Regularization

Machine Learning

(Largely based on slides from Andrew Ng)

Prof. Sandra Avila

Institute of Computing (IC/Unicamp)

MC886, August 26, 2019

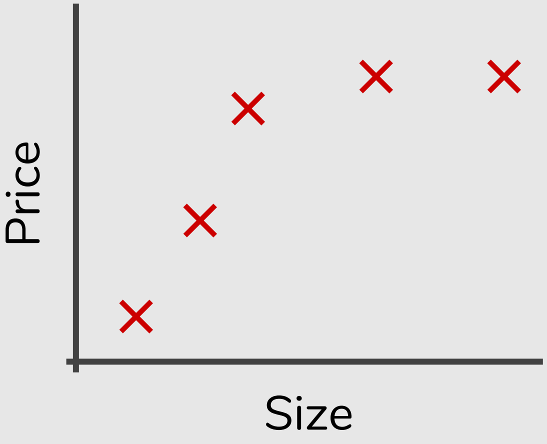
Today's Agenda

- Regularization
  - The Problem of Overfitting
  - Diagnosing Bias vs. Variance
  - Cost Function
  - Regularized Linear Regression
  - Regularized Logistic Regression

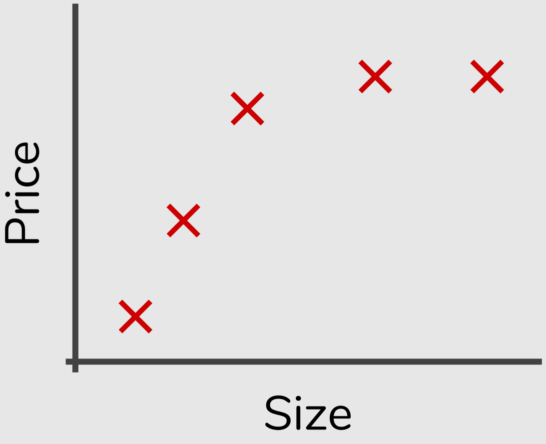
The Problem of Overfitting


THE BEST WAY TO
EXPLAIN OVERFITTING
imgflip.com
ADDTEXT.COM

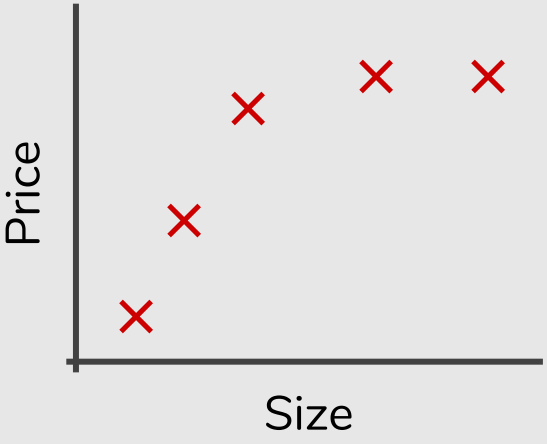
Example: Linear Regression



$$\theta_0 + \theta_1 x$$

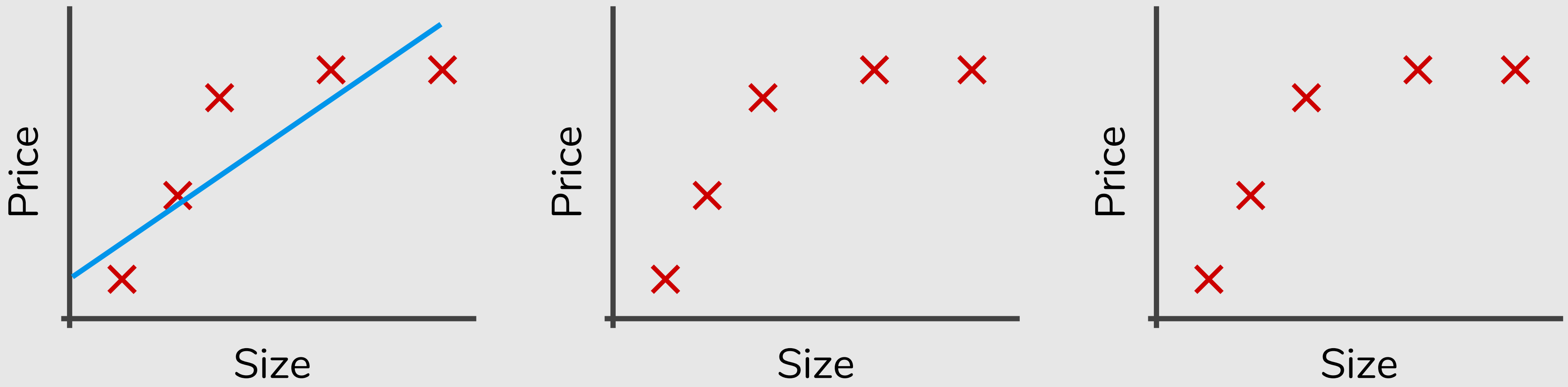


$$\theta_0 + \theta_1 x + \theta_2 x^2$$



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Example: Linear Regression

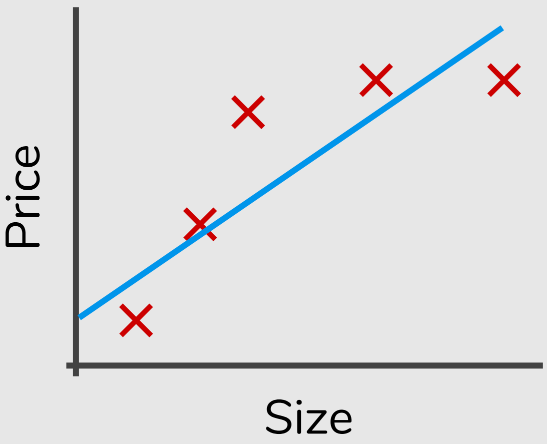


$\theta_0 + \theta_1 x$

$\theta_0 + \theta_1 x + \theta_2 x^2$

$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

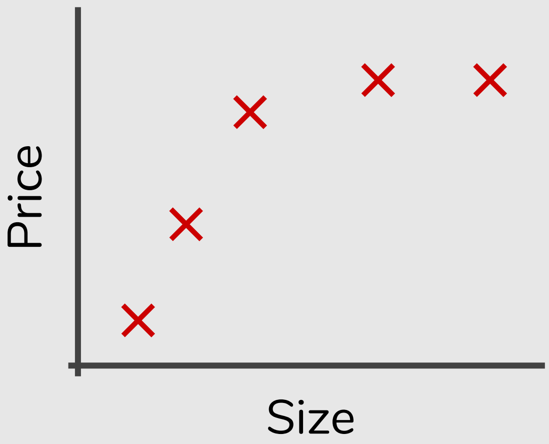
Example: Linear Regression



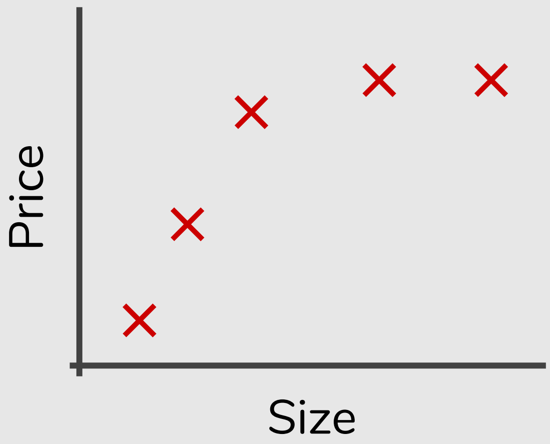
$$\theta_0 + \theta_1 x$$

Underfitting

High bias

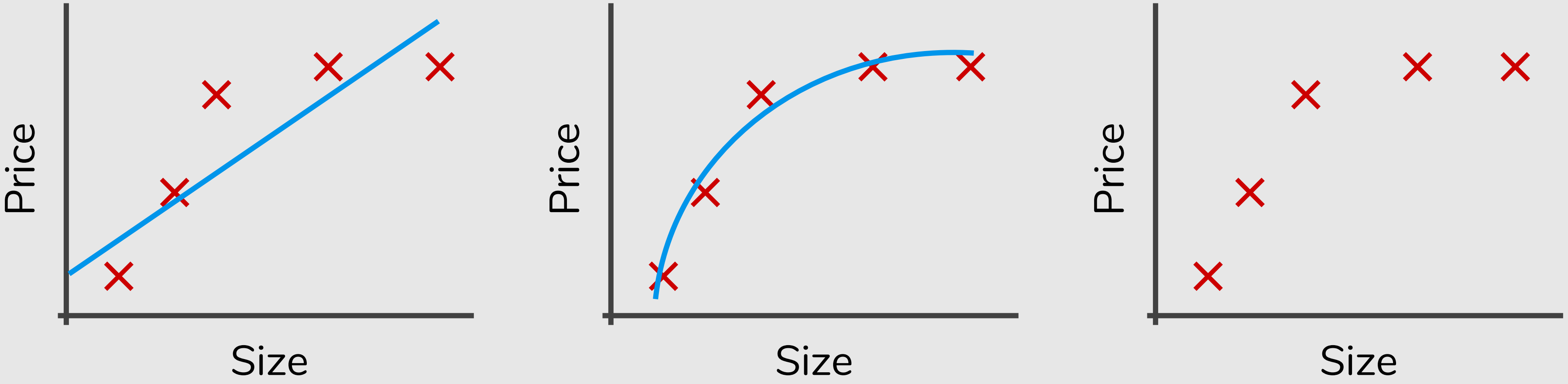


$$\theta_0 + \theta_1 x + \theta_2 x^2$$



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Example: Linear Regression



$\theta_0 + \theta_1 x$

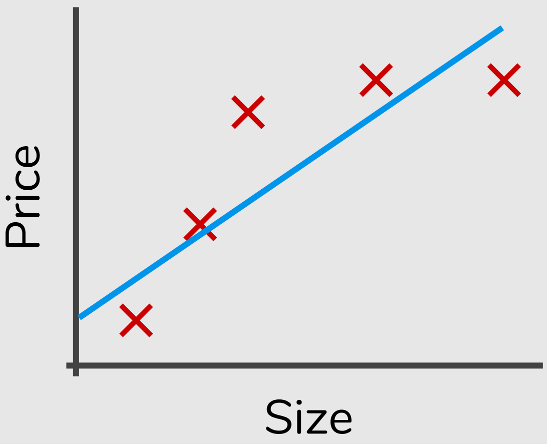
$\theta_0 + \theta_1 x + \theta_2 x^2$

$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

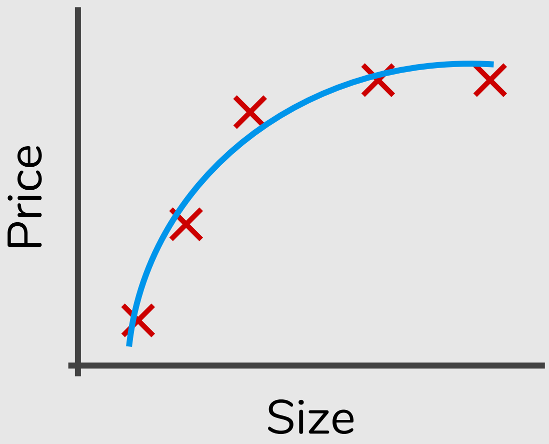
Underfitting

High bias

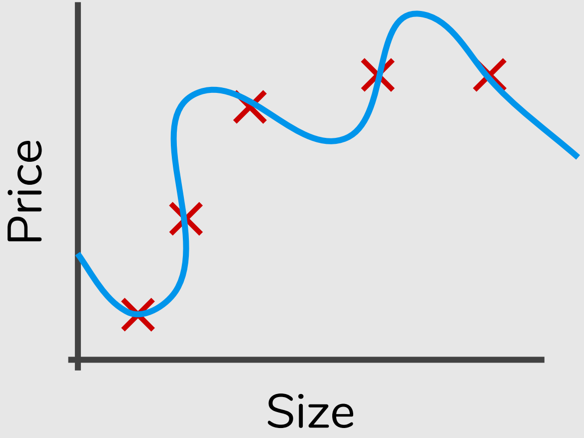
Example: Linear Regression



$\theta_0 + \theta_1 x$



$\theta_0 + \theta_1 x + \theta_2 x^2$

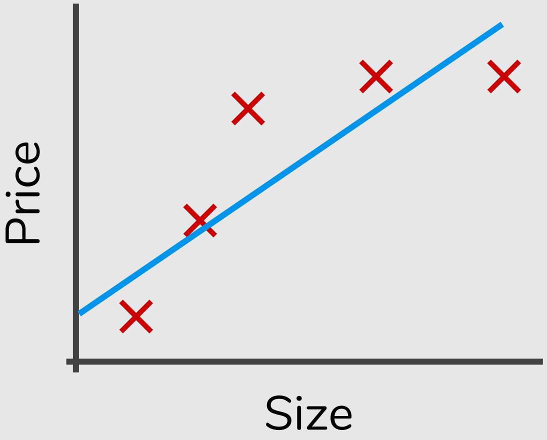


$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Underfitting

High bias

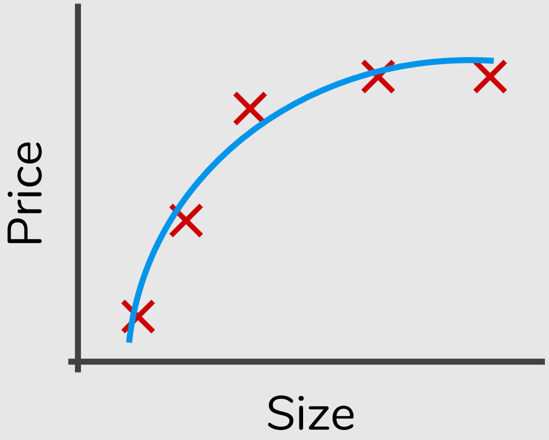
Example: Linear Regression



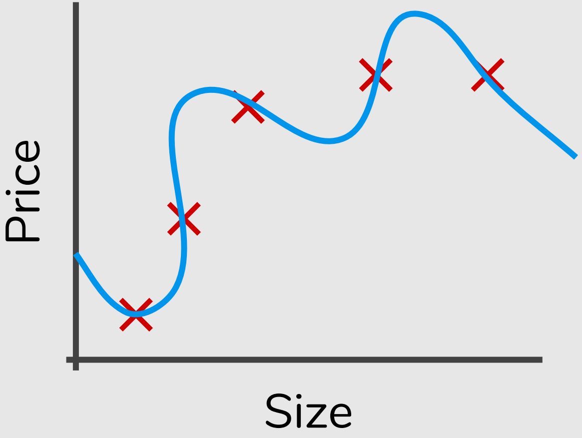
$$\theta_0 + \theta_1 x$$

Underfitting

High bias



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

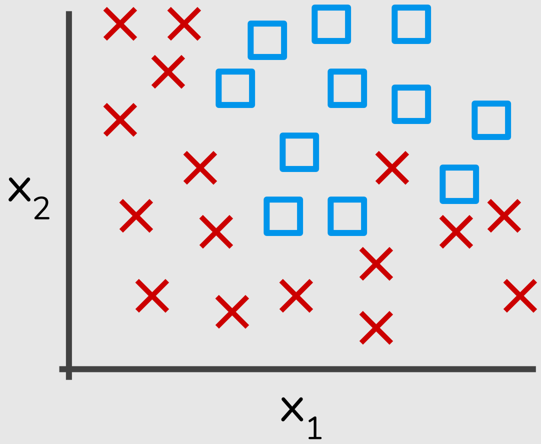


$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

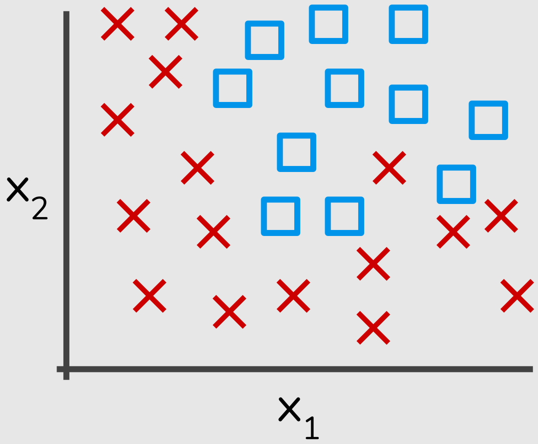
Overfitting

High variance

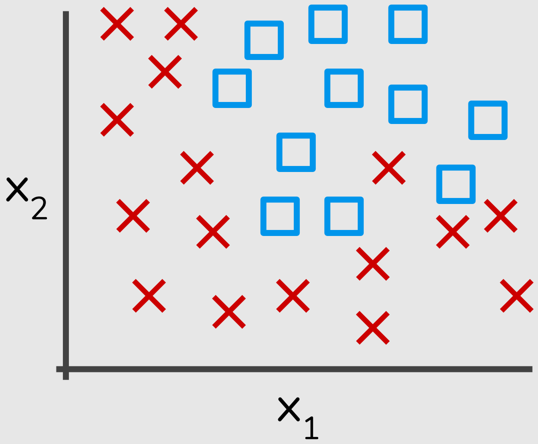
Example: Logistic Regression



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

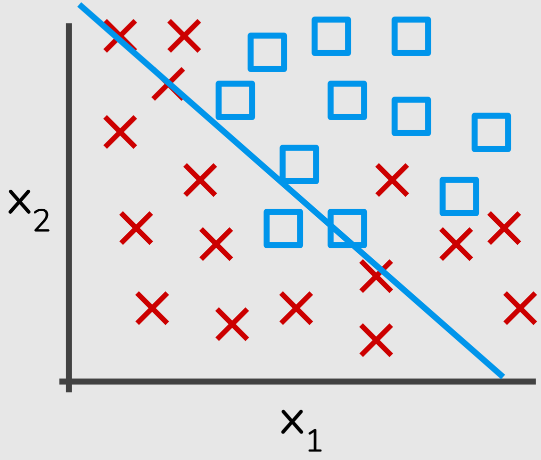


$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \\ + \theta_4 x_2^2 + \theta_5 x_1 x_2)$$



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

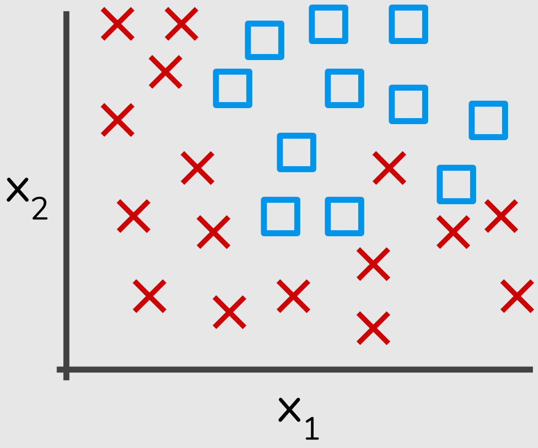
Example: Logistic Regression



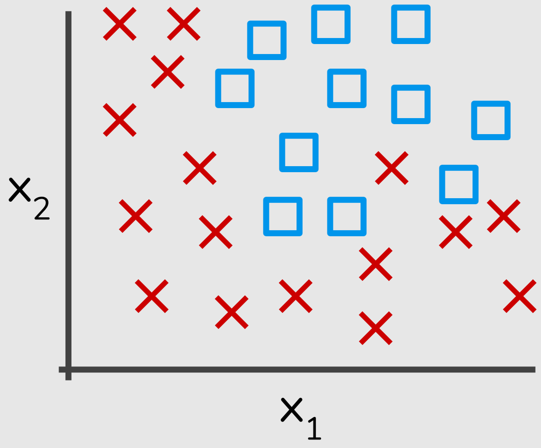
$$g(\theta_0+\theta_1 x_1+\theta_2 x_2)$$

Underfitting

High bias

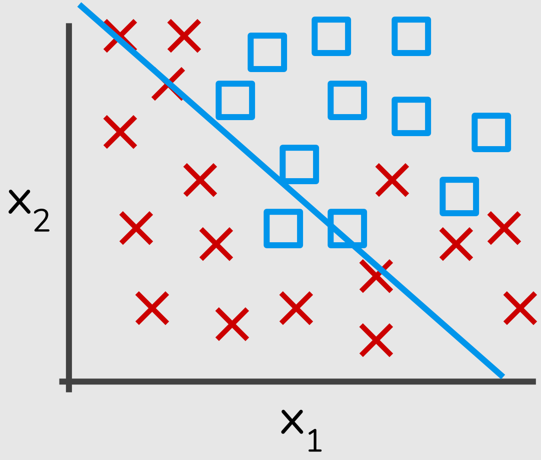


$$g(\theta_0+\theta_1 x_1+\theta_2 x_2+\theta_3 x_1^2+ + \theta_4 x_2^2+\theta_5 x_1 x_2)$$



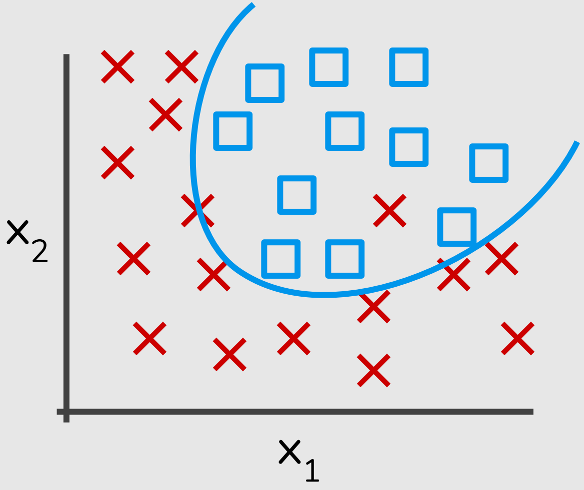
$$g(\theta_0+\theta_1 x_1+\theta_2 x_1^2 + + \theta_3 x_1^2 x_2+\theta_4 x_1^2 x_2^2 + + \theta_5 x_1^2 x_2^3+\ldots)$$

Example: Logistic Regression

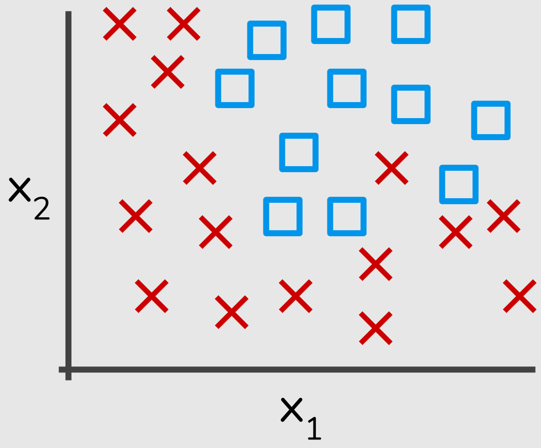


$g(\theta_0+\theta_1 x_1+\theta_2 x_2)$

Underfitting
High bias

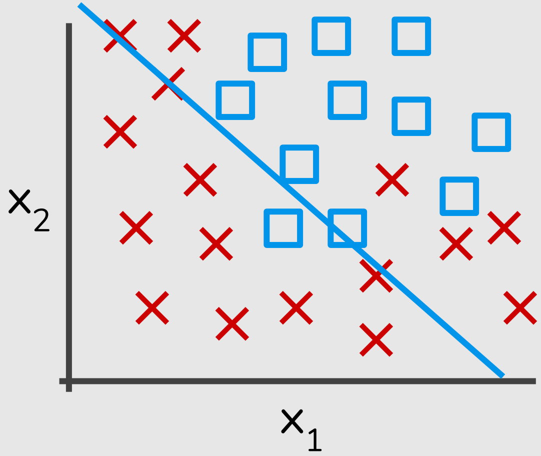


$g(\theta_0+\theta_1 x_1+\theta_2 x_2+\theta_3 x_1^2+$
$+\theta_4 x_2^2+\theta_5 x_1 x_2)$



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

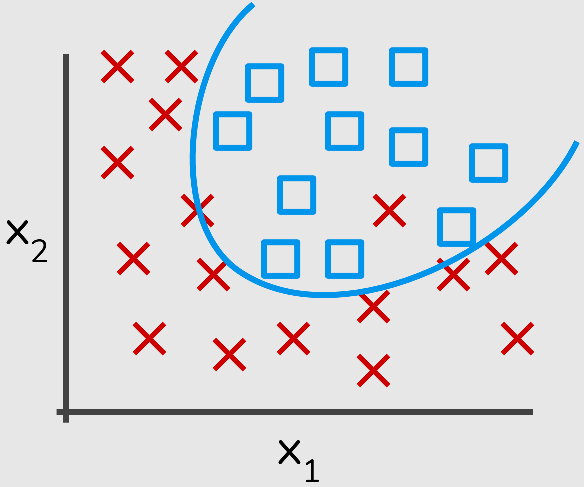
Example: Logistic Regression



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

Underfitting

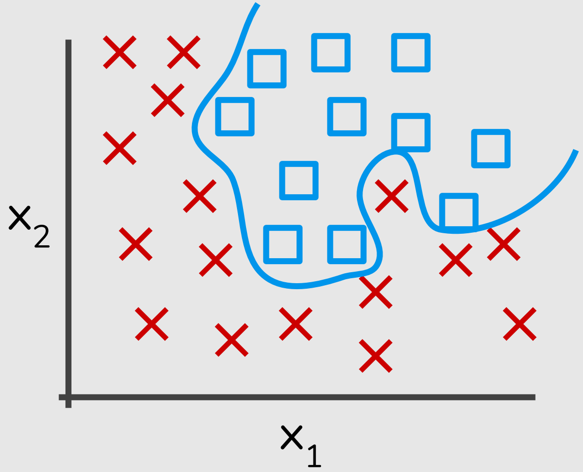
High bias



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

Overfitting

High variance



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

The Bias/Variance Tradeoff

The Bias/Variance Tradeoff

A model's generalization error can be expressed as the sum of **three** very different errors:

- Bias
- Variance
- Irreducible error

The Bias/Variance Tradeoff

A model's generalization error can be expressed as the sum of **three** very different errors:

- **Bias**
  - Due to wrong assumptions, such as assuming that the data is linear when it is actually quadratic.
  - A **high-bias** model is most likely to **underfit** the training data.
- Variance
- Irreducible error

The Bias/Variance Tradeoff

A model's generalization error can be expressed as the sum of **three** very different errors:

- Bias
- **Variance**
  - Due to the model's excessive sensitivity to small variations in the training data.
  - A model with many degrees of freedom is likely to have **high variance**, and thus to **overfit** the training data.
- Irreducible error

The Bias/Variance Tradeoff

A model's generalization error can be expressed as the sum of **three** very different errors:

- Bias
- Variance
- **Irreducible error**
  - Due to the noisiness of the data itself.
  - The only way to reduce this part of the error is to clean up the data.

The Bias/Variance Tradeoff

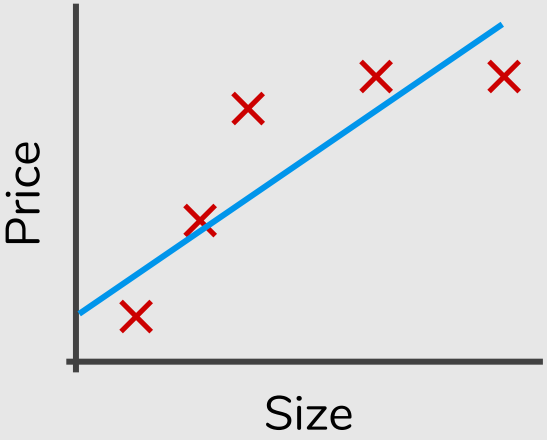
Increasing a model's complexity will typically increase its variance and reduce its bias.

Reducing a model's complexity increases its bias and reduces its variance.

This is why it is called a **tradeoff**.

Diagnosing
Bias vs. Variance

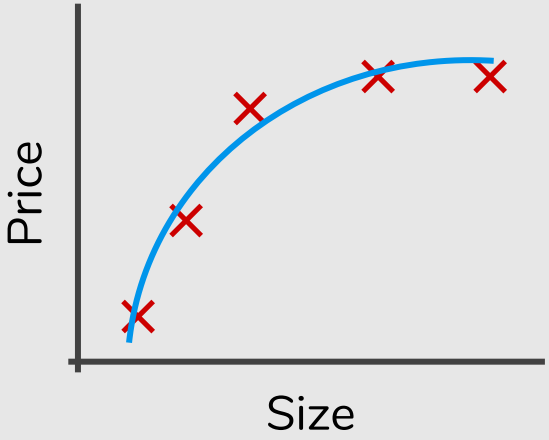
Bias/Variance



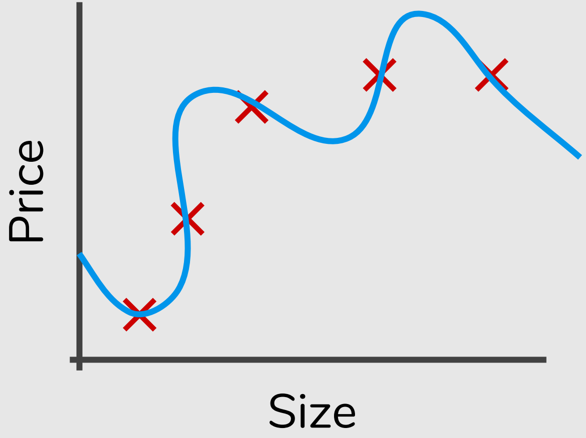
$$\theta_0 + \theta_1 x$$

Underfitting

High bias



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

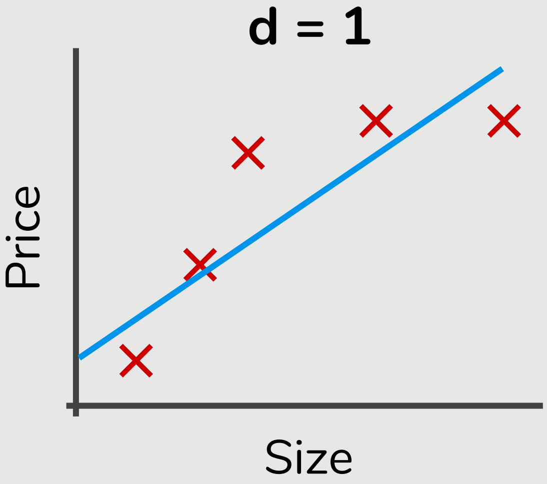


$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Overfitting

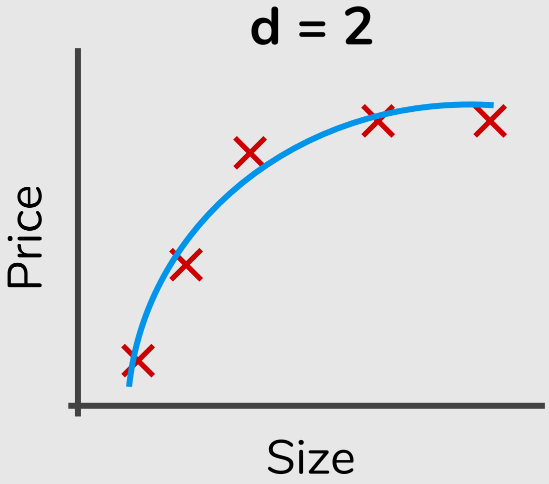
High variance

Bias/Variance

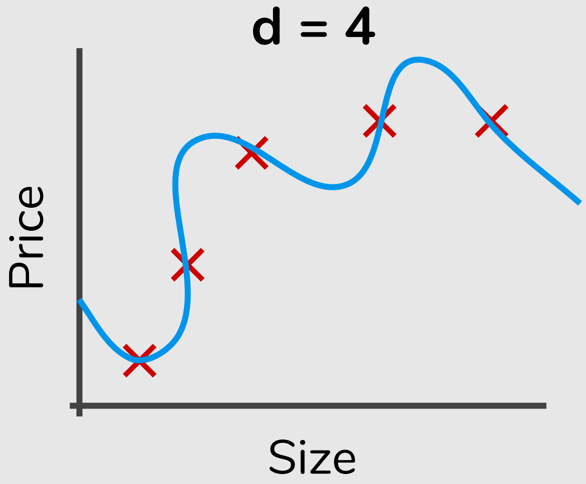


$$\theta_0 + \theta_1 x$$

Underfitting
High bias



$$\theta_0 + \theta_1 x + \theta_2 x^2$$



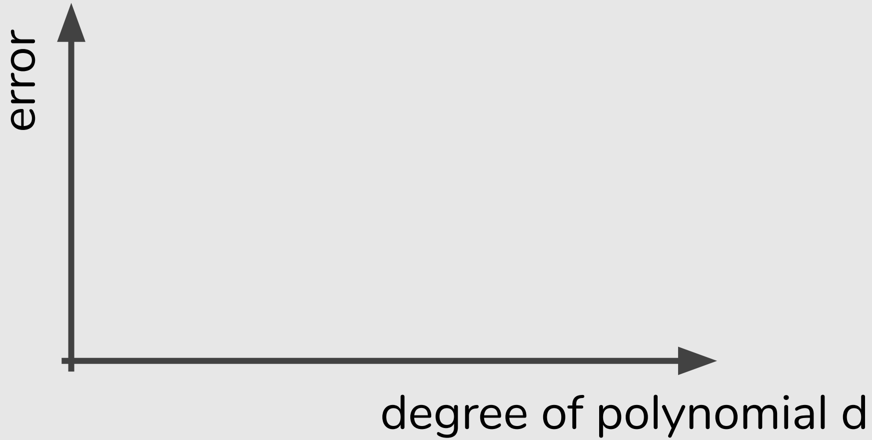
$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Overfitting
High variance

Bias/Variance

Training error: $J_{train}(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

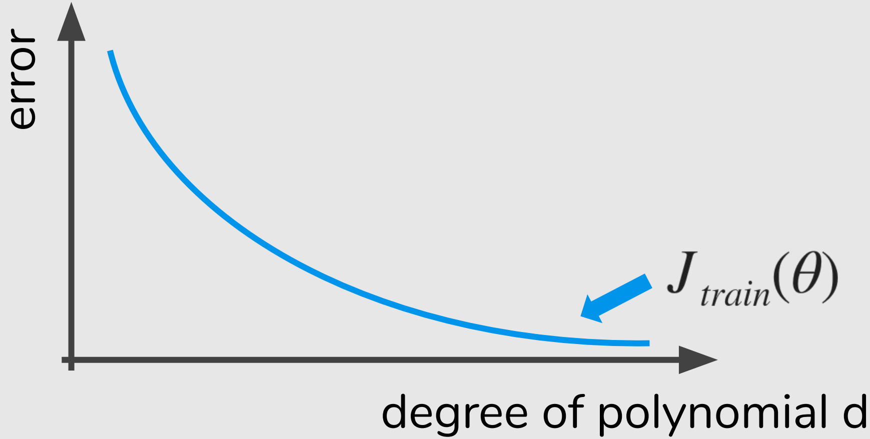
Cross-validation error: $J_{cv}(\theta) = \frac{1}{2m_{cv}} \sum_{i=1}^{m_{cv}} (h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$



Bias/Variance

Training error: $J_{train}(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

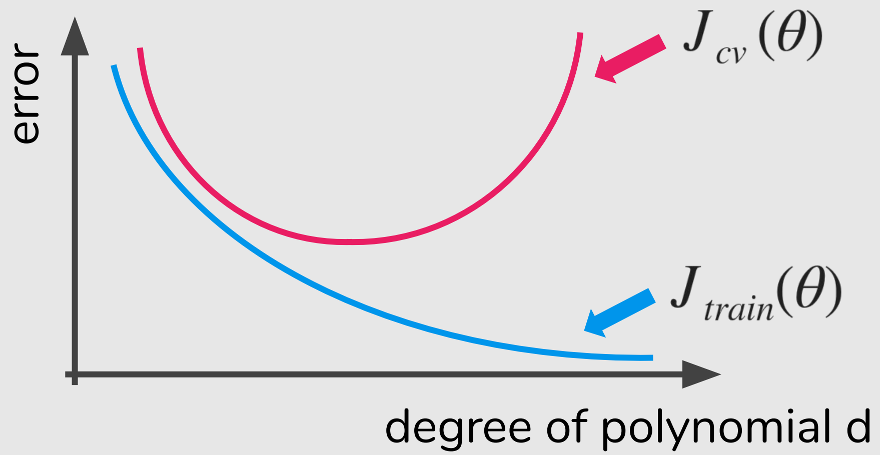
Cross-validation error: $J_{cv}(\theta) = \frac{1}{2m_{cv}} \sum_{i=1}^{m_{cv}} (h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$



Bias/Variance

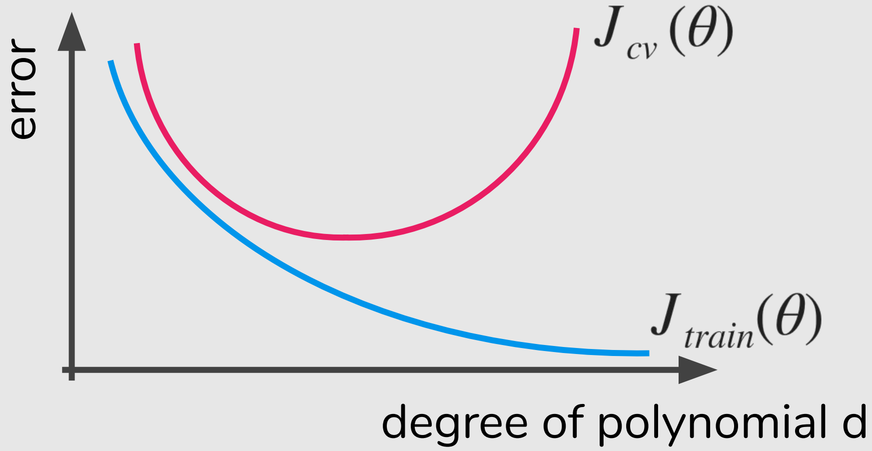
Training error: $J_{train}(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

Cross-validation error: $J_{cv}(\theta) = \frac{1}{2m_{cv}} \sum_{i=1}^{m_{cv}} (h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$



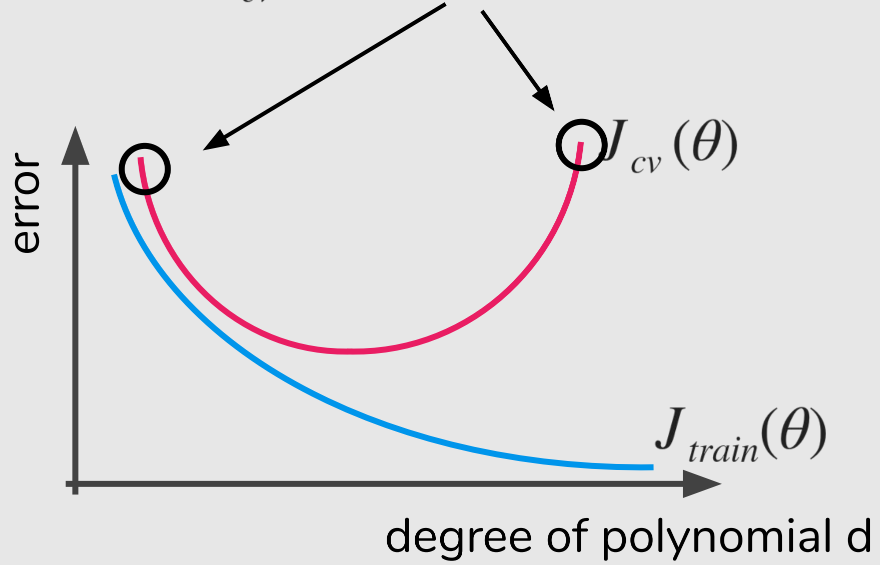
Diagnosing Bias vs. Variance

Suppose your learning algorithm is performing less well than you were hoping: $J_{cv}(\theta)$ is high. Is it a bias problem or a variance problem?



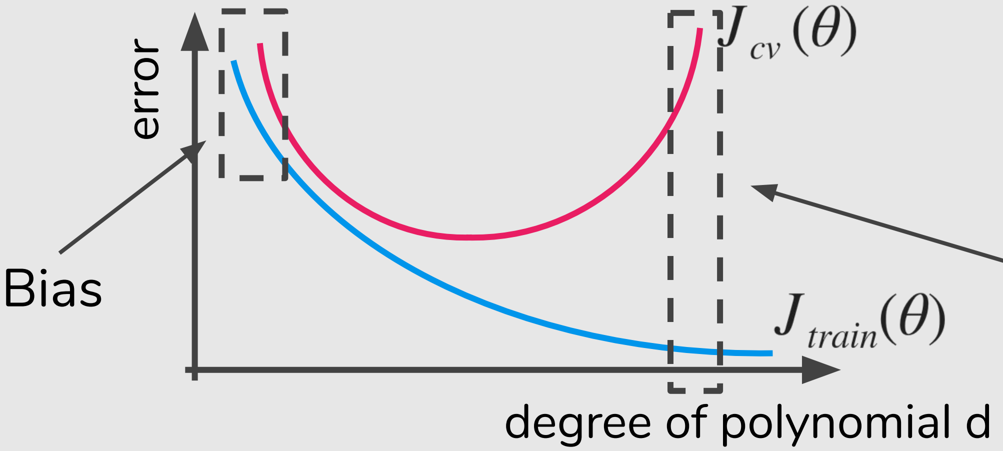
Diagnosing Bias vs. Variance

Suppose your learning algorithm is performing less well than you were hoping: $J_{cv}(\theta)$ is high. Is it a bias problem or a variance problem?



Diagnosing Bias vs. Variance

Suppose your learning algorithm is performing less well than you were hoping: $J_{cv}(\theta)$ is high. Is it a bias problem or a variance problem?

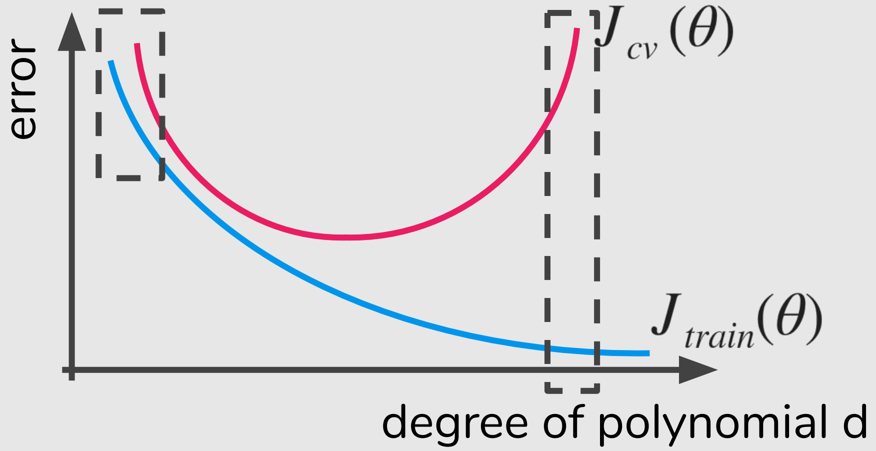


Bias (underfit):

Variance (overfit):

Diagnosing Bias vs. Variance

Suppose your learning algorithm is performing less well than you were hoping: $J_{cv}(\theta)$ is high. Is it a bias problem or a variance problem?



Bias (underfit):

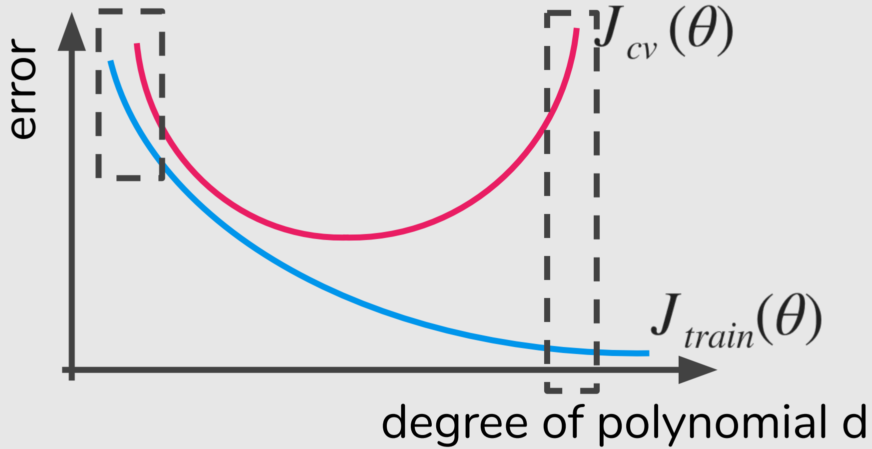
$J_{train}(\theta)$ will be high

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

Variance (overfit):

Diagnosing Bias vs. Variance

Suppose your learning algorithm is performing less well than you were hoping: $J_{cv}(\theta)$ is high. Is it a bias problem or a variance problem?



Bias (underfit):

$J_{train}(\theta)$ will be high

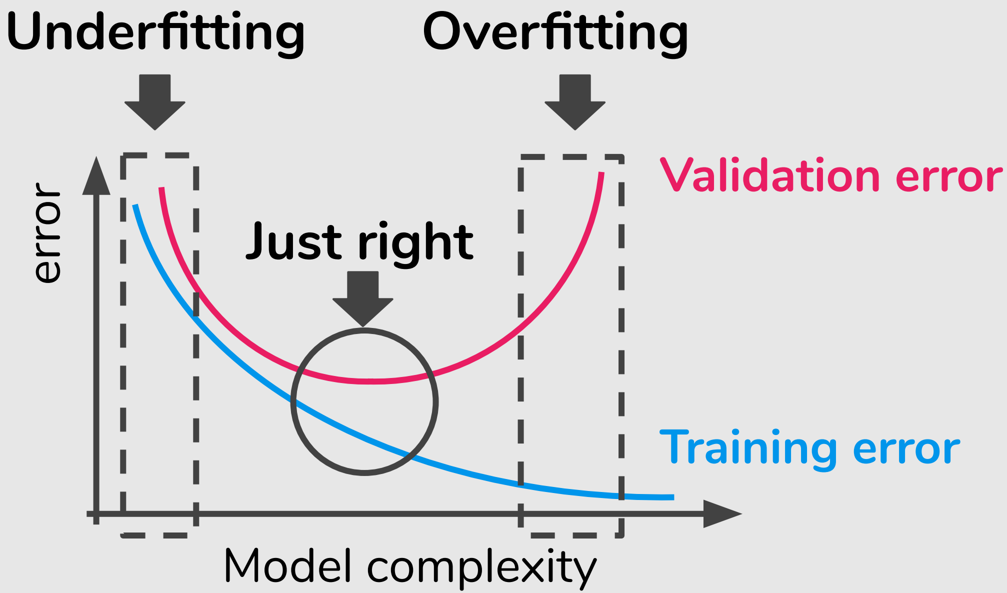
$J_{cv}(\theta) \approx J_{train}(\theta)$

Variance (overfit):

$J_{train}(\theta)$ will be low

$J_{cv}(\theta) \gg J_{train}(\theta)$

Diagnosing Bias vs. Variance



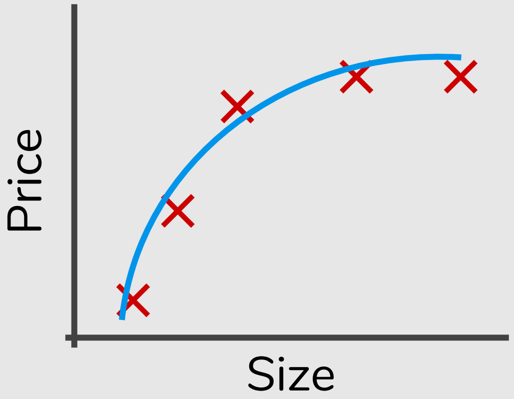
Understanding the Bias-Variance Tradeoff: http://scott.fortmann-roe.com/docs/BiasVariance.html

Underfitting

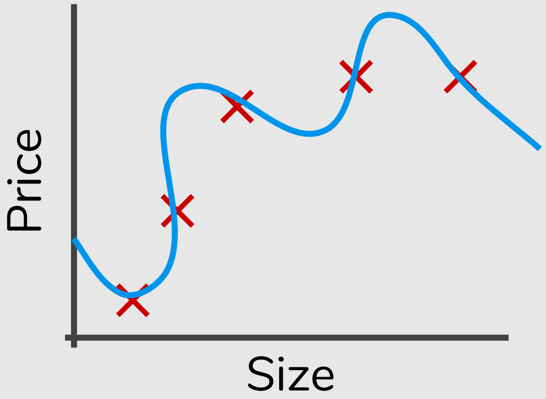
Overfitting

Cost Function

Intuition

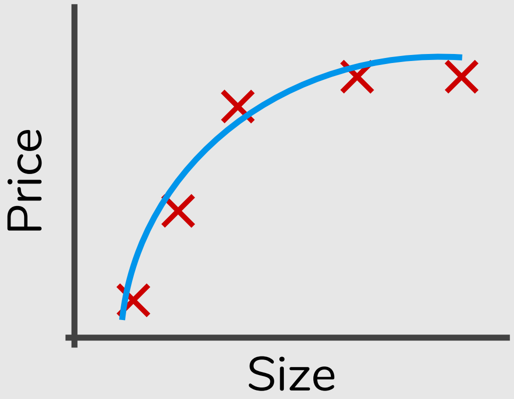


$$\theta_0 + \theta_1 x + \theta_2 x^2$$

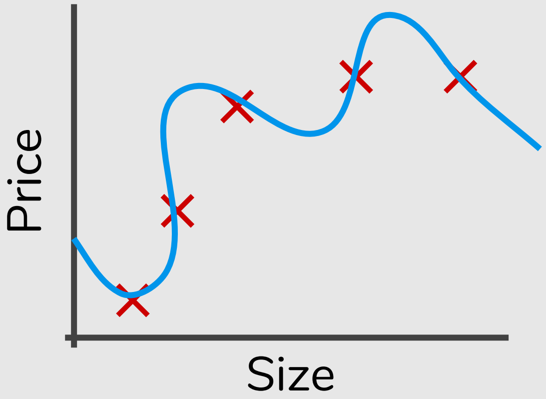


$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Intuition



$\theta_0 + \theta_1 x + \theta_2 x^2$

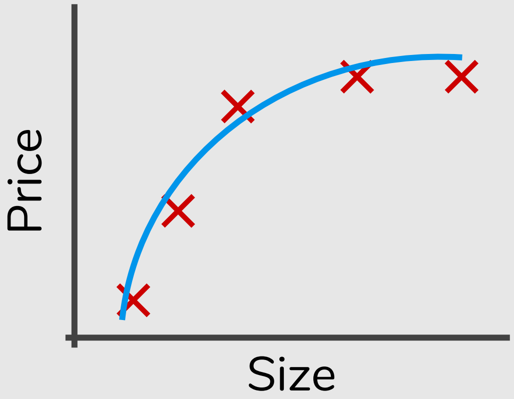


$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

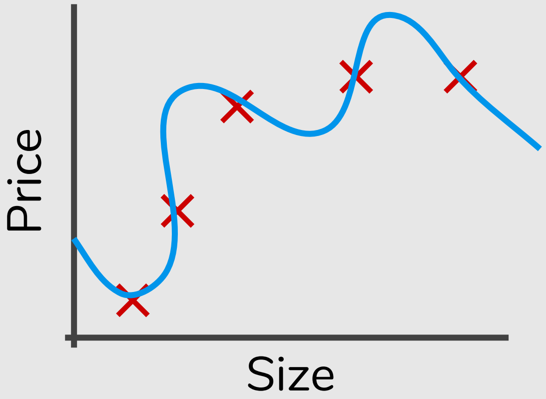
Suppose we penalize and make θ_3, θ_4 really small.

$$\min_\theta \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Intuition



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

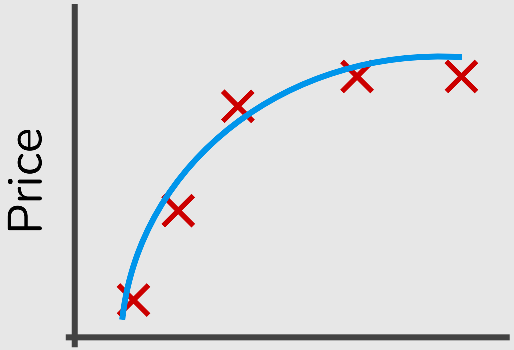


$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

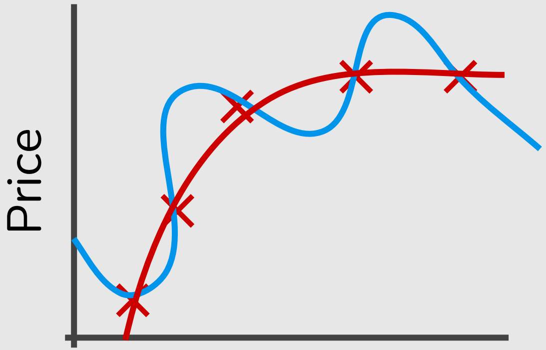
Suppose we penalize and make θ_3, θ_4 really small.

$$\min_\theta \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + 1000\ \theta_3^2 + 1000\ \theta_4^2$$

Intuition



$\theta_0 + \theta_1 x + \theta_2 x^2$



$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

$\theta_3 \approx 0$

$\theta_4 \approx 0$

Suppose we penalize and make θ_3, θ_4 really small.

$$\min_\theta \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + 1000\ \theta_3^2 + 1000\ \theta_4^2$$

Regularization

Small values for parameters $\theta_0, \theta_1, \ldots, \theta_n$

- “Simpler” hypothesis
- Less prone to overfitting

Regularization

Small values for parameters $\theta_0, \theta_1, \ldots, \theta_n$

- "Simpler" hypothesis
- Less prone to overfitting

Housing

- Features: $x_0, x_1, \ldots, x_{100}$
- Parameters: $\theta_0, \theta_1, \theta_2, \ldots, \theta_{100}$

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

Regularization

Small values for parameters $\theta_0, \theta_1, \ldots, \theta_n$

- "Simpler" hypothesis
- Less prone to overfitting

Housing

- Features: $x_0, x_1, \ldots, x_{100}$
- Parameters: $\theta_0, \theta_1, \theta_2, \ldots, \theta_{100}$

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

Regularization

Regularization parameter

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

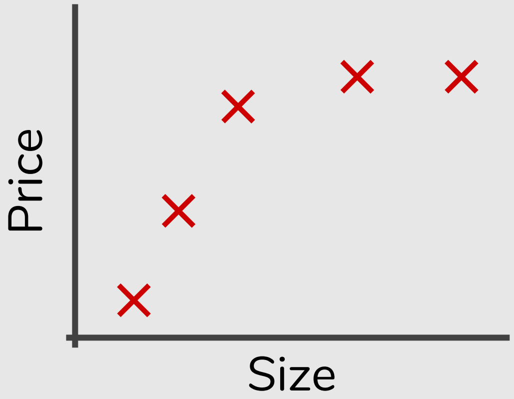
to fit the training data well

to keep the parameters small

In regularized linear regression, we choose θ to minimize

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2 \right]$$

What if λ is set to an extremely large value (perhaps for too large for our problem, say $\lambda = 10^{10}$)?

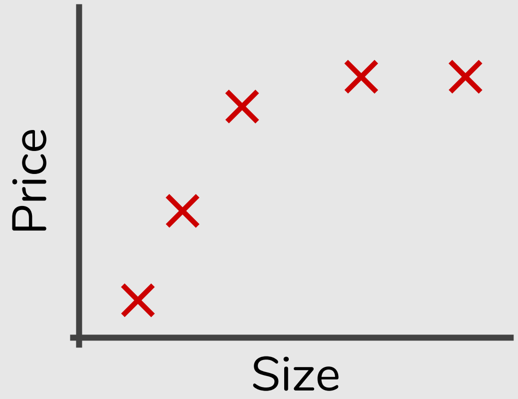


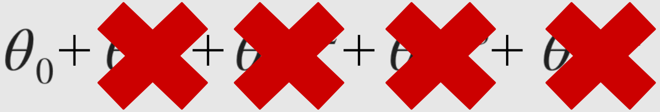
$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

In regularized linear regression, we choose θ to minimize

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

What if λ is set to an extremely large value (perhaps for too large for our problem, say $\lambda = 10^{10}$)?

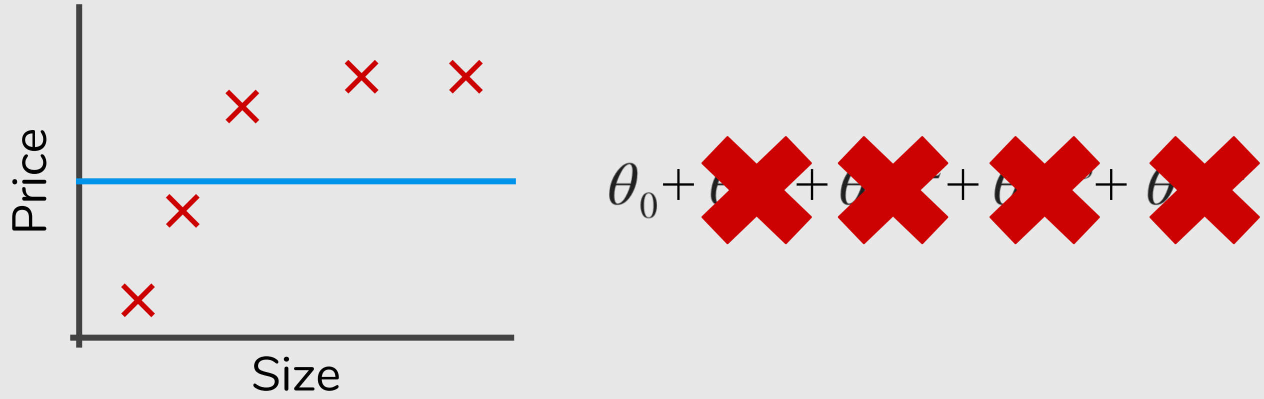




In regularized linear regression, we choose θ to minimize

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

What if λ is set to an extremely large value (perhaps for too large for our problem, say $\lambda = 10^{10}$)?



Regularized Linear Function

Gradient Descent

repeat {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

(simultaneously update θ_j for $j = 0, 1, \ldots, n$)

}

Gradient Descent

repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

} (simultaneously update θ_j for $j =$ ~~0,~~ $1, \ldots, n$)

Gradient Descent

repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

$$\theta_j := \theta_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} + \frac{\lambda}{m} \theta_j \right]$$

} (simultaneously update θ_j for $j =$ $1, \ldots, n$)

Gradient Descent

repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

$$\theta_j := \theta_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} + \frac{\lambda}{m} \theta_j \right]$$

} (simultaneously update θ_j for $j =$ ✗ $1, \ldots, n$)

$$\theta_j := \theta_j (1 - \alpha \frac{\lambda}{m}) - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

Gradient Descent

repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

$$\theta_j := \theta_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} + \frac{\lambda}{m} \theta_j \right]$$

} (simultaneously update θ_j for $j =$ ~~0~~ $1, \ldots, n$)

$$\theta_j := \theta_j (1 - \alpha \frac{\lambda}{m}) - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

Normal Equation

$$X = \begin{bmatrix} \text{---} & (x^{(1)})^{\mathrm{T}} & \text{---} \\ \text{---} & (x^{(2)})^{\mathrm{T}} & \text{---} \\ \text{---} & \vdots & \text{---} \\ \text{---} & (x^{(m)})^{\mathrm{T}} & \text{---} \end{bmatrix} \qquad y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix} \qquad \theta = (X^T X)^{-1} X^T y$$

Normal Equation

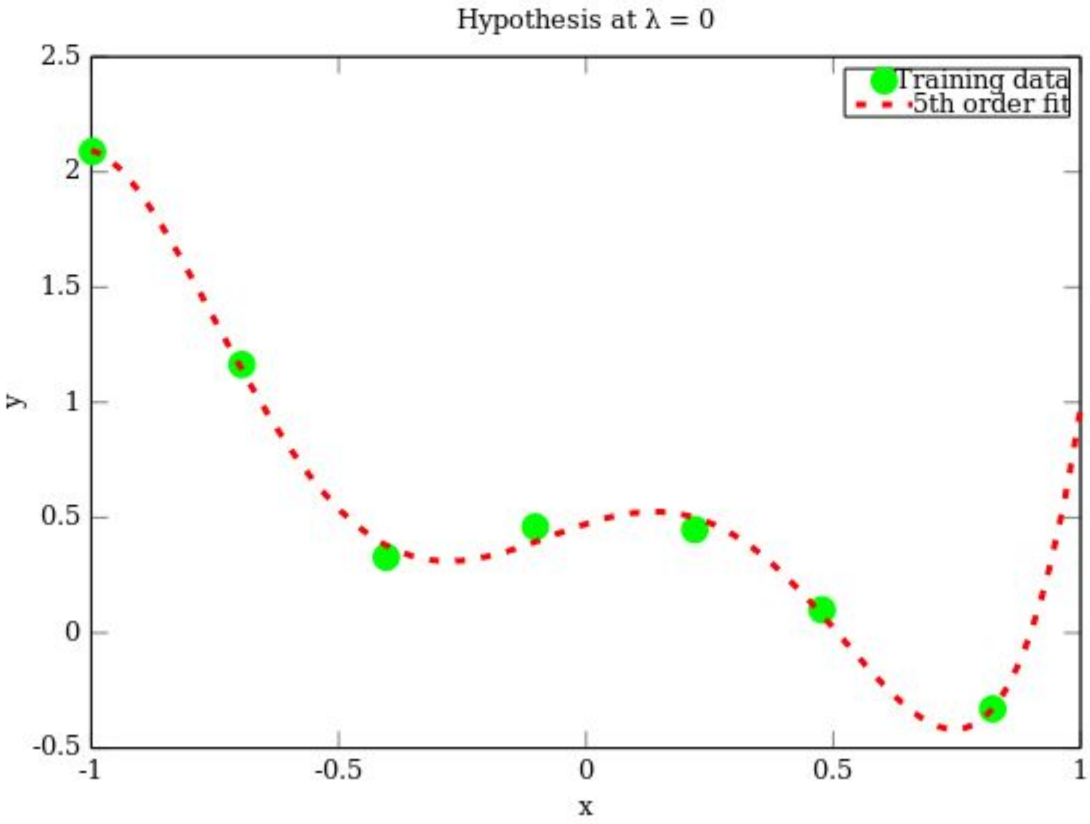
$$X = \begin{bmatrix} — (x^{(1)})^T — \\ — (x^{(2)})^T — \\ — \vdots — \\ — (x^{(m)})^T — \end{bmatrix} \quad y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix} \quad \theta = (X^T X)^{-1} X^T y$$

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

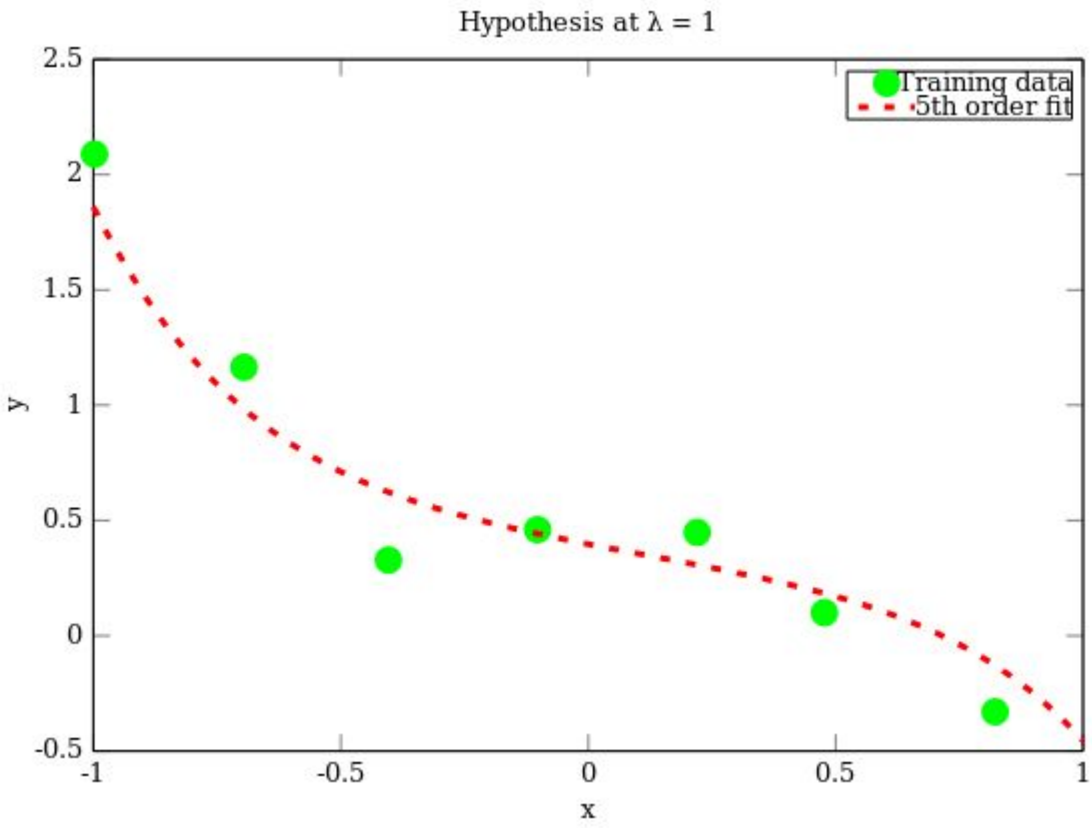
Normal Equation

$$X = \begin{bmatrix} — & (x^{(1)})^{\mathrm{T}} & — \\ — & (x^{(2)})^{\mathrm{T}} & — \\ — & \vdots & — \\ — & (x^{(m)})^{\mathrm{T}} & — \end{bmatrix} \qquad y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix} \qquad \theta = (X^T X)^{-1} X^T y$$

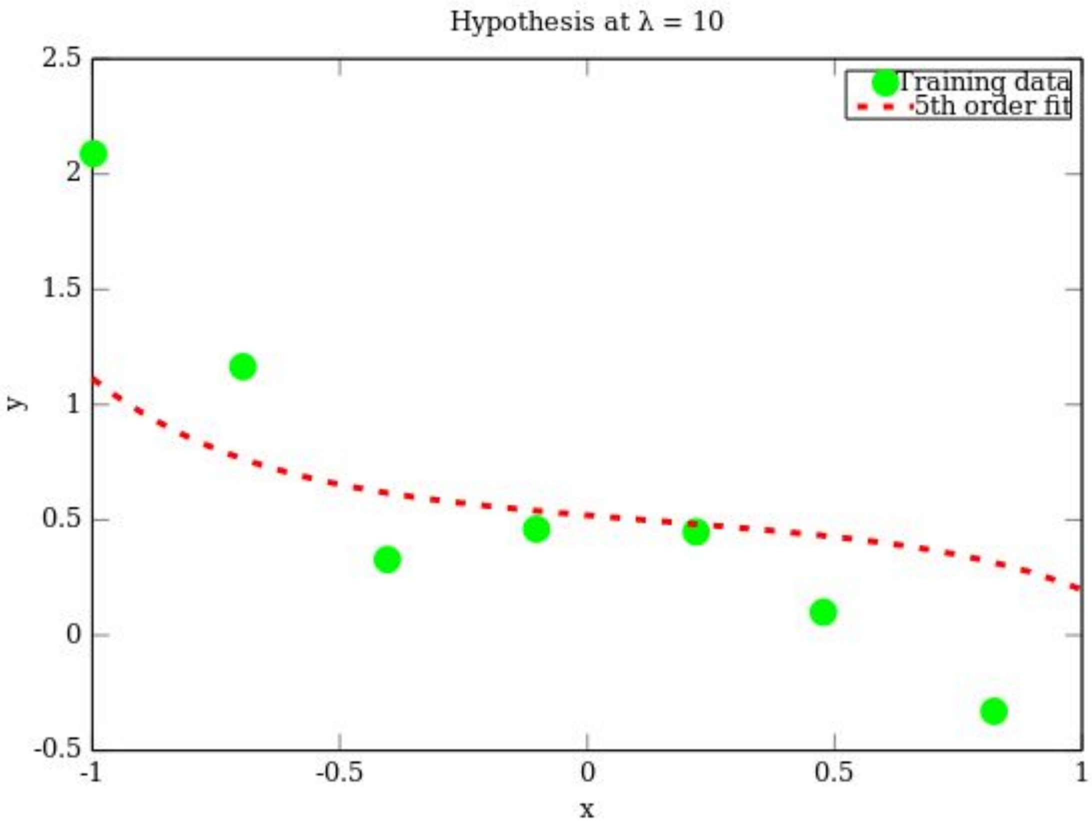
$$\theta = \left(X^T X + \lambda \begin{bmatrix} 0 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix} \right)^{-1} X^T y$$



http://melvincabatuan.github.io/Machine-Learning-Activity-4/



http://melvincabatuan.github.io/Machine-Learning-Activity-4/



http://melvincabatuan.github.io/Machine-Learning-Activity-4/

Regularized Logistic Function

Gradient Descent

repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

$$\theta_j := \theta_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} + \frac{\lambda}{m} \theta_j \right]$$

} (simultaneously update θ_j for $j =$ ~~0,~~ $1, \ldots, n$)

$$\theta_j := \theta_j (1 - \alpha \frac{\lambda}{m}) - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

Gradient Descent

$$h_\theta(x) = \theta^T x \;\rightarrow\; h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

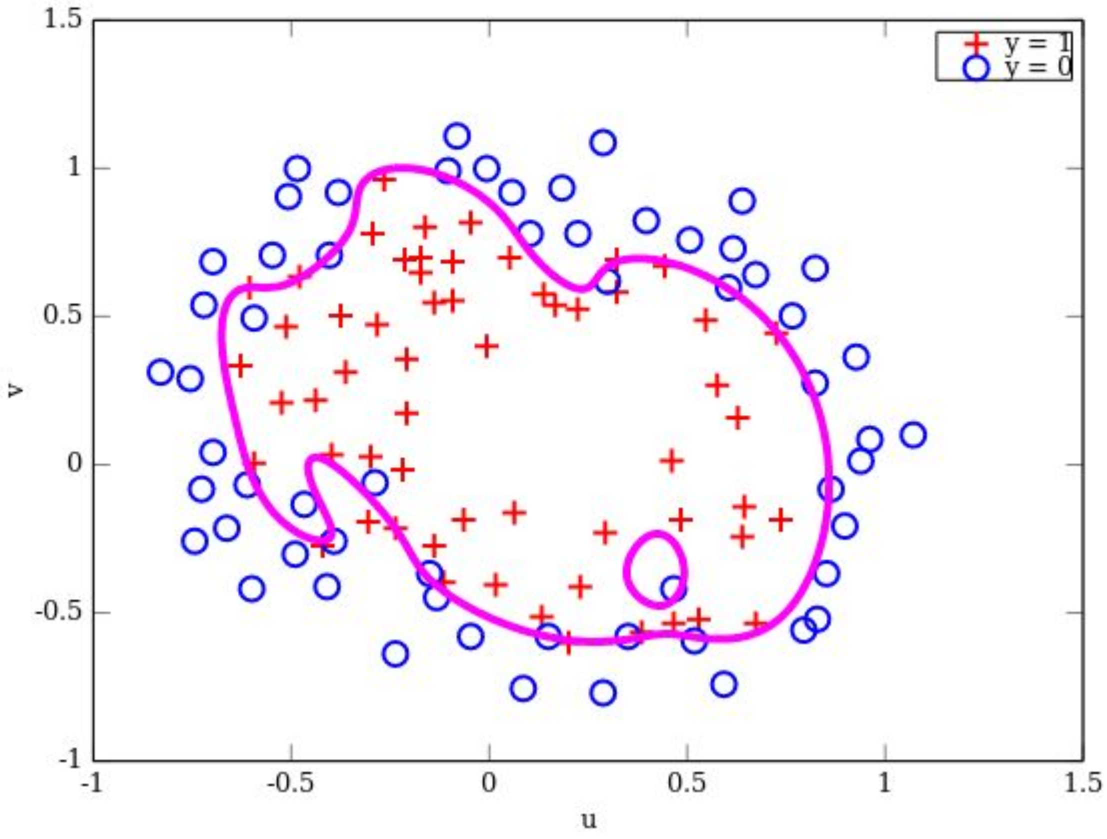
repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

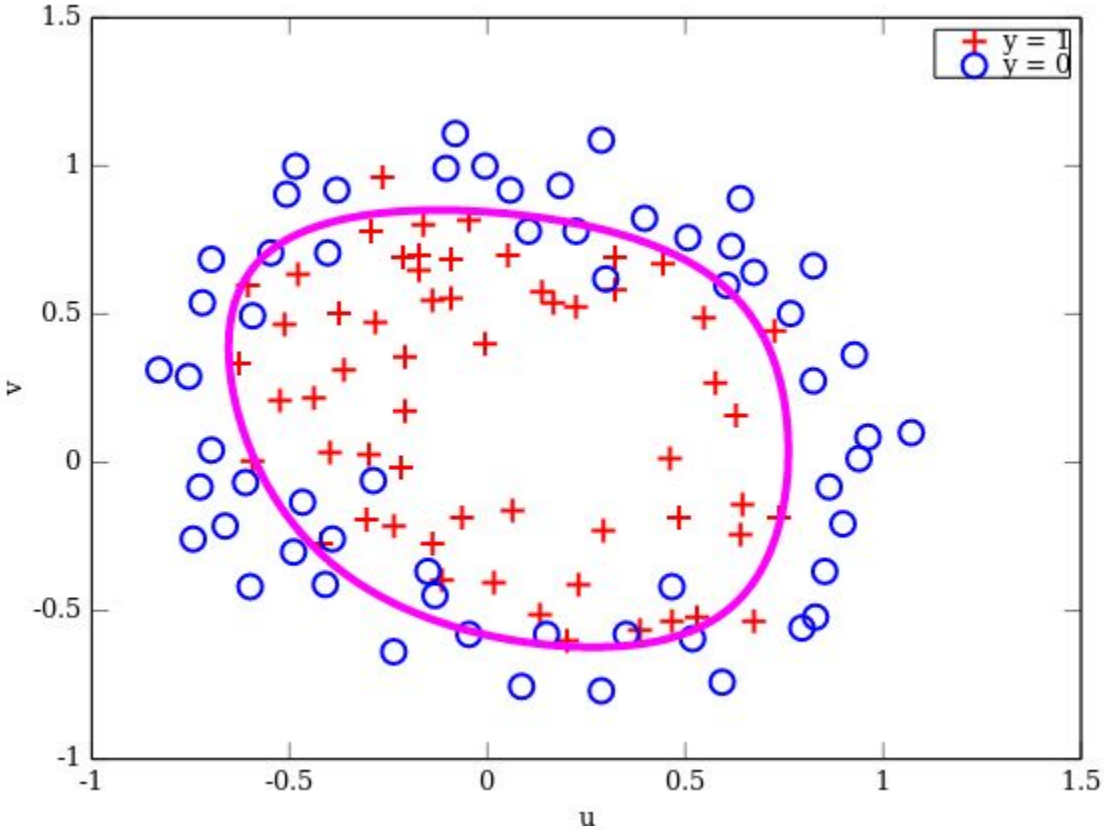
$$\theta_j := \theta_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} + \frac{\lambda}{m} \theta_j \right]$$

} (simultaneously update θ_j for $j =$ $1, \ldots, n$)

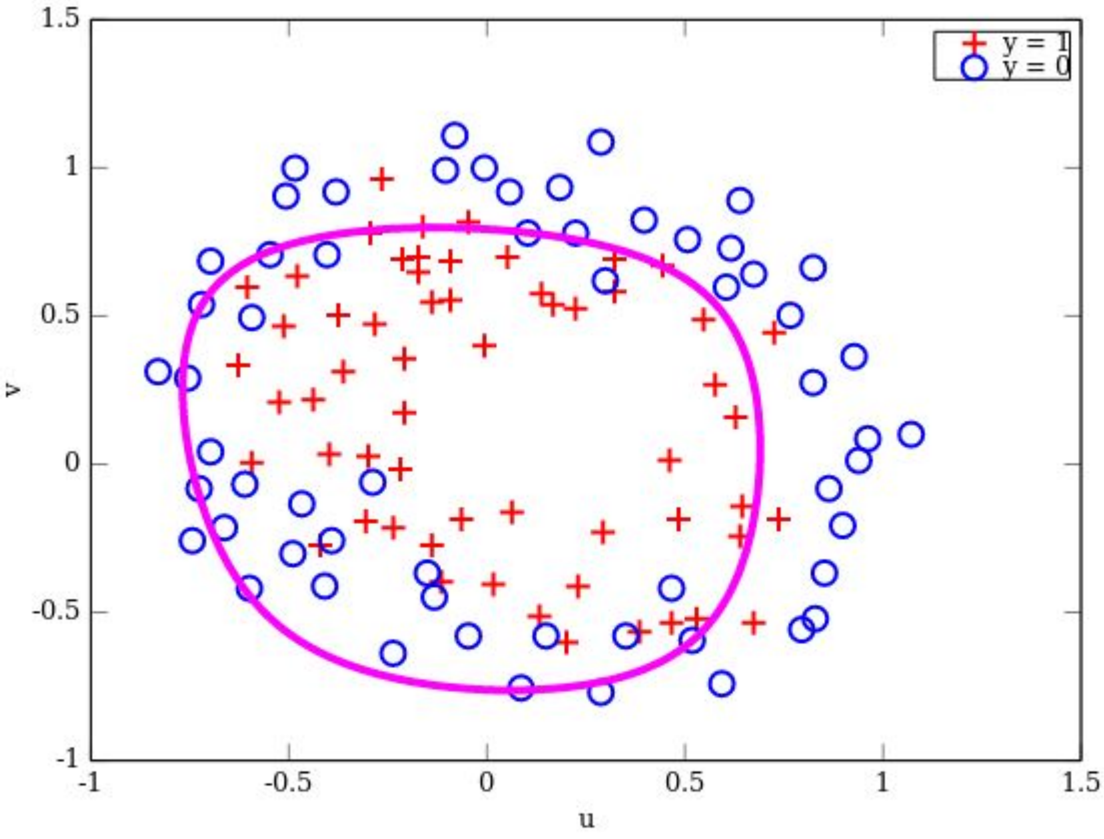
$$\theta_j := \theta_j (1 - \alpha \frac{\lambda}{m}) - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$



http://melvincabatuan.github.io/Machine-Learning-Activity-4/



http://melvincabatuan.github.io/Machine-Learning-Activity-4/



http://melvincabatuan.github.io/Machine-Learning-Activity-4/

References

Machine Learning Books

- Hands-On Machine Learning with Scikit-Learn and TensorFlow, Chap. 4
- Pattern Recognition and Machine Learning, Chap. 3

Machine Learning Courses

- https://www.coursera.org/learn/machine-learning, Week 3 & 6