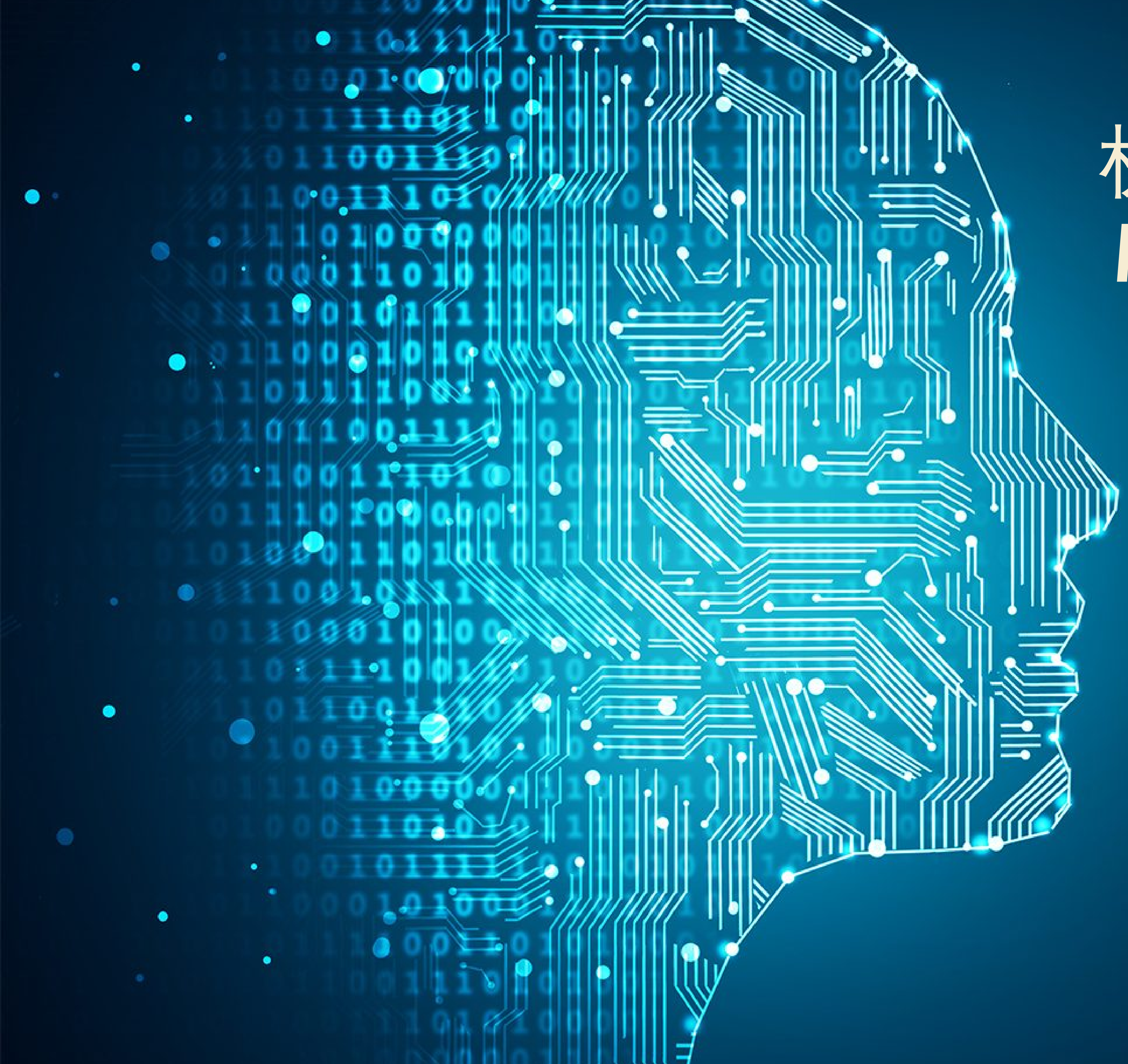


机器学习与人工智能
Machine Learning and Artificial Intelligence

Lecture 2 Regressions

Yingjie Zhang (张颖婕)

Peking University

yingjiezhang@gsm.pku.edu.cn

2021 Fall

Group Project

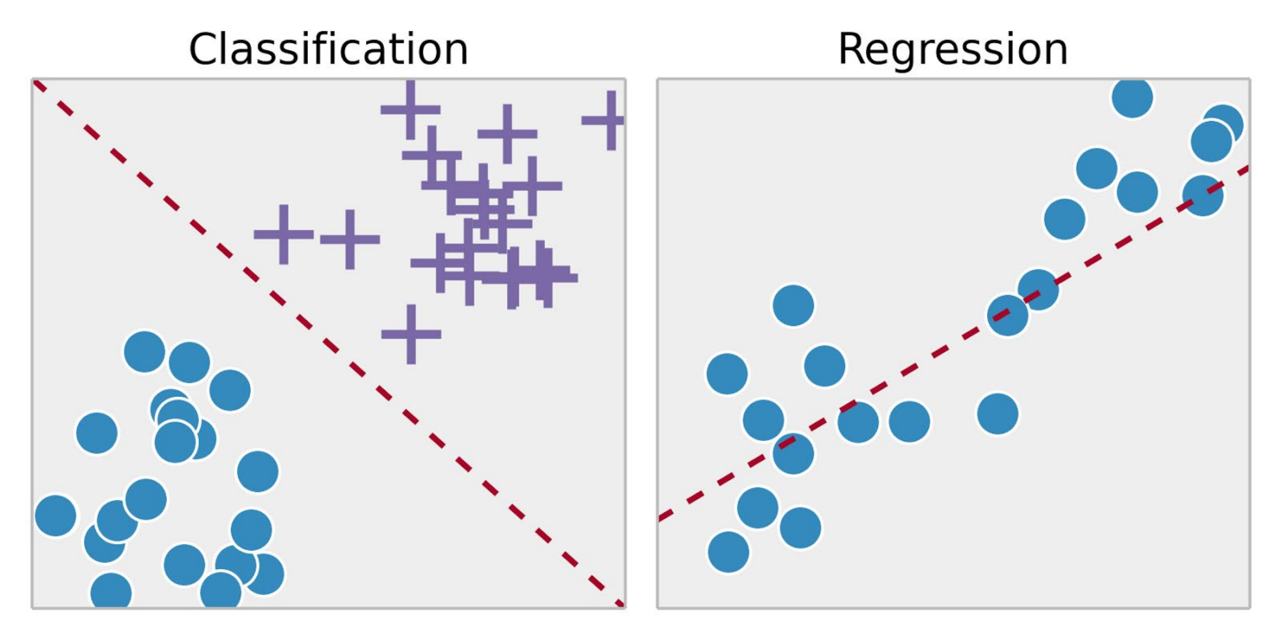
- Sign up the team by **Sept 26** (email/msg TAs)

- Proposal: Due on Oct 10, 2021, 11:59pm (one submission per team)
  - Team members (at most 5 students)
  - Project goals (with a real-world business question and available datasets)
  - Models to address the questions (at least one supervised and one unsupervised learning models)
  - Advanced model applications and deeper analyses are encouraged

- In-Class Presentation (Nov 25)
- Final reports (Due Dec 9, 2021)

Classification vs. Regression

- **Classification**: the goal is to predict a *class label*, which is a choice from a predefined list of possibilities
- **Regression**: the goal is to predict a continuous number, or a *floating-point number* (*real number*) in programming (math) terms



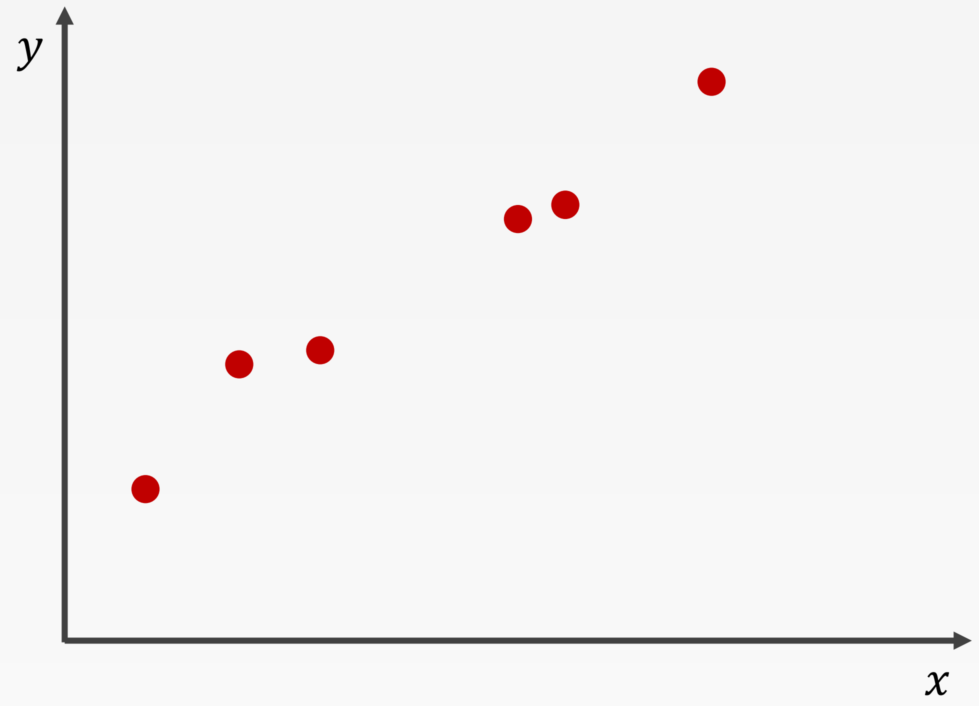
Regression

- Given the value of an input X, the output Y belongs to the set of real value R.
- Evaluation: predict output accurately
- Examples:
  - Predict housing price
  - Forecast precipitation

Source: CMU 10601

Regression

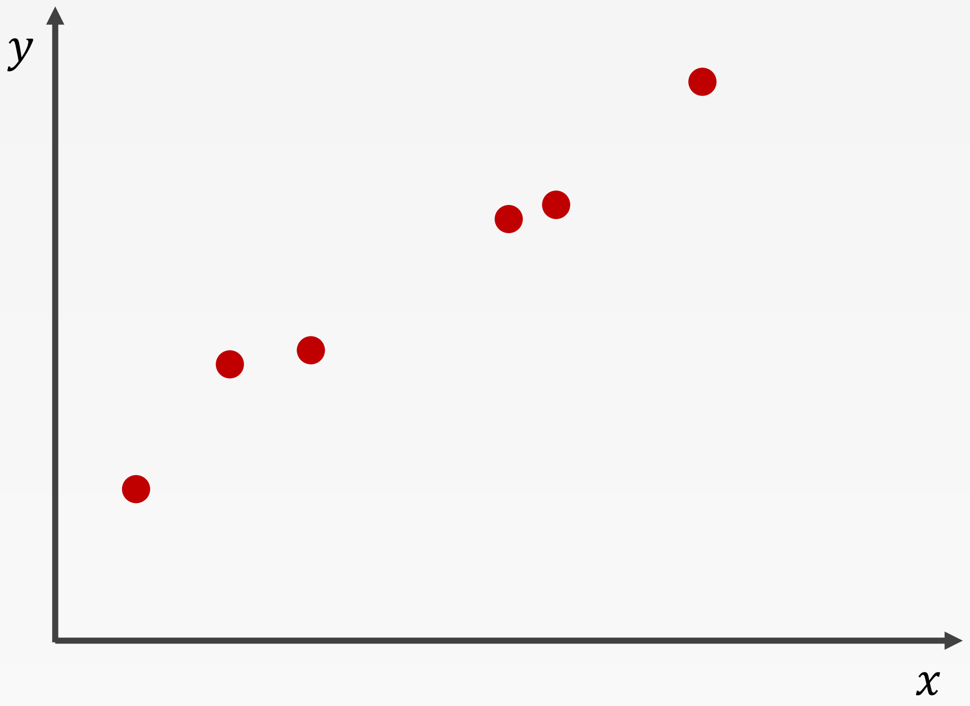
Example: Dataset with only one feature x and one scalar output y



Q: What is the function that best fits these points?

k-NN Regression

Example: Dataset with only one feature x and one scalar output y



$k = 1$

- *Train*: store all (x, y) pairs
- *Predict*: pick the nearest x in the training data and return its y

$k = 2$ Nearest Neighbor Distance Weighted Regression

- *Train*: store all (x, y) pairs
- *Predict*: pick the nearest two instances $x^{(n1)}$ and $x^{(n2)}$ in training data and return the weighted average of their y values

Linear Regression

Agenda

- Definition of Regression
- Linear functions
- Residuals
- Estimations (Optimization)
- Regularization

Linear Regression

- Linear relationship: outcome (dependent) variable is a linear combination of predictor (independent) variables.

$$\boldsymbol{Y = b + w_1X_1 + w_2X_2 + \cdots + w_nX_n}$$

Outcome

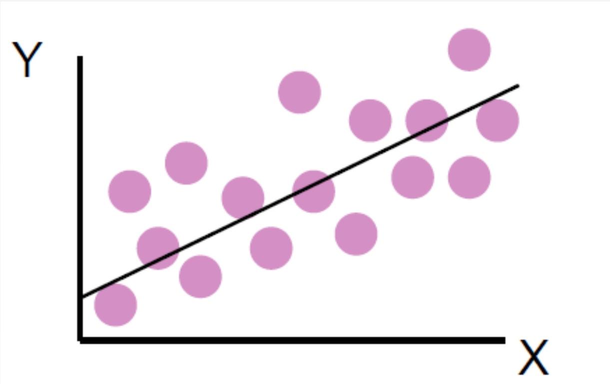
Intercept

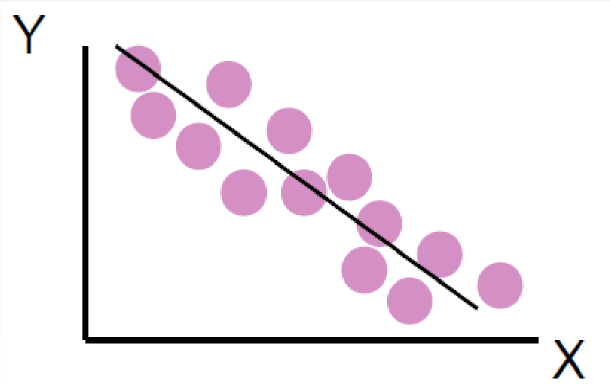
Coefficients

Predictors

Linear Model?

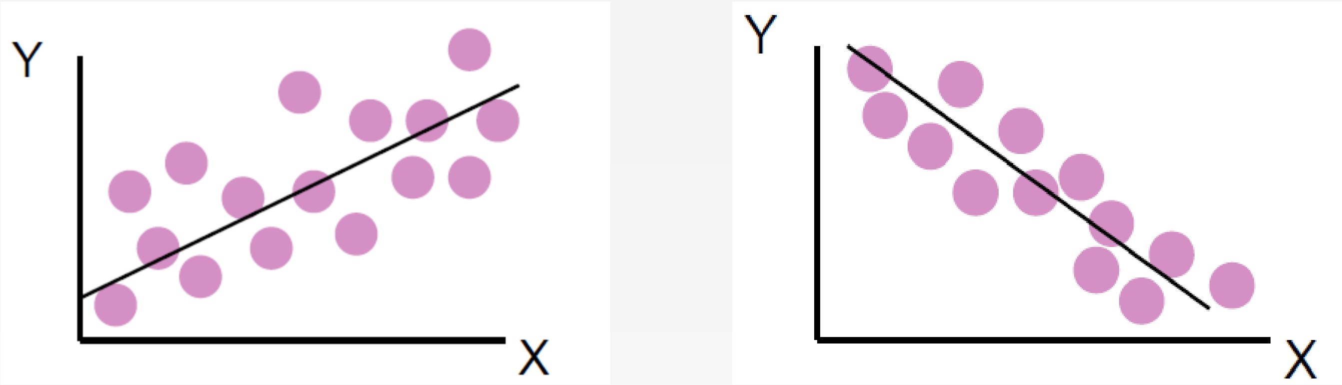
$$Y_i = b + wX_i$$



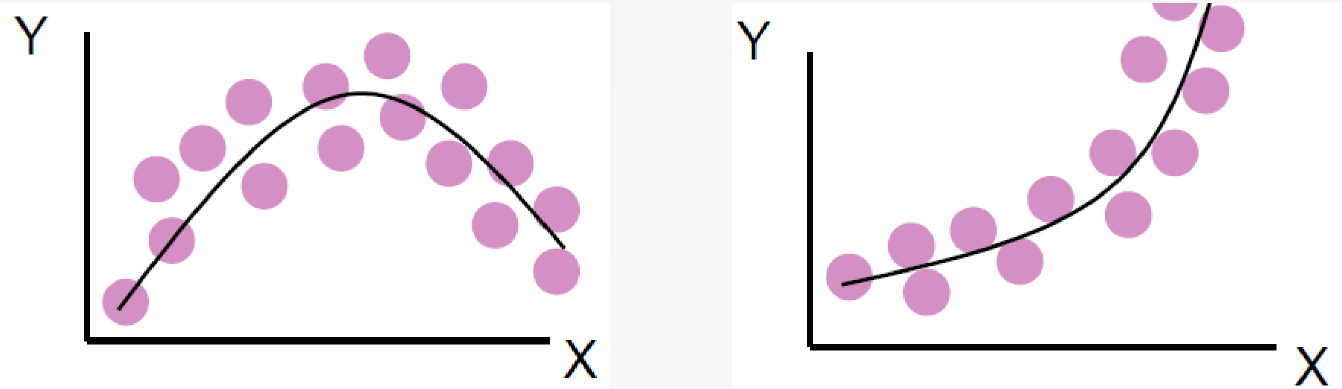


Linear Model?

$$Y_i = b + wX_i$$



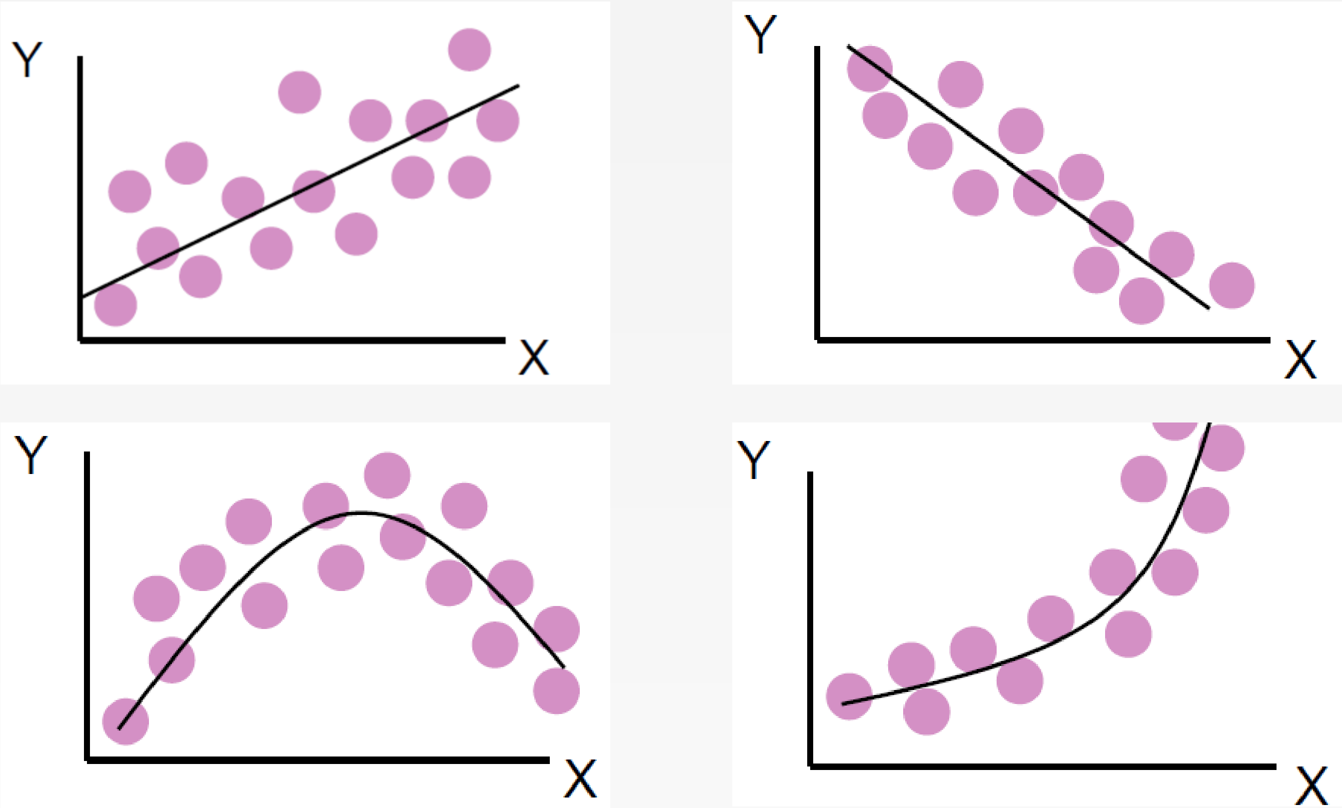
$$Y_i = b + w_1 X_i + w_2 X_i^2$$



Linear Model?

$$Y_i = b + wX_i$$

$$Y_i = b + w_1 X_i + w_2 X_i^2$$



A linear model means linear in parameters (not X)!

光华管理学院 Guanghua School of Management

Linear Regression?

- $y = \sum_i \omega_i f_i(x)$
- $y = \sum_i \omega_i x^i$
- $y = \sum_i e^{w_i x}$
- $y = \sum_i w_i \sin(i^2 x^7)$

Results Interpretation

Level-Level	$y = b + wx + \varepsilon$	One unit increase of x → w unit increase of y

Results Interpretation

Level-Level	$y = b + wx + \varepsilon$	One unit increase of x → w unit increase of y
Log-Level	$\log(y) = b + wx + \varepsilon$	One unit increase of x → $100w$% increase of y

Example: y – income; x – tenured year; $w = 0.04$. One more tenured year increases 4% in income

Results Interpretation

Level-Level

$$y = b + wx + \varepsilon$$

One unit increase of x
→ w unit increase of y

Log-Level

$$\log(y) = b + wx + \varepsilon$$

One unit increase of x →
$100w$% increase of y

Example: y – income; x – tenured year; $w = 0.04$. One more tenured year increases 4% in income

Log-Log

$$\log(y) = b + w\log(x) + \varepsilon$$

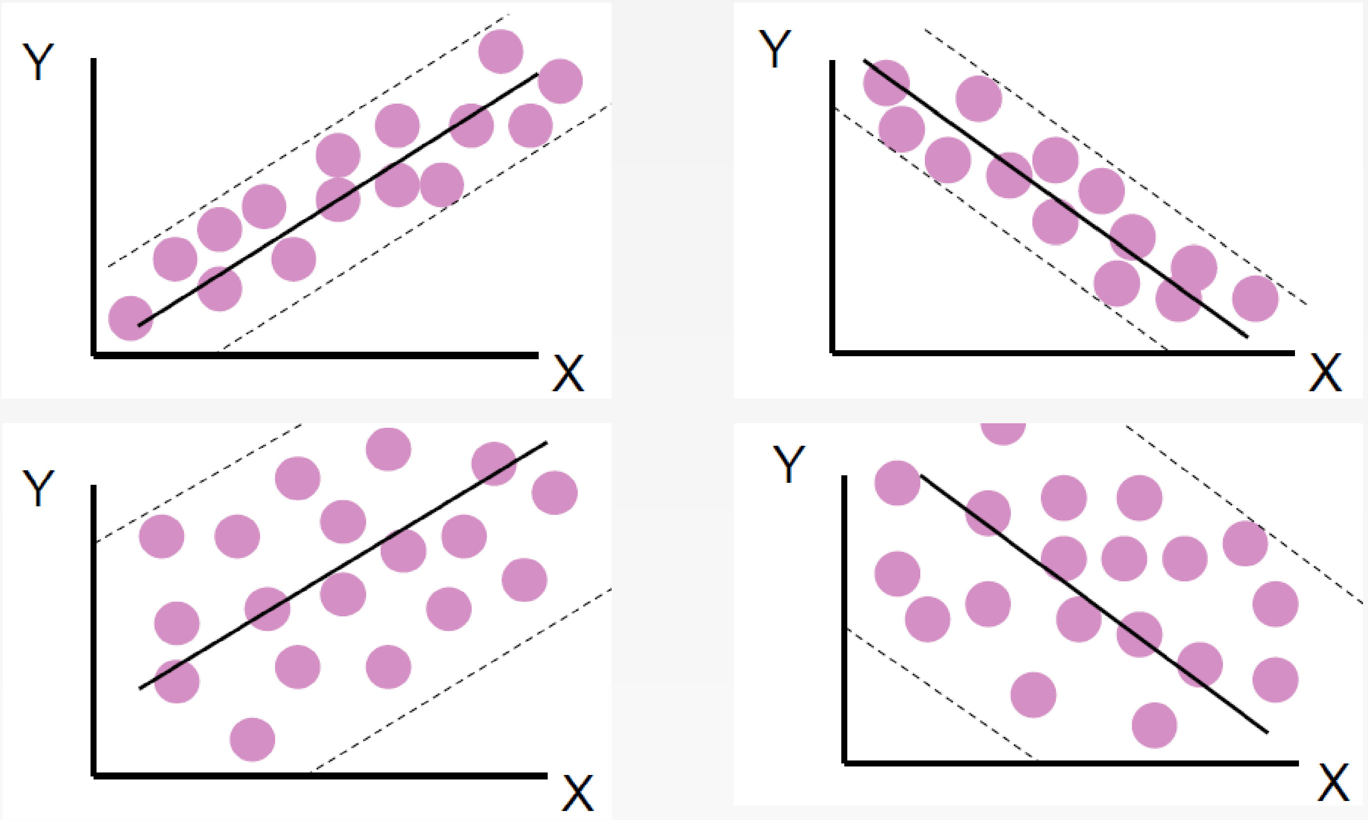
One percent increase of x
→ w% increase of y

Example: y – demand; x – price; $w = -0.6$. 1% increase in price leads to 0.6% decrease in demand

Linear Model?

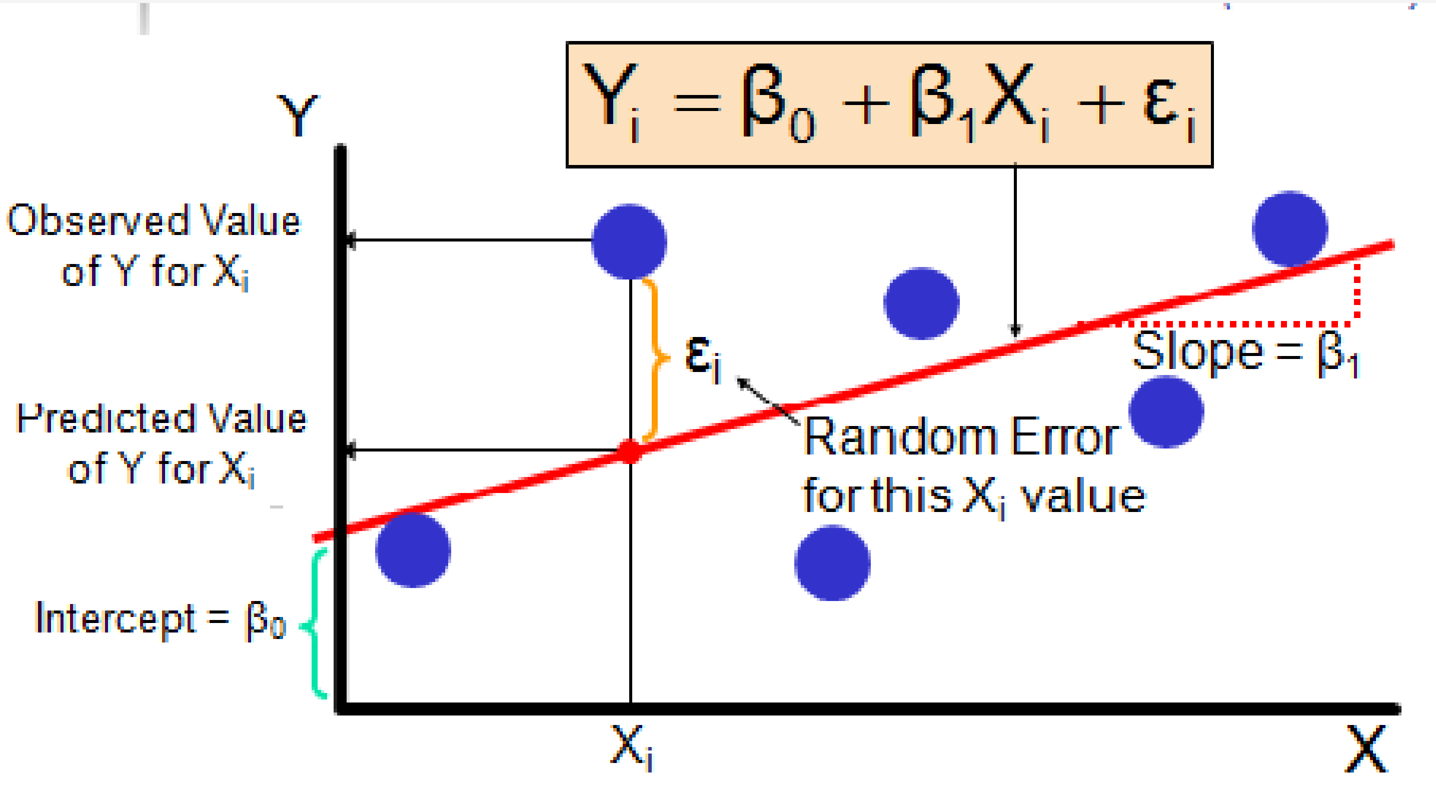
Strong relationship

Weak relationship



How to evaluation?

Residuals



Residuals e = observed (y) – predicted (y)

Train a Linear Model

- Goal: minimize the Error

- Potential ways:
  - Sum (mean) of absolute errors $|e_1| + |e_2| + |e_3| + \cdots$
  - Sum (mean) of squared errors $e_1^2 + e_2^2 + e_3^2 + \cdots$

Function Approximation

- Objective function: mean squared error (MSE)

$$J(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} e_i^2$$
$$= \frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - \boldsymbol{\theta}^T \boldsymbol{x}^{(i)}\right)^2$$

$$\boldsymbol{x}' = [1, x_1, x_2, \dots, x_M]^T$$
$$\boldsymbol{\theta} = [b, w_1, \dots, w_M]^T$$

- Solve the unconstrained optimization problem
  - Closed form
  - Gradient descent
  - Stochastic gradient descent

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, J(\boldsymbol{\theta})$$

- Test time:
  - Given a new x, make a prediction $\hat{y} = \widehat{\boldsymbol{\theta}}^T \boldsymbol{x}$

Closed-form Solution

Criteria Minimize the sum of squared errors $min\ \sum_n (y_i - \hat{y}_i)^2$

Solution $y_i = b + w_1 x_i + \varepsilon_i$

w_1: slope for the estimated regression equation

$$w_1 = \frac{\sum_n (x_i - \bar{x})(y_i - \bar{y})}{\sum_n (x_i - \bar{x})^2}$$

b: intercept for the estimated regression equation

$$b = \bar{y} - w_1 \bar{x}$$

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

Gradient Descent

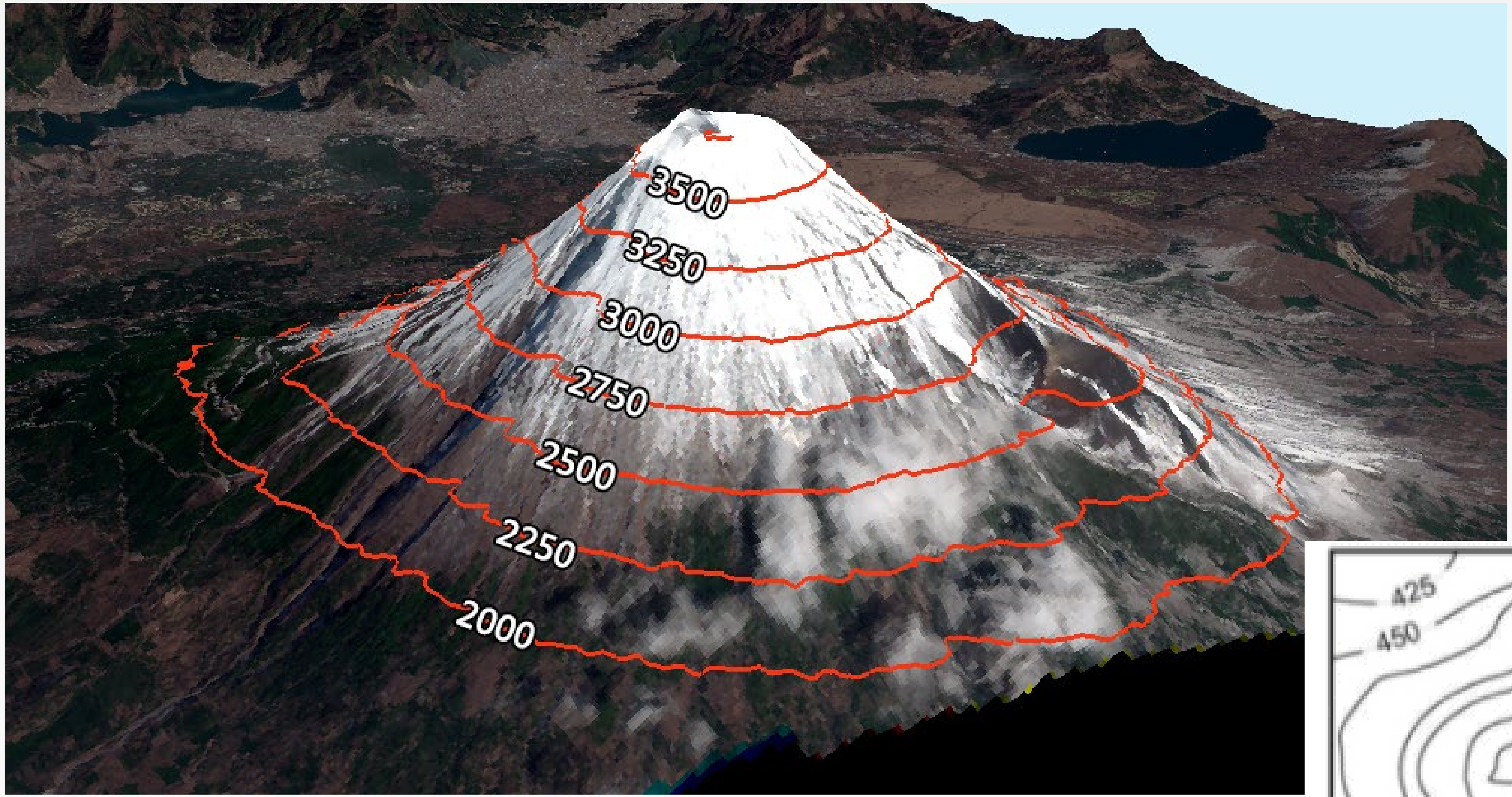
Linear Regression Solutions

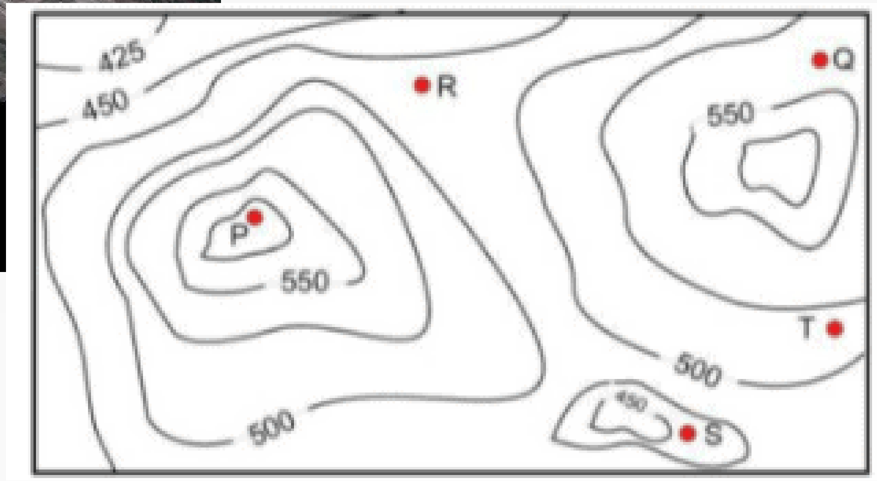
- Closed-form solution:
  - Computational complexity
  - Stability

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

- Gradient Descent for Linear Regression

Contour Plots





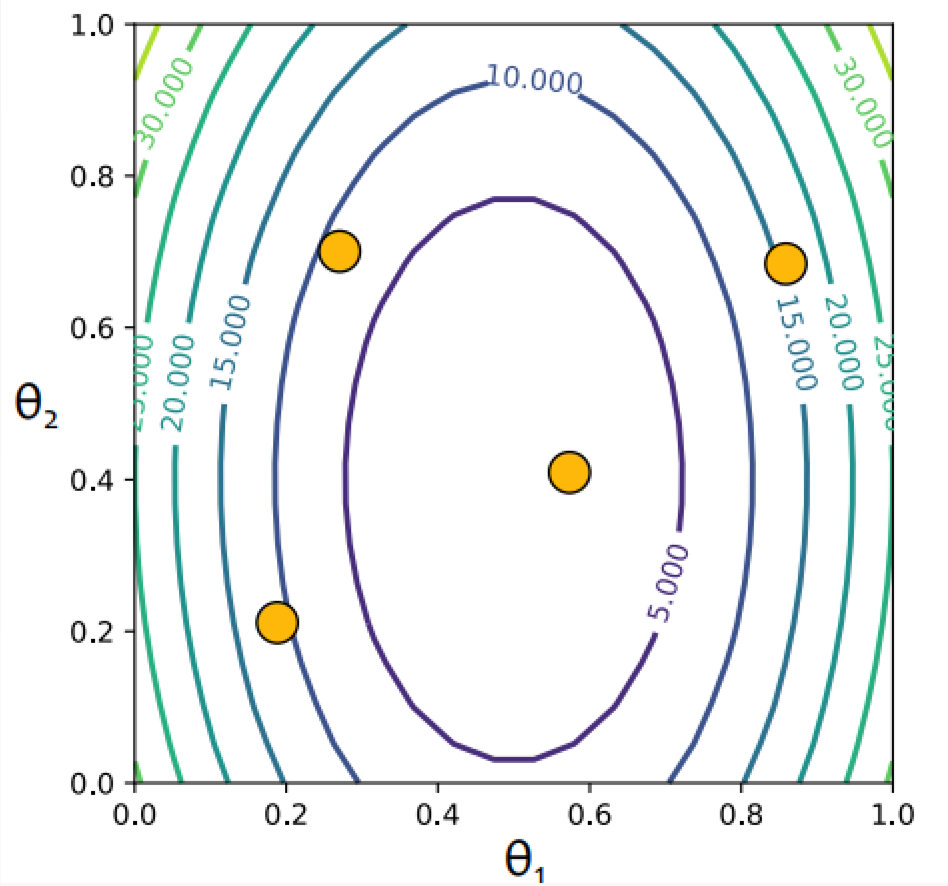
Contour Plots

1. Each level curve labeled with value
2. Value label indicates the value of the function for all points lying on that level curve

Optimization by Random Guessing

Random guessing:
1. Pick a random θ
2. Evaluate $J(\theta)$
3. Repeat steps 1 and 2 many times
4. Return θ that gives smallest $J(\theta)$

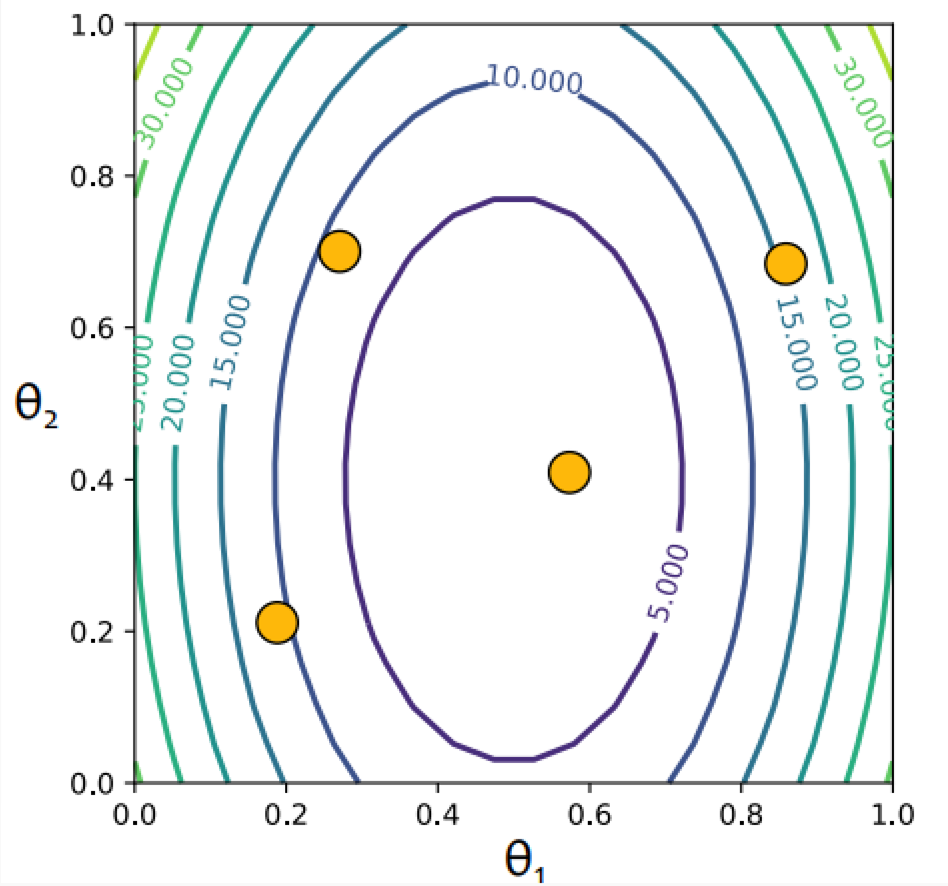
$$J(\boldsymbol{\theta}) = J(\theta_1, \theta_2) = (10(\theta_1 - 0.5))^2 + \left(6(\theta_2 - 0.4)\right)^2$$



Optimization by Random Guessing

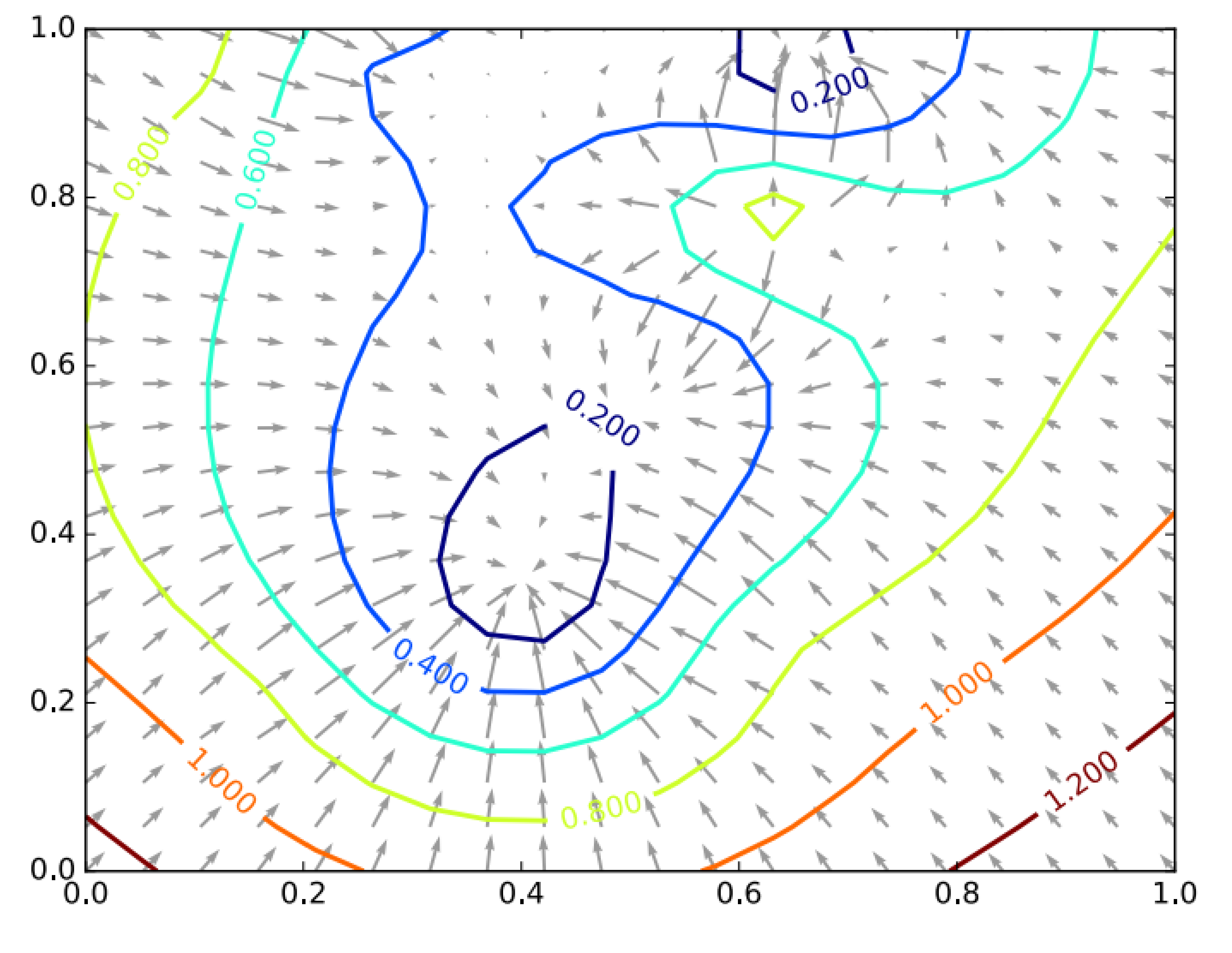
Random guessing:
1. Pick a random θ
2. Evaluate $J(\theta)$
3. Repeat steps 1 and 2 many times
4. Return θ that gives smallest $J(\theta)$

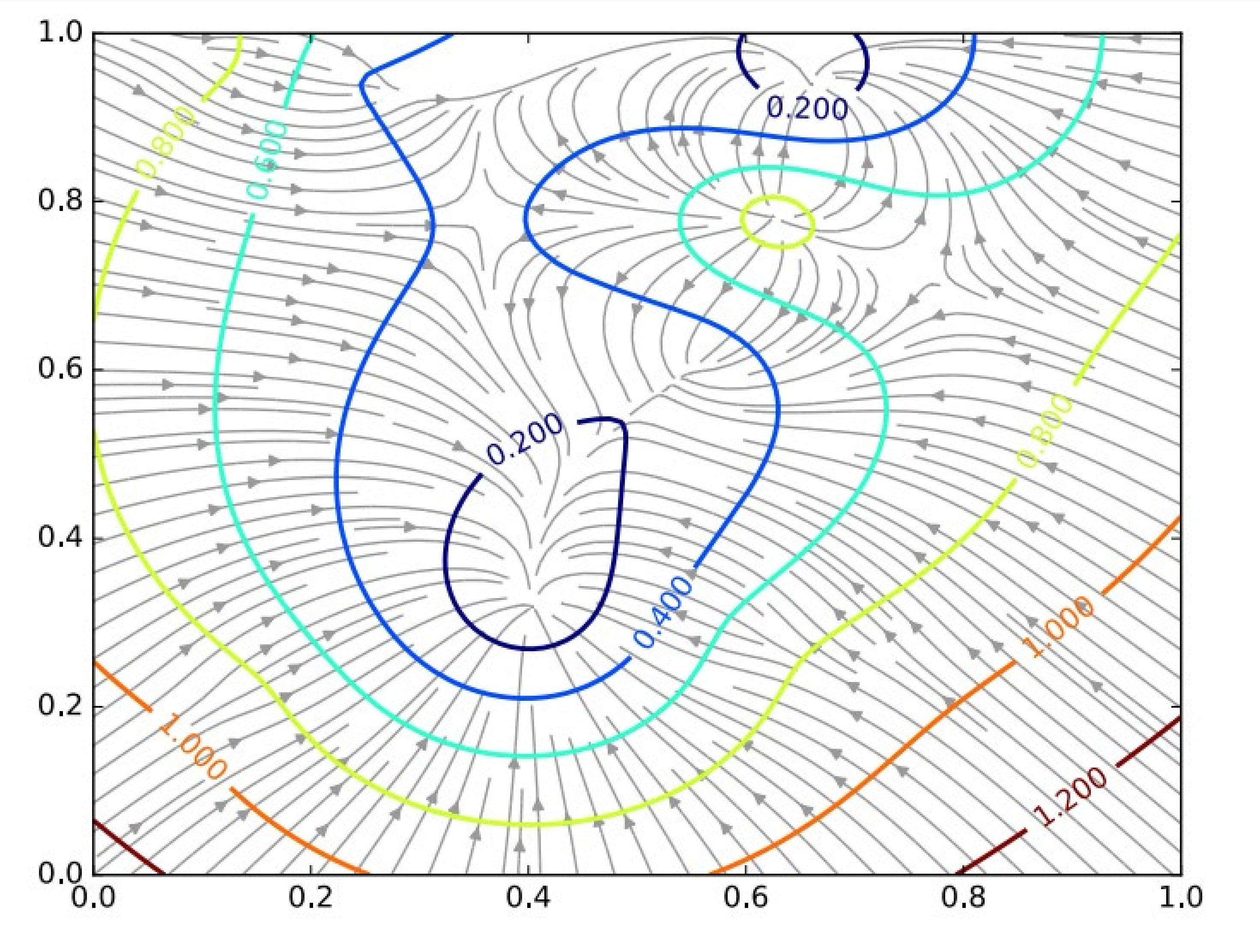
$$J(\boldsymbol{\theta}) = J(\theta_1, \theta_2) = (10(\theta_1 - 0.5))^2 + \left(6(\theta_2 - 0.4)\right)^2$$



Linear Regression:
1. Objective function: MSE
2. contour plot: each line labeled with MSE – lower means a better fit
3. **minimum** corresponds to parameters $(w, b) = (\theta_1, \theta_2)$ that **best fit** some training dataset

Gradient Descent

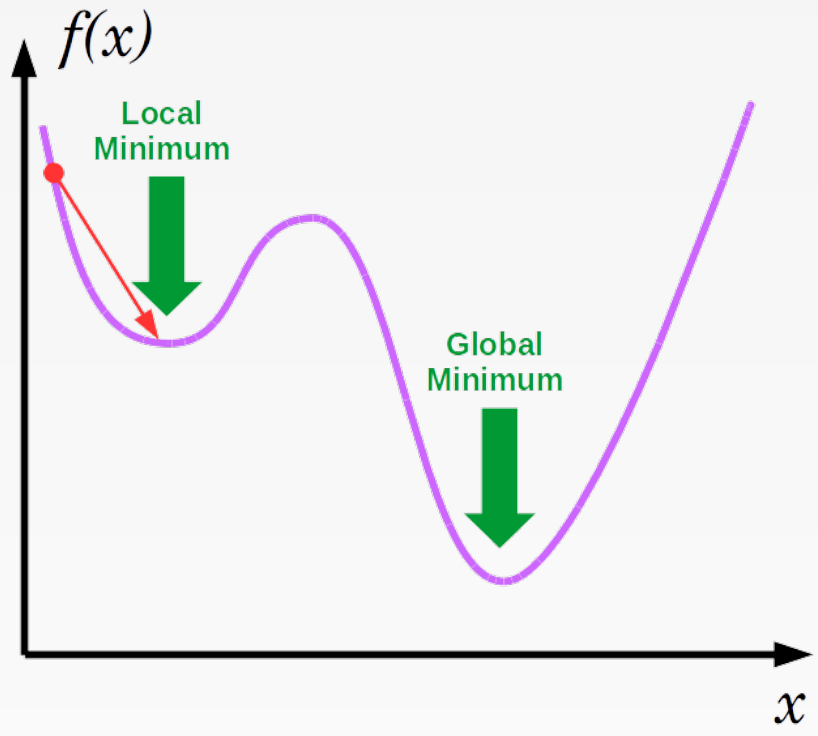



1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
0.200
0.200
0.400
0.600
0.800
0.800
1.000
1.000
1.200

Pros and Cons

- Advantages:
  - Simple and often quite effective on ML tasks
  - Often very scalable

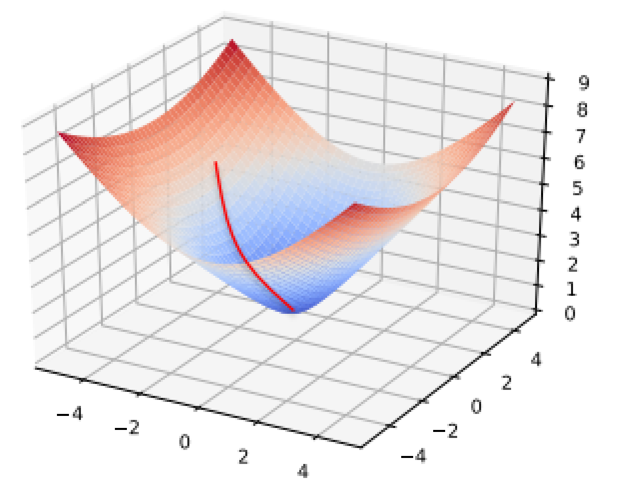
- Drawbacks
  - Might find a local minimum
  - Only applies to smooth function (differentiable)



Algorithm

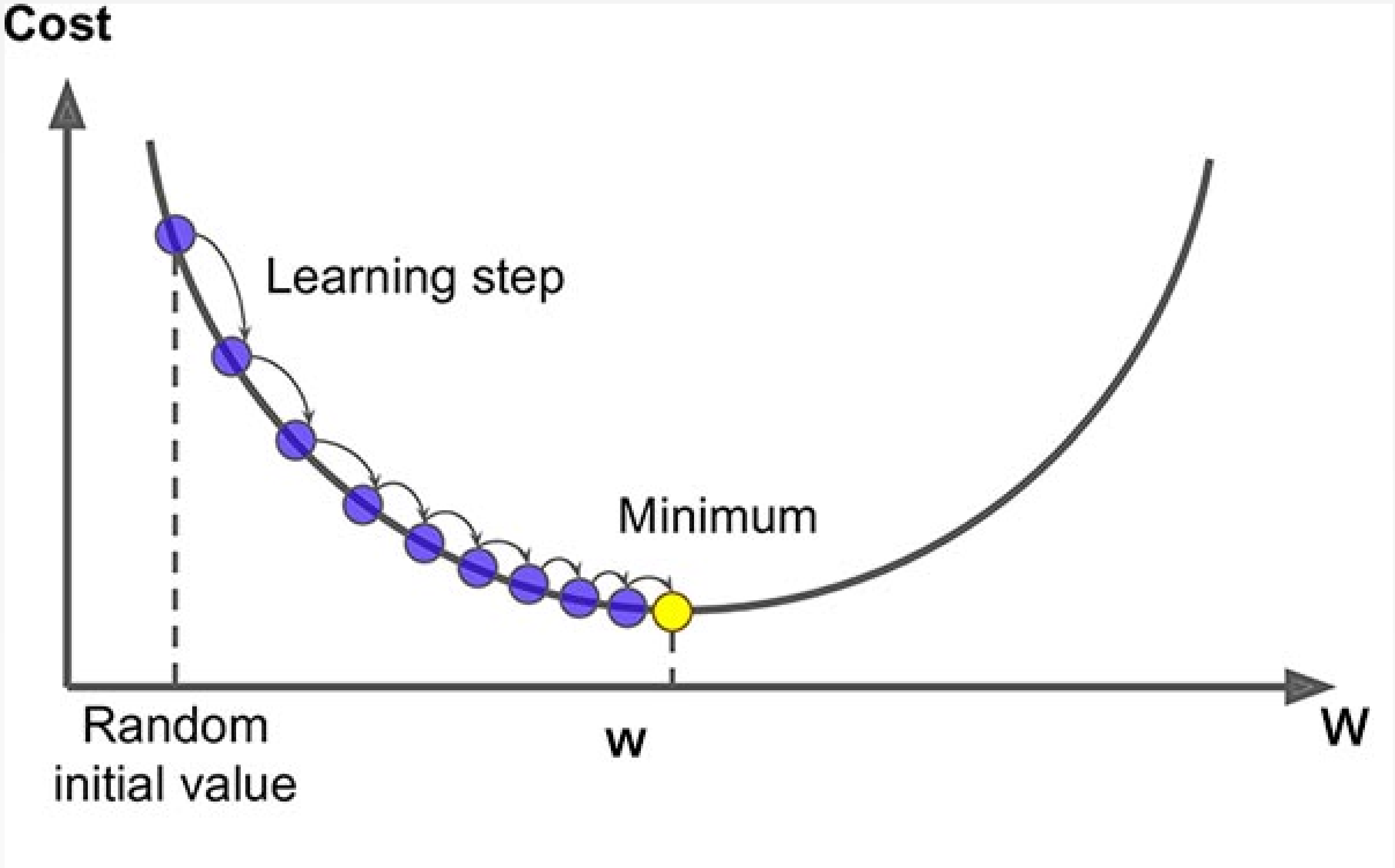
Algorithm 1 Gradient Descent

1: **procedure** GD($\mathcal{D}$, $\boldsymbol{\theta}^{(0)}$)
2: $\quad\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^{(0)}$
3: $\quad$**while** not converged **do**
4: $\quad\quad\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \gamma \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$
5: $\quad$**return** $\boldsymbol{\theta}$

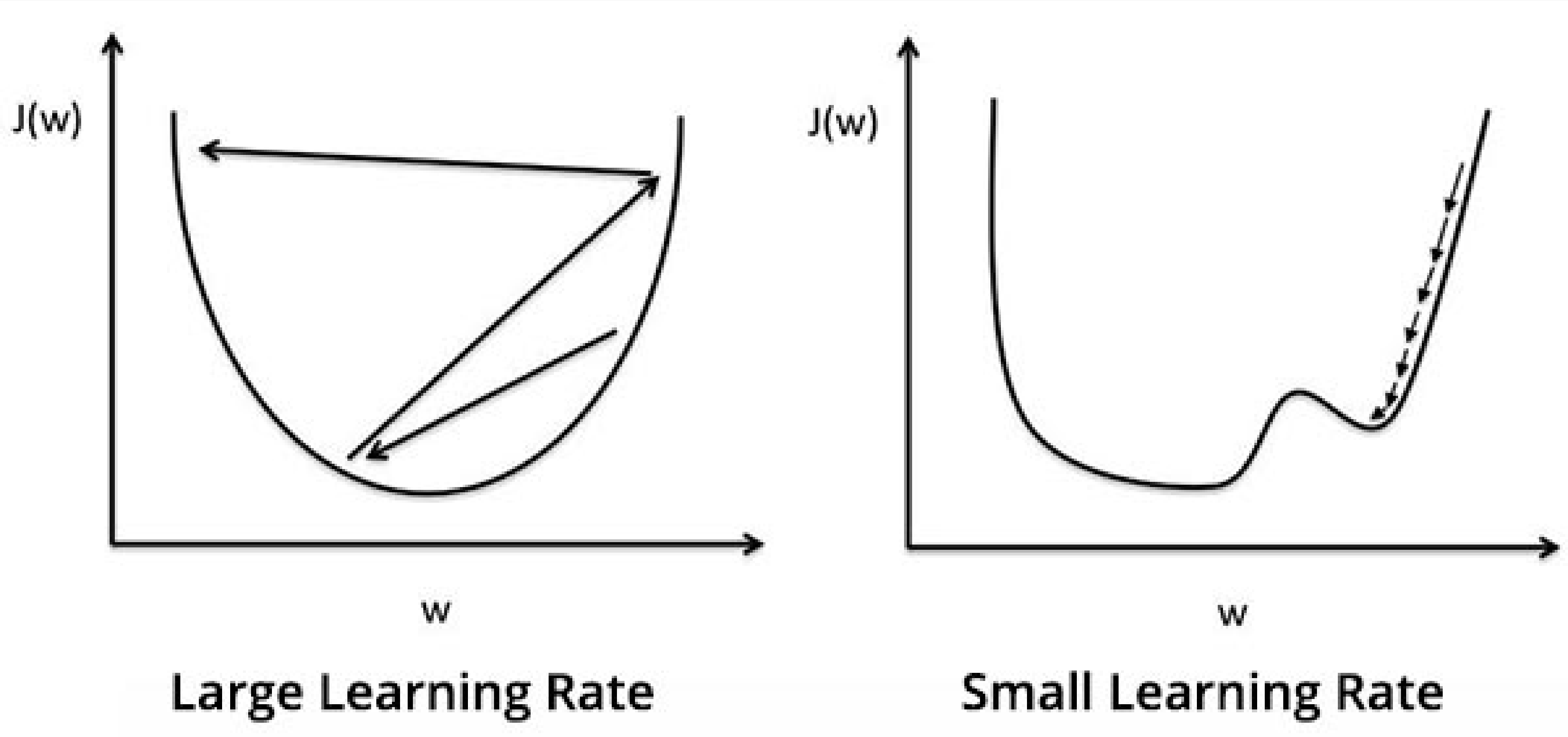


Convergence Criteria (one example): $\|\boldsymbol{\nabla}_{\theta} J(\boldsymbol{\theta})\|_2 \leq \epsilon$

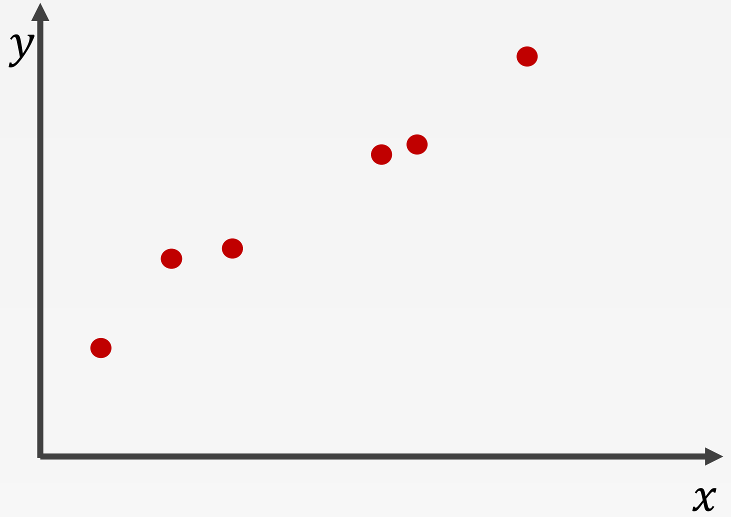
Gradient Descent



Learning Rate

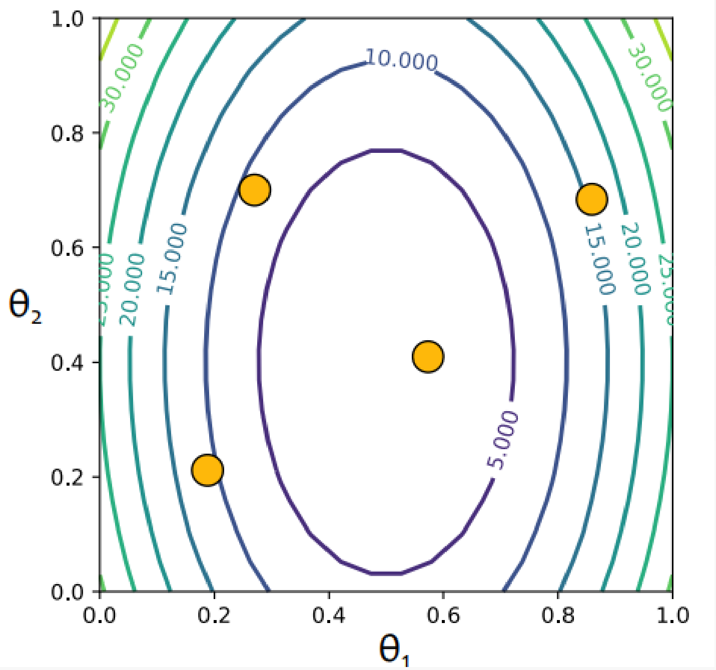


GD for Linear Regression

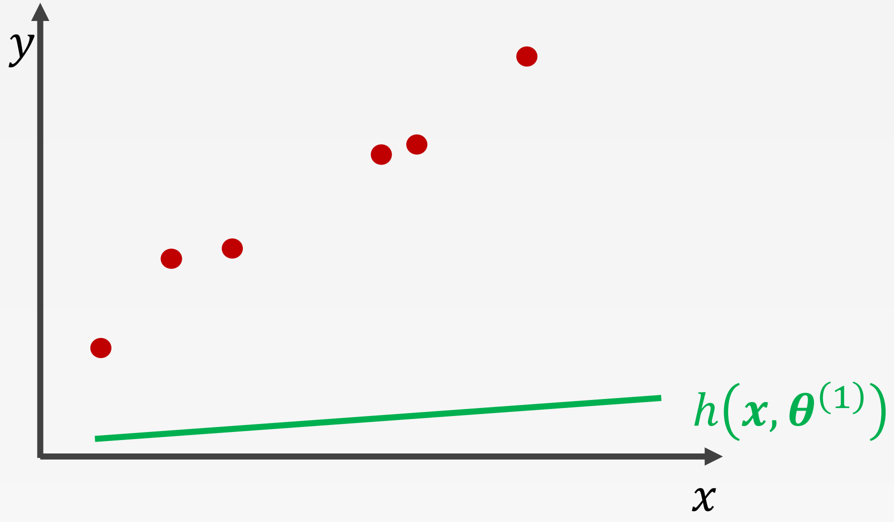


t	θ_1	θ_2	J

$$J(\boldsymbol{\theta}) = J(\theta_1, \theta_2) = \frac{1}{N}\sum\left(y^{(i)} - \boldsymbol{\theta}^T \boldsymbol{x}^{(i)}\right)^2$$

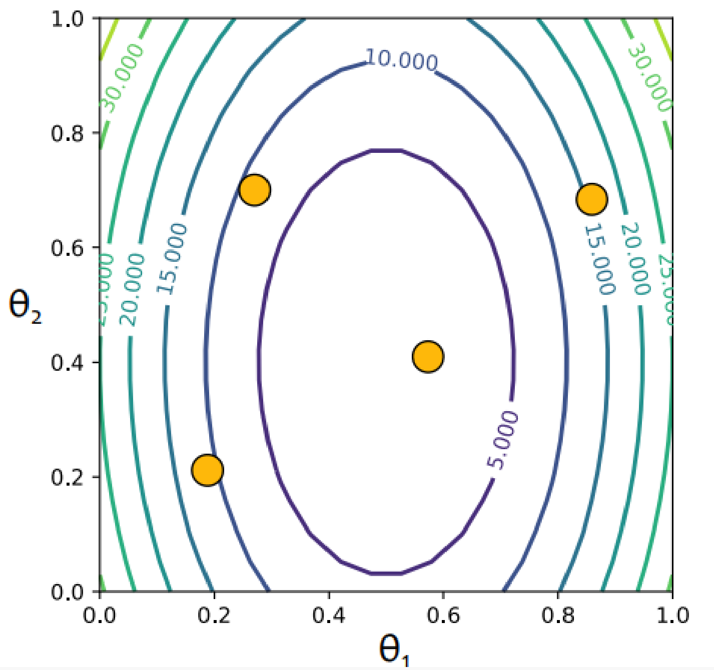


GD for Linear Regression

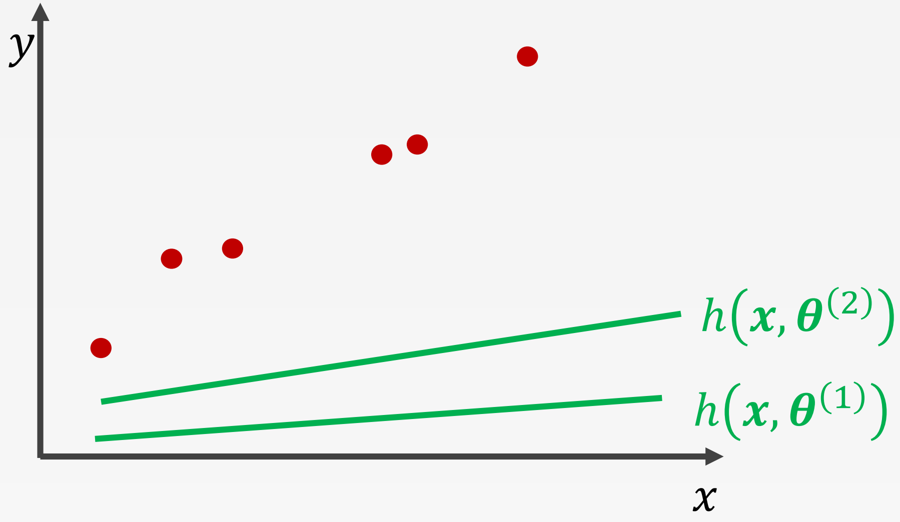


t	θ_1	θ_2	J
1	0.01	0.02	25.2

$$J(\boldsymbol{\theta}) = J(\theta_1, \theta_2) = \frac{1}{N}\sum\left(y^{(i)} - \boldsymbol{\theta}^T \boldsymbol{x}^{(i)}\right)^2$$

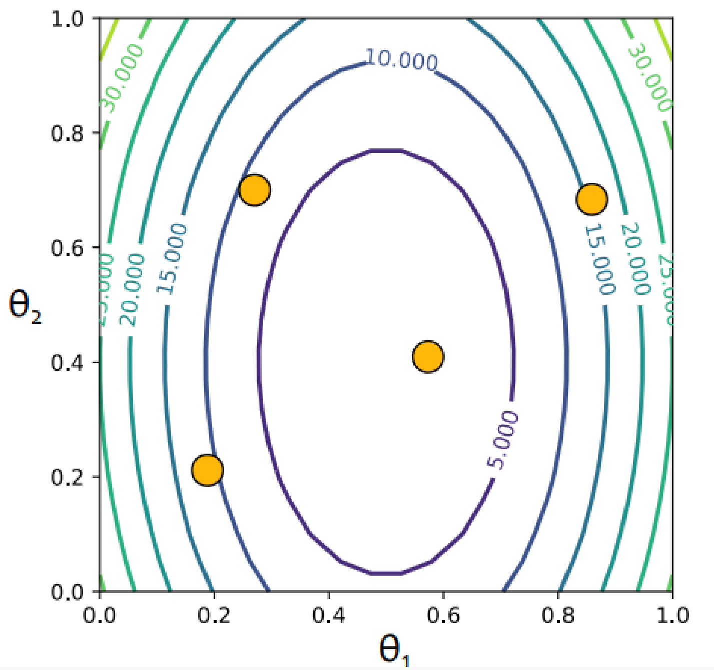


GD for Linear Regression

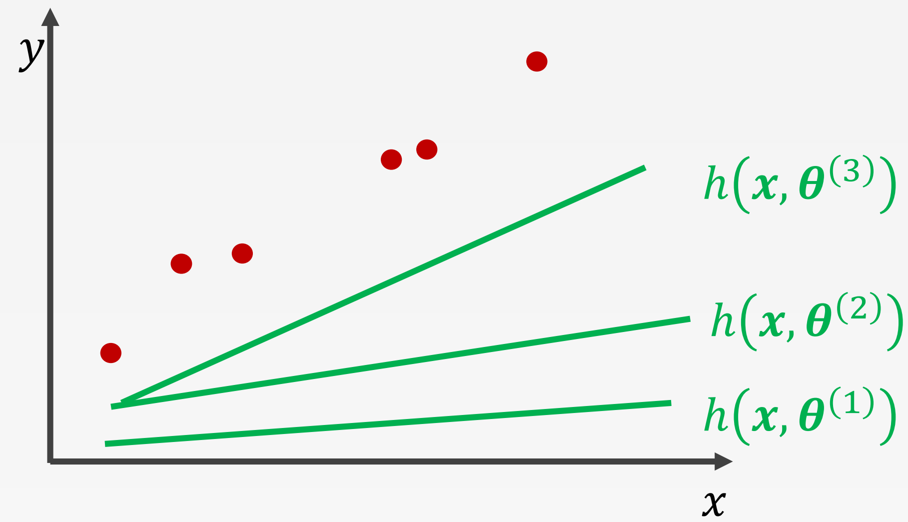


t	θ_1	θ_2	J
1	0.01	0.02	25.2
2	0.30	0.12	8.7

$$J(\boldsymbol{\theta}) = J(\theta_1, \theta_2) = \frac{1}{N}\sum\left(y^{(i)} - \boldsymbol{\theta}^T \boldsymbol{x}^{(i)}\right)^2$$

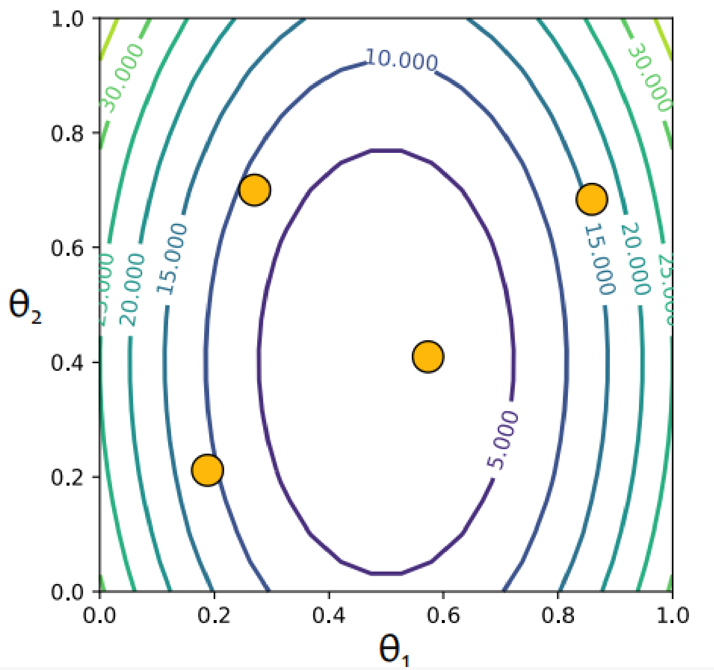


GD for Linear Regression

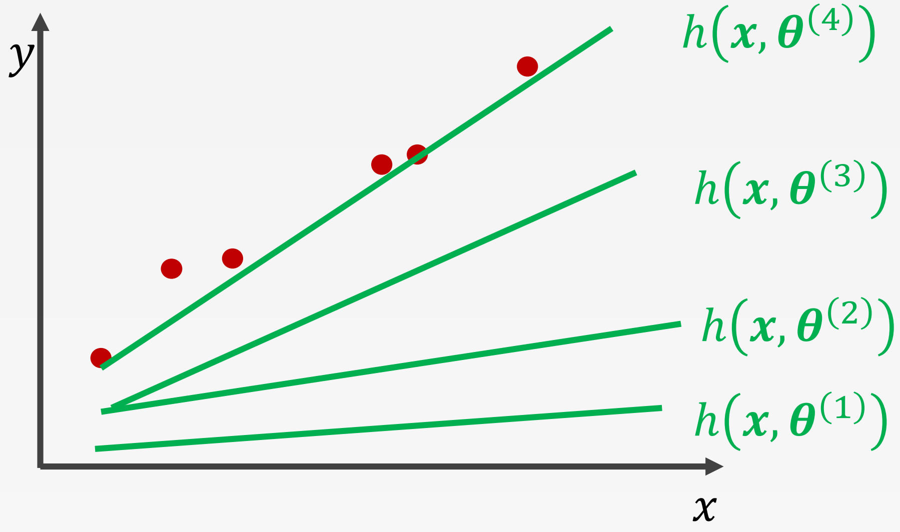


t	θ_1	θ_2	J
1	0.01	0.02	25.2
2	0.30	0.12	8.7
3	0.51	0.30	1.5

$$J(\boldsymbol{\theta}) = J(\theta_1, \theta_2) = \frac{1}{N}\Sigma\left(y^{(i)} - \boldsymbol{\theta}^T \boldsymbol{x}^{(i)}\right)^2$$

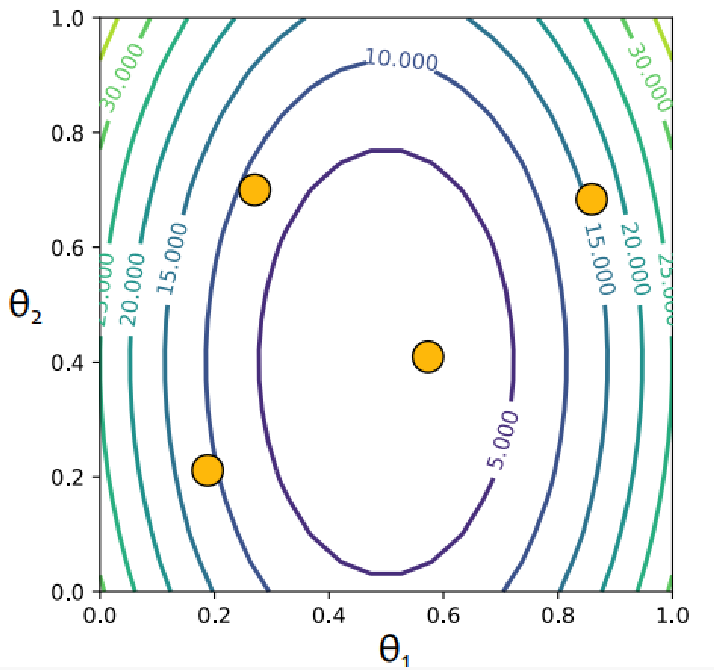


GD for Linear Regression

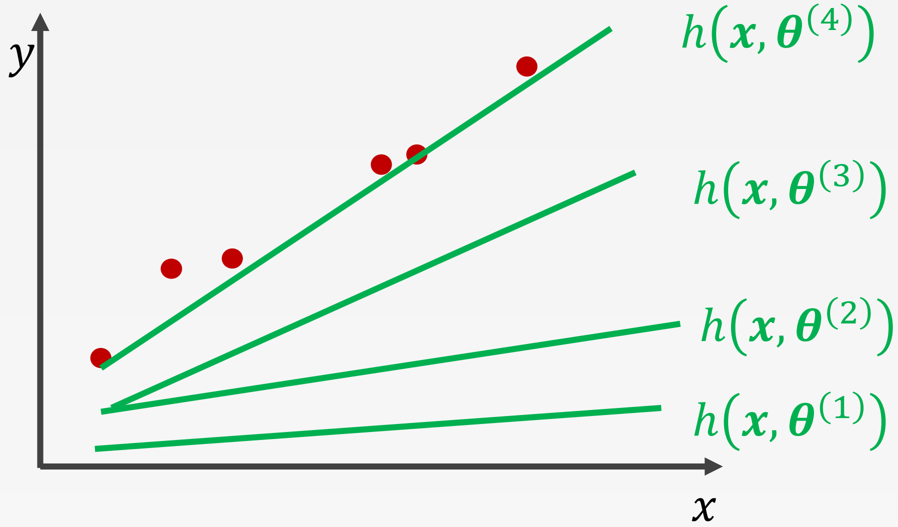


t	θ_1	θ_2	J
1	0.01	0.02	25.2
2	0.30	0.12	8.7
3	0.51	0.30	1.5
4	0.59	0.43	0.2

$$J(\boldsymbol{\theta}) = J(\theta_1, \theta_2) = \frac{1}{N}\sum\left(y^{(i)} - \boldsymbol{\theta}^T \boldsymbol{x}^{(i)}\right)^2$$

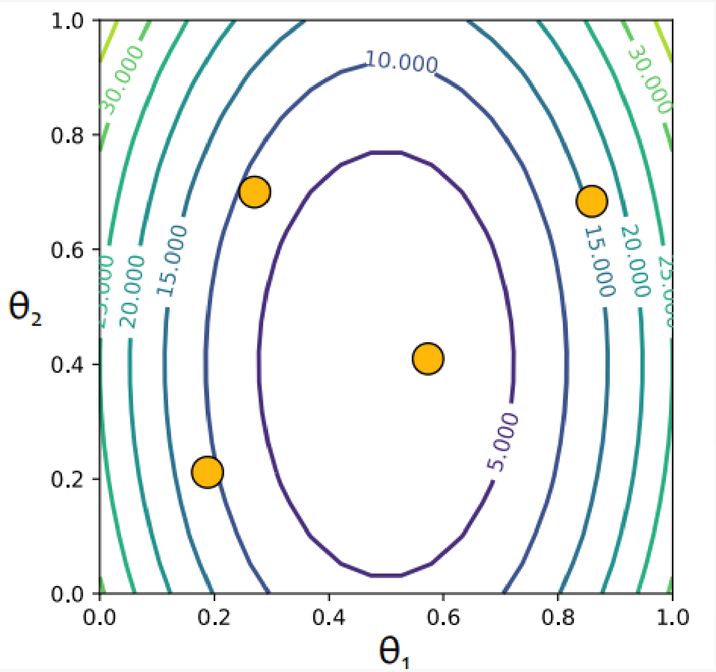


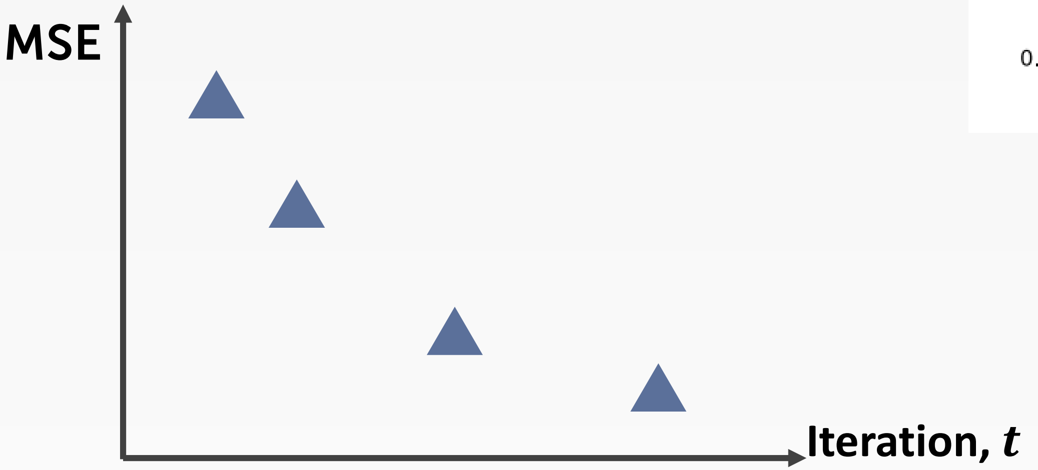
GD for Linear Regression



t	θ_1	θ_2	J
1	0.01	0.02	25.2
2	0.30	0.12	8.7
3	0.51	0.30	1.5
4	0.59	0.43	0.2

$$J(\boldsymbol{\theta}) = J(\theta_1, \theta_2) = \frac{1}{N}\sum\left(y^{(i)} - \boldsymbol{\theta}^T \boldsymbol{x}^{(i)}\right)^2$$





GD for Linear Regression

Algorithm 1 GD for Linear Regression

1: **procedure** GDLR($\mathcal{D}$, $\boldsymbol{\theta}^{(0)}$)
2: $\quad \boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^{(0)}$ ▷ Initialize parameters
3: $\quad$ **while** not converged **do**
4: $\quad\quad \mathbf{g} \leftarrow \sum_{i=1}^{N} (\boldsymbol{\theta}^T \mathbf{x}^{(i)} - y^{(i)}) \mathbf{x}^{(i)}$ ▷ Compute gradient
5: $\quad\quad \boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \gamma \mathbf{g}$ ▷ Update parameters
6: $\quad$ **return** $\boldsymbol{\theta}$

Stochastic Gradient Descent (SGD)

Algorithm 2 Stochastic Gradient Descent (SGD)

procedure SGD $(D, \theta^{(0)})$

  $\theta \leftarrow \theta^{(0)}$

  while not converged **do**

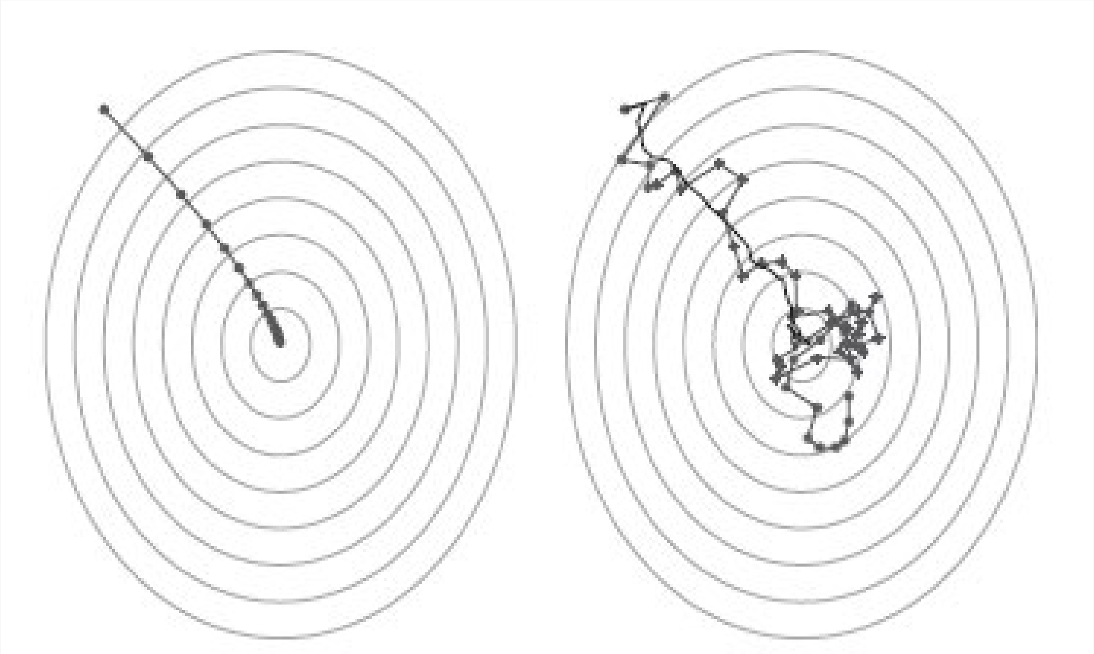
    for $i \sim \text{Uniform}(\{1,2,3,\ldots,N\})$

      $\theta \leftarrow \theta - \gamma \nabla_\theta J^{(i)}(\theta)$

  return θ

Stochastic Gradient Descent

- Just picks a random instance (or sampling) in the training set to compute the gradient



Mini-Batch SGD

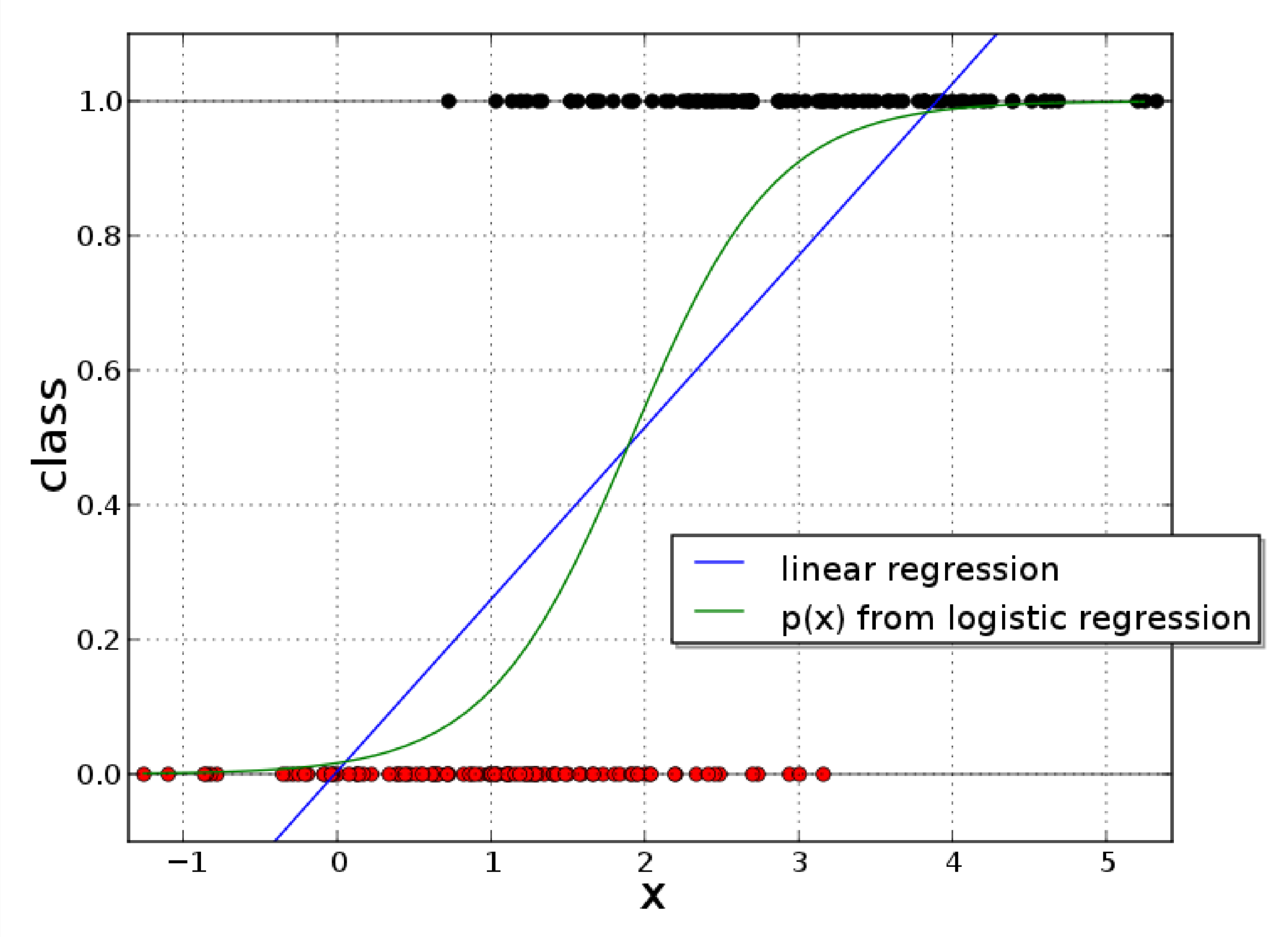
- Gradient Descent:
  - Compute true gradient exactly from all N examples
- Stochastic Gradient Descent (SGD):
  - Approximate true gradient by the gradient of one randomly chosen example
- Mini-Batch SGD:
  - Approximate true gradient by the average gradient of K randomly chosen examples

Logistic Regression

General View

- Logistic regression is a classification algorithm
- It predicts the probability of occurrence of an event by fitting data to a *logit* function
- Outcome: categorial variables

Logistic vs. Linear



Logistic Regression

- **Data**: Inputs are continuous vectors of length M. Outputs are discrete.

$$D = \{\boldsymbol{x}^{(i)}, y^{(i)}\}_{i=1}^{N}$$

- **Model**: Logistic function applied to dot product of parameters with input vector.

$$p_\theta(y = 1|\boldsymbol{x}) = \frac{1}{1 + \exp(-\boldsymbol{\theta}^T \boldsymbol{x})}$$

- **Learning**: finds the parameters that minimize some objective function.

$$\boldsymbol{\theta}^* = \underset{\theta}{\operatorname{argmin}}\, J(\boldsymbol{\theta})$$

- **Prediction**: Output is the most probable class.

$$\hat{y} = \underset{y \in \{0,1\}}{\operatorname{argmax}}\, p_{\boldsymbol{\theta}}(y|\boldsymbol{x})$$

Additional Information

- Estimates of coefficients (θ) are derived through an iterative process called Maximum Likelihood Estimation (MLE)
- If estimated probability > Cutoff → Classify as class "1"

- Probability of success (Odds) $\hat{\pi} = P(y = 1|x) = \frac{e^u}{1+e^u}$
- Odds-Ratio for success $\frac{\hat{\pi}}{1-\hat{\pi}} = e^u$
- Log Odds-Ratio $\ln\left(\frac{\hat{\pi}}{1-\hat{\pi}}\right) = u = b + w_1 X_1 + w_2 X_2 + \cdots$

MLE

Principle of Maximum Likelihood Estimation:
Choose the parameters that maximize the likelihood of the data.

$$\boldsymbol{\theta}^{\mathsf{MLE}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \prod_{i=1}^{N} p(\mathbf{x}^{(i)}|\boldsymbol{\theta})$$

Maximum Likelihood Estimate (MLE)

MLE tries to allocate as much probability mass as possible to the things we have observed… …at the expense of the things we have not observed

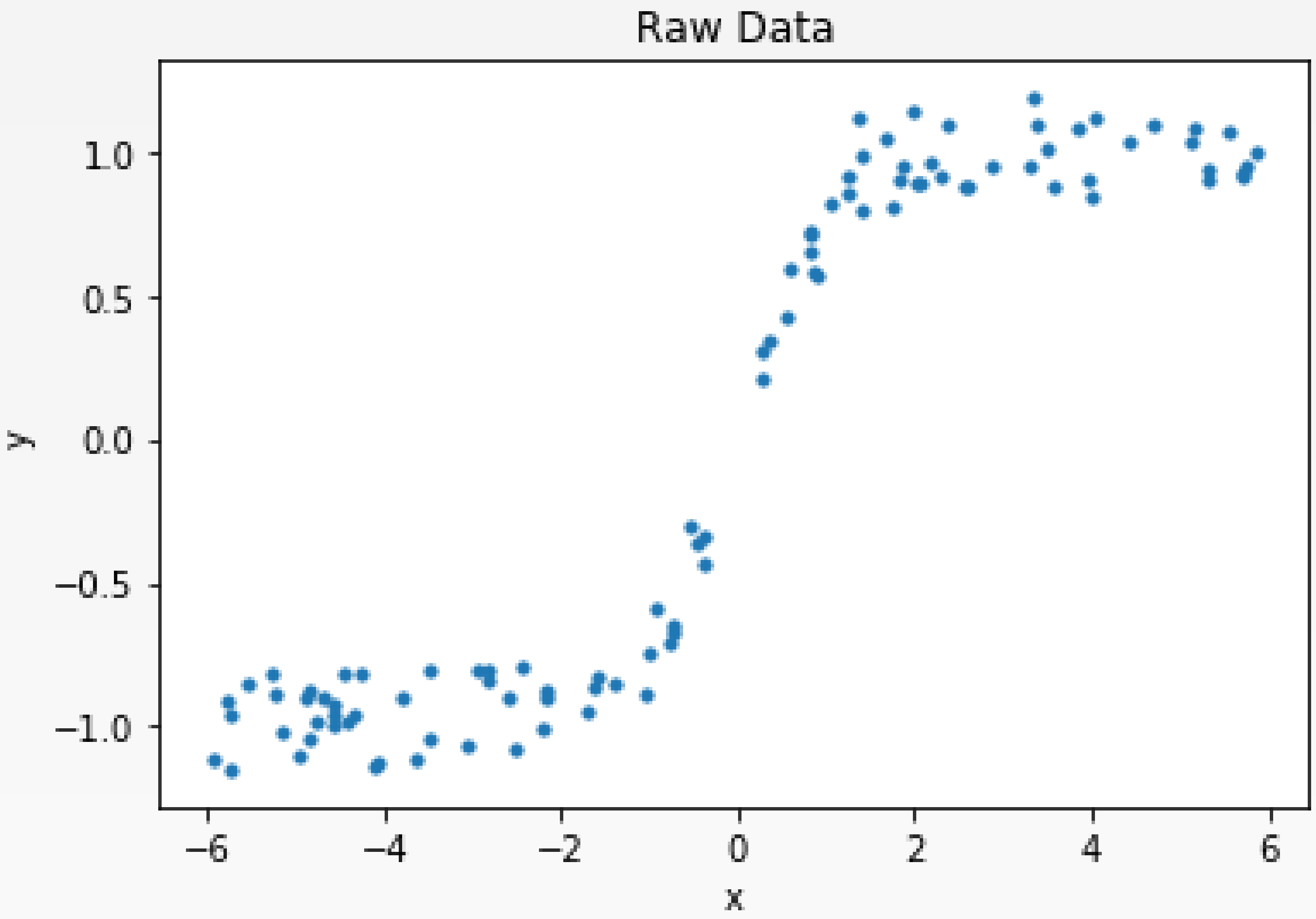
Polynomial Regression

Polynomial Regression

- Generate new features consisting of all polynomial combinations of the original features
- A linear model
- An application of non-linear transformations

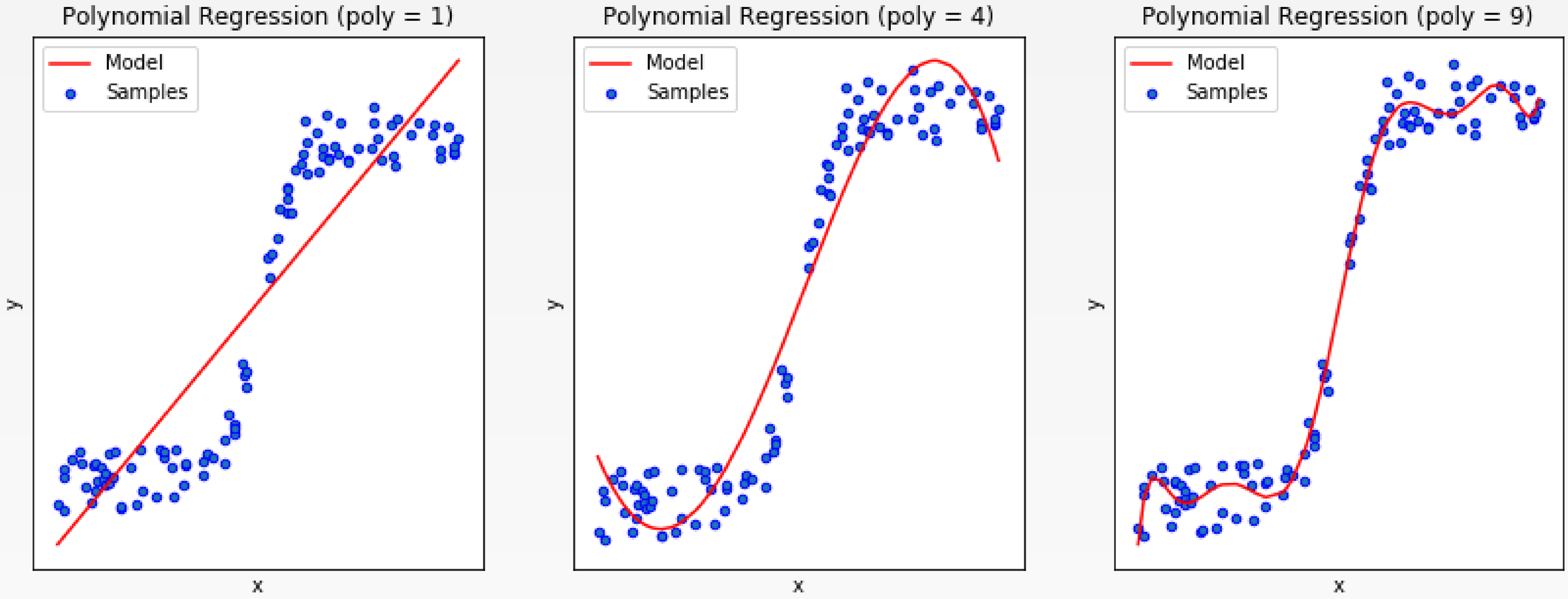
$$X = (x_0, x_1) \longrightarrow X' = (x_0, x_1, x_0x_1, x_0^2, x_1^2)$$
$$Y = w_0x_0 + w_1x_1 + w_{01}x_0x_1 + w_{00}x_0^2 + w_{11}x_1^2$$

Example I: Polynomial Features



True: $y = \tanh(x) + randn$

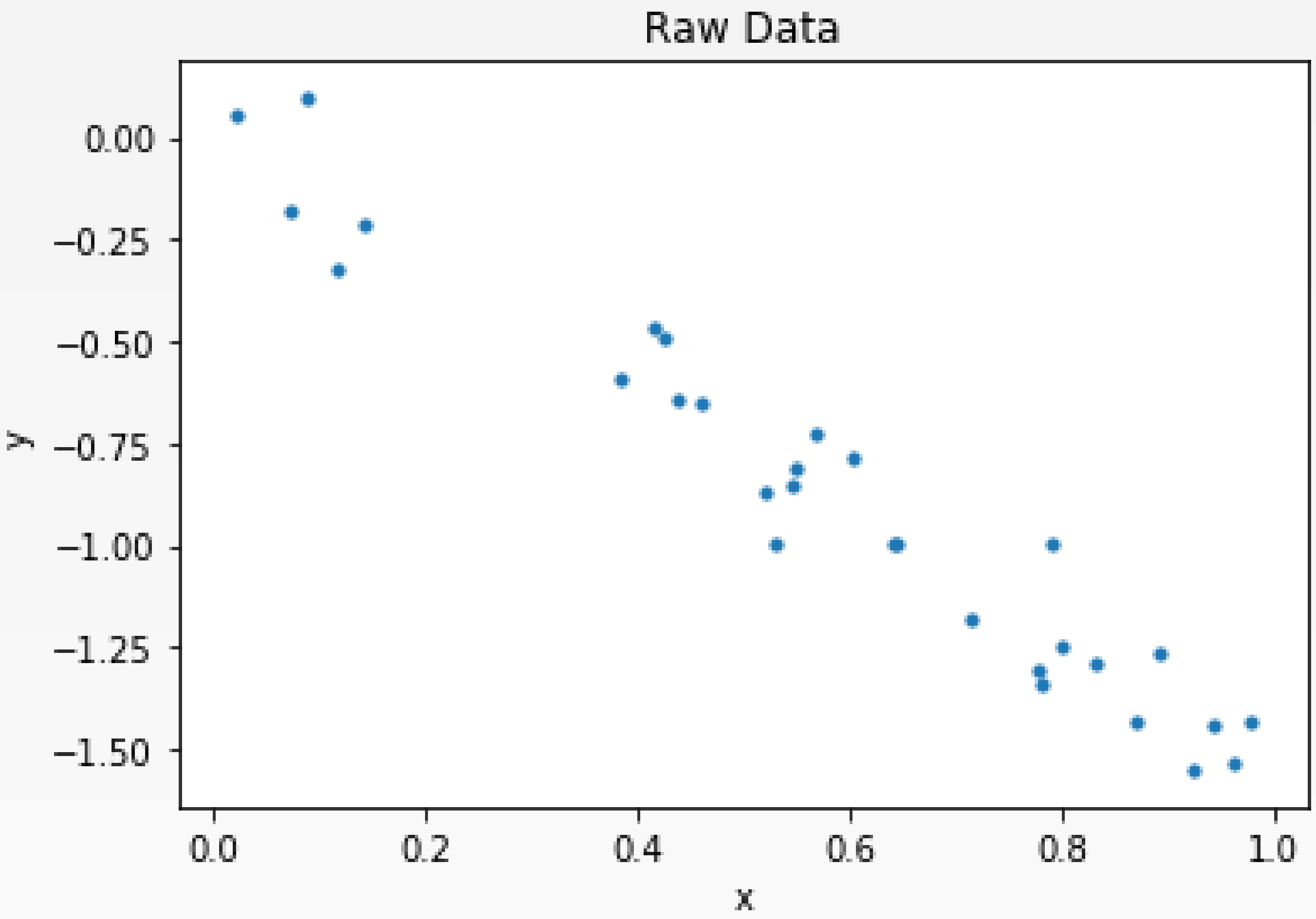
Example I: Polynomial Features



True: $y = \tanh(x) + randn$

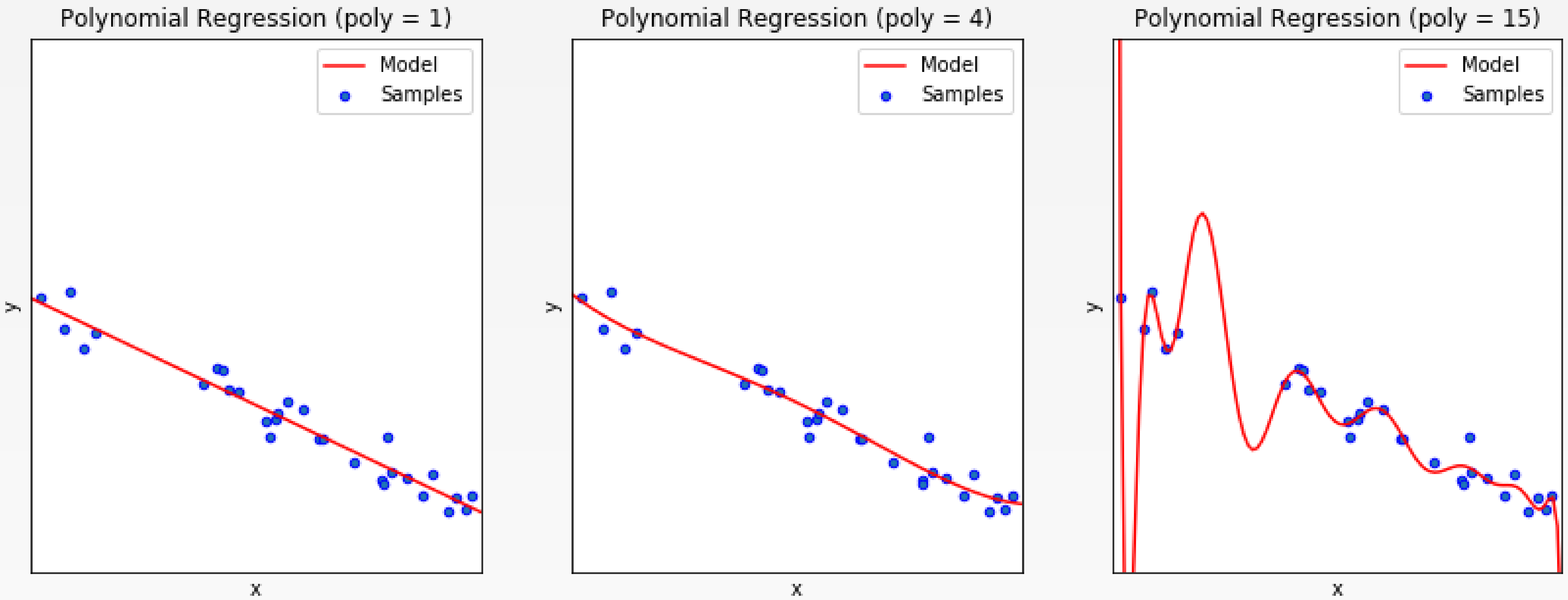
光华管理学院
Guanghua School of Management

Example II: Polynomial Features



True: $y = -1.5 \cdot x + randn$

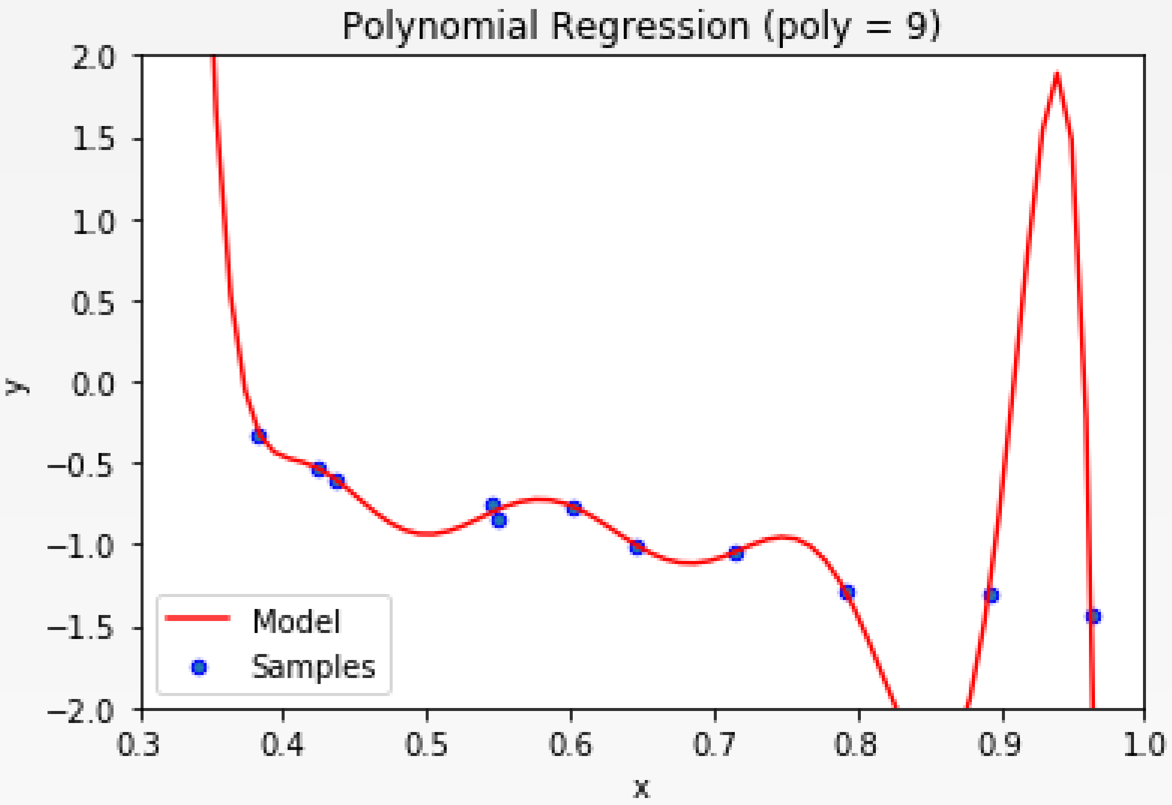
Example II: Polynomial Features



True: $y = -1.5 \cdot x + randn$

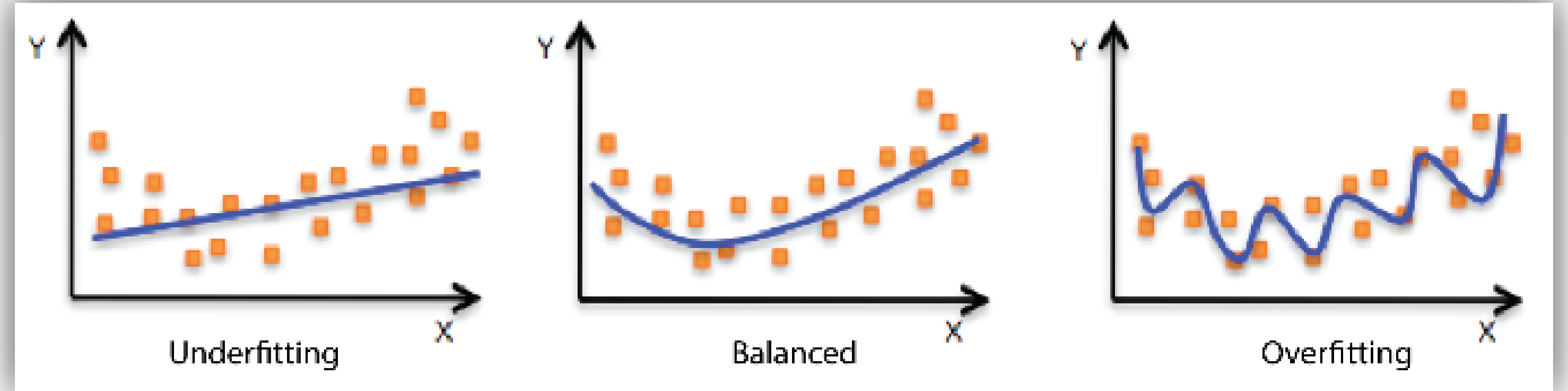
Overfitting

True:
$y = -1.5 \cdot x + randn$



Overfitting Definition: when the model captures the noise in the training data instead of the underlying structure.

Guanghua School of Management


Y
X
Underfitting
Y
X
Balanced
Y
X
Overfitting

Regularization

Regularization

- Goal: optimize some combination of fit and simplicity
  - Penalize the magnitude of coefficients of features
  - Minimize the error between predicted and actual examples

- Ridge Regression:
  - L2-norm: adds penalty equivalent to square of the magnitude of coefficients
- Lasso Regression:
  - L1-norm: adds penalty equivalent to absolute value of the magnitude of coefficients

Ridge Regression

- Recall: in a linear regression with the least square estimation

$$RSS = \sum_{i=1}^{n} \left(y_i - (\omega \cdot x_i + b)\right)^2$$

- Ridge regression

$$RSS = \sum_{i=1}^{n} \left(y_i - (\omega \cdot x_i + b)\right)^2 + \boxed{\alpha \sum_{j=1}^{p} \omega_j^2}$$

Shrinkage penalty

Ridge Regression: λ

- Ridge regression

$$RSS = \sum_{i=1}^{n}\left(y_i - (\omega \cdot x_i + b)\right)^2 + \alpha \sum_{j=1}^{p} \omega_j^2$$

α: tuning parameter

- $\alpha = 0$:
  - A simple linear regression
- $\alpha = \infty$:
  - Coefficients ω will be zero

- As α increases, the flexibility of the model fit decreases

LASSO

Least Absolute Shrinkage and Selection Operator Regression

- L1 Regularization

$$RSS = \sum_{i=1}^{n} \left(y_i - (\omega \cdot x_i + b)\right)^2 + \alpha \sum_{j=1}^{p} |\omega_j|$$

- Lasso combines some of the shrinking advantages of ridge with **variable selection**
  - The L1 penalty has the effects of forcing some coefficient estimates to be exactly equal to zero

Tip: Techniques such as cross validation are recommended to determine which approach is better on a particular dataset

光华管理学院
Guanghua School of Management

Questions?