Recall from last time ...

Logistic Regression

Classification

Email: **Spam** / **Not Spam**?

Content Video: **Sensitive** / **Non-sensitive**?

Skin Lesion: **Malignant** / **Benign**?

$$y \in \{0,1\}$$

0: "Negative Class" (e.g., Benign skin lesion)

1: "Positive Class" (e.g., Malignant skin lesion)

Hypothesis Representation

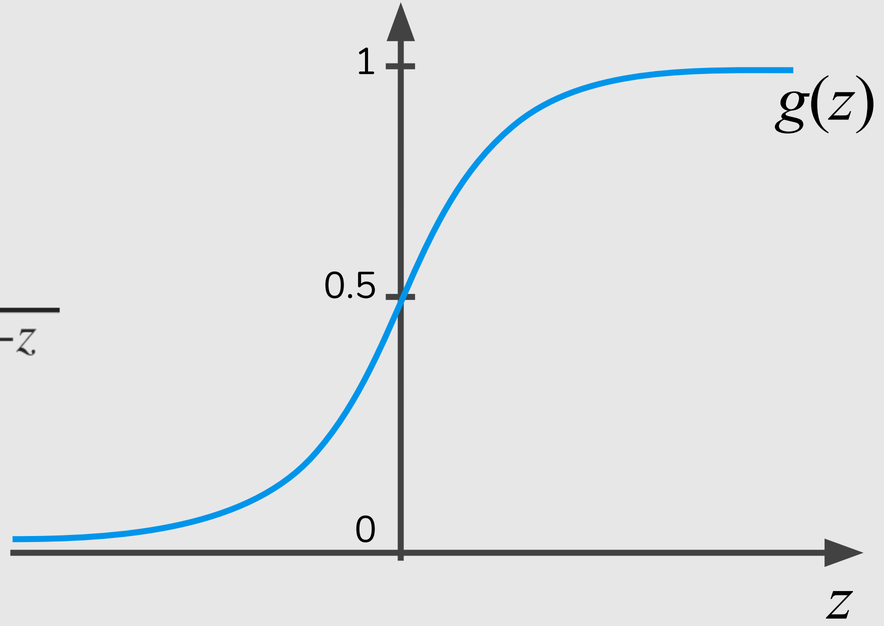
Logistic Regression Model

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

Want $0 \leq h_\theta(x) \leq 1$

$$h_\theta(x) = g(\theta^T x)$$

$$g(z) = \frac{1}{1+e^{-z}}$$



Sigmoid Function

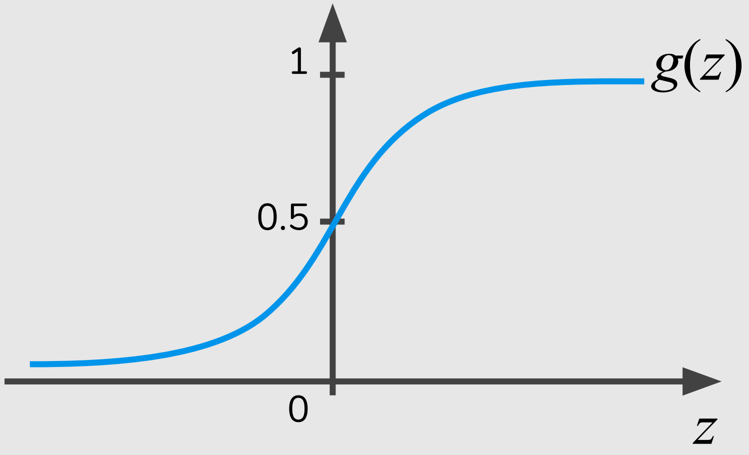
Logistic Function

Decision Boundary

Logistic Regression

$$h_\theta(x) = g(\theta^{\mathrm{T}}x)$$

$$g(z) = \frac{1}{1+e^{-z}}$$



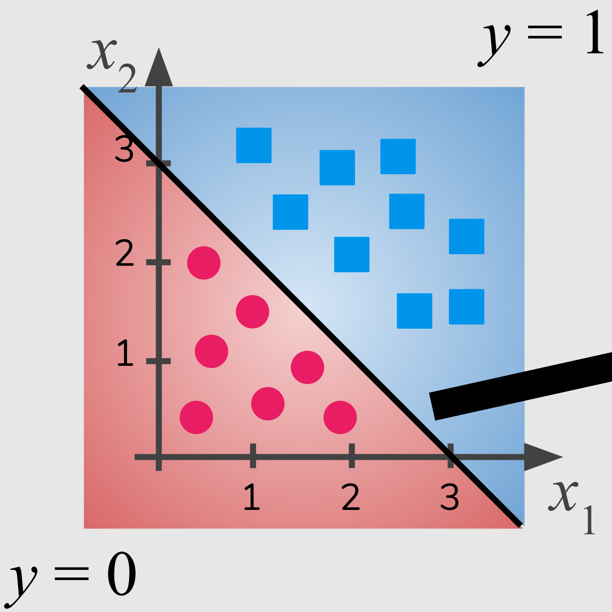
Suppose predict "$y = 1$" if $h_\theta(x) \geq 0.5$

predict "$y = 0$" if $h_\theta(x) < 0.5$

$g(z) \geq 0.5$ when $z \geq 0$

$g(z) < 0.5$ when $z < 0$

Decision Boundary



$$\begin{array}{ccc} -3 & 1 & 1 \\ \Uparrow & \Uparrow & \Uparrow \end{array}$$

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

Decision Boundary $\quad x_1 + x_2 = 3$

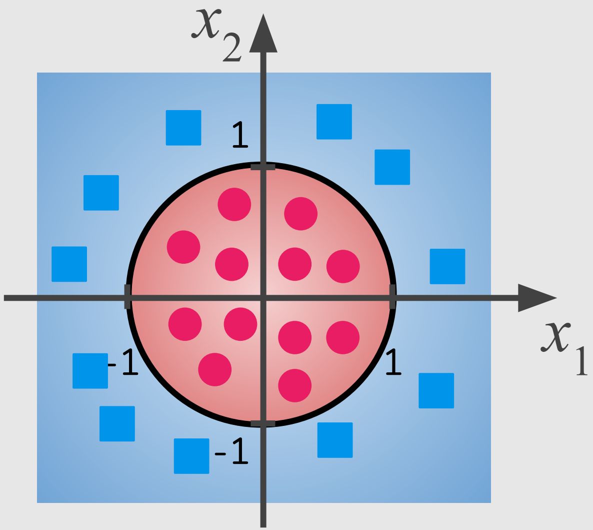
$$h_\theta(x) = 0.5$$

Predict "$y = 1$" if $-3 + x_1 + x_2 \geq 0$

$$x_1 + x_2 \geq 3$$

$$y = 0,\ x_1 + x_2 < 3$$

Non-linear Decision Boundaries



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$$

$$\theta_0 \to -1,\quad \theta_1 \to 0,\quad \theta_2 \to 0,\quad \theta_3 \to 1,\quad \theta_4 \to 1$$

Predict "$y = 1$" if $-1 + x_1^2 + x_2^2 \geq 0$

$$x_1^2 + x_2^2 \geq 1$$

Cost Function

Training set: $\{(x^{(1)},y^{(1)}),\ (x^{(2)},y^{(2)}),\ \ldots,\ (x^{(m)},y^{(m)})\}$

$$h_\theta(x) = \frac{1}{1+e^{-\theta^{T}x}} \qquad x \in \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \quad x_0 = 1,\ y \in \{0,1\}$$

How to choose parameters θ ?

Cost Function

$$\mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

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

Logistic

$$\mathrm{Cost}(h_\theta(x), y) = \frac{1}{2} (h_\theta(x) - y)^2 \qquad h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

Derivative of Logistic Function

$$\boxed{g(z) = \frac{1}{1+e^{-z}}}$$

$$\begin{aligned}
g'(z) &= \frac{d}{dz}\,\frac{1}{1+e^{-z}} \\
&= \frac{0\cdot(1+e^{-z}) - 1\cdot(-e^{-z})}{(1+e^{-z})^2} \qquad \text{(quotient rule)} \\
&= \frac{e^{-z}}{(1+e^{-z})^2} \\
&= \left(\frac{1}{1+e^{-z}}\right)\left(1 - \frac{1}{1+e^{-z}}\right) \\
&= g(z)(1-g(z))
\end{aligned}$$

Logistic Regression Cost Function

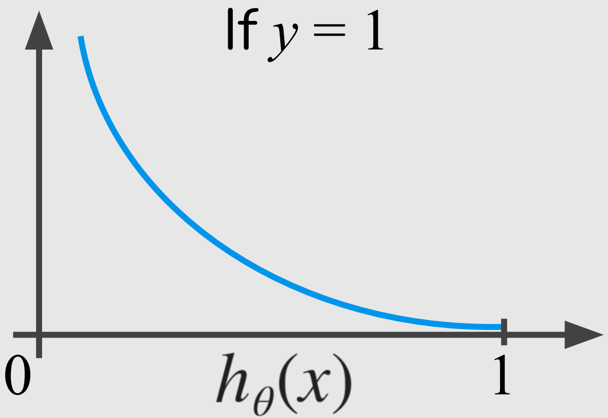
$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

Why use log function?

https://math.stackexchange.com/questions/886555/deriving-cost-function-using-mle-why-use-log-function

Logistic Regression Cost Function

$$\text{Cost}(h_\theta(x),\ y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$



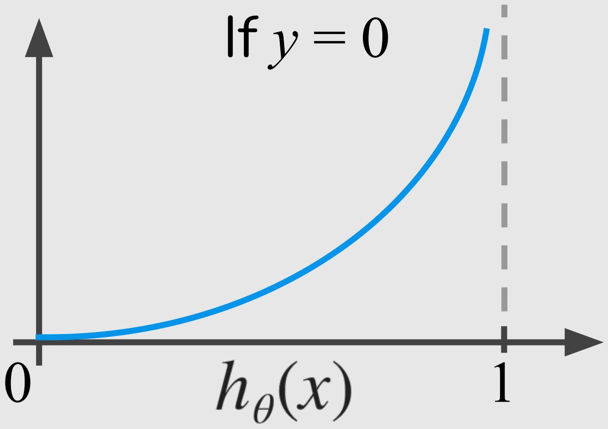
Cost $= 0$ if $y = 1,\ h_\theta(x) = 1$

But as $h_\theta(x) \rightarrow 0$

$\text{Cost} \rightarrow \infty$

Logistic Regression Cost Function

$$\mathrm{Cost}(h_\theta(x),\ y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$



Simplified Cost Function and Gradient Descent

Logistic Regression Cost Function

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\mathrm{Cost}(h_\theta(x^{(i)}),\ y^{(i)})$$

$$= -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right]$$

To fit parameters θ : $\min_\theta J(\theta)$

To make a new prediction given new x: Output $h_\theta(x) = \frac{1}{1+\mathrm{e}^{-\theta^{\mathrm{T}}x}}$

Gradient Descent

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

Want $\min_\theta J(\theta)$:

repeat {

$$\theta_j := \theta_j - \alpha \boxed{\frac{\partial}{\partial\theta_j}J(\theta)}$$

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

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

Gradient Descent

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

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

Want $\min_\theta J(\theta)$:

repeat {

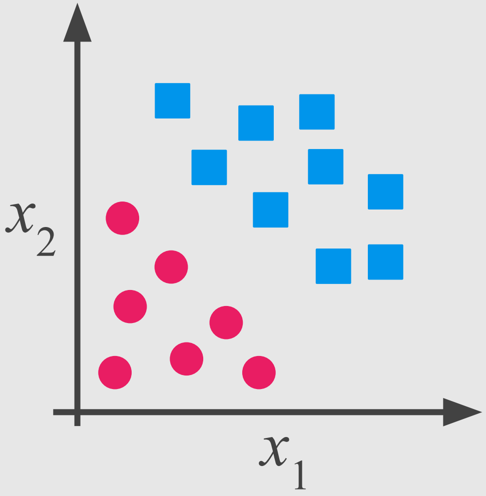
$$\theta_j := \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

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

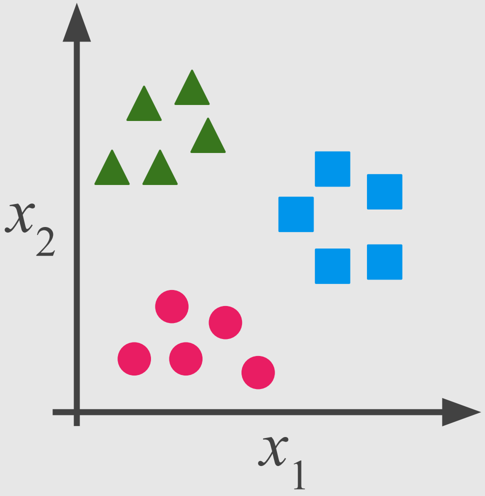
Algorithm looks identical to linear regression!

Multiclass Classification: One-vs-all

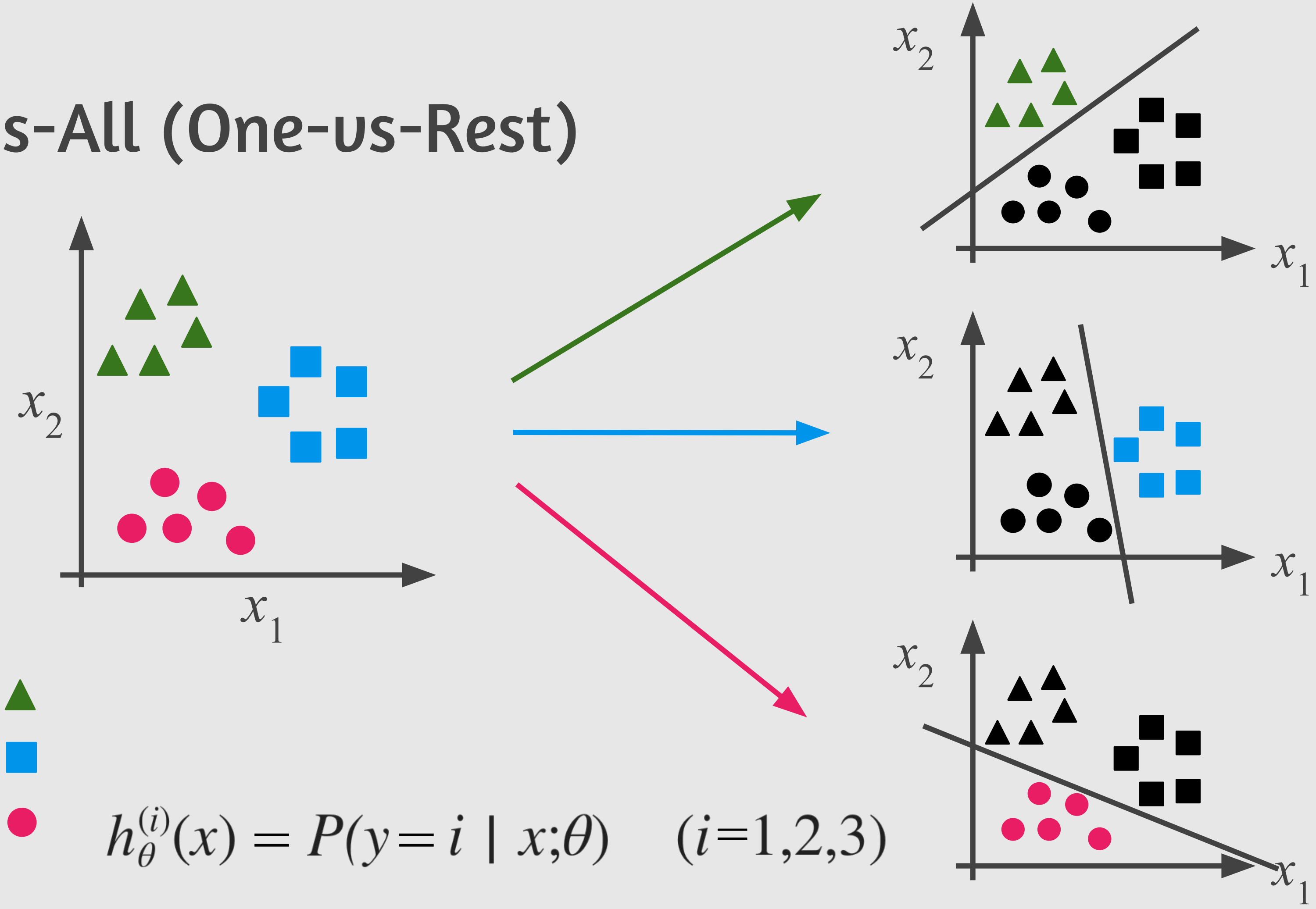
Binary Classification



Multi-class Classification



One-vs-All (One-vs-Rest)



Class 1: ▲

Class 2: ■

Class 3: ●

$$h_\theta^{(i)}(x) = P(y=i \mid x;\theta) \quad (i=1,2,3)$$

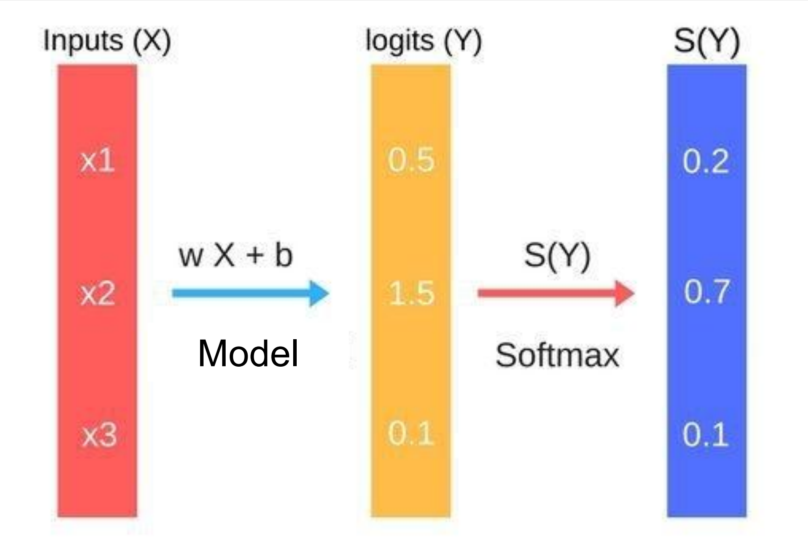
One-vs-All (One-vs-Rest)

Train a logistic regression classifier $h_\theta^{(i)}(x)$ for each class i to predict the probability that $y = i$.

One a new input x, to make a prediction, pick the class i that maximizes

$$\max_i \; h_\theta^{(i)}(x)$$

Multinomial Logistic Regression



http://dataaspirant.com/2017/03/14/multinomial-logistic-regression-model-works-machine-learning/

References

Machine Learning Books

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

Machine Learning Courses

- https://www.coursera.org/learn/machine-learning, Week 3
- Logistic Regression — The Math of Intelligence (Week 2): https://youtu.be/D8alok2P468
- http://cs229.stanford.edu/notes/cs229-notes1.pdf

Regularization

Machine Learning and Pattern Recognition

(Largely based on slides from Andrew Ng)

Prof. Sandra Avila

Institute of Computing (IC/Unicamp)

MC886/MO444, August 28, 2018

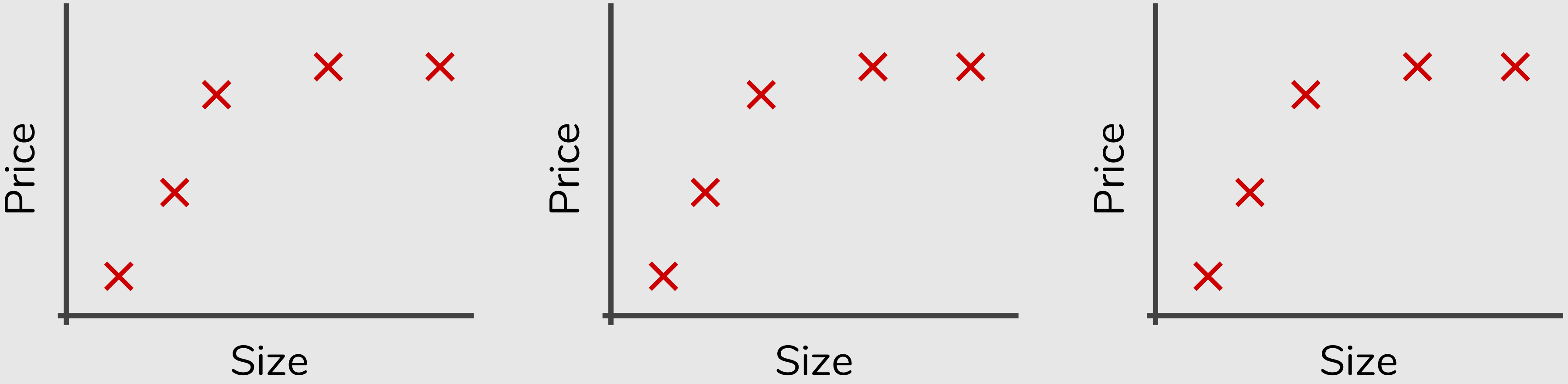
Today's Agenda

- 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

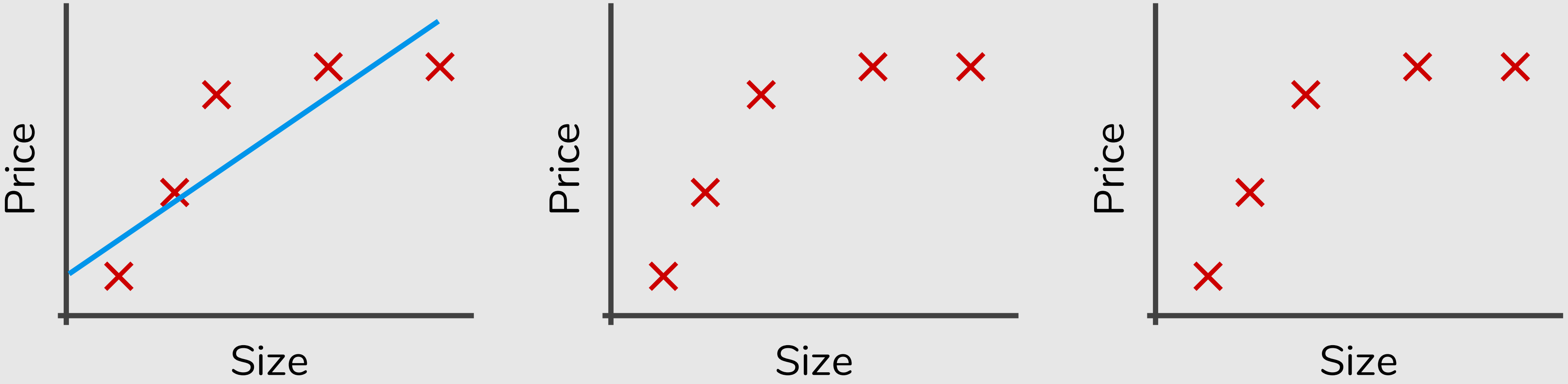


$$\theta_0 + \theta_1 x$$

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

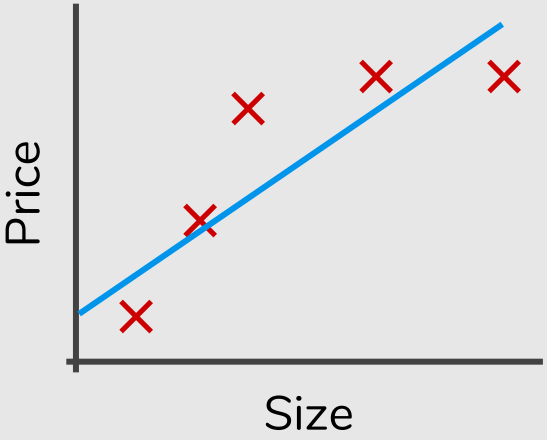
$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Example: Linear Regression



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

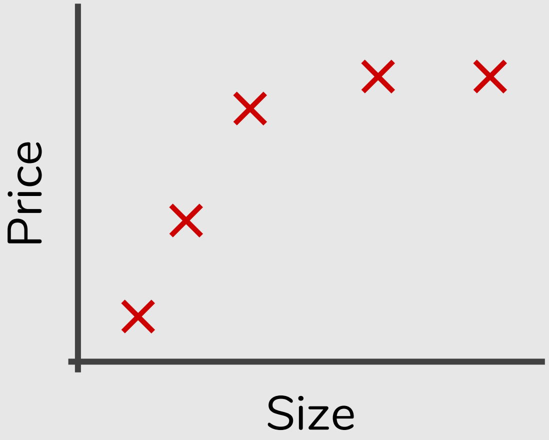
Example: Linear Regression



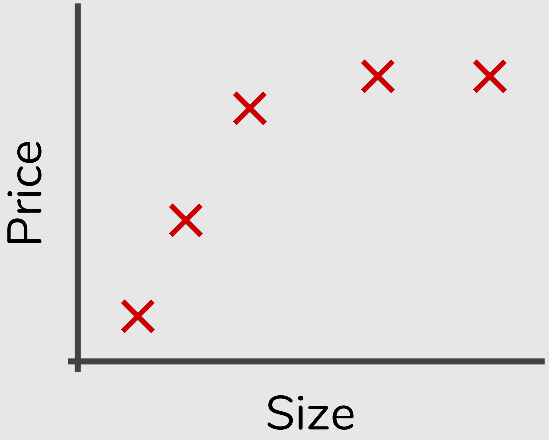
$$\theta_0 + \theta_1 x$$

Underfitting

High bias

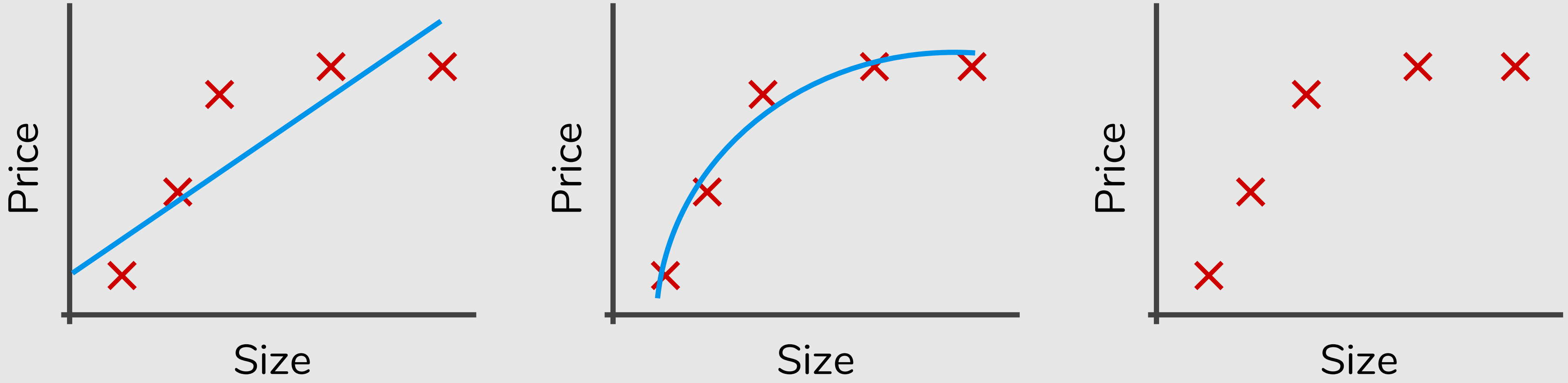


$$\theta_0 + \theta_1 x + \theta_2 x^2$$



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

Example: Linear Regression



$\theta_0 + \theta_1 x$

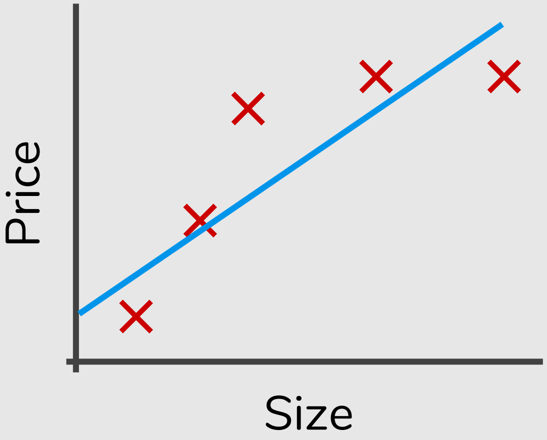
$\theta_0 + \theta_1 x + \theta_2 x^2$

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

Underfitting

High bias

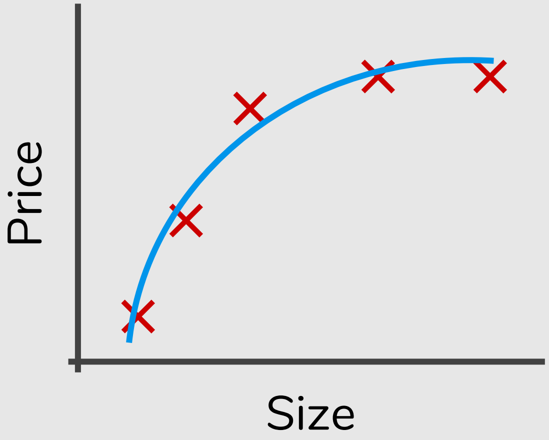
Example: Linear Regression



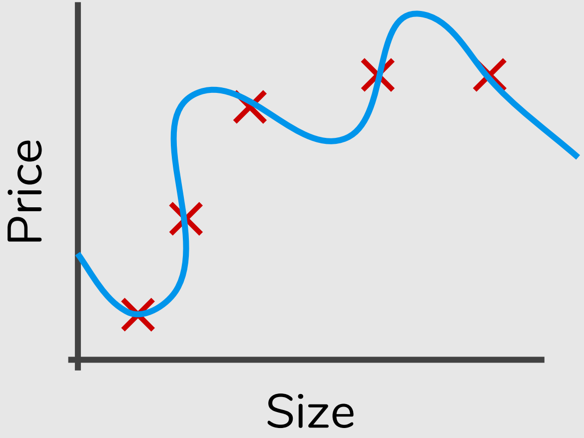
$$\theta_0 + \theta_1 x$$

Underfitting

High bias

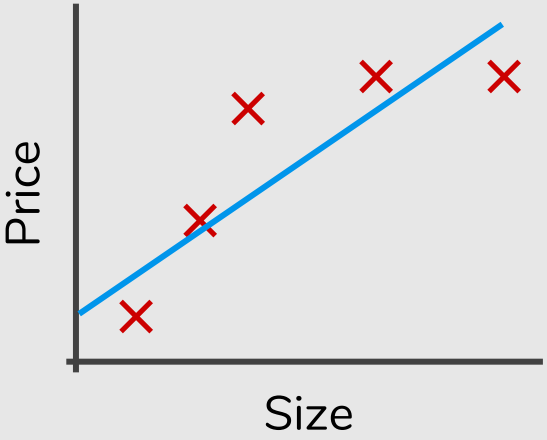


$$\theta_0 + \theta_1 x + \theta_2 x^2$$



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

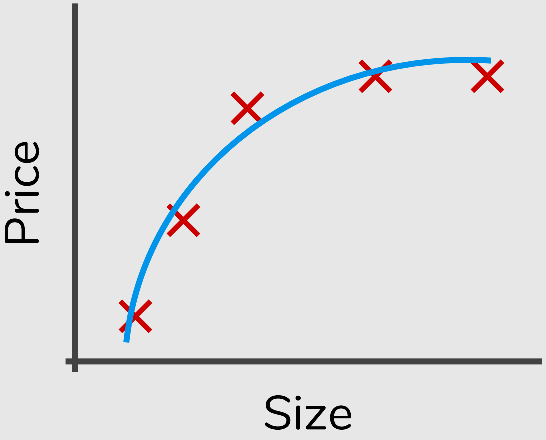
Example: Linear Regression



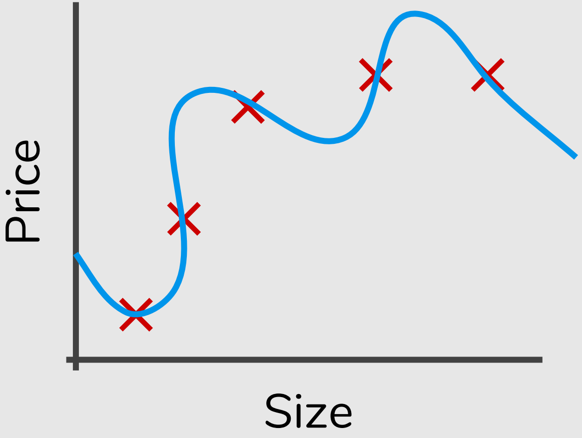
$$\theta_0 + \theta_1 x$$

Underfitting

High bias



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

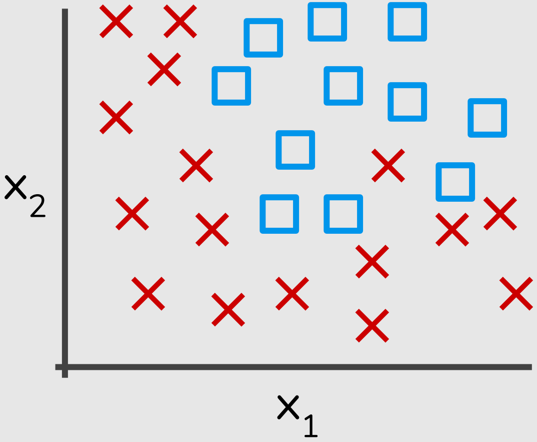


$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

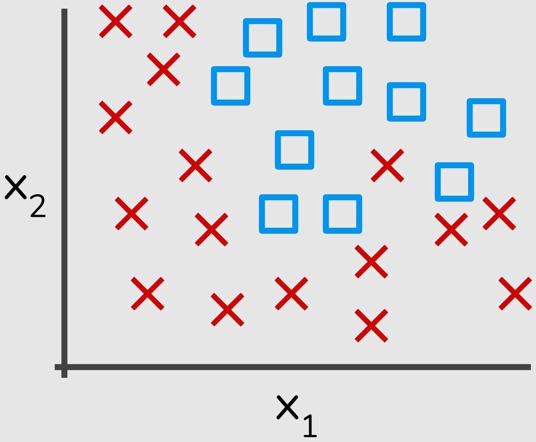
Overfitting

High variance

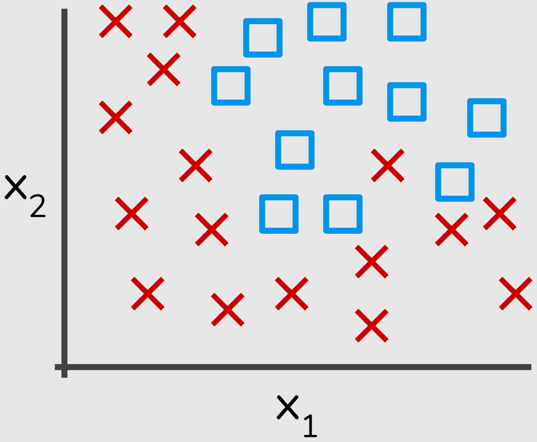
Example: Logistic Regression



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

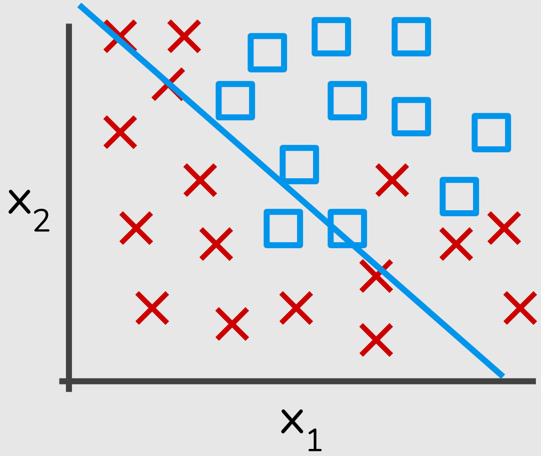


$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \\ + \theta_4 x_2^2 + \theta_5 x_1 x_2)$$



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

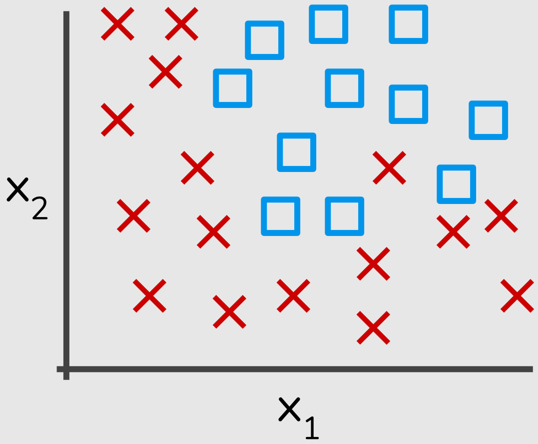
Example: Logistic Regression



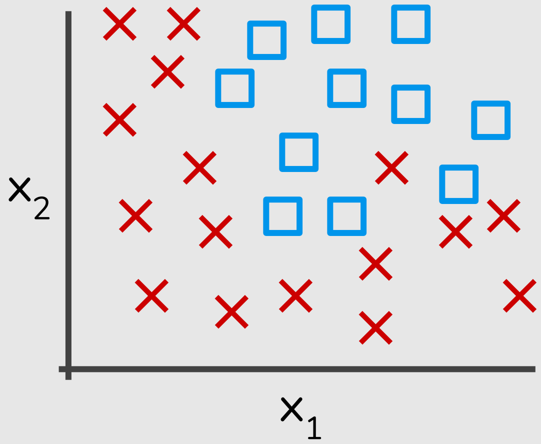
$$g(\theta_0+\theta_1x_1+\theta_2x_2)$$

Underfitting

High bias

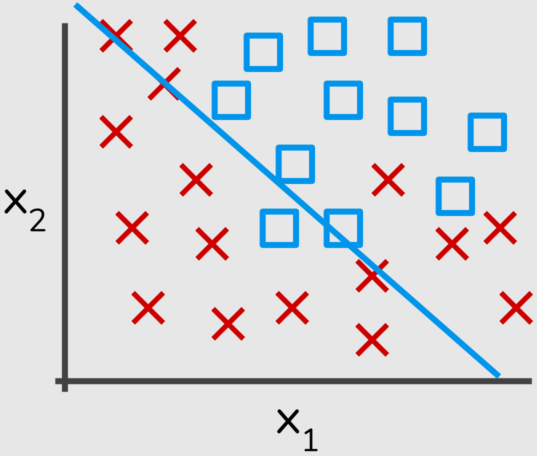


$$g(\theta_0+\theta_1x_1+\theta_2x_2+\theta_3x_1^2+\\+\theta_4x_2^2+\theta_5x_1x_2)$$



$$g(\theta_0+\theta_1x_1+\theta_2x_1^2+\\+\theta_3x_1^2x_2+\theta_4x_1^2x_2^2+\\+\theta_5x_1^2x_2^3+\dots)$$

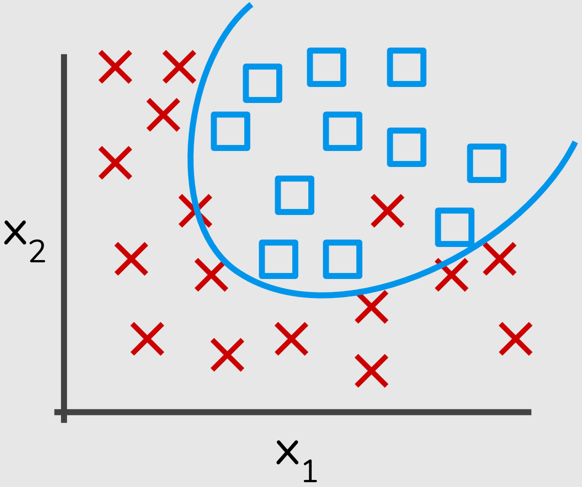
Example: Logistic Regression



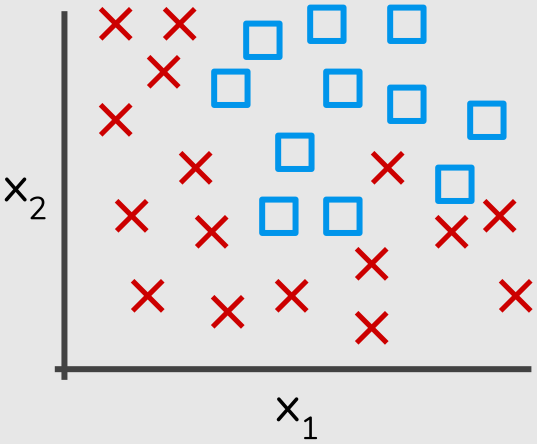
$$g(\theta_0+\theta_1x_1+\theta_2x_2)$$

Underfitting

High bias

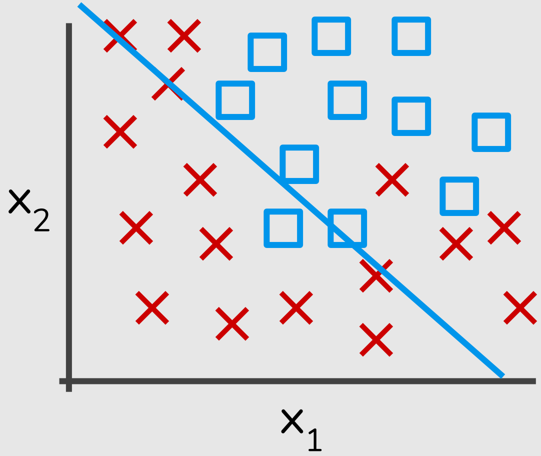


$$g(\theta_0+\theta_1x_1+\theta_2x_2+\theta_3x_1^2+ \\ +\theta_4x_2^2+\theta_5x_1x_2)$$



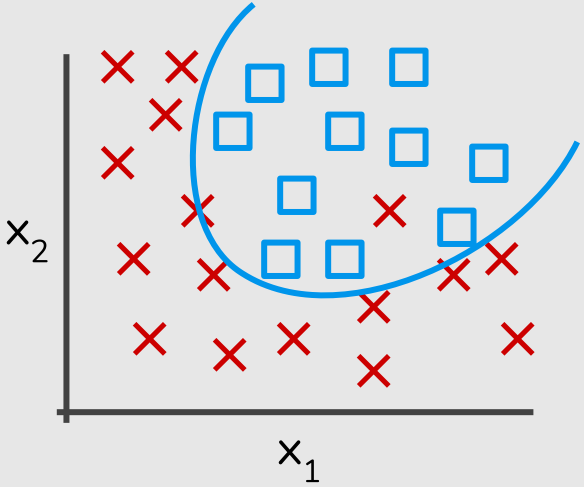
$$g(\theta_0+\theta_1x_1+\theta_2x_1^2+ \\ +\theta_3x_1^2x_2+\theta_4x_1^2x_2^2+ \\ +\theta_5x_1^2x_2^3+\dots)$$

Example: Logistic Regression



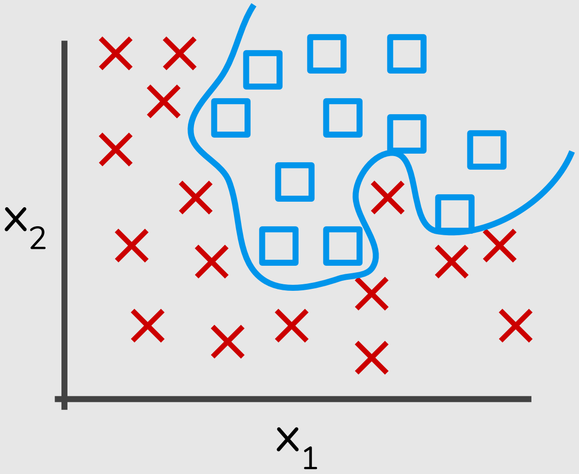
$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

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



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

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

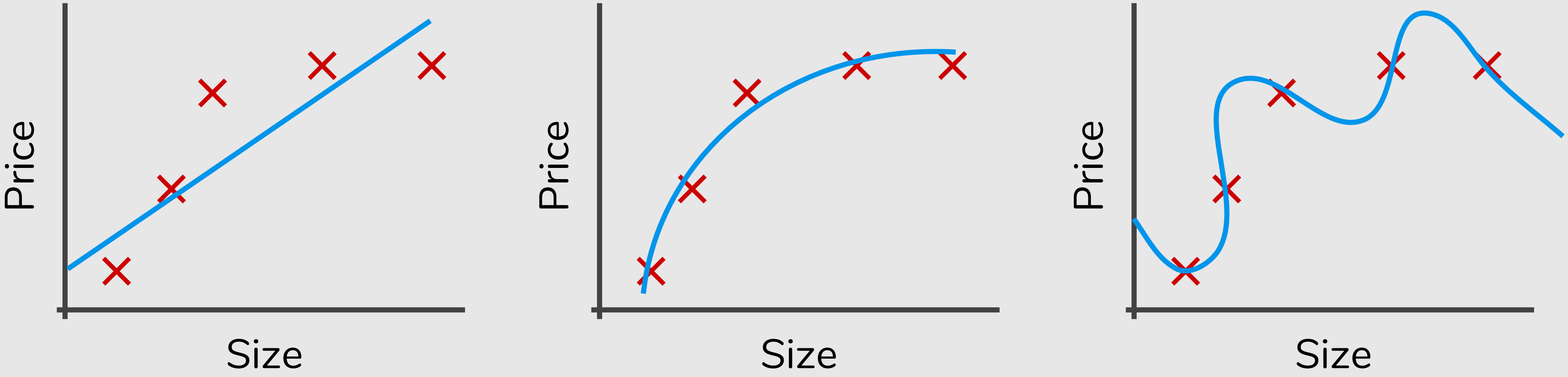
Increasing a model's complexity will typically increase its variance and reduce its bias.

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

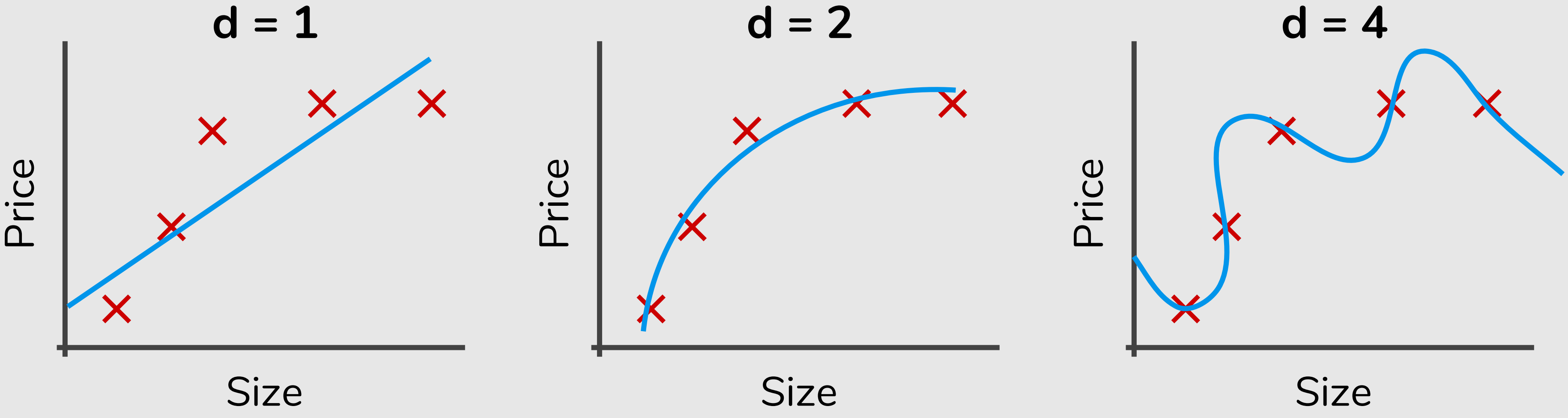
$\theta_0 + \theta_1 x + \theta_2 x^2$

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

Overfitting

High variance

Bias/Variance



$\theta_0 + \theta_1 x$

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

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

Underfitting
High bias

Overfitting
High variance

Bias/Variance

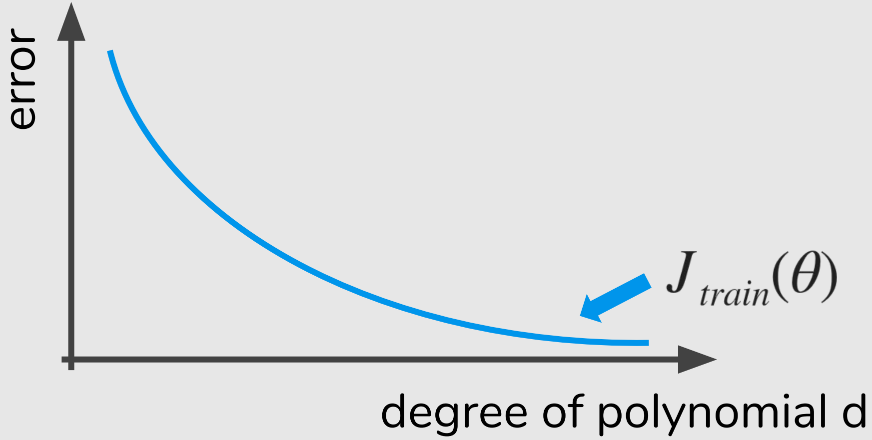
Training error: $J_{train}(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

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

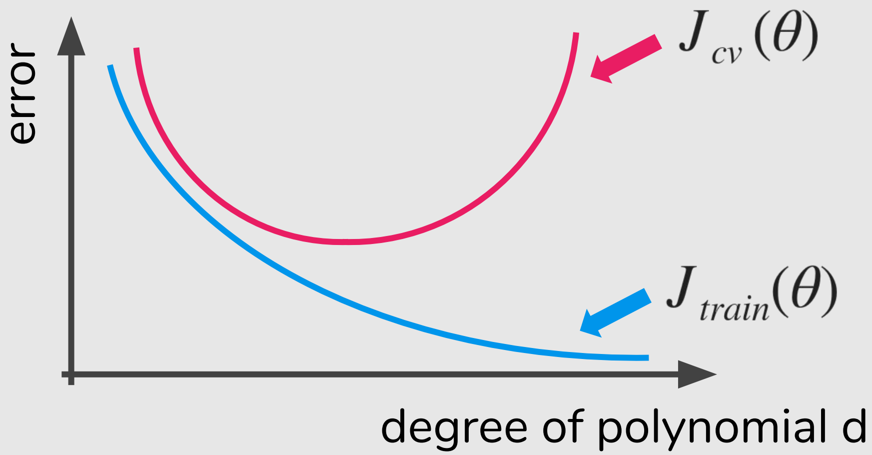
Cross-validation error: $J_{cv}(\theta) = \frac{1}{2m_{cv}} \sum_{i=1}^{m_{cv}} (h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$



Bias/Variance

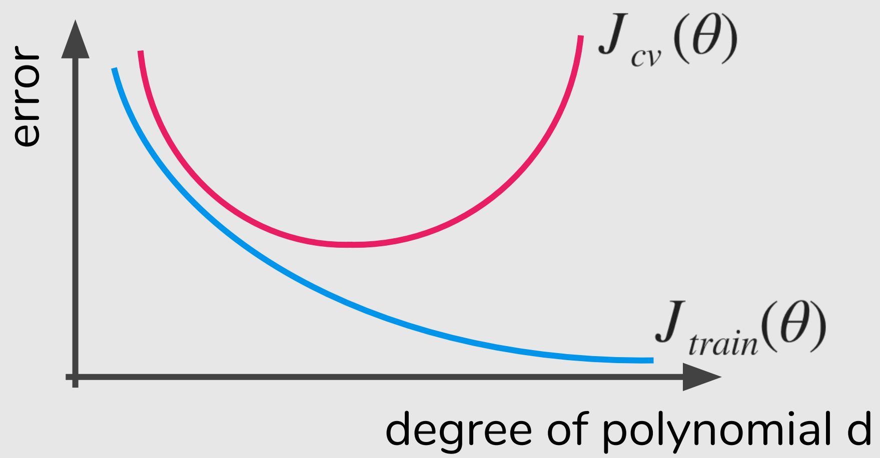
Training error: $J_{train}(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

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



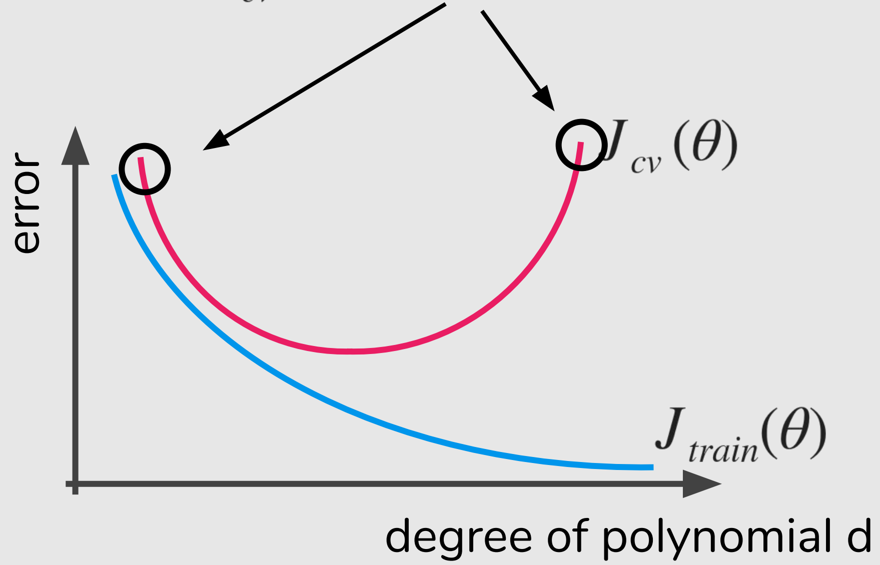
Diagnosing Bias vs. Variance

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?



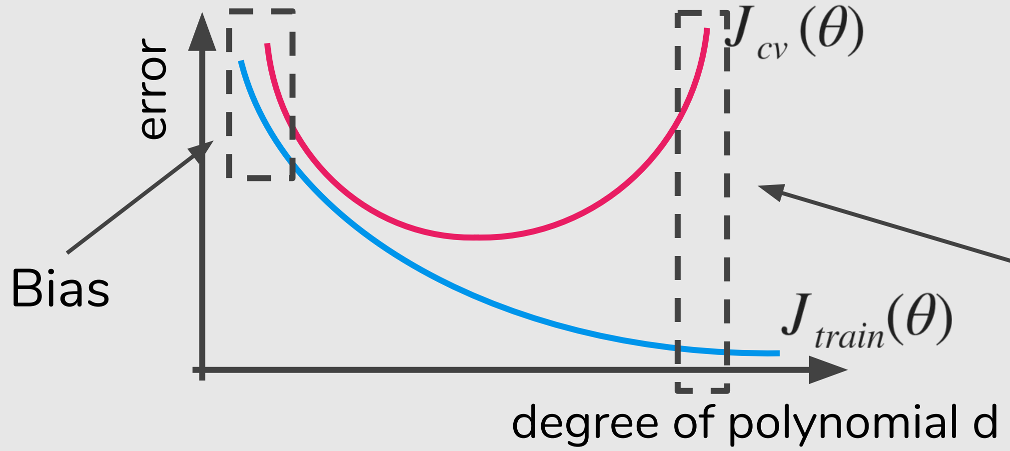
Diagnosing Bias vs. Variance

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?

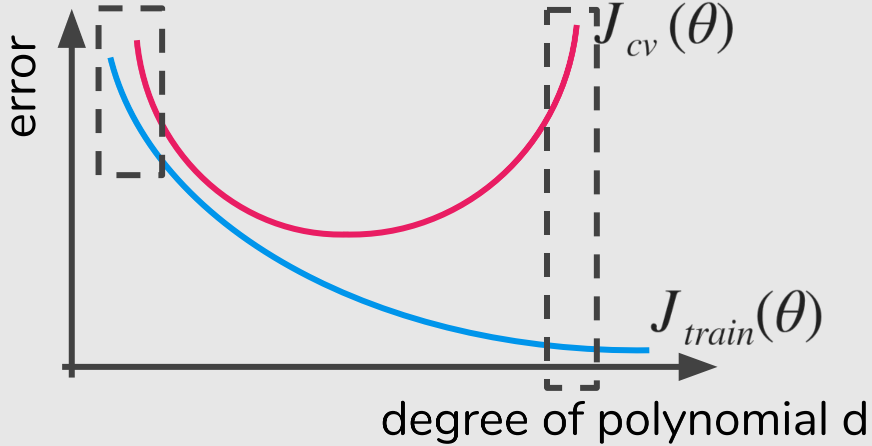


Bias (underfit):

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):

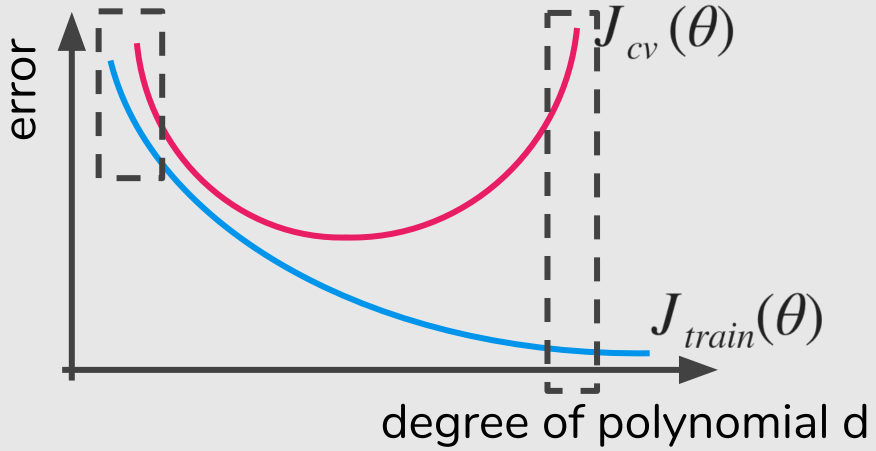
$J_{train}(\theta)$ will be high

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

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

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

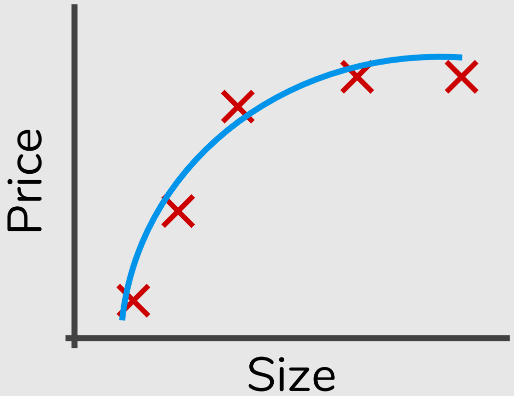
Variance (overfit):

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

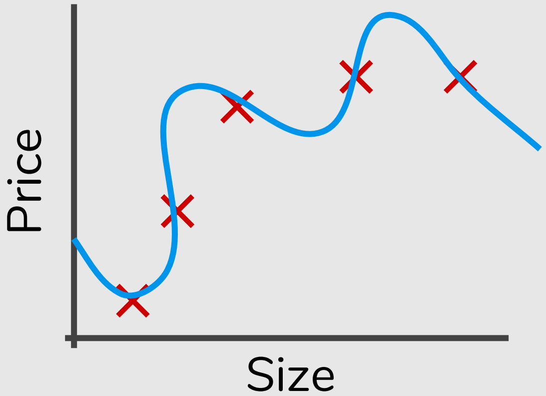
$J_{cv}(\theta) \gg J_{train}(\theta)$

Cost Function

Intuition

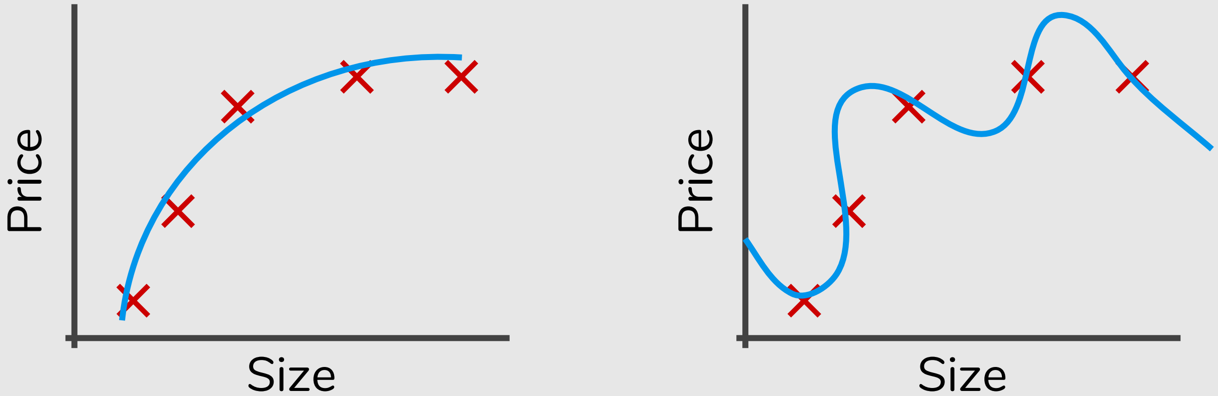


$$\theta_0 + \theta_1 x + \theta_2 x^2$$



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

Intuition

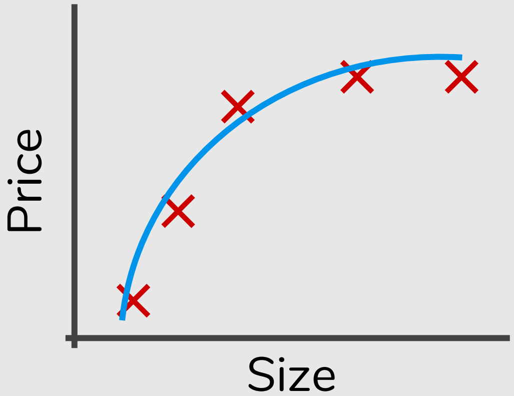


$$\theta_0 + \theta_1 x + \theta_2 x^2 \qquad \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

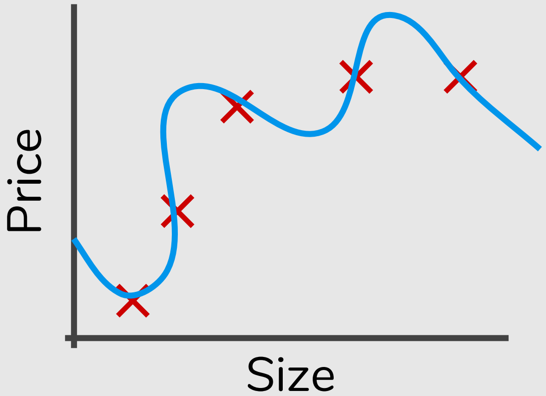
Suppose we penalize and make θ_3, θ_4 really small.

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

Intuition



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

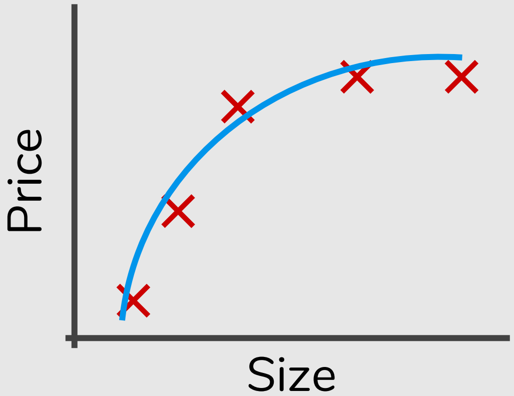


$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

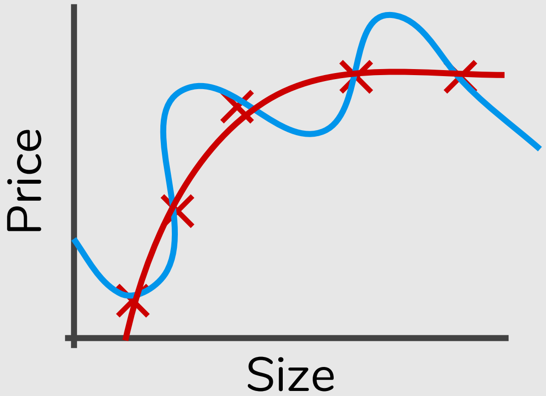
Suppose we penalize and make θ_3, θ_4 really small.

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

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

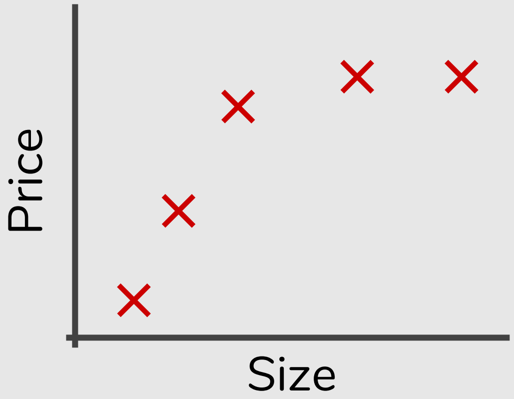
Regularization parameter

$$J(\theta) = \frac{1}{2m}\left[\underbrace{\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2}_{\text{to fit the training data well}} + \boxed{\lambda}\underbrace{\sum_{j=1}^{n}\theta_j^2}_{\text{to keep the parameters small}}\right]$$

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 $\lambda = 10^{10}$)?

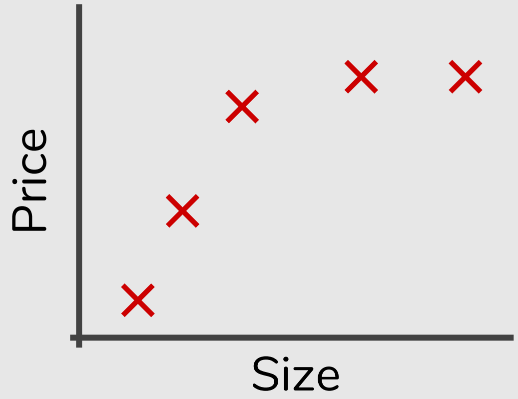


$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

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 $\lambda = 10^{10}$)?

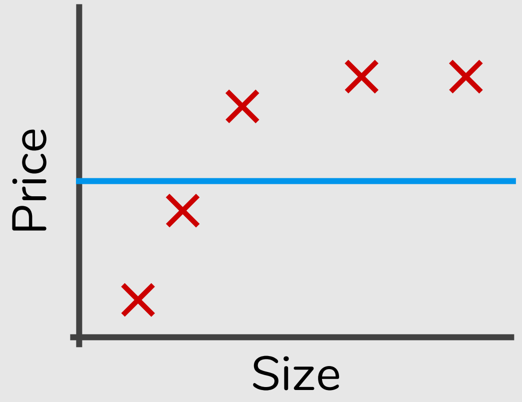


$$\theta_0 + \cancel{\theta_1 x} + \cancel{\theta_2 x^2} + \cancel{\theta_3 x^3} + \cancel{\theta_4 x^4}$$

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 $\lambda = 10^{10}$)?



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

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 $j = \cancel{0}, 1, \ldots, n$)

Gradient Descent

repeat {

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

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

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

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 $j = \cancel{0}\ 1, \ldots, n$)

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

Gradient Descent

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 $j = 1, \ldots, n$)

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

Normal Equation

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

Normal Equation

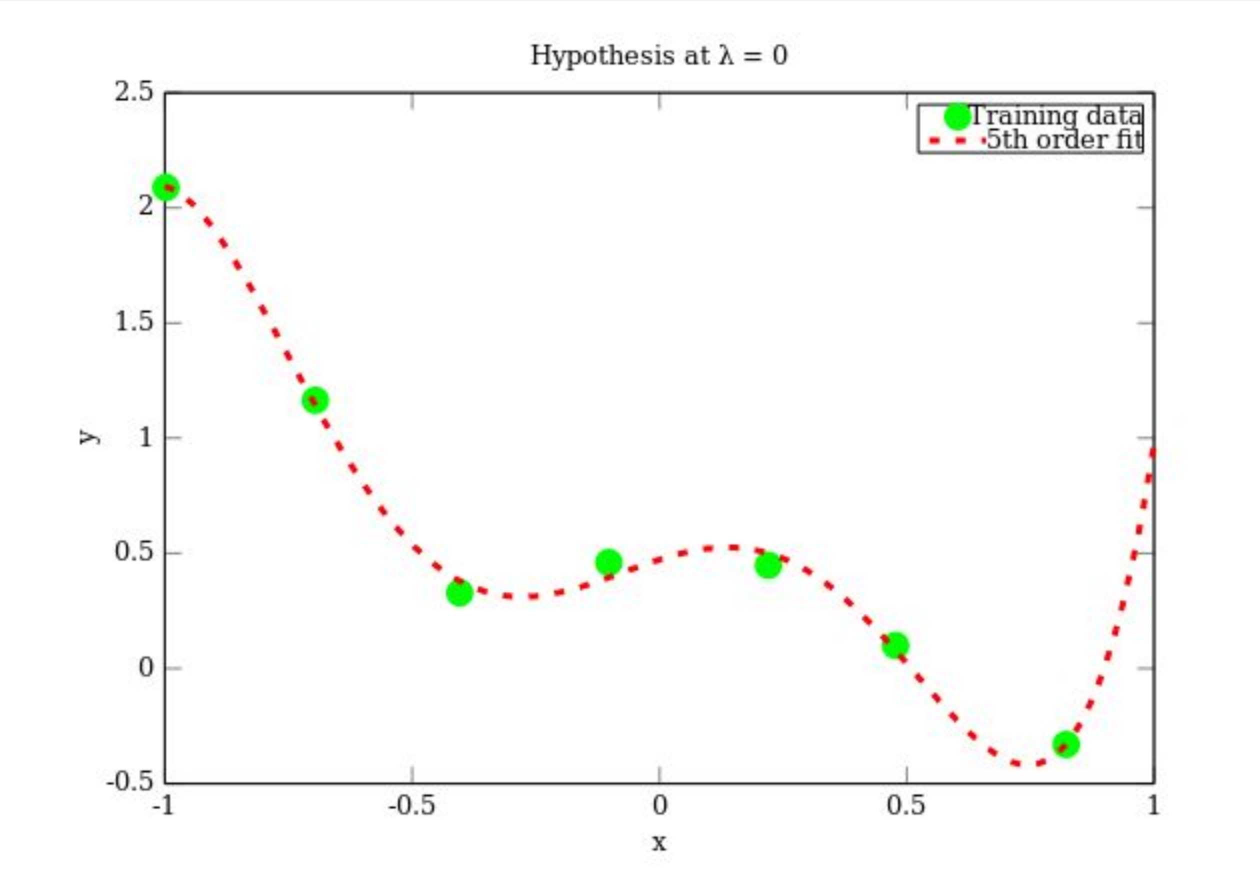
$$X = \begin{bmatrix} — (x^{(1)})^{\mathrm{T}} — \\ — (x^{(2)})^{\mathrm{T}} — \\ — \vdots — \\ — (x^{(m)})^{\mathrm{T}} — \end{bmatrix} \quad y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix} \quad \theta = (X^T X)^{-1} X^T y$$

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

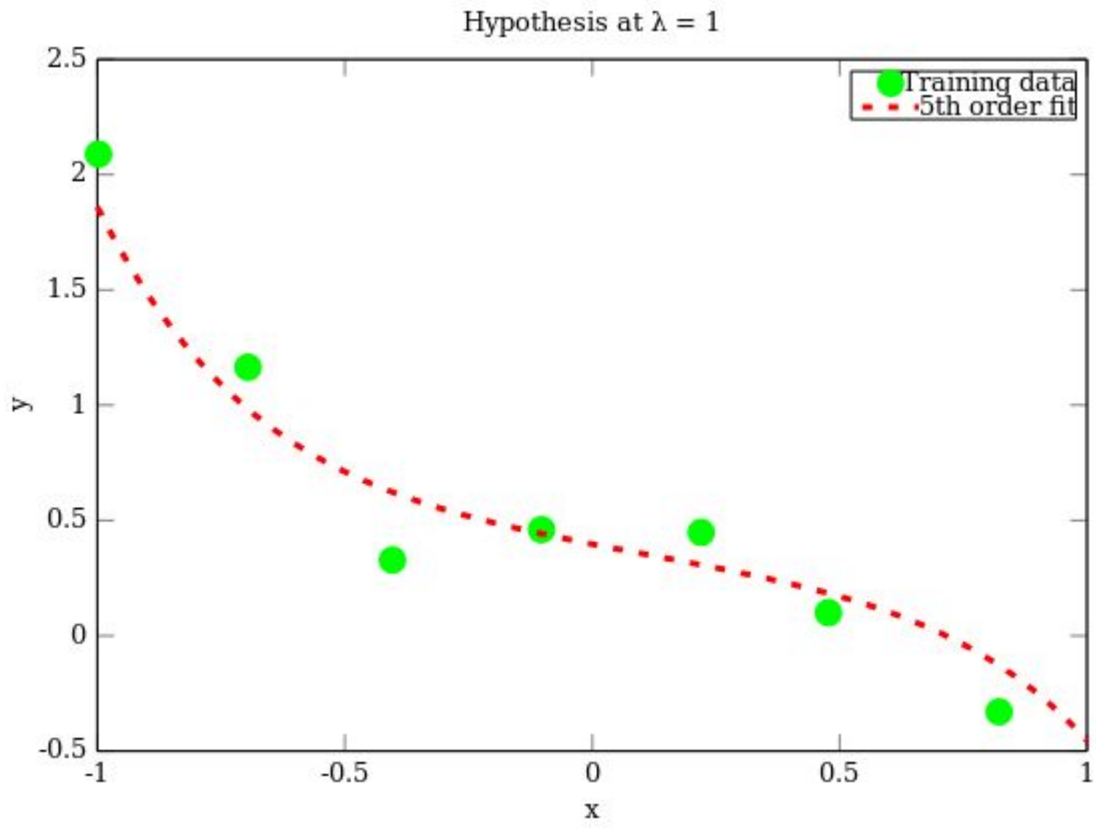
Normal Equation

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

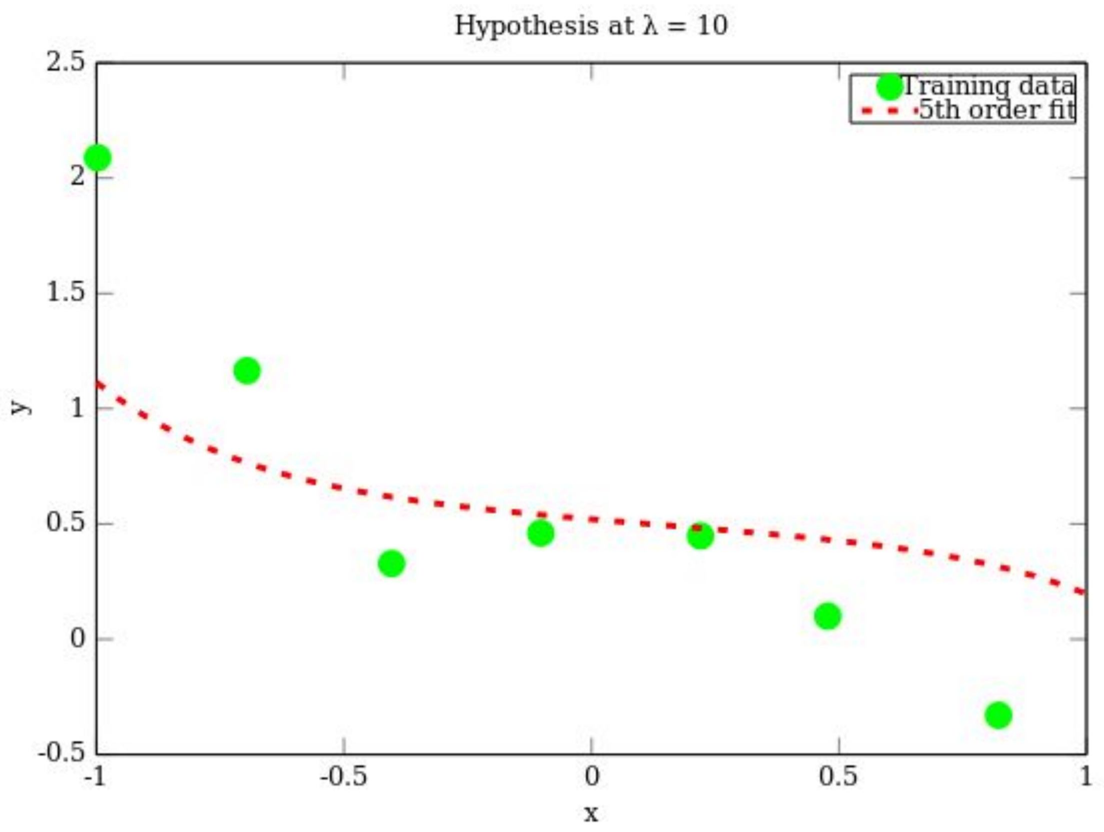
$$\theta = \left(X^T X + \lambda \begin{bmatrix} 0 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix} \right)^{-1} X^T y$$



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 $j = \cancel{0,}\ 1, \ldots, n$)

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

Gradient Descent

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

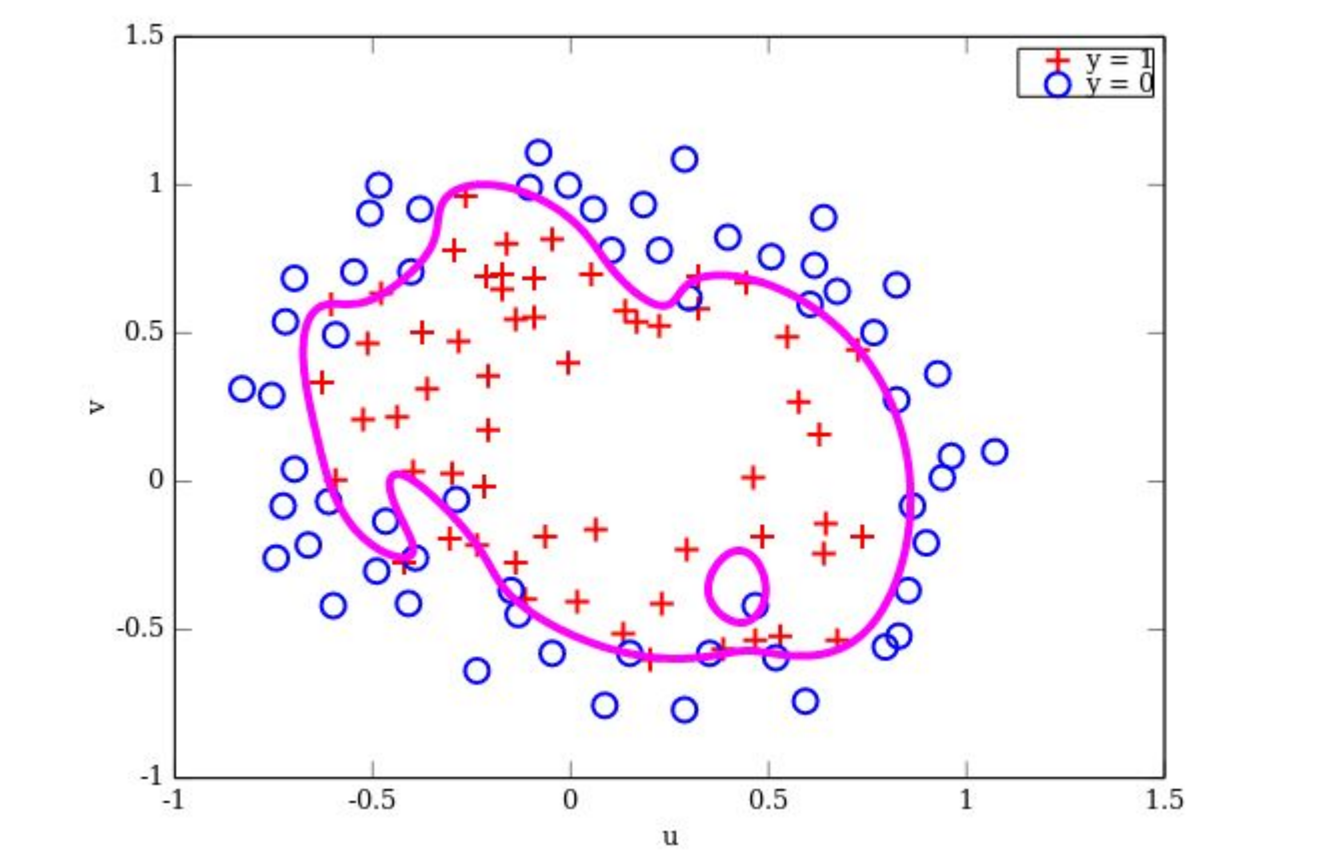
repeat {

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

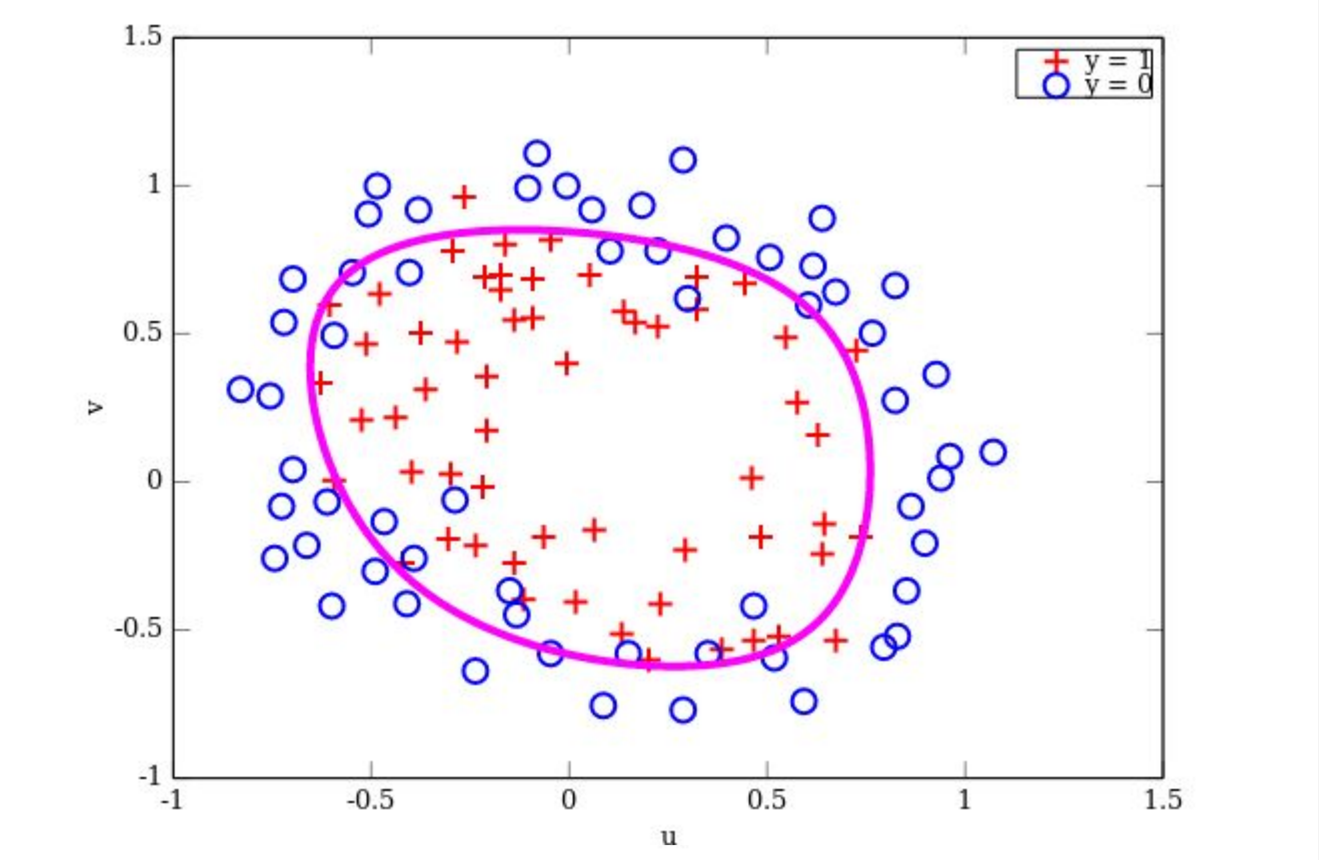
$$\theta_j := \theta_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} + \frac{\lambda}{m} \theta_j \right]$$

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

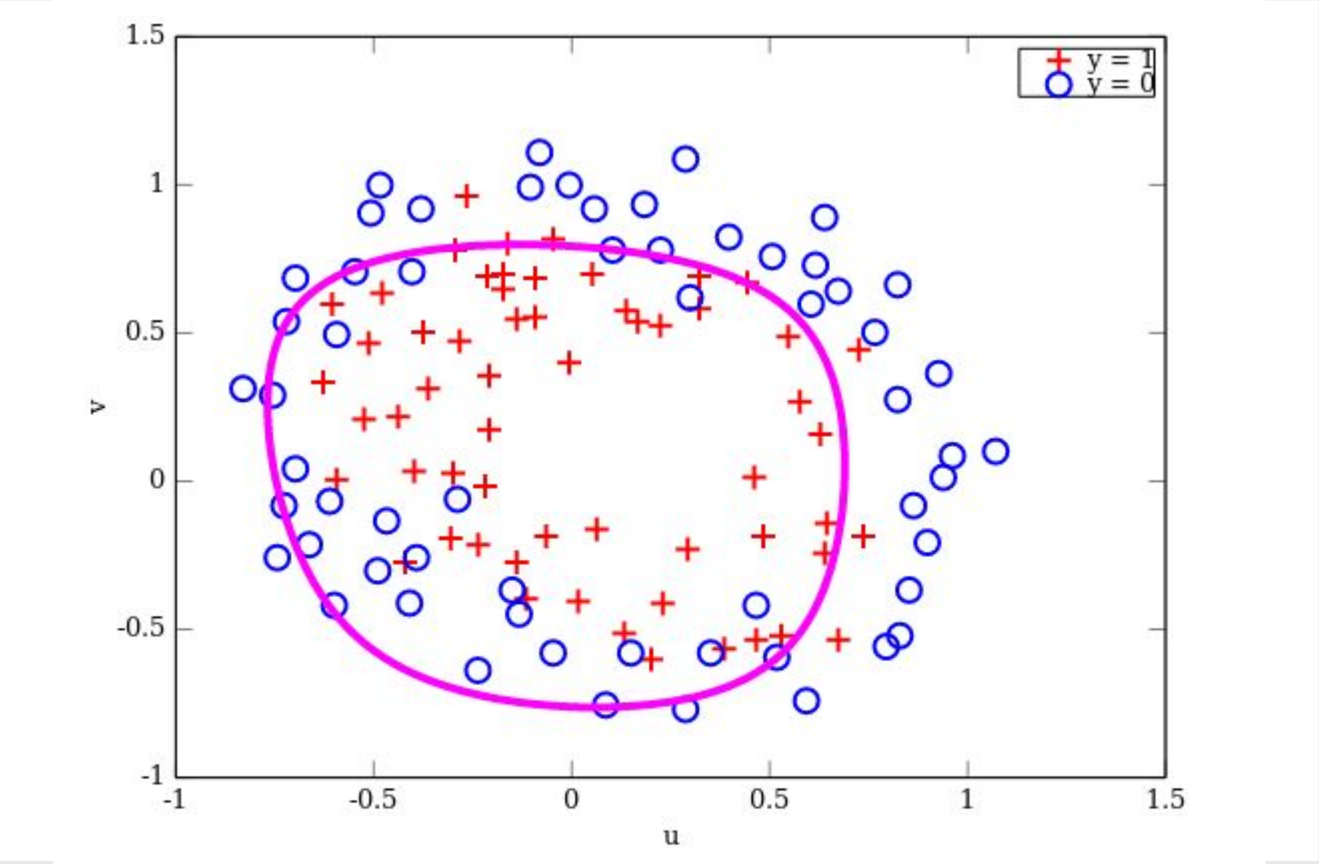
$$\theta_j := \theta_j (1 - \alpha \frac{\lambda}{m}) - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$



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