CoE202
Fundamentals of Artificial intelligence
<Big Data Analysis and Machine Learning>

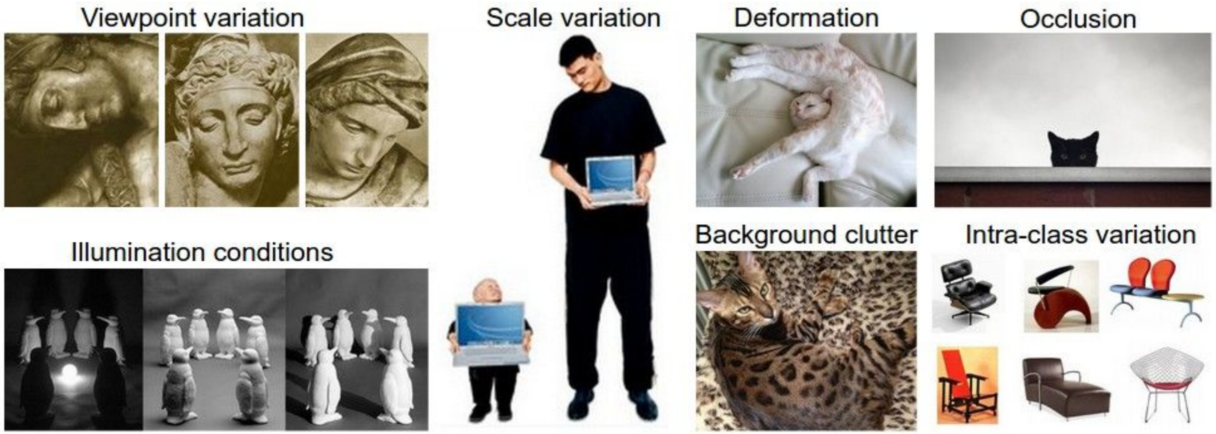
Linear and Polynomial Regression

Prof. Young-Gyu Yoon
School of EE, KAIST

Contents

- Recap
- Machine learning
- Supervised learning
- Linear regression
- Polynomial regression
- House price prediction problem

Recap: Challenges in image classification



- **Viewpoint variation**
- **Scale variation**
- **Deformation**
- **Occlusion**
- **Illumination conditions**
- **Background clutter**
- **Intra-class variation**

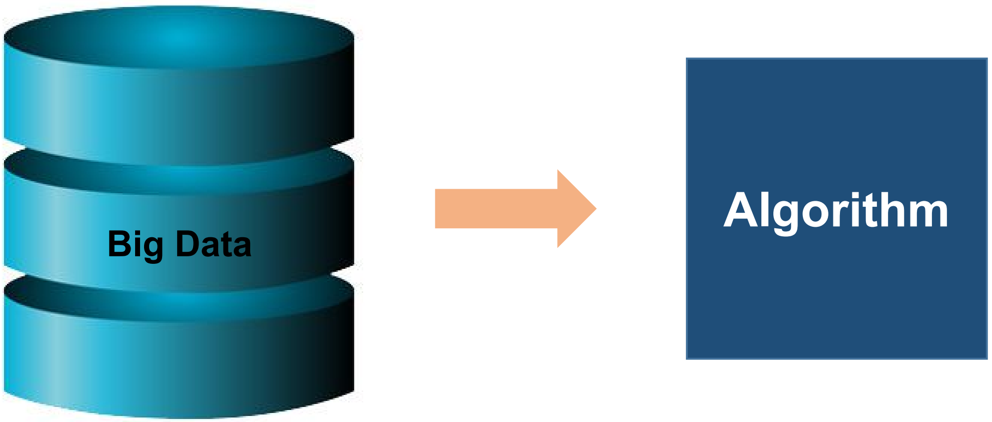
Recap: Question






- Despite all these issues, we (human) have no problem in recognizing that these are cats
- How can our algorithms do the same?

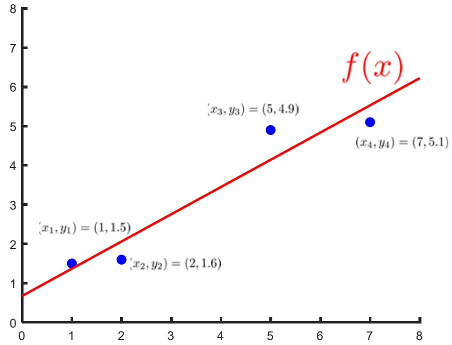
Recap: Data-driven approach



- What if we can **design a program** <u>that can analyze the data and make its own algorithm</u>?
  - Give all possible variations (viewpoint, scale, etc) and just let the program make the algorithm

Recap: Goal

Regression



$$f(x) = 0.6934x + 0.6747$$

$$\approx$$

Classification

CAT

Types of machine learning

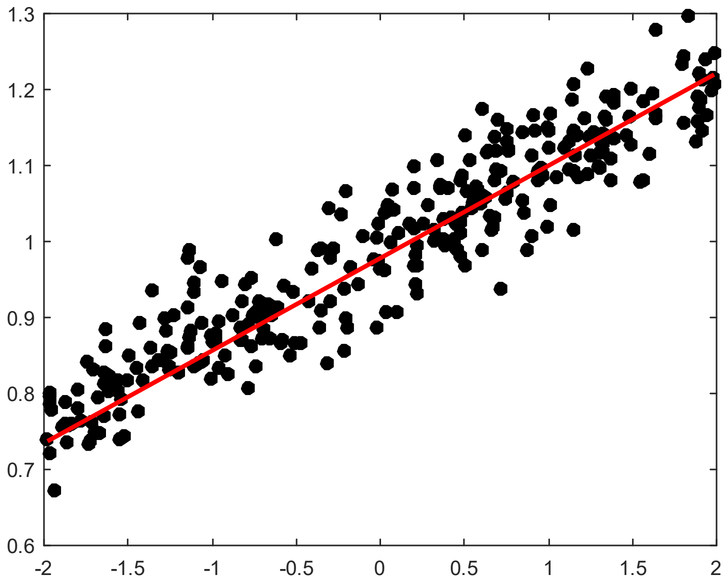
- **Supervised learning**: learning a function that maps an input to an output based on example input-output pairs

- **Unsupervised learning**: looking for previously undetected patterns in a data set with no pre-existing labels and without human supervision

- **Reinforcement learning**: enabling an agent to learn in an interactive environment by trial and error using feedback from its own actions and experiences

Question

- How many of you has ever done linear fitting?



Where we are going

- Discuss the definition of machine learning & supervised learning
- Show that linear fitting is a “perfect” example of supervised learning
- Discuss how linear fitting works

- Then, we (pretty much) understand supervised learning ☺

Supervised learning

- **Supervised learning**: learning a function that maps an input to an output based on example input-output pairs

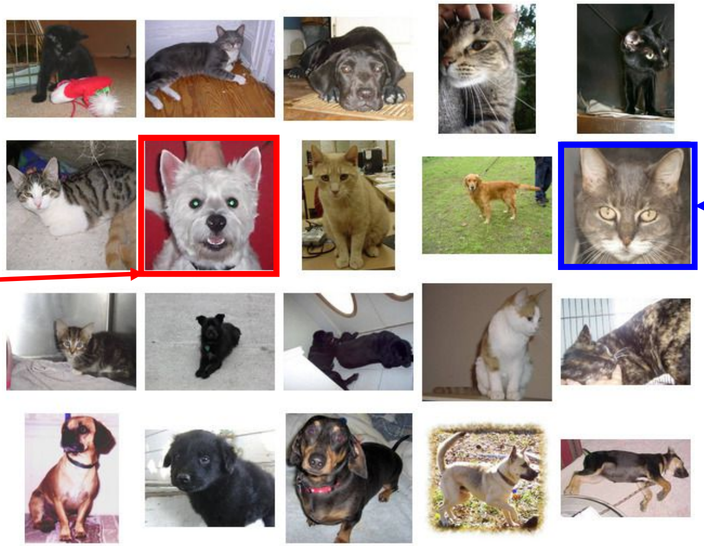
$$\text{For a data set } \mathcal{D} = \{(\vec{x_1}, \vec{y_1}), (\vec{x_2}, \vec{y_2}), \cdots, (\vec{x_N}, \vec{y_N})\}$$

$$\text{Seeks a function } f : X \to Y$$

$$\text{Such that a loss function } \mathcal{L} : X \times Y \to \mathcal{R} \text{ is minimized}$$

Supervised learning: image classification

For a data set $\mathcal{D} = \{(\vec{x_1}, \vec{y_1}), (\vec{x_2}, \vec{y_2}), \cdots, (\vec{x_N}, \vec{y_N})\}$



$\overrightarrow{x_7}$

$\overrightarrow{y_7} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

$\overrightarrow{x_{10}}$

$\overrightarrow{y_{10}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

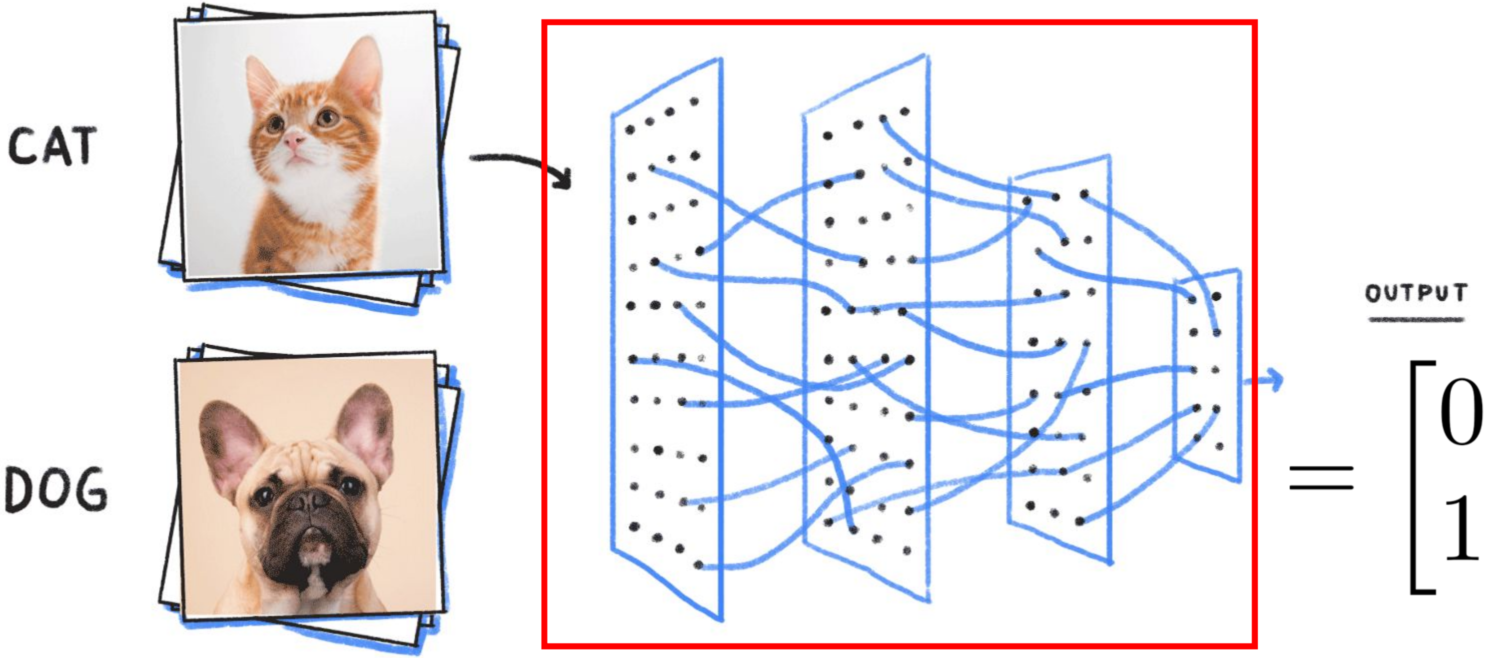
Supervised learning: image classification

Seeks a function $f : X \to Y$

$f\Big($  $\Big) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

$f\Big($  $\Big) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Supervised learning: image classification



$$= \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Neural network can be used to construct the "function" f

Supervised learning: Linear regression

- **Linear regression** is an approach to model the relationship between a dependent variable and one or more independent variables as a linear function

$$f(\vec{x};\vec{\theta},\theta_0) = \vec{\theta}\cdot\vec{x} + \theta_0 = \sum_{i=1}^{d} \theta_i x_i + \theta_0$$

this is NOT sample index

- **Simple linear regression**: one dependent variable and one independent variable

$$f(x;\theta_0,\theta_1) = \theta_1 x + \theta_0$$

Simple linear regression

For a data set $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$

Seeks a function $f : X \to Y$

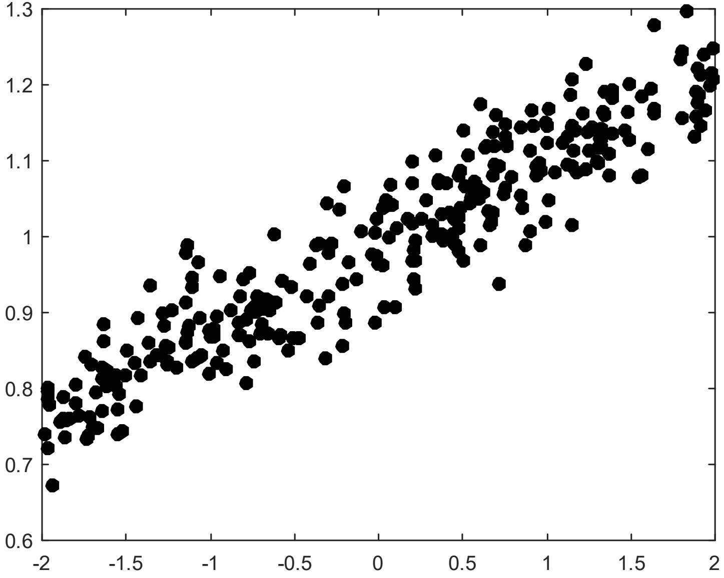
$$f(x; \theta_0, \theta_1) = \theta_1 x + \theta_0$$

Such that a loss function $\mathcal{L} : X \times Y \to \mathcal{R}$ is minimized

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} (f(x_i) - y_i)^2$$

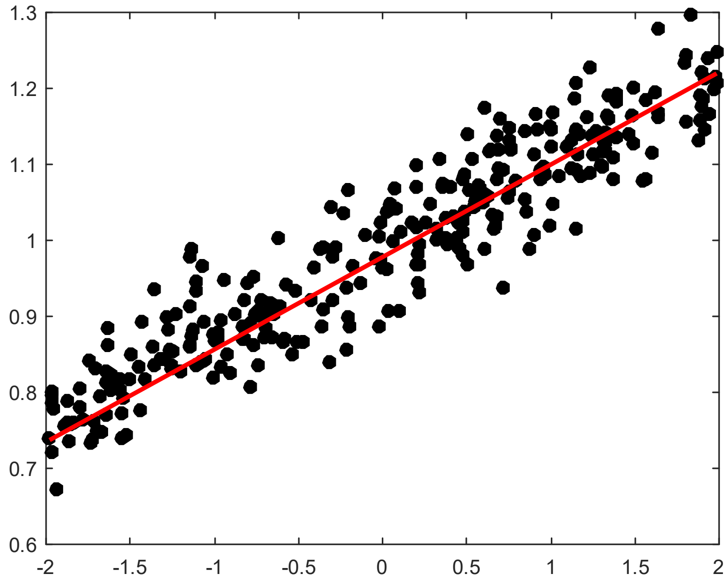
Simple linear regression

For a data set $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$



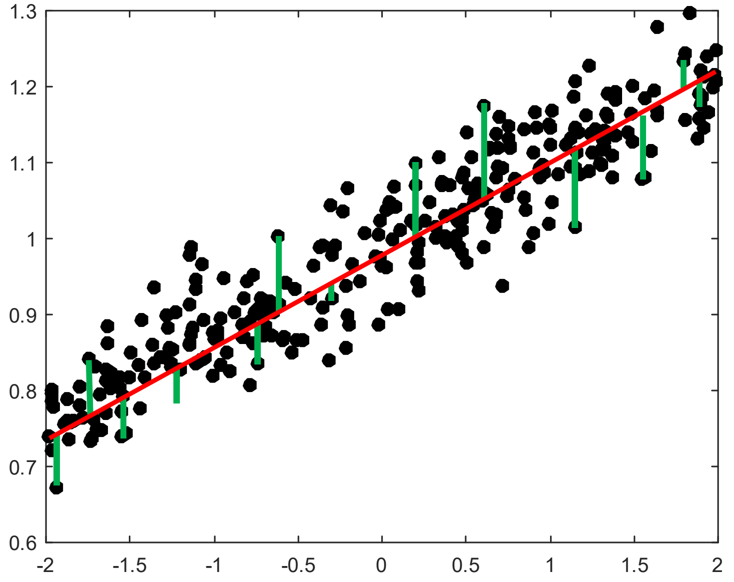
Simple linear regression

Seeks a function $f(x; \theta_0, \theta_1) = \theta_1 x + \theta_0$



Simple linear regression

Such that the mean squared error,
$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(f(x_i) - y_i)^2$, is minimized



The “training”: finding the function

Seeks a function $f(x;\theta_0,\theta_1) = \theta_1 x + \theta_0$

Such that the mean squared error is minimized

- Finding f is equivalent to finding θ_1 and θ_0
- Training: finding f (or θ_1 and θ_0) from

$$\text{data set } \mathcal{D} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$$

- How can we find θ_1 and θ_0?

*In machine learning, training refers to the process of finding the model parameters that minimizes the loss function

Training (simple example)

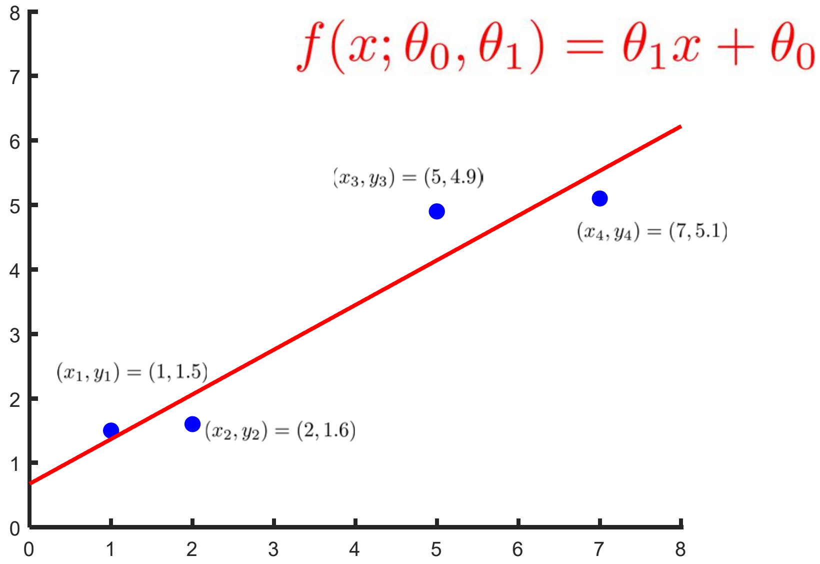
$$\text{data set } \mathcal{D} = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\}$$

$$(x_1, y_1) = (1, 1.5)$$

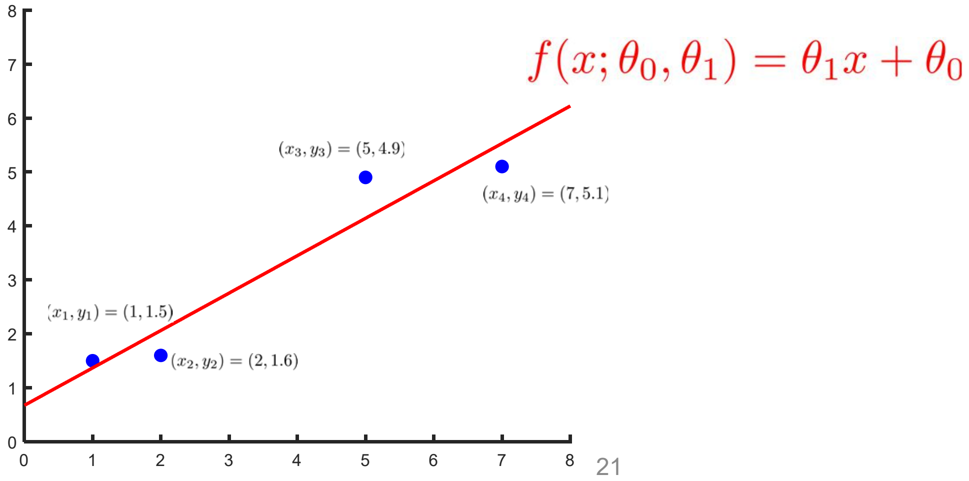
$$(x_2, y_2) = (2, 1.6)$$

$$(x_3, y_3) = (5, 4.9)$$

$$(x_4, y_4) = (7, 5.1)$$



Training (simple example)



$$\mathcal{L} = \tfrac{1}{N}\textstyle\sum_{i=1}^{N}(f(x_i) - y_i)^2$$

$$= \tfrac{1}{4}\textstyle\sum_{i=1}^{4}(\theta_1 x_i + \theta_0 - y_i)^2$$

$$= \tfrac{1}{4}\{(\theta_1\cdot 1+\theta_0-1.5)^2+(\theta_1\cdot 2+\theta_0-1.6)^2+(\theta_1\cdot 5+\theta_0-4.9)^2+(\theta_1\cdot 7+\theta_0-5.1)^2\}$$

Training (simple example)

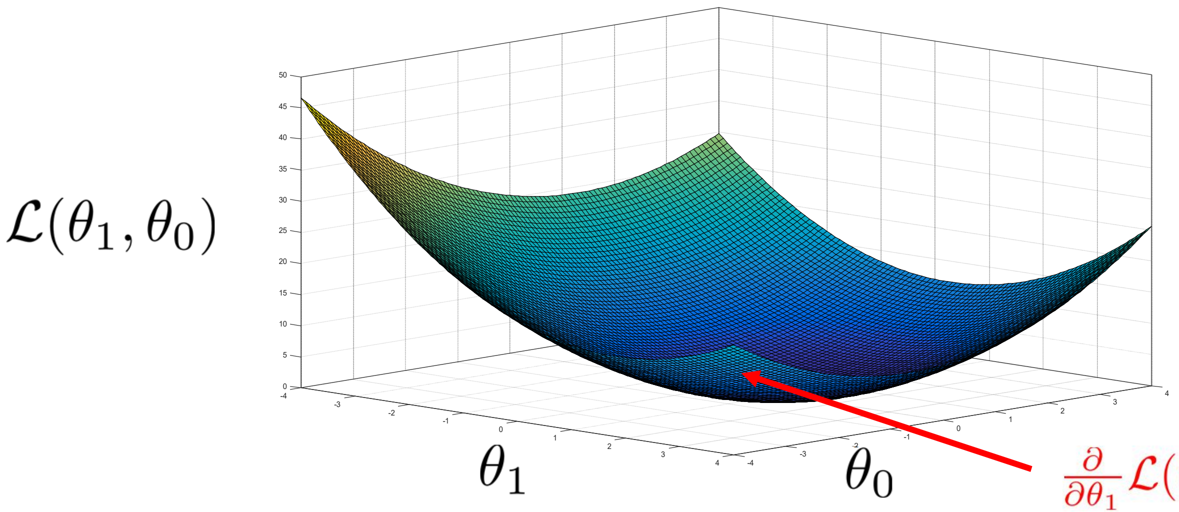
$$\mathcal{L}(\theta_1, \theta_0)$$
$$= \frac{1}{4}\{(\theta_1 \cdot 1 + \theta_0 - 1.5)^2 + (\theta_1 \cdot 2 + \theta_0 - 1.6)^2 + (\theta_1 \cdot 5 + \theta_0 - 4.9)^2 + (\theta_1 \cdot 7 + \theta_0 - 5.1)^2\}$$

Training: finding θ_1 and θ_0 that minimizes this loss function

$$\mathcal{L}(\theta_1, \theta_0) = \theta_0^2 + 7.5\theta_0\theta_1 - 6.55\theta_0 + 19.75\theta_1^2 - 32.45\theta_1 + 13.7075$$

Training (simple example)

$$\mathcal{L}(\theta_1, \theta_0) = \theta_0^2 + 7.5\theta_0\theta_1 - 6.55\theta_0 + 19.75\theta_1^2 - 32.45\theta_1 + 13.7075$$



$$\frac{\partial}{\partial \theta_1}\mathcal{L}(\theta_1, \theta_0) = 0$$

$$\frac{\partial}{\partial \theta_0}\mathcal{L}(\theta_1, \theta_0) = 0$$

Training (simple example): partial derivative

$$\mathcal{L}(\theta_1, \theta_0) = \theta_0^2 + 7.5\theta_0\theta_1 - 6.55\theta_0 + 19.75\theta_1^2 - 32.45\theta_1 + 13.7075$$

$$\frac{\partial}{\partial\theta_1}\mathcal{L}(\theta_1, \theta_0) = \frac{\partial}{\partial\theta_1}\{\theta_0^2 + 7.5\theta_0\theta_1 - 6.55\theta_0 + 19.75\theta_1^2 - 32.45\theta_1 + 13.7075\}$$

$$= 39.5\theta_1 + 7.5\theta_0 - 32.45$$

$$\frac{\partial}{\partial\theta_0}\mathcal{L}(\theta_1, \theta_0) = \frac{\partial}{\partial\theta_0}\{\theta_0^2 + 7.5\theta_0\theta_1 - 6.55\theta_0 + 19.75\theta_1^2 - 32.45\theta_1 + 13.7075\}$$

$$= 7.5\theta_1 + 2\theta_0 - 6.55$$

Training (simple example): partial derivative

$$\frac{\partial}{\partial \theta_1} \mathcal{L}(\theta_1, \theta_0) = 0$$

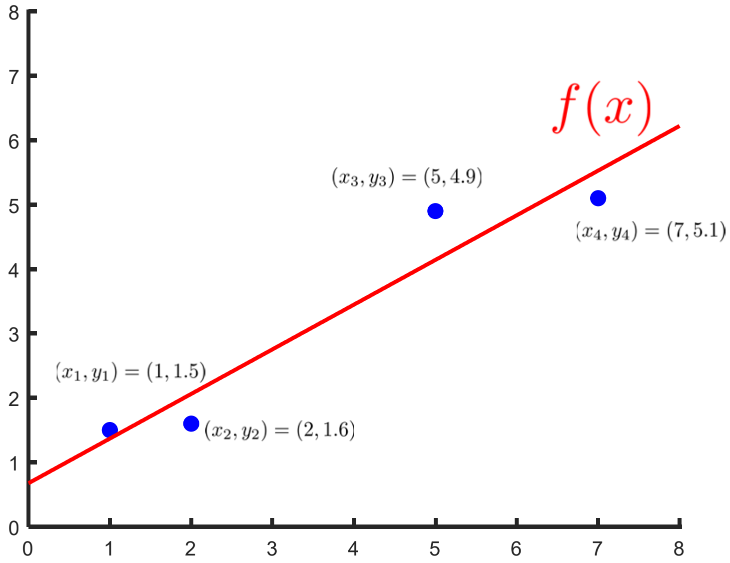
$$\frac{\partial}{\partial \theta_0} \mathcal{L}(\theta_1, \theta_0) = 0$$

$$39.5\theta_1 + 7.5\theta_0 = 32.45$$

$$7.5\theta_1 + 2\theta_0 = 6.55$$

$$\theta_0 = 0.6747$$

$$\theta_1 = 0.6934$$



$$f(x) = 0.6934x + 0.6747$$

Training (simple example): matrix representation

$$
\begin{aligned}
f(x_1) &= \theta_1 x_1 + \theta_0 \\
f(x_2) &= \theta_1 x_2 + \theta_0 \\
f(x_3) &= \theta_1 x_3 + \theta_0 \\
f(x_4) &= \theta_1 x_4 + \theta_0
\end{aligned}
$$

$$
\begin{bmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \\ f(x_4) \end{bmatrix} = \theta_0 \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + \theta_1 \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}
$$

$$
\begin{bmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \\ f(x_4) \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ 1 & x_4 \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}
$$

Training (simple example): matrix representation

$$(x_1, y_1) = (1, 1.5)$$
$$(x_2, y_2) = (2, 1.6)$$
$$(x_3, y_3) = (5, 4.9)$$
$$(x_4, y_4) = (7, 5.1)$$

$$\Rightarrow \quad X = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 5 \\ 1 & 7 \end{bmatrix} \qquad Y = \begin{bmatrix} 1.5 \\ 1.6 \\ 4.9 \\ 5.1 \end{bmatrix} \qquad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$$

$$X\theta - Y = \begin{bmatrix} \theta_1 + \theta_0 - 1.5 \\ 2\theta_1 + \theta_0 - 1.6 \\ 5\theta_1 + \theta_0 - 4.9 \\ 7\theta_1 + \theta_0 - 5.1 \end{bmatrix}$$

Training (simple example): matrix representation

$$(X\theta - Y)^T(X\theta - Y) = \begin{bmatrix} \theta_1 + \theta_0 - 1.5 \\ 2\theta_1 + \theta_0 - 1.6 \\ 5\theta_1 + \theta_0 - 4.9 \\ 7\theta_1 + \theta_0 - 5.1 \end{bmatrix}^T \begin{bmatrix} \theta_1 + \theta_0 - 1.5 \\ 2\theta_1 + \theta_0 - 1.6 \\ 5\theta_1 + \theta_0 - 4.9 \\ 7\theta_1 + \theta_0 - 5.1 \end{bmatrix}$$

$$= \{(\theta_1 \cdot 1 + \theta_0 - 1.5)^2 + (\theta_1 \cdot 2 + \theta_0 - 1.6)^2 + (\theta_1 \cdot 5 + \theta_0 - 4.9)^2 + (\theta_1 \cdot 7 + \theta_0 - 5.1)^2\}$$

$$\therefore \mathcal{L}(\theta) = \frac{1}{4}(X\theta - Y)^T(X\theta - Y)$$

Training (simple example): matrix representation

$$\mathcal{L}(\theta) = \frac{1}{4}(X\theta - Y)^T(X\theta - Y)$$

$$\nabla_\theta \mathcal{L}(\theta) = \begin{bmatrix} \frac{\partial}{\partial \theta_0}\mathcal{L}(\theta_1, \theta_0) \\ \frac{\partial}{\partial \theta_1}\mathcal{L}(\theta_1, \theta_0) \end{bmatrix} = 0$$

$$\nabla_\theta \mathcal{L}(\theta) = \frac{1}{2}X^T(X\theta - Y) = 0$$

$$X^TY = X^TX\theta$$

$$\theta = (X^TX)^{-1}X^TY$$

Training (simple example): matrix representation

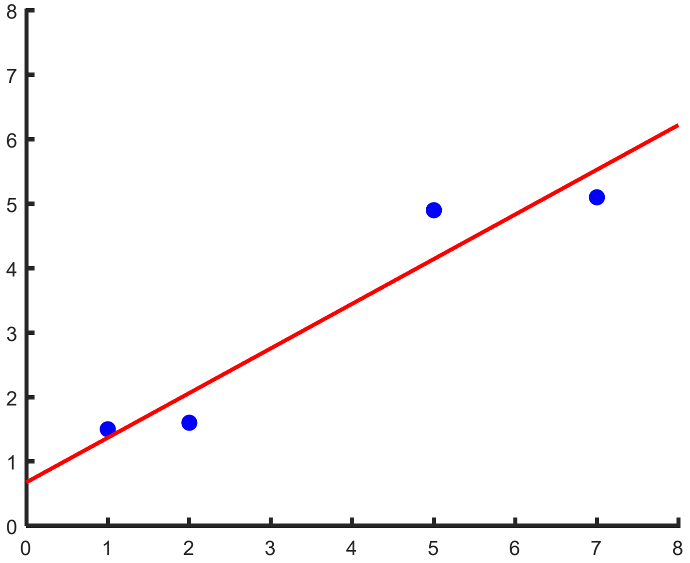
$$\theta = (X^T X)^{-1} X^T Y$$

$$= \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 5 & 7 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 5 \\ 1 & 7 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 5 & 7 \end{bmatrix} \begin{bmatrix} 1.5 \\ 1.6 \\ 4.9 \\ 5.1 \end{bmatrix}$$

$$= \begin{bmatrix} 4 & 15 \\ 15 & 79 \end{bmatrix}^{-1} \begin{bmatrix} 13.1 \\ 64.9 \end{bmatrix} = \begin{bmatrix} 0.8681 & -0.1648 \\ -0.1648 & 0.0440 \end{bmatrix} \begin{bmatrix} 13.1 \\ 64.9 \end{bmatrix} = \begin{bmatrix} 0.6747 \\ 0.6934 \end{bmatrix}$$

$$\theta_0 = 0.6747 \qquad \theta_1 = 0.6934$$

Training (simple example): matrix representation



$$\theta_0 = 0.6747$$

$$\theta_1 = 0.6934$$

$$f(x) = 0.6934x + 0.6747$$

Training (simple example): matrix representation

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

$$X : m \times 2$$

$$X^T : 2 \times m$$

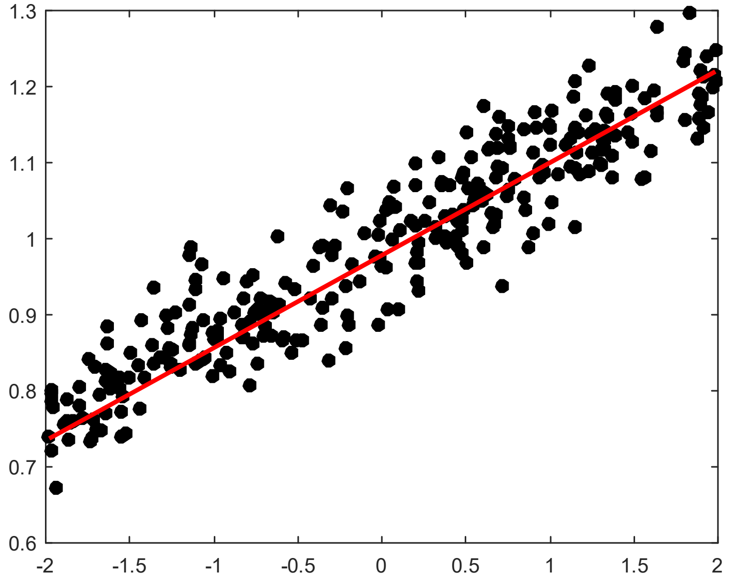
$$X^T X : 2 \times 2$$

$$(X^T X)^{-1} : 2 \times 2$$

$$(X^T X)^{-1} X^T : 2 \times m$$

$$Y : m \times 1$$

$$\theta = (X^T X)^{-1} X^T Y : 2 \times 1$$



- This relation can be used to find θ for a large m (number of data points)!

Training (simple example): matrix representation

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

Q) Pop quiz: what happens if m < 2

$$X : m \times 2$$

$$X^T : 2 \times m$$

$$X^T X : 2 \times 2$$

$$(X^T X)^{-1} : 2 \times 2$$

$$(X^T X)^{-1} X^T : 2 \times m$$

$$Y : m \times 1$$

$$\theta = (X^T X)^{-1} X^T Y : 2 \times 1$$

- This relation can be used to find θ for a large m (number of data points)!

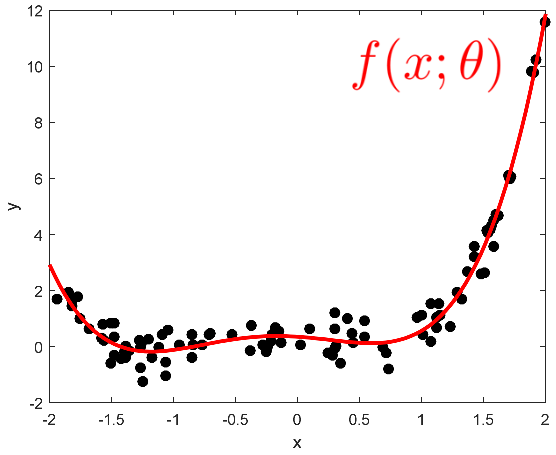
Polynomial regression

- **Polynomial regression** is an approach to model the relationship between a dependent variable and one or more independent variables as an nth order polynomial function

$$f(x;\theta) = \sum_{l=1}^{k} \theta_l x^l + \theta_0$$

Polynomial regression

- Almost everything is the same as linear regression



For a data set $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$

Seeks a function $f : X \to Y$

$$f(x;\theta) = \sum_{l=1}^{k} \theta_l x^l + \theta_0$$

Such that a loss function $\mathcal{L} : X \times Y \to \mathcal{R}$ is minimized

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} (f(x_i) - y_i)^2$$

Polynomial regression: matrix representation

- 2nd order polynomial regression for the same data set

$$f(x_1) = \theta_2 x_1^2 + \theta_1 x_1 + \theta_0$$

$$f(x_2) = \theta_2 x_2^2 + \theta_1 x_2 + \theta_0$$

$$f(x_3) = \theta_2 x_3^2 + \theta_1 x_3 + \theta_0$$

$$f(x_4) = \theta_2 x_4^2 + \theta_1 x_4 + \theta_0$$

$$\begin{bmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \\ f(x_4) \end{bmatrix} = \theta_0 \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + \theta_1 \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \theta_2 \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_3^2 \\ x_4^2 \end{bmatrix}$$

$$\begin{bmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \\ f(x_4) \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \\ 1 & x_4 & x_4^2 \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix}$$

Feature matrix

$$\Phi = \begin{bmatrix} \phi(x_1) & \phi(x_2) & \phi(x_3) & \cdots & \phi(x_n) \end{bmatrix}^T$$

$$\phi(x) = \begin{bmatrix} 1 \\ x^1 \\ x^2 \\ \vdots \\ x^K \end{bmatrix}$$

$$(x_1, y_1) = (1, 1.5)$$

$$(x_2, y_2) = (2, 1.6)$$

$$(x_3, y_3) = (5, 4.9)$$

$$(x_4, y_4) = (7, 5.1)$$

$$\xrightarrow{\text{K=2}}$$

$$\Phi = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 5 & 25 \\ 1 & 7 & 49 \end{bmatrix}$$

Polynomial regression

$$\begin{bmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \\ f(x_4) \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \\ 1 & x_4 & x_4^2 \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix}$$

$$\Phi = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 5 & 25 \\ 1 & 7 & 49 \end{bmatrix} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix} \quad Y = \begin{bmatrix} 1.5 \\ 1.6 \\ 4.9 \\ 5.1 \end{bmatrix}$$

$$\begin{aligned} \mathcal{L} &= \tfrac{1}{N} \textstyle\sum_{i=1}^{N} (f(x_i) - y_i)^2 \;\; = \tfrac{1}{4} \textstyle\sum_{i=1}^{4} (\theta_2 x_i^2 + \theta_1 x_i + \theta_0 - y_i)^2 \\ &= \tfrac{1}{4} \{ (\theta_2 \cdot 1 + \theta_1 \cdot 1 + \theta_0 - 1.5)^2 + (\theta_2 \cdot 4 + \theta_1 \cdot 2 + \theta_0 - 1.6)^2 \\ &\quad + (\theta_2 \cdot 25 + \theta_1 \cdot 5 + \theta_0 - 4.9)^2 + (\theta_2 \cdot 49 + \theta_1 \cdot 7 + \theta_0 - 5.1)^2 \} \end{aligned}$$

$$\Phi\theta - Y = \begin{bmatrix} \theta_2 + \theta_1 + \theta_0 - 1.5 \\ 4\theta_2 + 2\theta_1 + \theta_0 - 1.6 \\ 25\theta_2 + 5\theta_1 + \theta_0 - 4.9 \\ 49\theta_2 + 7\theta_1 + \theta_0 - 5.1 \end{bmatrix}$$

Polynomial regression

$$\theta = (\Phi^T \Phi)^{-1} \Phi^T Y$$

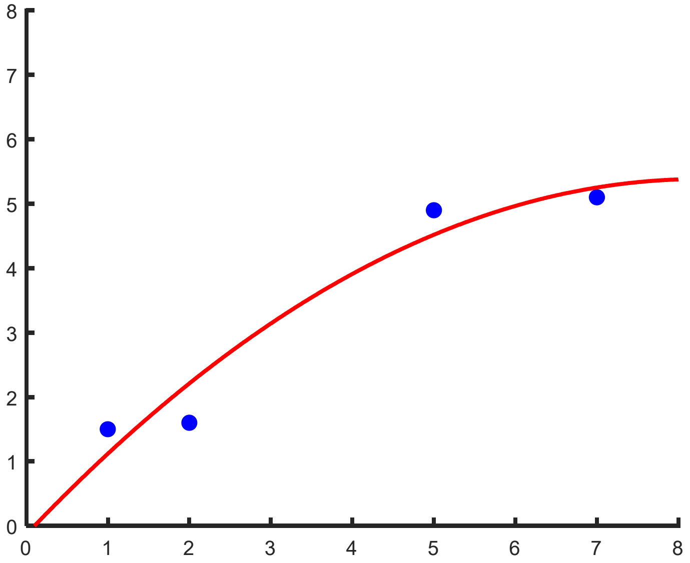
$$= \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 5 & 7 \\ 1 & 4 & 25 & 49 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 5 & 25 \\ 1 & 7 & 49 \end{bmatrix} \right)^{-1} \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 5 & 7 \\ 1 & 4 & 25 & 49 \end{bmatrix} \begin{bmatrix} 1.5 \\ 1.6 \\ 4.9 \\ 5.1 \end{bmatrix}$$

$$= \begin{bmatrix} 4 & 15 & 79 \\ 15 & 79 & 477 \\ 79 & 477 & 3043 \end{bmatrix}^{-1} \begin{bmatrix} 13.1 \\ 64.9 \\ 380.3 \end{bmatrix} = \begin{bmatrix} 3.0292 & -1.8743 & 0.2152 \\ -1.8743 & 1.3962 & -0.1702 \\ 0.2152 & -0.1702 & 0.0214 \end{bmatrix} \begin{bmatrix} 13.1 \\ 64.9 \\ 380.3 \end{bmatrix}$$

$$= \begin{bmatrix} -0.1339 \\ 1.3331 \\ -0.0805 \end{bmatrix}$$

$$\theta_0 = -0.1339 \quad \theta_1 = 1.3331 \quad \theta_2 = -0.0805$$

Polynomial regression



$$\theta_0 = -0.1339$$

$$\theta_1 = 1.3331$$

$$\theta_2 = -0.0805$$

$$f(x) = -0.0805x^2 + 1.3331x - 0.1339$$

Polynomial regression

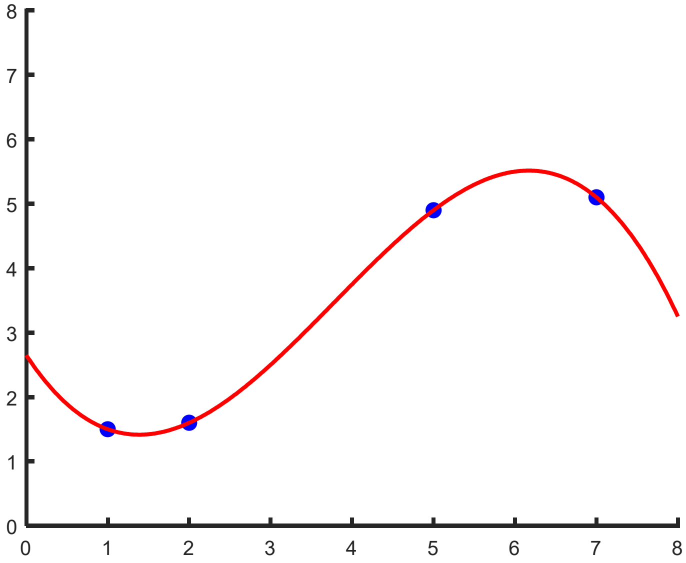
- 3rd order polynomial regression?

$$(x_1, y_1) = (1, 1.5)$$
$$(x_2, y_2) = (2, 1.6)$$
$$(x_3, y_3) = (5, 4.9)$$
$$(x_4, y_4) = (7, 5.1)$$

$$\Rightarrow \quad \Phi = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 5 & 25 & 125 \\ 1 & 7 & 49 & 343 \end{bmatrix} \qquad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} \quad Y = \begin{bmatrix} 1.5 \\ 1.6 \\ 4.9 \\ 5.1 \end{bmatrix}$$

$$\theta = (\Phi^T \Phi)^{-1} \Phi^T Y \quad = \begin{bmatrix} 2.6500 \\ -1.9250 \\ 0.8500 \\ -0.0750 \end{bmatrix}$$

Polynomial regression



$$\theta_0 = 2.6500$$

$$\theta_1 = -1.9250$$

$$\theta_2 = 0.8500$$

$$\theta_3 = -0.0750$$

$$f(x) = -0.075x^3 + 0.85x^2 - 1.925x + 2.65$$

Linear regression vs. Polynomial regression

Simple linear regression

$$f(x; \theta_0, \theta_1) = \theta_1 x + \theta_0$$

$$\begin{bmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \\ f(x_4) \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ 1 & x_4 \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$$

Linear regression

$$f(\vec{x}; \vec{\theta}, \theta_0) = \vec{\theta} \cdot \vec{x} + \theta_0 = \sum_{i=1}^{d} \theta_i x_i + \theta_0$$

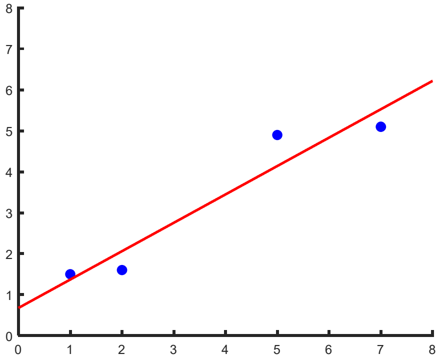
compare!

Polynomial regression

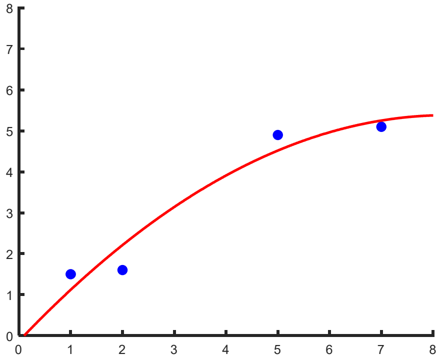
$$f(x; \theta) = \sum_{l=1}^{k} \theta_l x^l + \theta_0$$

$$\begin{bmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \\ f(x_4) \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \\ 1 & x_4 & x_4^2 \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix}$$

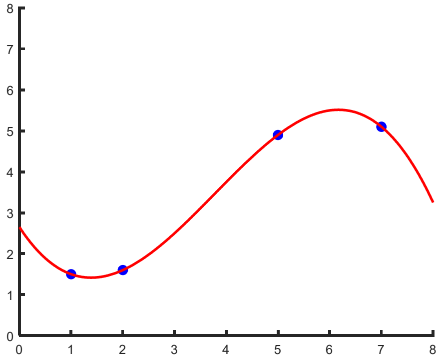
Choosing the “model”



vs.

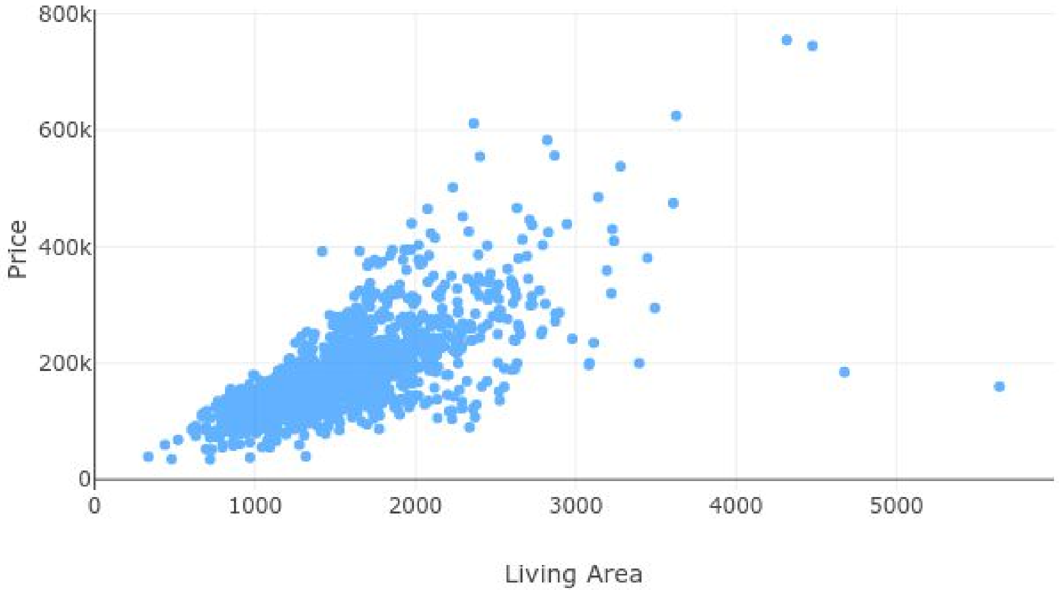


vs.



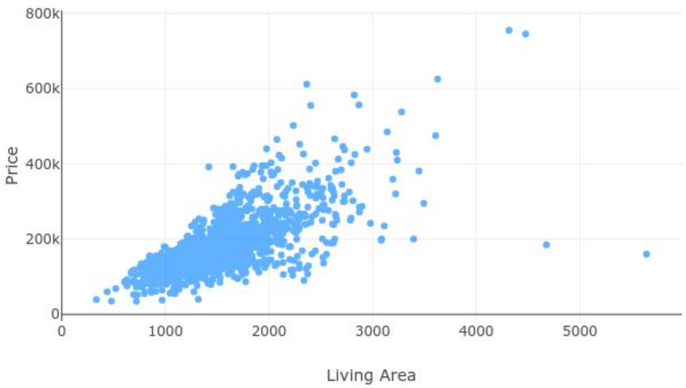
- Linear regression vs. 2nd order polynomial regression vs. 3rd order polynomial regression
- We have a smaller loss with a higher order function
  - More complex model has more ability (we call this “capacity”) represent more complicated relationship between the input and the output
- Does this mean higher order function is better?
  - We will revisit and talk more about this later…but the short conclusion is that it is important to choose the “right” model

House price prediction problem



- What regression do we want to do?

House price prediction problem



- We can try to minimize the MSE loss by choosing a model and applying what we have learned
- In the plot, it seems like the data shows linear correlation between the area and the price (which makes us want to use linear fitting)
- In the plot, it seems that the data has a lot of “noise”
- We know that there are lots of factors, other than area, that can affect the house price
- No matter how well we do the regression, our prediction will not be very accurate
- We have to provide “enough” information

Summary

- **Machine learning** refers to algorithms that improve their performance at some task with experience
- There are three types of machine learning: supervised learning, unsupervised learning and reinforcement learning
- **Supervised learning** is about learning a function that maps an input to an output based on example input-output pairs
- **Linear regression** is an approach to model the relationship between a dependent variable and one or more independent variables as a linear function
  - …and linear regression is a perfect example of supervised learning
- **Polynomial regression** is an approach to model the relationship between a dependent variable and one or more independent variables as an nth order polynomial function

References

- Lecture notes
  - CC229 lecture note
    - http://cs229.stanford.edu/notes/cs229-notes-all/cs229-notes1.pdf
  - MIT 6.036 Intro to Machine Learning (Chapter 7)
    - https://www.mit.edu/~lindrew/6.036.pdf