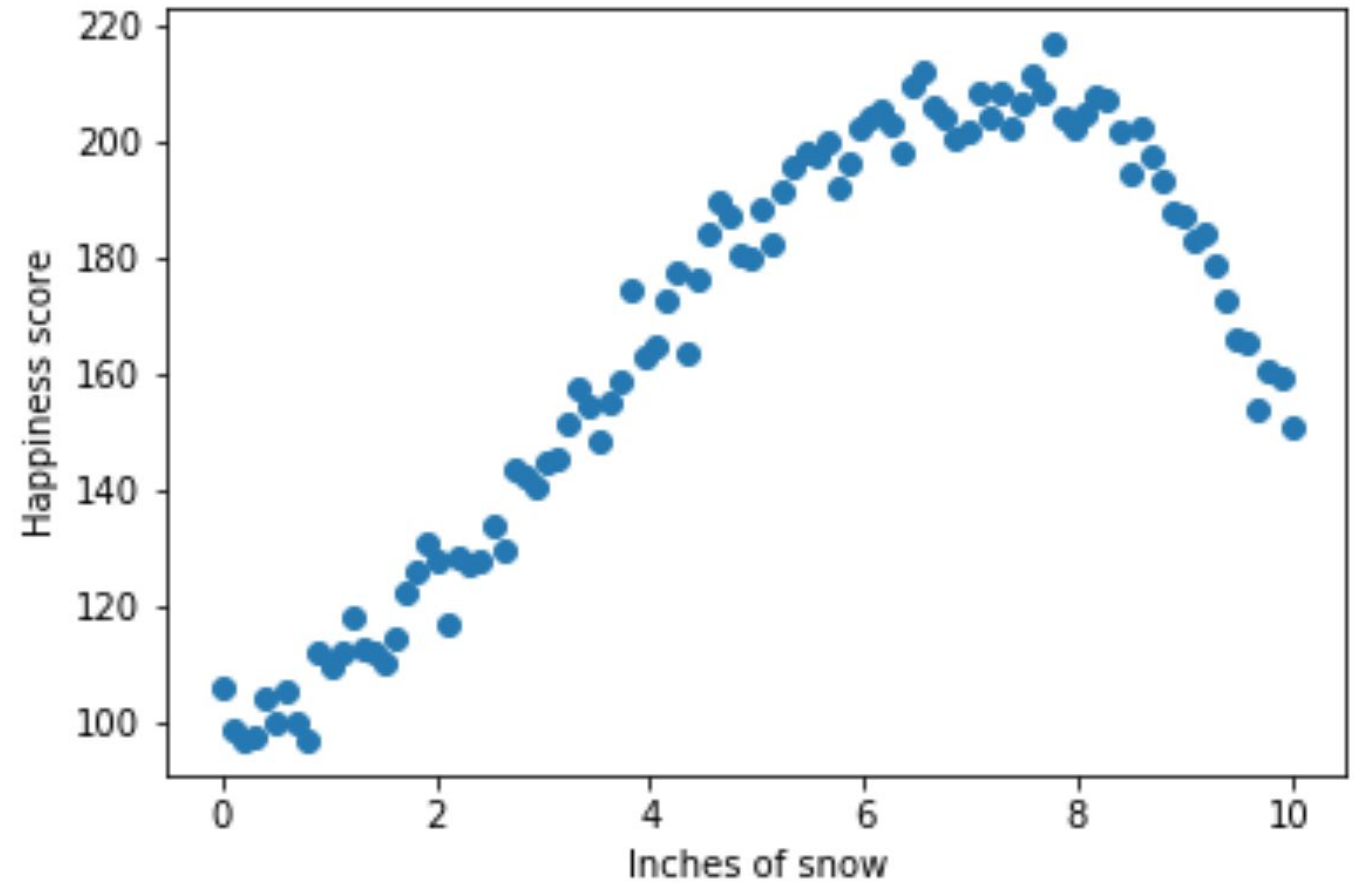
$h_{\theta}(x)$





# Modeling Non-linear relationships

What if we want to model this relation?



# Modeling Non-linear relationships

The best Linear Estimator

Obtained by solving the Normal Equation

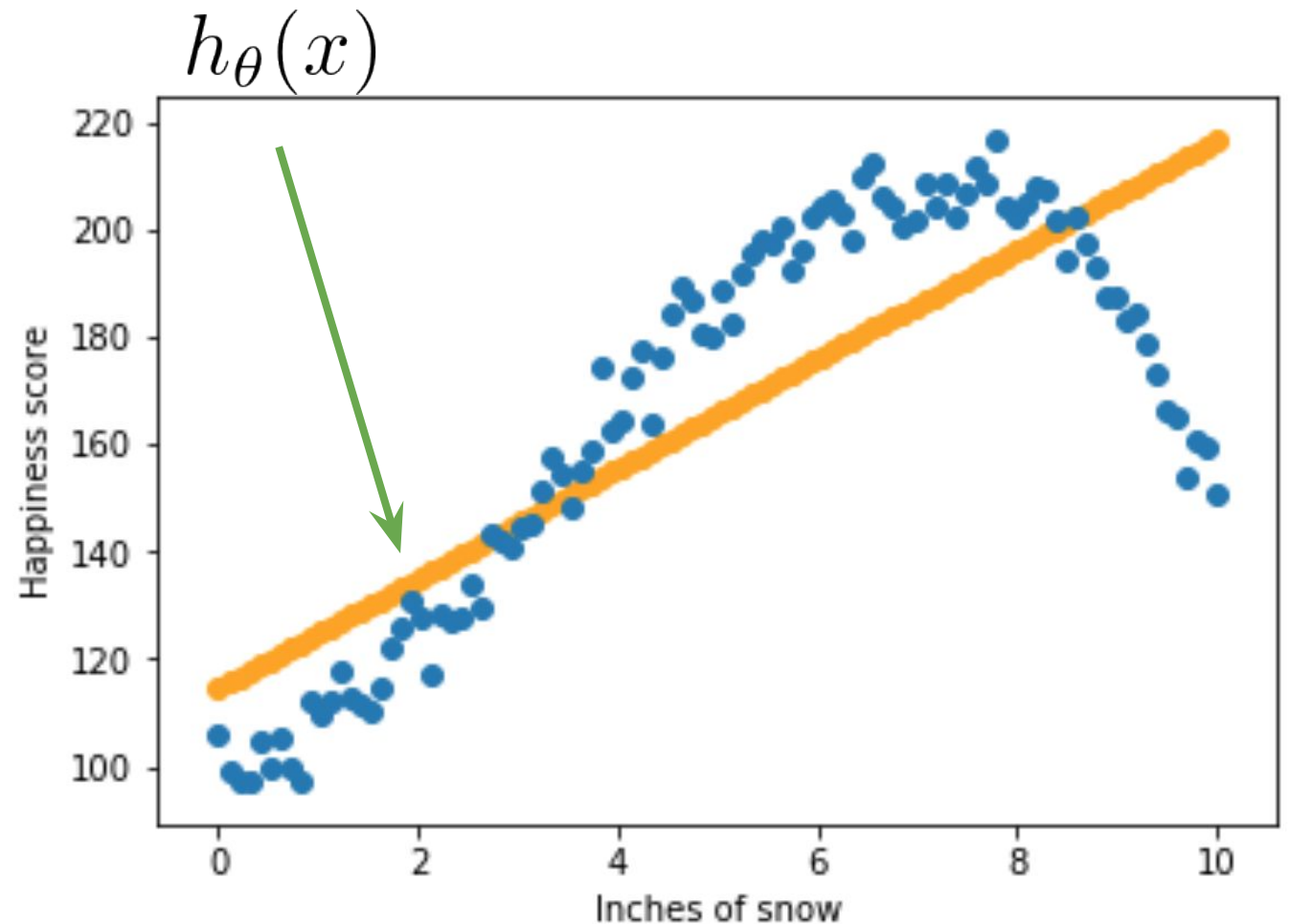
$$\theta = (X^T X)^{-1} X^T y$$

gives us:

$$\begin{aligned}\hat{y} &= h_{\theta}(x) = \theta_0 + \theta_1 x_1 \\ &\approx 117 + 10x_1\end{aligned}$$

**We are clearly underfitting!**

(Our model has high bias)



# Introducing: Polynomial Regression

(Simple) Polynomial Regression:

$$\hat{y} = h_{\theta}(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \dots + \theta_n x^n$$

It is still a Linear Regression model, we have just transformed some of the predictors.

Rewrite the predictors, as:

$$x \rightarrow x_1$$

$$x^2 \rightarrow x_2$$

$$\vdots$$

$$x^n \rightarrow x_n$$

to see that it's still a Linear function for the parameters.

# Exponential / Logarithmic / Square root ... Regression

$$\hat{y} = h_{\theta}(x) = \theta_0 + \theta_1 \sqrt{x} + \theta_2 \log(x) + \theta_3 e^x$$

You can also transform the logarithmic, exponential, the square root of predictors etc.

$$\sqrt{x} \rightarrow x_1$$

$$\log(x) \rightarrow x_2$$

$$e^x \rightarrow x_3$$

$\vdots$

*Since it still can be cast as a multiple linear regression problem we can find the optimal parameters by using the Normal Equations or Gradient Descent (as shown in lecture 5)!*



# Polynomial Regression

**The best polynomial function for prediction** (of degree 3)

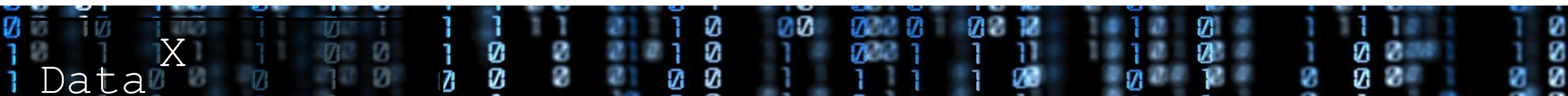
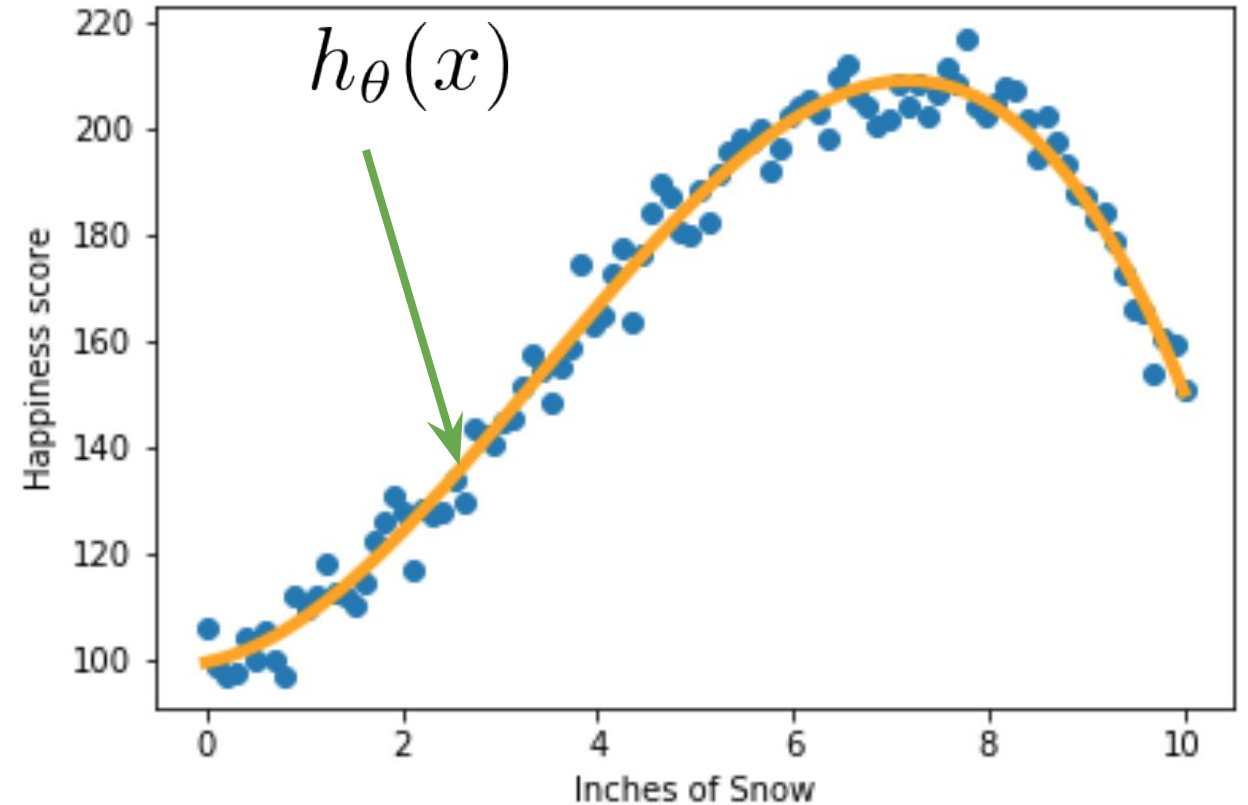
Obtained by solving the Normal Equation

$$\theta = (X^T X)^{-1} X^T y$$

is given by:

$$\begin{aligned}\hat{y} = h_{\theta}(x) &= \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 \\ &\approx 98 + 7x + 4.6x^2 - 0.5x^3\end{aligned}$$

*which is a much better fit to our training data!*



Overfitting

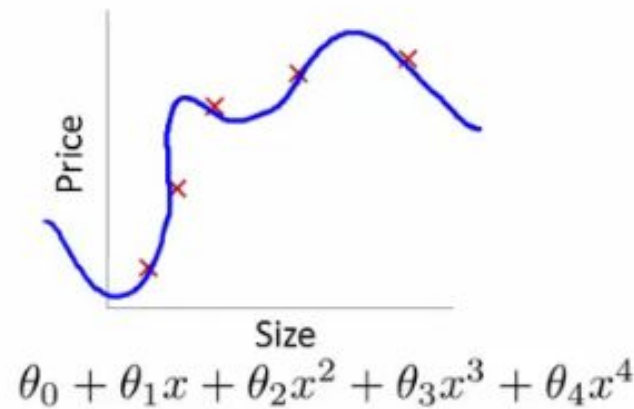


# Overfitting

*Why don't we fit polynomial functions of very high degrees that always fit our data perfectly so that we get an error that approaches zero?*

$$\hat{y} = h_{\theta}(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \dots + \theta_{\infty} x^{\infty} \qquad J(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 \rightarrow \mathbf{0}$$

It leads to **overfitting** (we won't predict well on new data that our model hasn't seen)



High variance  
(overfit)

Data X

# Overfitting / High variance

## Characteristics of overfitting:

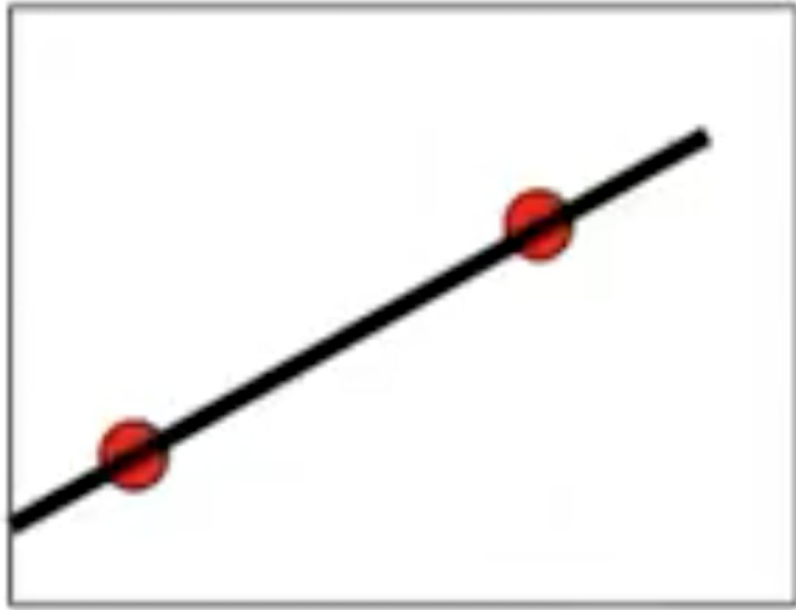
- Our model is too complex (sees patterns not in data)
- High variance, but low bias
- Use too many features, our model parameters are too big
- We are able to perfectly predict training data, but not test data



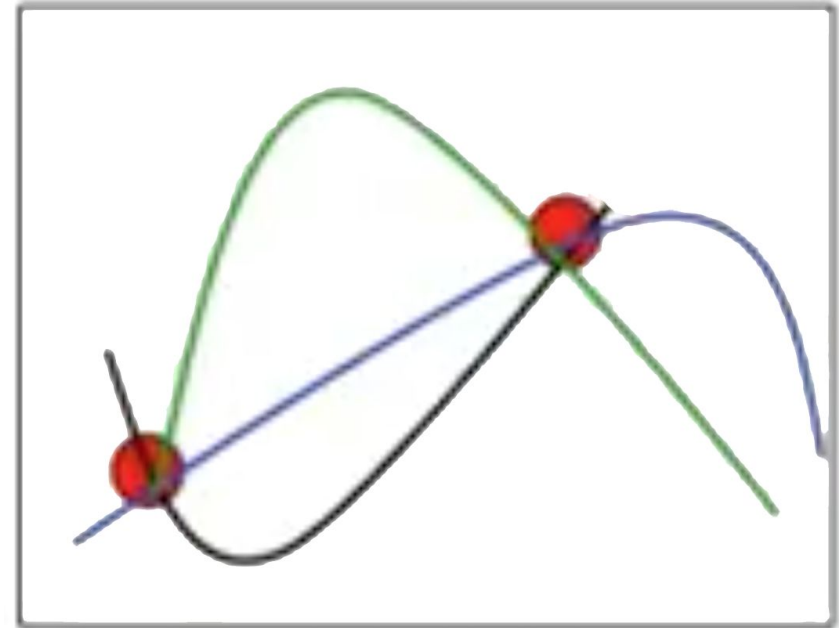


# Overfitting / High variance

**Examples of overfitting, two data points: Choose simplest model!**



**Best (Simple) Model!**

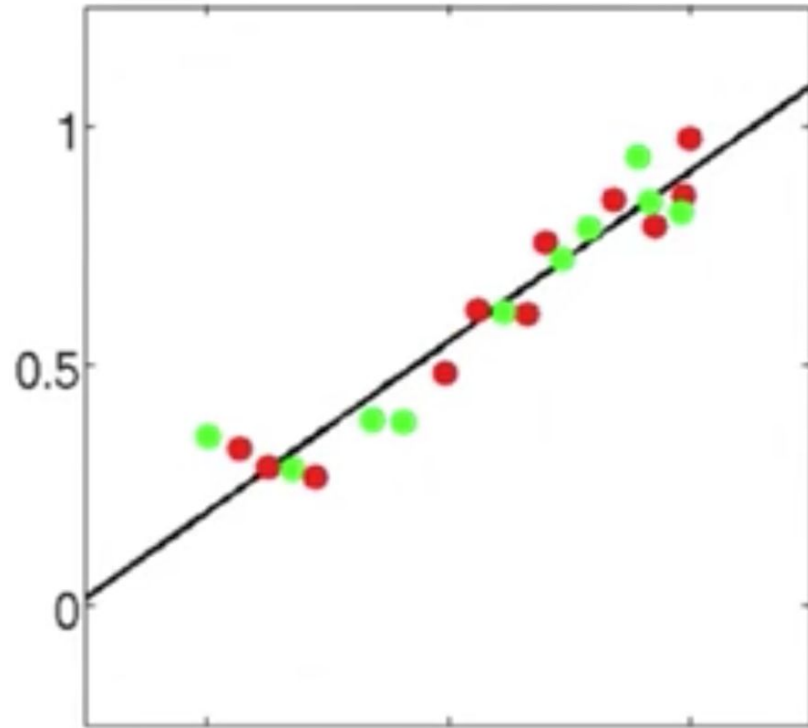


**Worse (too complex) models!**

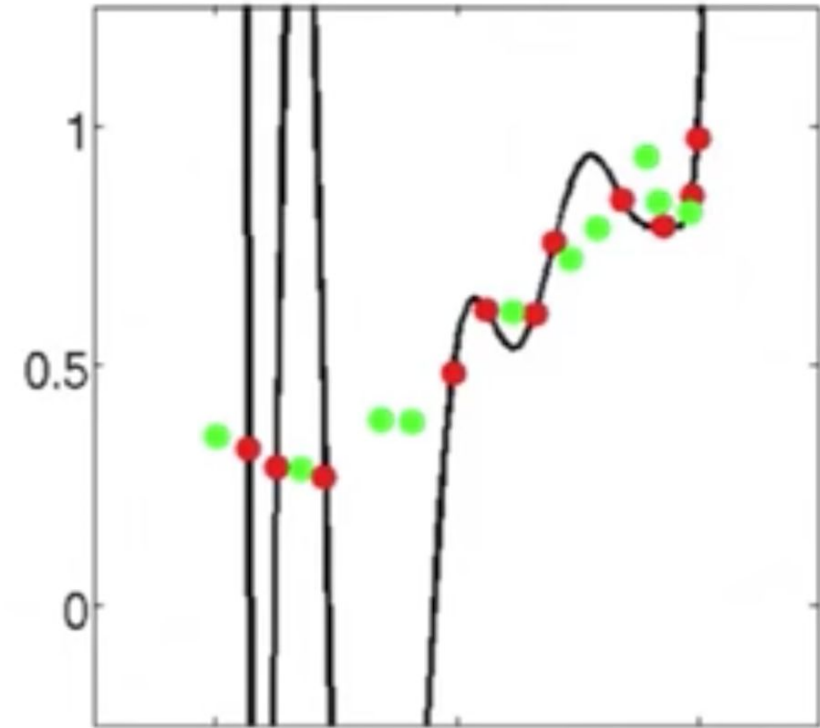


# Overfitting / High variance

Examples of overfitting: Red = Training data    Test data = green



Best (Simple) Model!



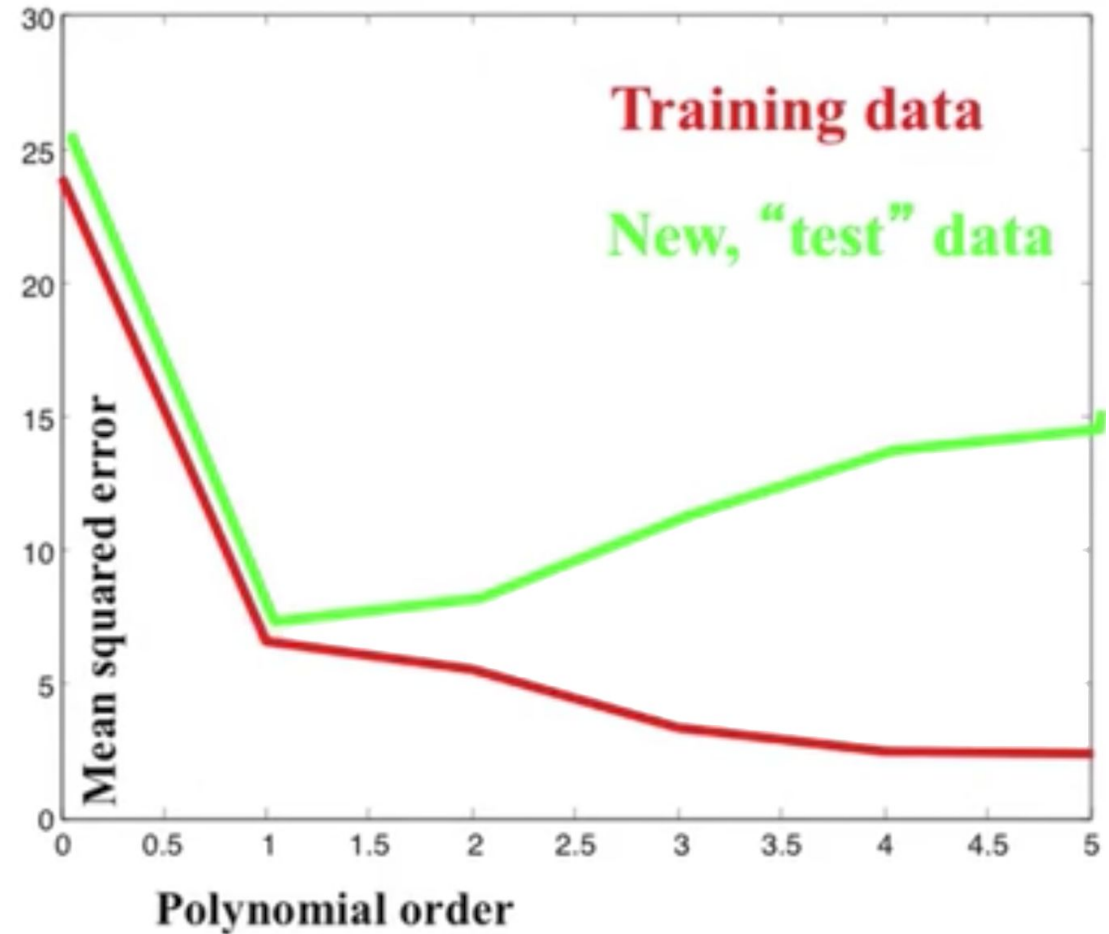
Too high variance, bad model!



# Check if you are overfitting

Plot MSE against model complexity measure

When MSE starts to increase for the test data, that is the point where we are starting to overfit.



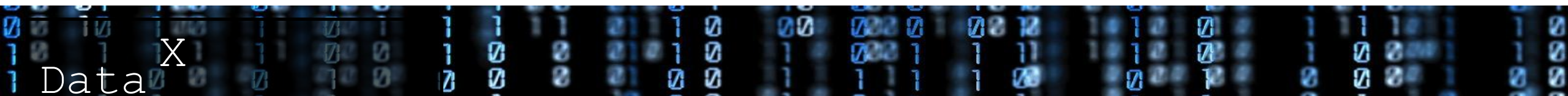
# Overfitting data / High variance

## How to check if you're overfitting:

- Use cross-validation predict on test data
- Plot MSE against model complexity for training and test set, see how the error changes.

## How to prevent overfitting:

- Exclude some features from the model (feature engineering)
- Introduce a stopping criteria in the optimization algorithm
- ***Regularization!***



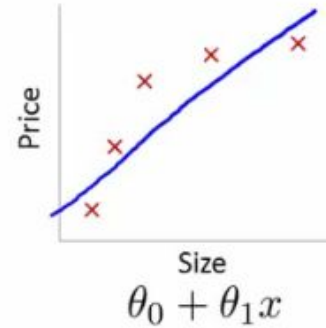


Bias (underfitting) - Variance (Overfitting) tradeoff

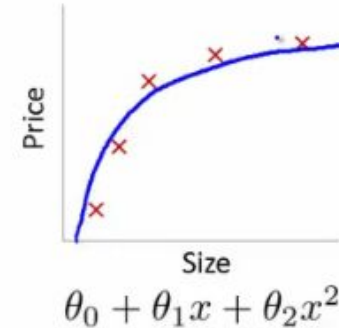


# Bias-Variance Tradeoff:

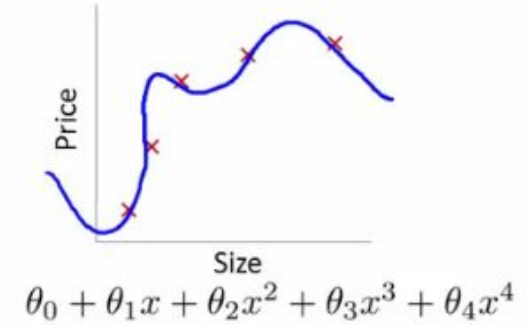
## REGRESSION CASE:



High bias  
(underfit)

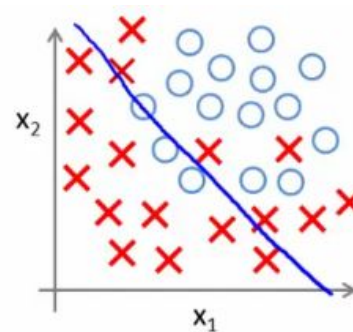


“Just right”



High variance  
(overfit)

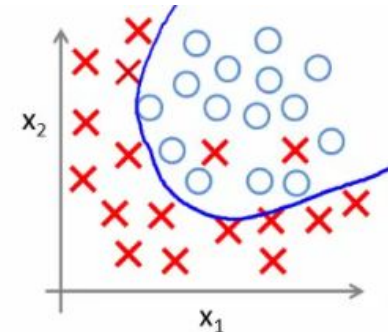
## CLASSIFICATION CASE:



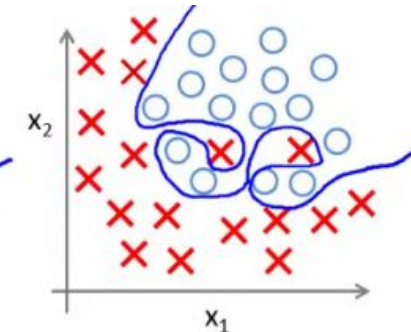
$$h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

( $g$  = sigmoid function)

UNDERFITTING  
(high bias)



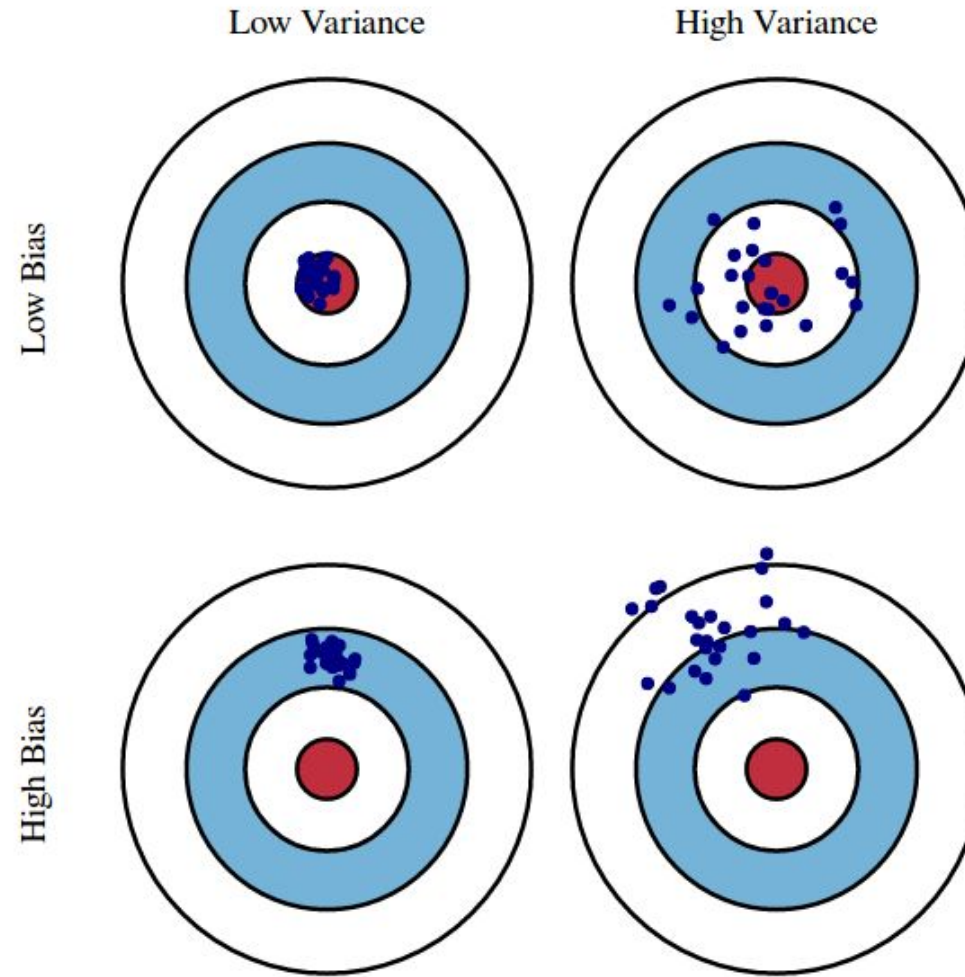
$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$$



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots)$$

OVERFITTING  
(high variance)

# Bias-Variance Tradeoff:



# Regularization





# Regularization

## Why:

Avoid overfitting (and perform feature selection, LASSO only)

## How:

Increase bias by penalizing the model for large parameter values.

Add a multiple of an L1 (LASSO) or an L2 (Ridge) norm of the model parameters  $\theta$  to the cost function

## New cost function:

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 + \lambda \cdot \|\theta\|_p^p$$

- $\lambda$  is the regularization parameter, basically a tuning parameter
- $\|\theta\|_p^p$  is the  $p$ :th matrix norm on the parameters

# Regularization (increase error if we have too many or too big parameters)

**Non-regularized COST FUNCTION:**  $J_{old}(\theta) = MSE(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$

**RIDGE REGRESSION (L2 NORM):**  $J(\theta) = MSE(\theta) + \lambda \sum_{j=1}^n \theta_j^2$

**LASSO (L1 NORM):**  $J(\theta) = MSE(\theta) + \lambda \sum_{j=1}^n |\theta_j|$

**Find optimal regularization term  $\lambda$  by tuning it and using Cross-validation:**

- Divide your training data,
- Train your model for a fixed value of  $\lambda$  and test it on the remaining subsets
- Repeat this procedure while varying  $\lambda$ .

Then you select the best  $\lambda$  that minimizes your cost function.



# Regularization (increase error if we have too many parameters)

The optimal estimates of the model parameters,  $\beta$ , could be denoted as shown below.

This shows us the difference between Ridge and Lasso Regression

$$\hat{\beta}^{\text{lasso}} = \operatorname{argmin} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1$$

$$\hat{\beta}^{\text{ridge}} = \operatorname{argmin} \|y - X\beta\|_2^2 + \lambda \|\beta\|_2^2$$

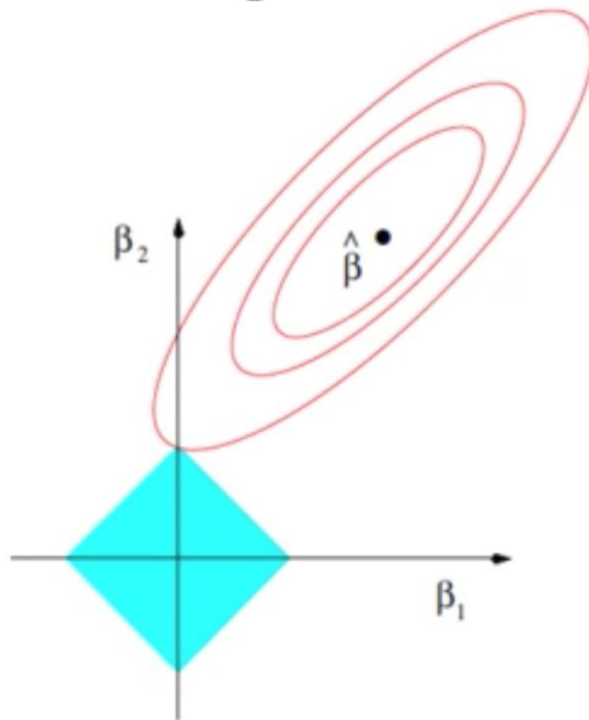
$$\hat{\beta}^{\text{lasso}} = \operatorname{argmin} \|y - X\beta\|_2^2 \quad \text{subject to } \|\beta\|_1 \leq t$$

$$\hat{\beta}^{\text{ridge}} = \operatorname{argmin} \|y - X\beta\|_2^2 \quad \text{subject to } \|\beta\|_2^2 \leq t$$

# Regularization (increase error if we have too many parameters)

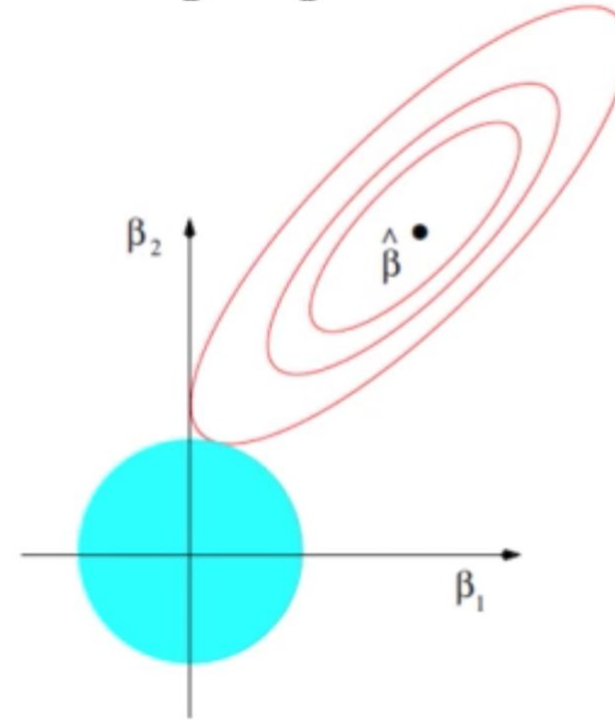
We can visualize the difference between Ridge and Lasso Regression for two parameters. Note, there is a trade-off between the Least Square error and the size of the parameters (which are constrained, to the blue areas).

Lasso Regression



$$\hat{\beta}^{\text{lasso}} = \operatorname{argmin} \|y - X\beta\|_2^2 \quad \text{subject to } \|\beta\|_1 \leq t$$

Ridge Regression



$$\hat{\beta}^{\text{ridge}} = \operatorname{argmin} \|y - X\beta\|_2^2 \quad \text{subject to } \|\beta\|_2 \leq t$$

$$t \propto \frac{1}{\lambda}$$

## Example Code: Regularization





**End**



# References

- The material presented in this lecture references lecture material draws on the materials the following courses:
- Derek Kane's Data Science Tutorials:  
<https://www.youtube.com/channel/UC33qFpcu7eHFtpZ6dp3FFXw>
- Stanford – CS229 (Machine Learning) & Andrew Ng's Machine Learning at Coursera: <http://cs229.stanford.edu/> & <https://www.coursera.org/learn/machine-learning>
- Professor Alexander Ihler, UC Davis: <youtube.com/watch?v=sO4ZirJh9ds>

