

# Machine Learning and Data Analytics

## ME 5013- Fall 2019

### Lectures 13

- Regularization



The University of Texas at San Antonio™

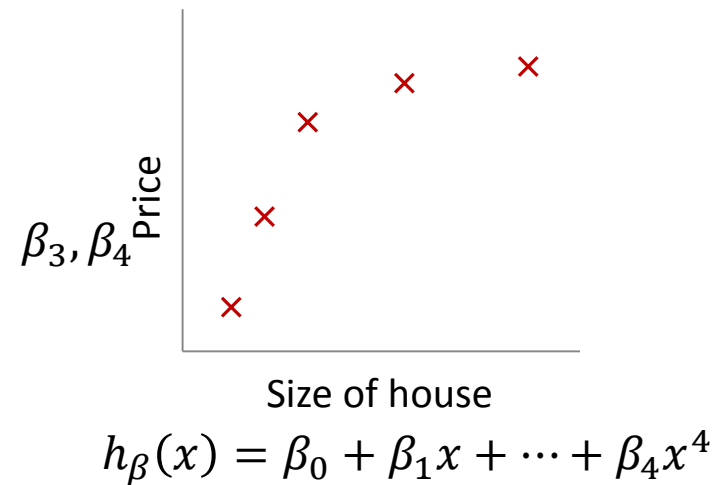
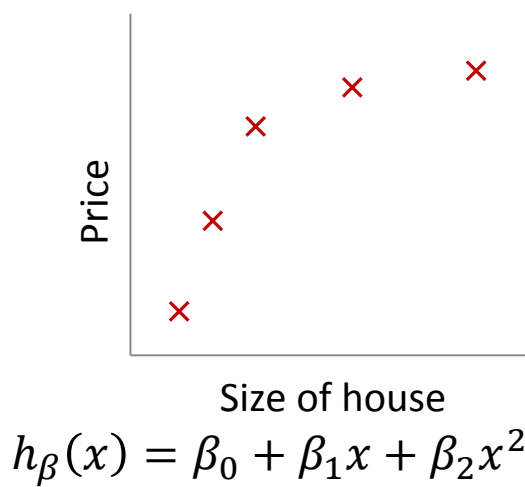
Adel Alaeddini, PhD

Associate Professor of Mechanical Engineering

Advanced Data Engineering Lab

[adel.alaeddini@utsa.edu](mailto:adel.alaeddini@utsa.edu)

Keep all the features, but reduce magnitude/values of parameters  
 Works well when we have a lot of features, each of which contributes a bit to predicting



Suppose we penalize and make  $\beta_3, \beta_4$  really small.

$$\min_{\beta} \frac{1}{2n} \sum_{i=1}^n \left( y^{(i)} - h_{\beta}(x^{(i)}) \right)^2$$

Small values for parameters  $\beta_0, \beta_1, \dots, \beta_p$

- “Simpler” hypothesis
- Less prone to overfitting

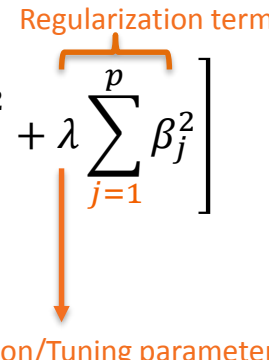
Housing:

- Features:  $x_1, x_2, \dots, x_{100}$
- Parameters:  $\beta_0, \beta_1, \dots, \beta_{100}$

## Regression Problem

$$J(\beta) = \frac{1}{2n} \sum_{i=1}^n (y^{(i)} - h_{\beta}(x^{(i)}))^2$$

$$J(\beta) = \frac{1}{2n} \left[ \sum_{i=1}^n (y^{(i)} - h_{\beta}(x^{(i)}))^2 + \underbrace{\lambda \sum_{j=1}^p \beta_j^2}_{\text{Regularization term}} \right]$$


  
 Regularization/Tuning parameter

$$\min_{\beta} J(\beta)$$

## Multiple Linear Regression

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

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

*The tuning parameter serves to control the relative impact of the two terms on the regression coefficient estimates.*

**Hypothesis:**  $h_{\beta}(x) = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p$

**Cost Function:**  $J(\beta) = \frac{1}{2n} \left[ \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_1^{(i)} - \dots - \beta_p x_p^{(i)})^2 + \lambda \sum_{j=1}^p \beta_j^2 \right]$

**Parameters:**  $\beta_0, \beta_1, \dots, \beta_p$

$\min_{\beta} J(\beta)$

**Gradient descent:**

Repeat until convergence{

$$\beta_0 := \beta_0 + \alpha \frac{1}{n} \left[ \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_1^{(i)} - \dots - \beta_p x_p^{(i)}) \right]$$

$$\beta_1 := \beta_1 + \alpha \frac{1}{n} \left[ \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_1^{(i)} - \dots - \beta_p x_p^{(i)}) x_1^{(i)} - \lambda \beta_1 \right]$$

...

$$\beta_p := \beta_p + \alpha \frac{1}{n} \left[ \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_1^{(i)} - \dots - \beta_p x_p^{(i)}) x_p^{(i)} - \lambda \beta_p \right]$$

}

**NOTE:**

$$\beta_p := \beta_p \left( 1 - \alpha \frac{\lambda}{n} \right) + \alpha \frac{1}{n} \left[ \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_1^{(i)} - \dots - \beta_p x_p^{(i)}) x_p^{(i)} \right]$$

$$y = X\beta + \epsilon$$

The diagram illustrates the components of the normal equation  $y = X\beta + \epsilon$ . Arrows point from the equation to the definitions of each term:

- $y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$
- $X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$
- $\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p+1} \end{bmatrix}$
- $\epsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$

Multiple linear regression:  $\beta = (X'X)^{-1}X'y$

Regularized multiple linear regression:  $\beta = (X'X + \lambda I)^{-1}X'y$

where  $I = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}_{(p+1) \times (p+1)}$

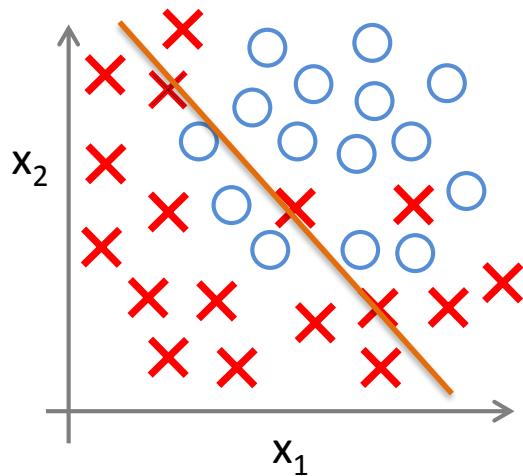
Suppose  $n \leq p$

(#examples) (#features)

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

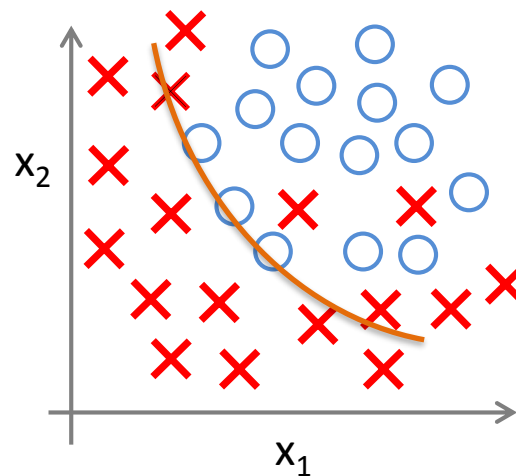
If  $\lambda > 0$ ,

$$\beta = \left( X^T X + \lambda \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \right)^{-1} X^T y$$

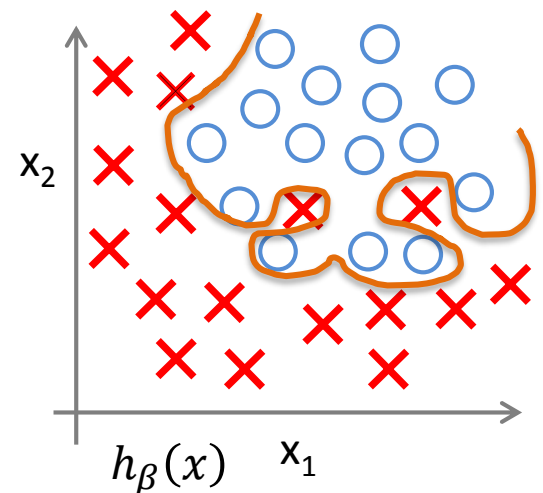


$$h_{\beta}(x) = g(\beta_0 + \beta_1 x_1 + \beta_2 x_2)$$

(  $g$  = sigmoid function)



$$h_{\beta}(x) = g(\beta_0 + \beta_1 x_1 + \beta_2 x_2)$$



$$h_{\beta}(x) = g(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2)$$

$$J(\beta) = -\frac{1}{n} \left[ \sum_{i=1}^n y^{(i)} \log h_{\beta}(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_{\beta}(x^{(i)})) \right] + \frac{\lambda}{n} \sum_{j=1}^p \beta_j^2$$

Want  $\min_{\beta} J(\beta)$  :

Repeat {

$$\beta_j := \beta_j - \alpha \frac{\partial}{\partial \beta_j} J(\beta_0, \beta_1) \quad (\text{for } j = 0 \text{ and } j = 1)$$

}

(simultaneously update all parameters )

Closed form formula of the gradients

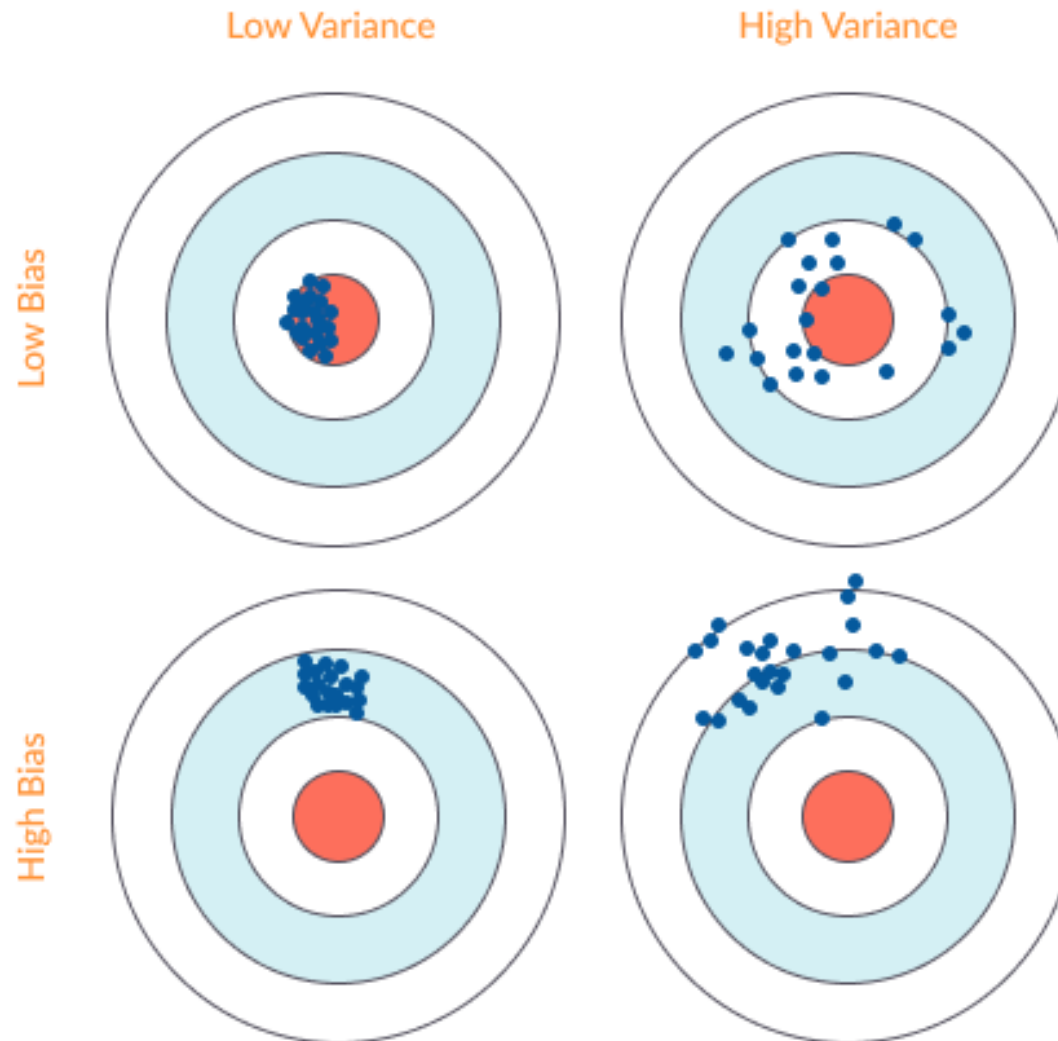
$$\beta_0 := \beta_0 + \alpha \frac{1}{n} \sum_{i=1}^n (y^{(i)} - h_{\beta}(x^{(i)})) x^{(i)}$$

$$\beta_{j \neq 0} := \beta_j + \alpha \frac{1}{n} \sum_{i=1}^n (y^{(i)} - h_{\beta}(x^{(i)})) x^{(i)} - \frac{\lambda}{n} \beta_j$$

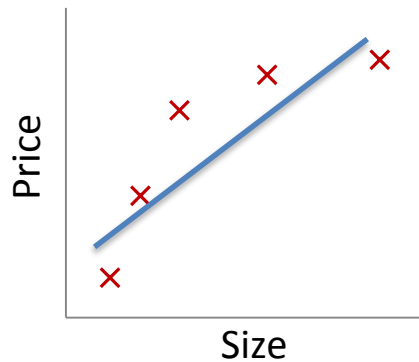
$$h_{\beta}(x) = \frac{1}{1 + e^{-x\beta}}$$

$$x = [x_0 = 1 \quad x_1 \quad \dots]$$



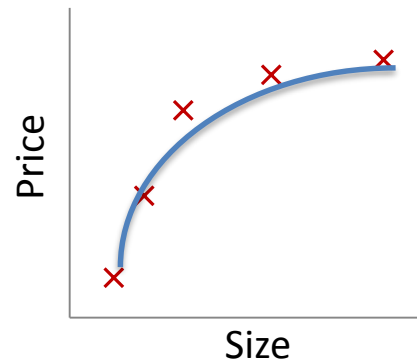


## Bias/variance



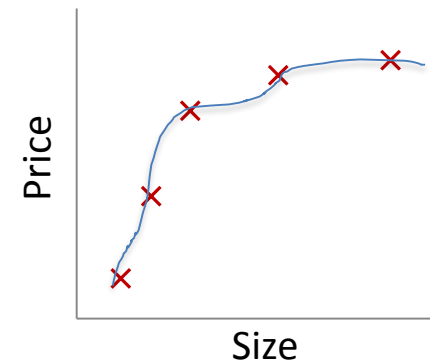
$$h_{\beta}(x) = \beta_0 + \beta_1 x$$

High bias  
(underfit)



$$h_{\beta}(x) = \beta_0 + \beta_1 x + \beta_2 x^2$$

“Just right”

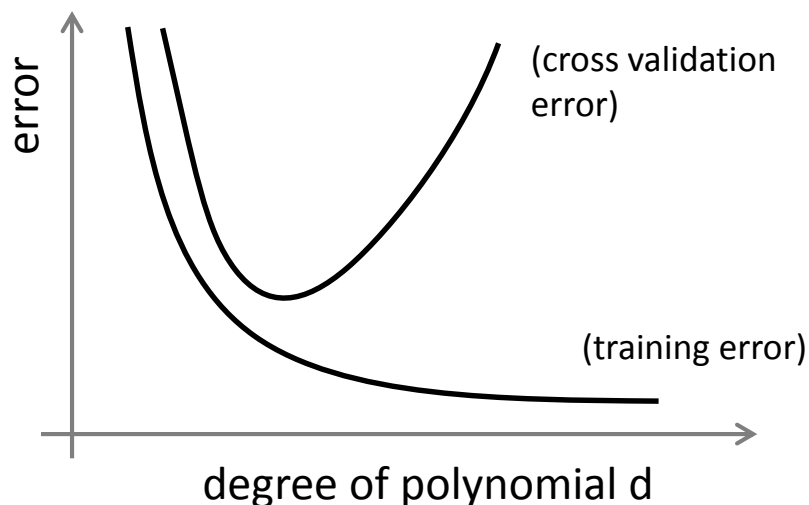


$$h_{\beta}(x) = \beta_0 + \beta_1 x + \dots + \beta_4 x^4$$

High variance  
(overfit)

*Fail to generalize to new examples*

Suppose your learning algorithm is performing less well than you were hoping. Is it a bias problem or a variance problem?



Bias (underfit):

$$J_{train}(\beta) \text{ high}$$

$$J_{train}(\beta) \approx J_{cv}(\beta)$$

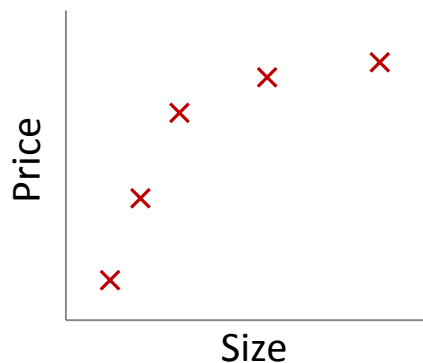
Variance (overfit):

$$J_{train}(\beta) \text{ low}$$

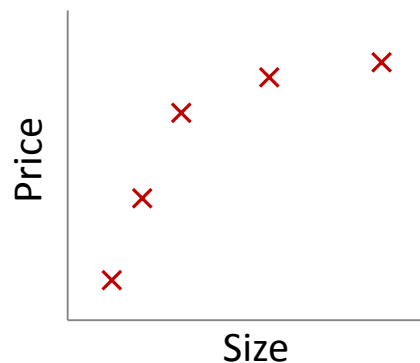
$$J_{train}(\beta) \ll J_{cv}(\beta)$$

## Linear regression with regularization

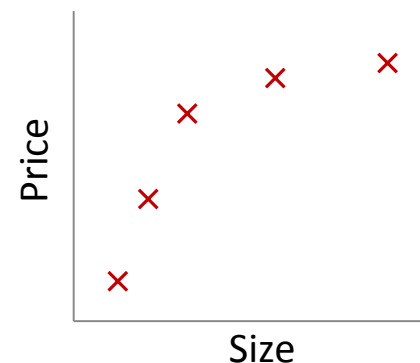
$$J(\beta) = \frac{1}{2n} \left[ \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_1^{(i)} - \dots - \beta_p x_p^{(i)})^2 + \lambda \sum_{j=1}^p \beta_j^2 \right]$$



Large  $\lambda$   
High bias (underfit)



Intermediate  $\lambda$   
"Just right"



Small  $\lambda$   
High variance (overfit)

$$h_{\beta}(x) = \beta_0 + \beta_1 x + \cdots + \beta_4 x^4$$

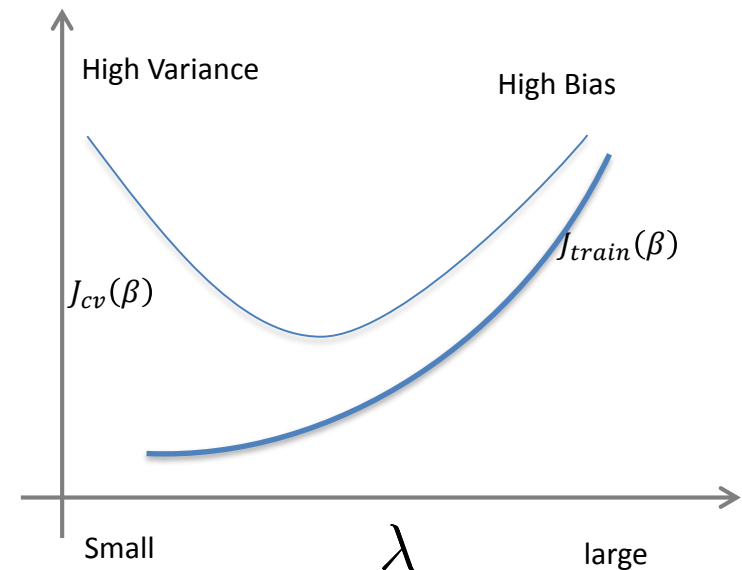
$$J(\beta) = \frac{1}{2n} \left[ \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_1^{(i)} - \cdots - \beta_p x_p^{(i)})^2 + \lambda \sum_{j=1}^p \beta_j^2 \right]$$

$$J_{train}(\beta) = \frac{1}{2n_{train}} \sum_{i=1}^{n_{train}} (y_i - \beta_0 - \beta_1 x_1^{(i)} - \cdots - \beta_p x_p^{(i)})^2$$

$$J_{cv}(\beta) = \frac{1}{2n_{cv}} \sum_{i=1}^{n_{cv}} (y_i - \beta_0 - \beta_1 x_1^{(i)} - \cdots - \beta_p x_p^{(i)})^2$$

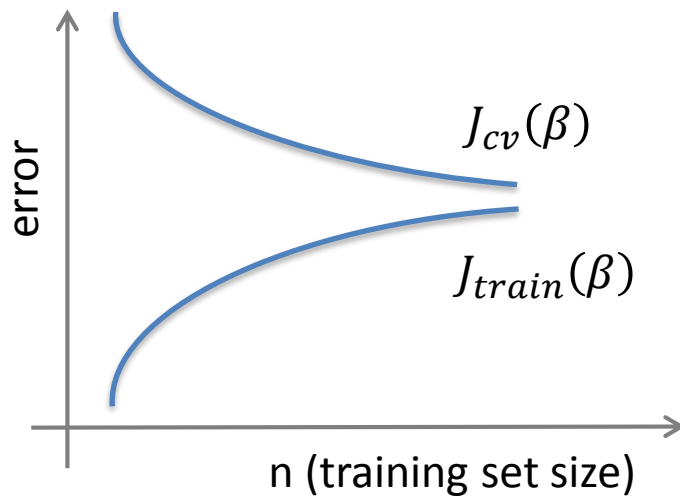
$$J_{test}(\beta) = \frac{1}{2n_{test}} \sum_{i=1}^{n_{test}} (y_i - \beta_0 - \beta_1 x_1^{(i)} - \cdots - \beta_p x_p^{(i)})^2$$

1. Try  $\lambda = 0$
2. Try  $\lambda = 0.01$
3. Try  $\lambda = 0.02$
4. Try  $\lambda = 0.04$
5. Try  $\lambda = 0.08$
- $\vdots$
12. Try  $\lambda = 10$

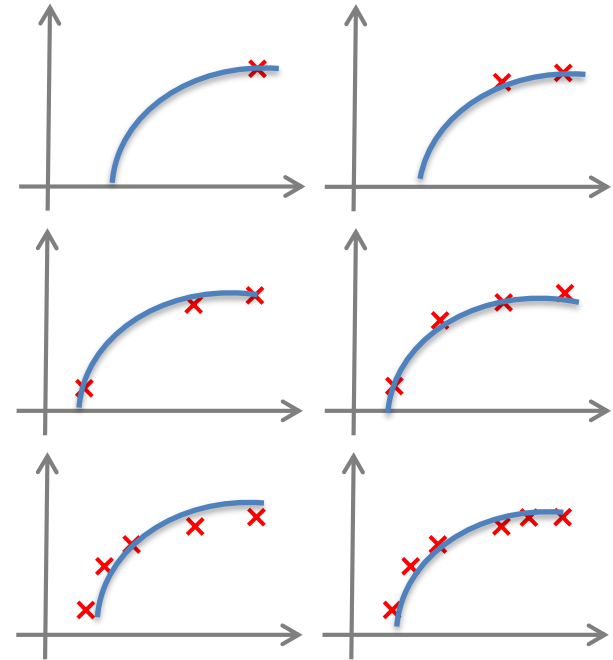


$$J_{train}(\beta) = \frac{1}{2n_{train}} \sum_{i=1}^{n_{train}} (y_i - \beta_0 - \beta_1 x_1^{(i)} - \dots - \beta_2 x_2^{(i)})^2$$

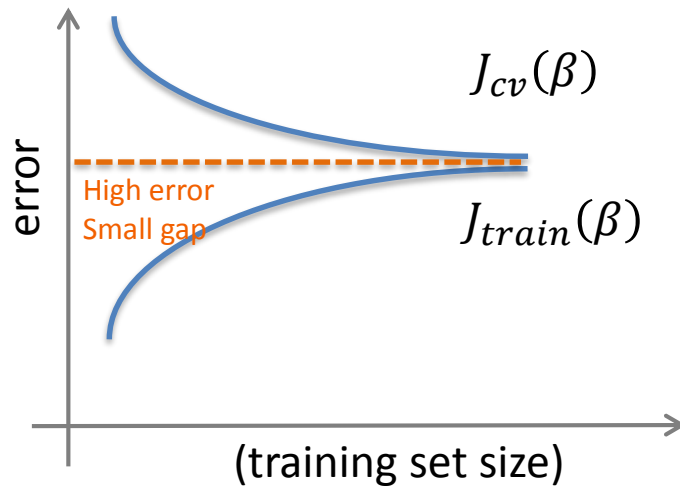
$$J_{cv}(\beta) = \frac{1}{2n_{cv}} \sum_{i=1}^{n_{cv}} (y_i - \beta_0 - \beta_1 x_1^{(i)} - \dots - \beta_2 x_2^{(i)})^2$$



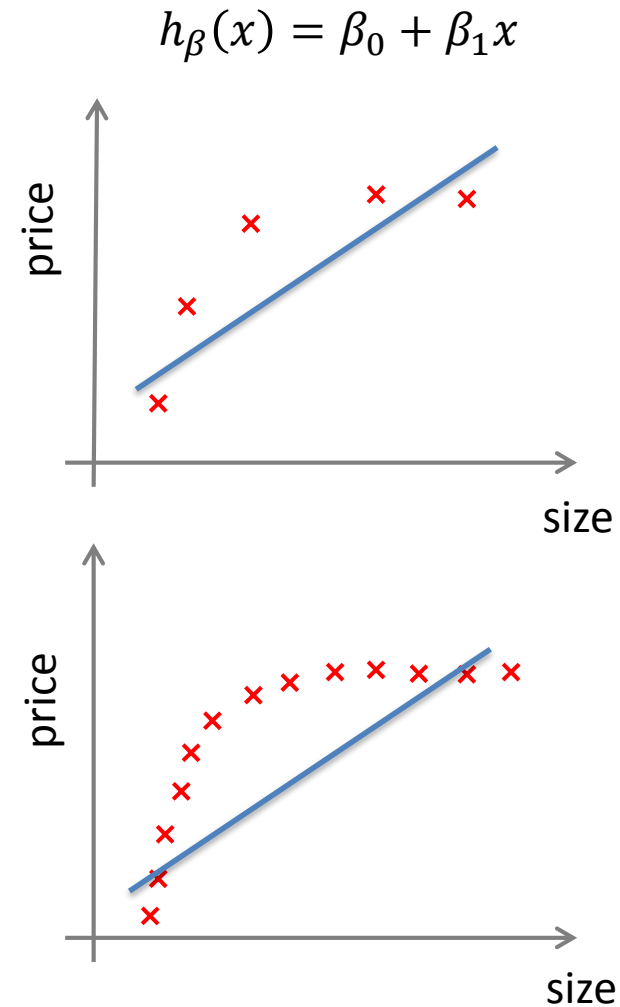
$$h_{\beta}(x) = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p$$

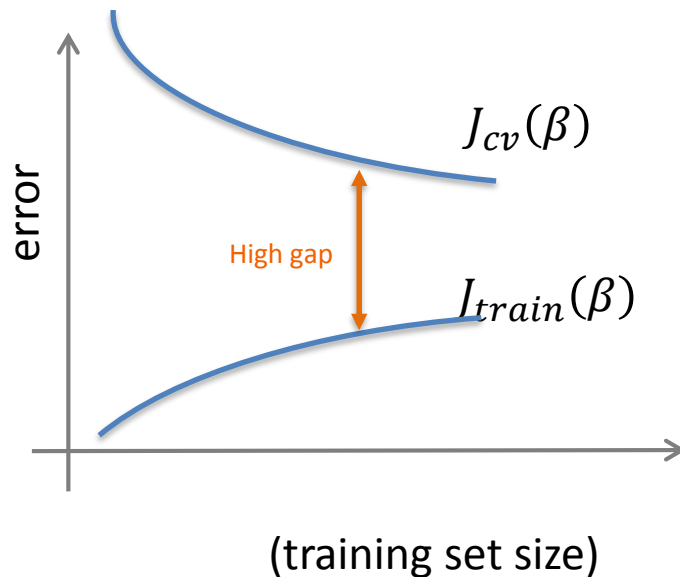


## High bias

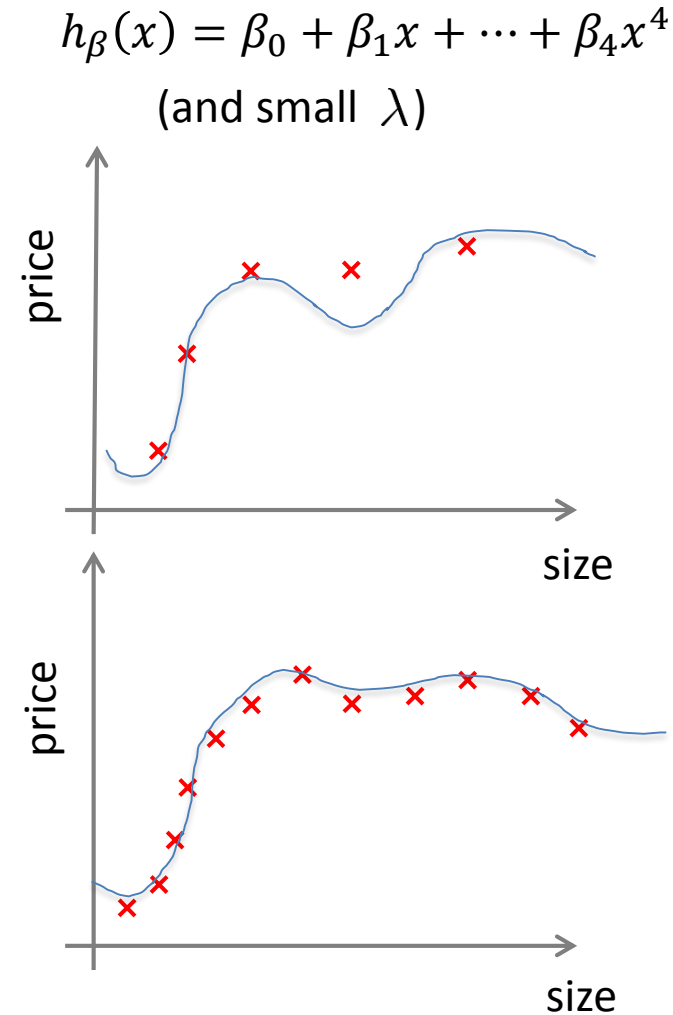


If a learning algorithm is suffering from high bias, getting more training data will not (by itself) help much.





If a learning algorithm is suffering from high variance, getting more training data is likely to help.





**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 ( $x_1^2, x_2^2, x_1x_2$ , etc)
- Try decreasing  $\lambda$
- Try increasing  $\lambda$