Multivariate Linear Regression

Teeradaj Racharak (เอ็กซ์)
r.teeradaj@gmail.com

Single Feature (Recap)

In the previous version of linear regression,

Size (feet^2) x	Price ($1000) y
2104	460
1416	232
1534	315
852	178
…	…

$$h_\theta(x) = \theta_0 + \theta_1 x$$

Multiple Features (Variables)

x_1 Size (feet^2)	x_2 Number of bedrooms	x_3 Number of floors	x_4 Age of home (years)	y Price (\$1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

Notation :

- n : number of features
- $x^{(i)}$: input (features) of i^{th} training example
- $x^{(i)}_j$: value of feature j in i^{th} training example

Multiple Features (Variables)

x_1 Size (feet^2)	x_2 Number of bedrooms	x_3 Number of floors	x_4 Age of home (years)	y Price (\$1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

Notation :

- n : number of features
- $x^{(i)}$: input (features) of i^{th} training example
- $x^{(i)}_j$: value of feature j in i^{th} training example

$$n = 4 \qquad x^{(2)} = \begin{bmatrix} 1416 \\ 3 \\ 2 \\ 40 \end{bmatrix} \qquad x_3^{(2)} = 2$$

Question

Size (feet^2)	Number of	Number of	Age of home	Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

- In the training set above, what is $x_1^{(4)}$?

  (i) The size (in feet^2) of the 1st home in the training set
  (ii) The age (in years) of the 1st home in the training set
  (iii) The size (in feet^2) of the 4th home in the training set
  (iv) The age (in years) of the 4th home in the training set

Hypothesis Function

Previously,

$$h_\theta(x) = \theta_0 + \theta_1 x$$

Hypothesis (Reviewed)

Previously,

$$\cancel{h_\theta(x) = \theta_0 + \theta_1 x}$$

In our example,

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$$

E.g. $h_\theta(x) = 80 + 0.1x_1 + 0.01x_2 + 3x_3 - 2x_4$

Hypothesis (Multiple Features)

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n$$

For convenience of notation, define $\boxed{x_0 = 1}$

$$\iff x_0^{(i)} = 1$$

Hypothesis (Multiple Features)

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n$$

For convenience of notation, define $x_0 = 1$

$$\iff x_0^{(i)} = 1$$

$$\iff \boldsymbol{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1}$$

Hypothesis (Multiple Features)

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n$$

Then, we have

$$\boldsymbol{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1} \quad \text{and} \quad \boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \in \mathbb{R}^{n+1}$$

Hypothesis (Multiple Features)

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n$$

Then, we have

$$\boldsymbol{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1} \quad \text{and} \quad \boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \in \mathbb{R}^{n+1}$$

Therefore, we can rewrite as:

$$\begin{bmatrix} \theta_0 & \theta_1 & \ldots & \theta_n \end{bmatrix} \times \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$$

Hypothesis (Multiple Features)

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n$$

$$= \begin{bmatrix} \theta_0 & \theta_1 & \ldots & \theta_n \end{bmatrix} \times \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$$

$$= \boldsymbol{\theta}^T \boldsymbol{x}$$

‘Multivariate Linear Regression’

Gradient Descent for Multiple Features

Things we have

Hypothesis: $h_\theta(x) = \boldsymbol{\theta}^T \boldsymbol{x} = \theta_0 x_0 + \theta_1 x_1 + \ldots \theta_n x_n$

Parameters: $\theta_0, \theta_1, \ldots, \theta_n$

Cost Function: $J(\theta_0, \theta_1, \ldots, \theta_n) = \frac{1}{2m} \Sigma_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

Let's vectorize them

Hypothesis: $h_\theta(x) = \boldsymbol{\theta}^T \boldsymbol{x} = \theta_0 x_0 + \theta_1 x_1 + \ldots + \theta_n x_n$

Parameters: $\cancel{\theta_0, \theta_1, \ldots, \theta_n}$ $\boldsymbol{\theta} \in \mathbb{R}^{n+1}$

Cost Function: $\cancel{J(\theta_0, \theta_1, \ldots, \theta_n)} = \frac{1}{2m} \Sigma_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

$J(\boldsymbol{\theta})$

Question

- When there are n features, we define the cost function as

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

For linear regression, which of the following are also equivalent and correct definitions of $J(\theta)$?

$$\boxed{\begin{gathered} J(\theta) = \frac{1}{2m}\Sigma_{i=1}^{m}(\boldsymbol{\theta}^T\mathbf{x}^{(i)} - y^{(i)})^2 \\ J(\theta) = \frac{1}{2m}\Sigma_{i=1}^{m}((\Sigma_{j=0}^{n}\theta_j x_j^{(i)}) - y^{(i)})^2 \end{gathered}}$$

$$J(\theta) = \frac{1}{2m}\Sigma_{i=1}^{m}((\Sigma_{j=1}^{n}\theta_j x_j^{(i)}) - y^{(i)})^2$$

$$J(\theta) = \frac{1}{2m}\Sigma_{i=1}^{m}((\Sigma_{j=1}^{n}\theta_j x_j^{(i)}) - (\Sigma_{j=0}^{n}y_j^{(i)}))^2$$

Vectorizing Gradient Descent

Gradient Descent:

cost function

Repeat {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \ldots, \theta_n)$$

}

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

Vectorizing Gradient Descent

Gradient Descent:

Repeat {

$$J(\boldsymbol{\theta})$$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\cancel{\theta_0, \ldots, \theta_n})$$

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

Single Feature (Recap)

Single feature (n = 1):

Repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \underbrace{\Sigma_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})}_{\frac{\partial}{\partial \theta_0} J(\theta)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \Sigma_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x^{(i)}$$

}

(simultaneously update θ_0, θ_1)

Single Feature (Recap)

Single feature (n = 1):

Repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \underbrace{\Sigma_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})}_{\frac{\partial}{\partial \theta_0} J(\theta)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \Sigma_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cancel{x^{(i)}} \, x_1^{(i)}$$

}

(simultaneously update θ_0, θ_1)

Single *vs.* Multiple Features

Single feature (n = 1):

Repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \underbrace{\Sigma_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})}_{\frac{\partial}{\partial \theta_0} J(\theta)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \Sigma_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_1^{(i)}$$

(simultaneously update θ_0, θ_1)

}

Single *vs.* Multiple Features

Single feature (n = 1):

Repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \underbrace{\Sigma_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})}_{\frac{\partial}{\partial \theta_0} J(\theta)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \Sigma_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)}) x_1^{(i)}$$

} (simultaneously update θ_0, θ_1)

Multiple features (n ≥ 1):

Repeat {

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

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

Single *vs.* Multiple Features

Multiple features (n ≥ 1):

Repeat {

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

}

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

E.g.

$$\boxed{\begin{aligned} \theta_0 &:= \theta_0 - \alpha \frac{1}{m} \Sigma_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)} \\ \theta_1 &:= \theta_1 - \alpha \frac{1}{m} \Sigma_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_1^{(i)} \\ \theta_2 &:= \theta_2 - \alpha \frac{1}{m} \Sigma_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_2^{(i)} \end{aligned}}$$

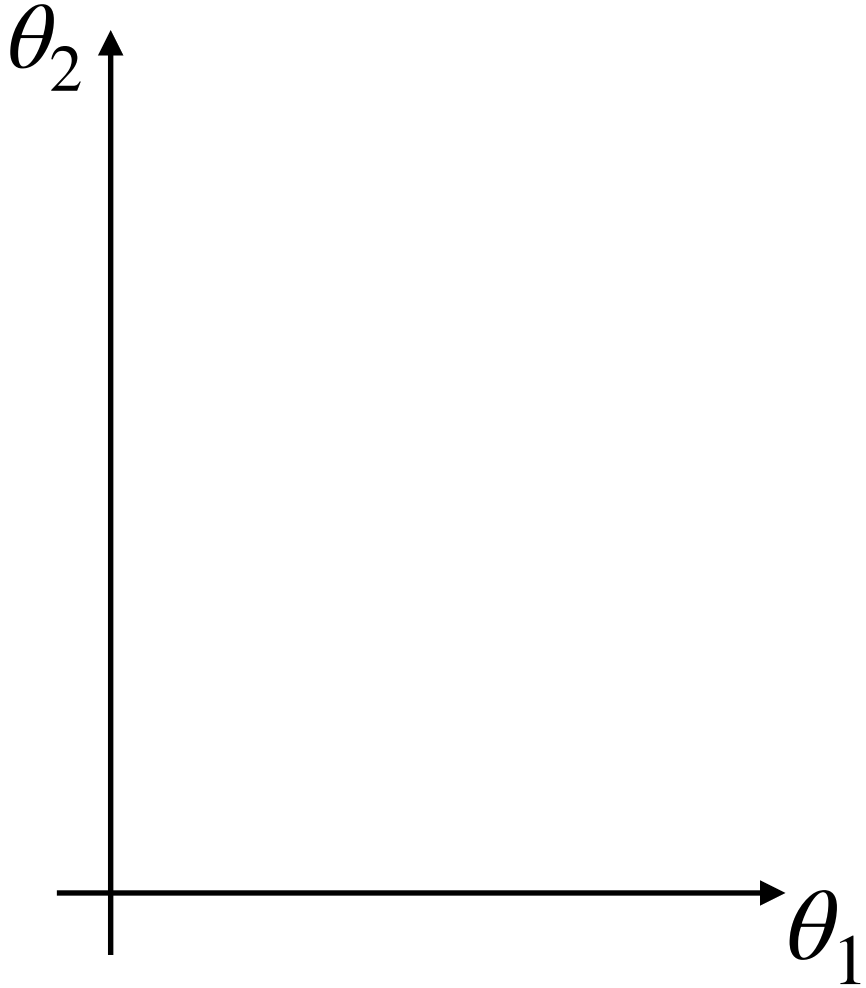
Gradient Descent in Practice I - Feature Scaling

Feature Scaling (Intuitively)

Idea: Make sure that features are on a similar scale

e.g. x_1 = size (0 - 2000 feet^2)

x_2 = number of bedrooms (1 - 5)

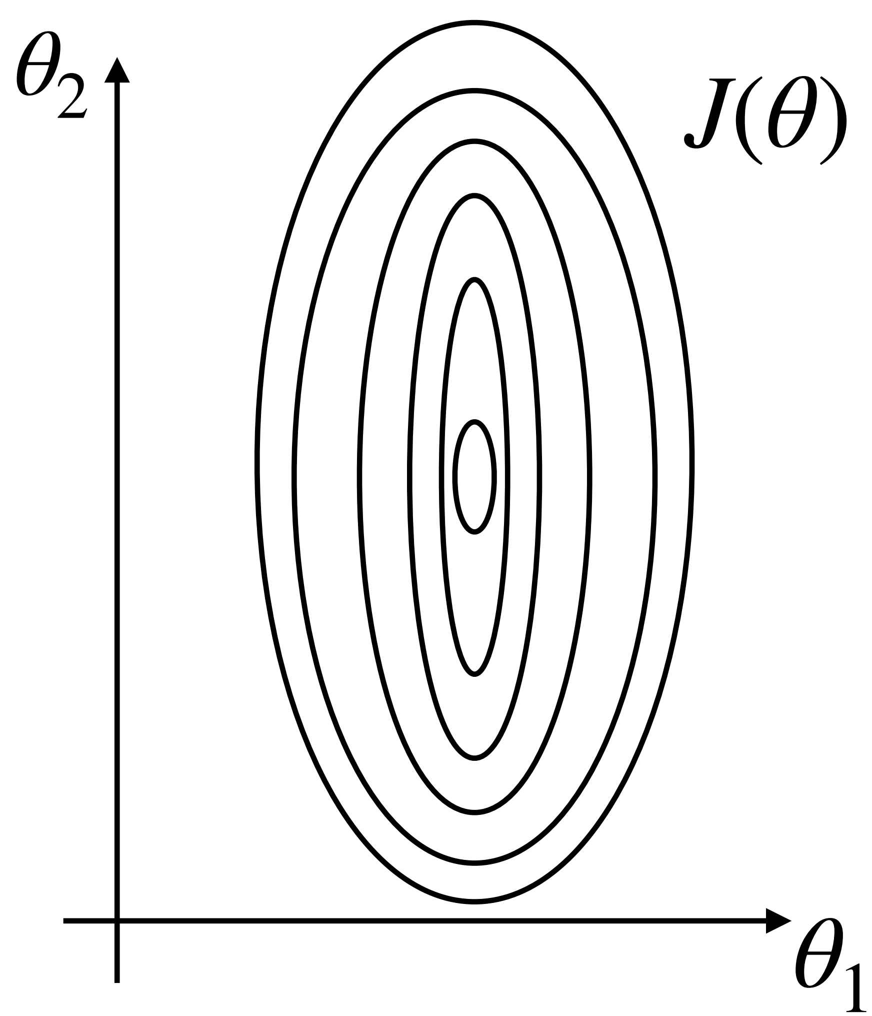


Feature Scaling (Intuitively)

Idea: Make sure that features are on a similar scale

e.g. x_1 = size (0 - 2000 feet^2)

x_2 = number of bedrooms (1 - 5)

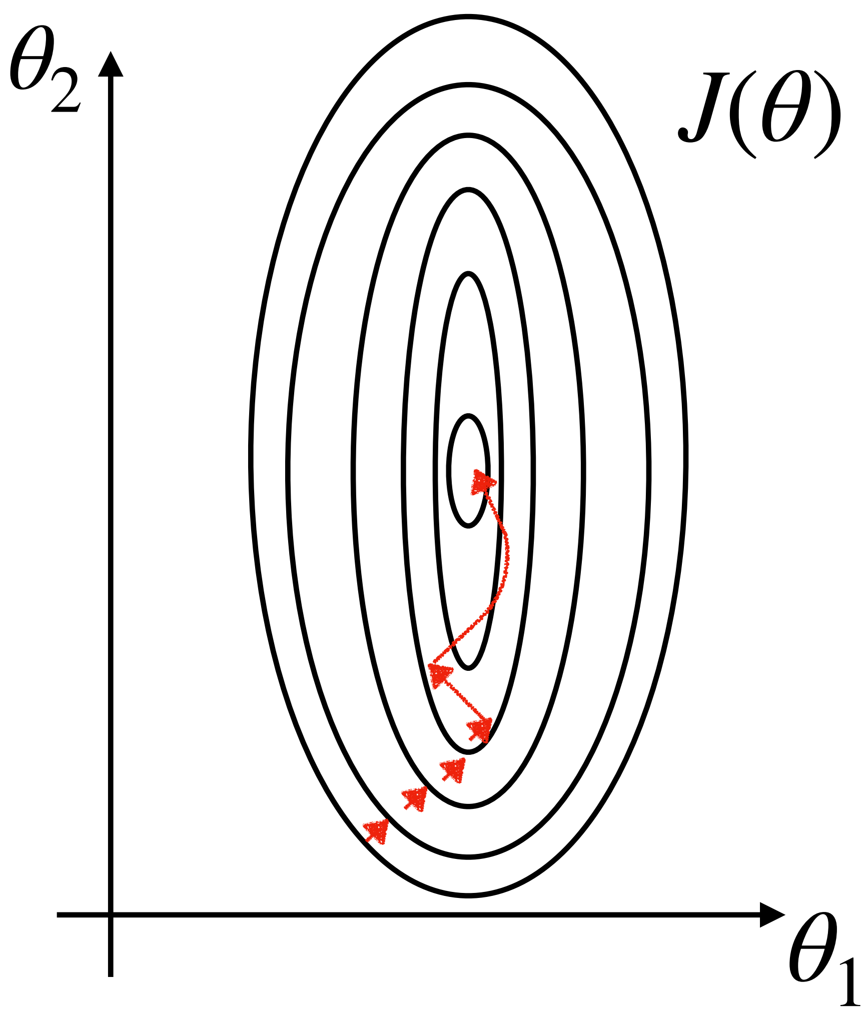


Feature Scaling (Intuitively)

Idea: Make sure that features are on a similar scale

e.g. x_1 = size (0 - 2000 feet^2)

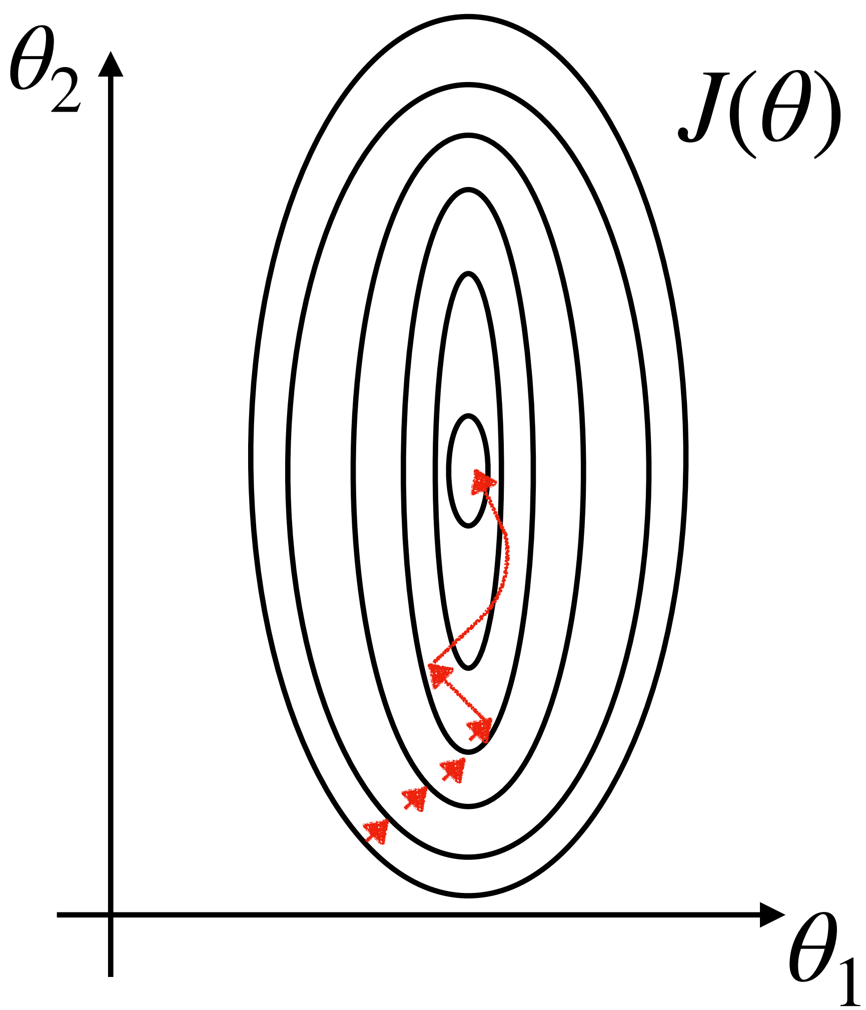
x_2 = number of bedrooms (1 - 5)



Feature Scaling (Intuitively)

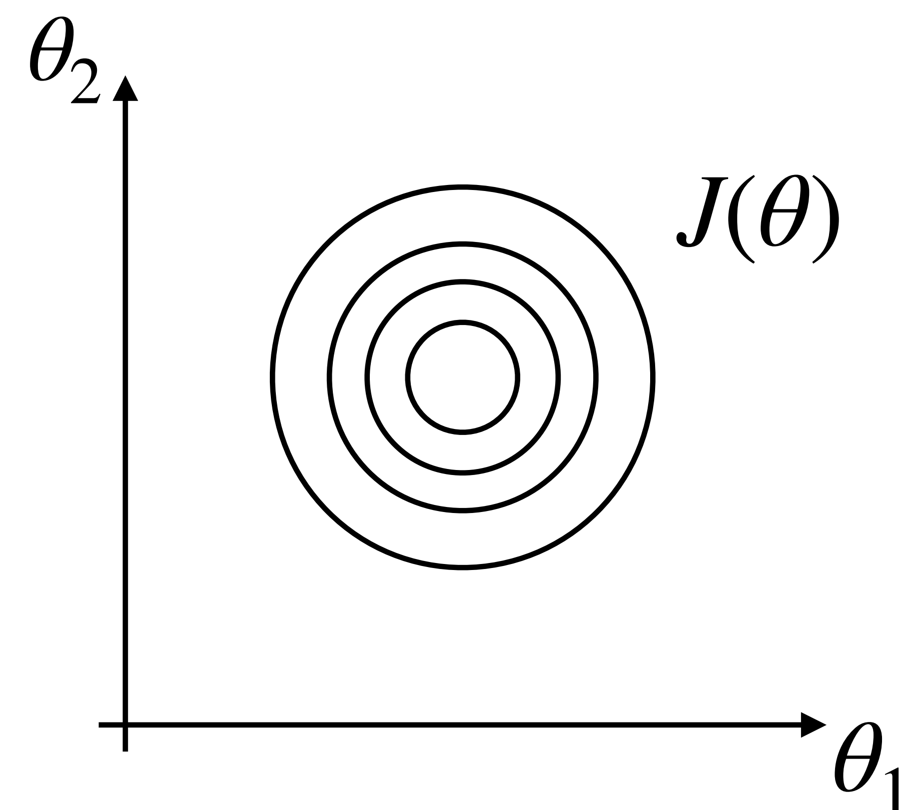
Idea: Make sure that features are on a similar scale

e.g. x_1 = size (0 - 2000 feet^2)
x_2 = number of bedrooms (1 - 5)



x_1 = size (feet^2) / 2000

x_2 = #bedrooms / 5

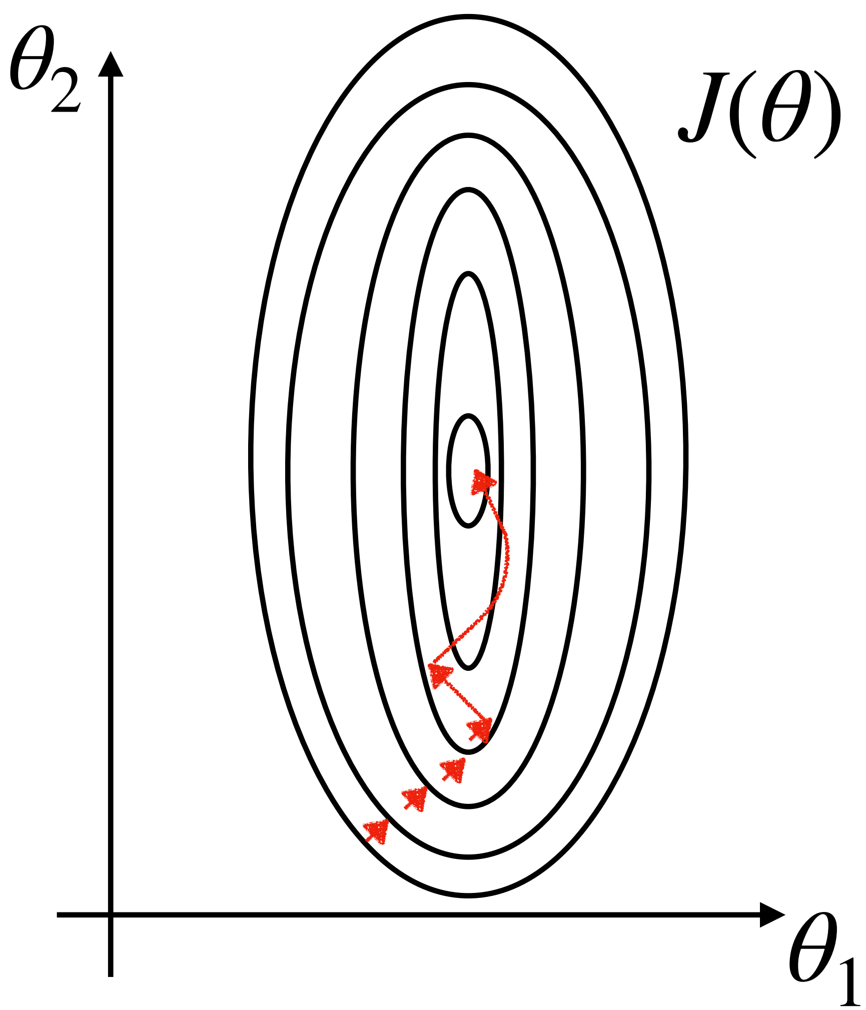


Feature Scaling (Intuitively)

Idea: Make sure that features are on a similar scale

e.g. x_1 = size (0 - 2000 feet^2)

x_2 = number of bedrooms (1 - 5)

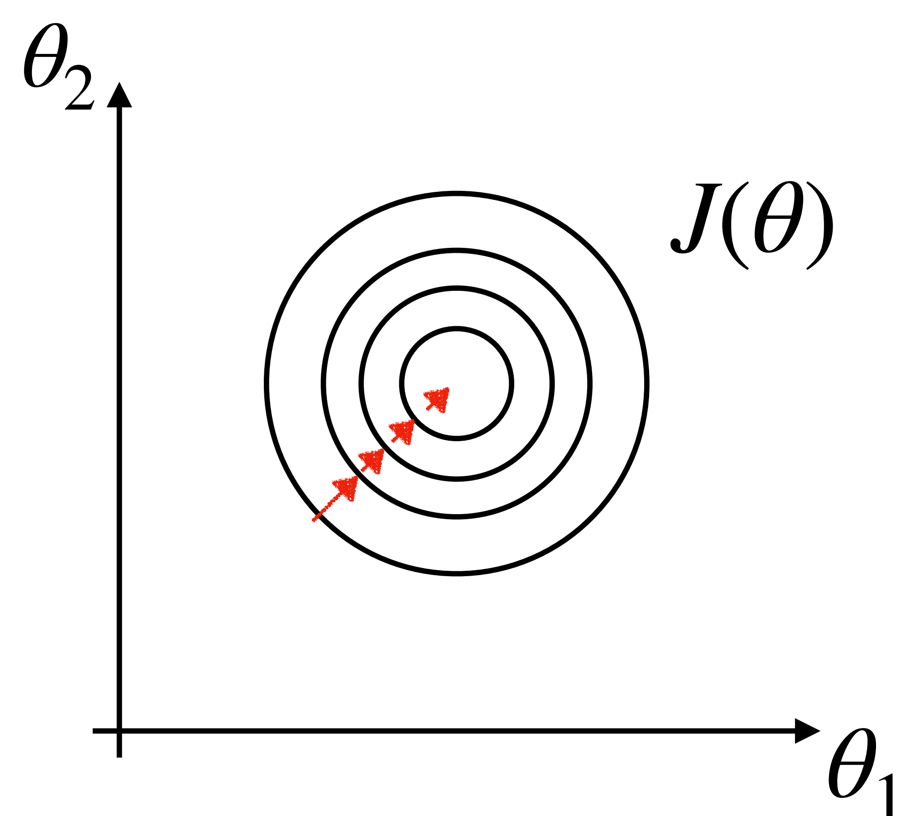


$0 \leq x_1 \leq 1$

x_1 = size (feet^2) / 2000

$0 \leq x_2 \leq 1$

x_2 = #bedrooms / 5



Feature Scaling (in Practice)

- Get every feature into approximately a $-1 \leq x_i \leq 1$ range
- In fact, the range [-1, 1] is not so necessary *e.g.*

$$0 \leq x_1 \leq 3 \qquad \text{(fine)}$$

$$-2 \leq x_2 \leq 0.5 \qquad \text{(fine)}$$

$$-100 \leq x_3 \leq 100 \qquad \text{(too big)}$$

$$-0.0001 \leq x_4 \leq 0.0001 \qquad \text{(too small)}$$

Feature Scaling (in Practice)

Mean Normalization

$(S_i := \max - \min)$ or the standard deviation.

Redefine x_i as $(x_i - \mu_i)/S_i$
to make features have approximately zero mean
(do not apply to $x_0 = 1$)

e.g. $x_1 = (size - 1000) / 2000$ (given that $\mu_1 = 1000$)
$x_2 = (\#bedrooms - 2) / 4$ (given that $\mu_2 = 2$)

Question

- Suppose you are using a learning algorithm to estimate the price of houses in a city. You want one of your features x_i to capture the age of the house. In your training set, all of your houses have an age between 30 and 50 years, with an average age of 38 years. Which of the following would you use as features, assuming you use feature scaling and mean normalization?

(i) x_i = age of house
(ii) x_i = age of house / 50
(iii) x_i = (age of house - 38) / 50
(iv) x_i = (age of house - 38) / 20

Gradient Descent in Practice II - Learning Rate

Gradient Descent (Recap)

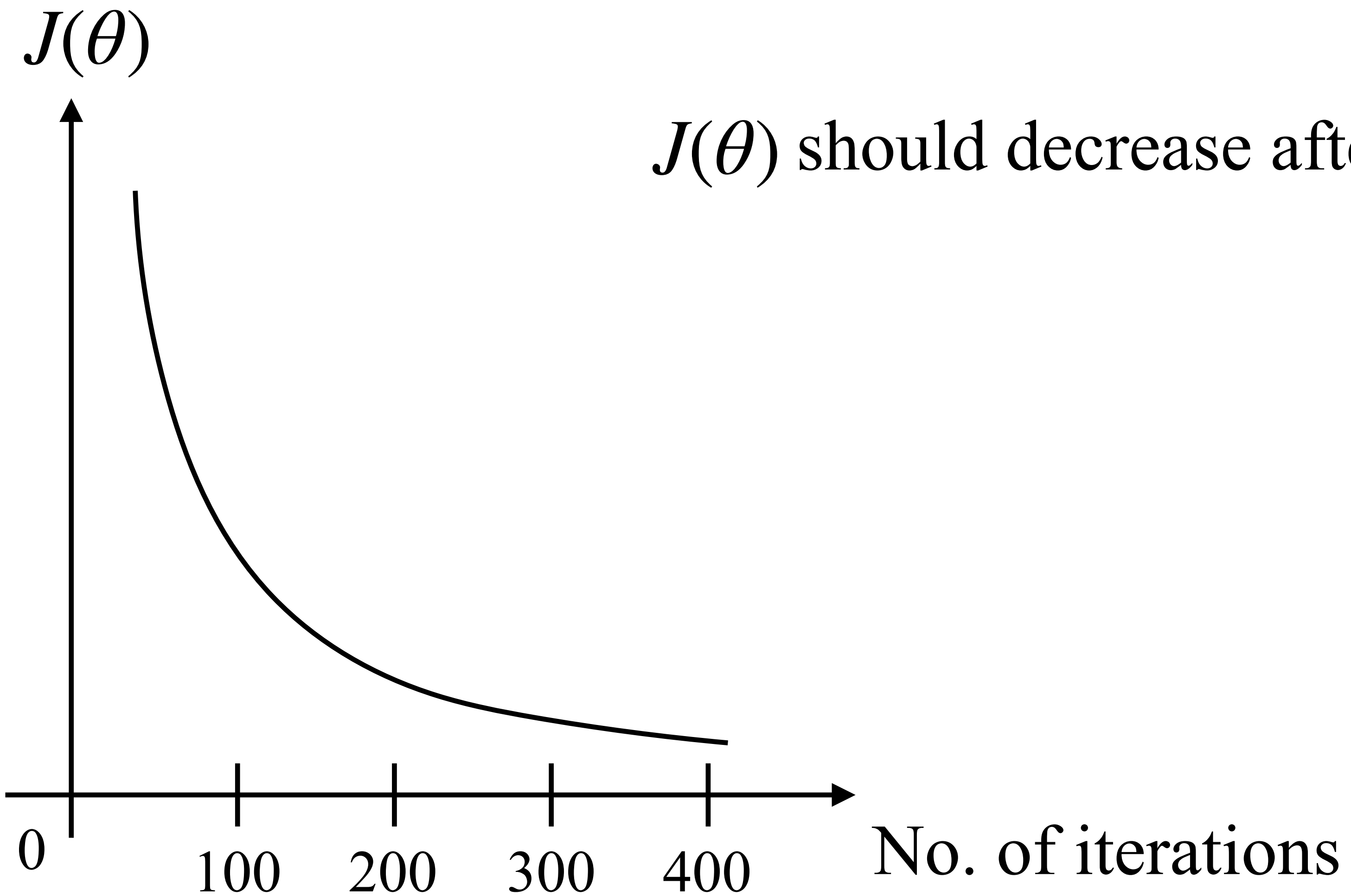
Gradient Descent Update Rule

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

- 'Debugging' : How to make sure gradient descent is working correctly.
- How to choose learning rate

Is descent working correctly?

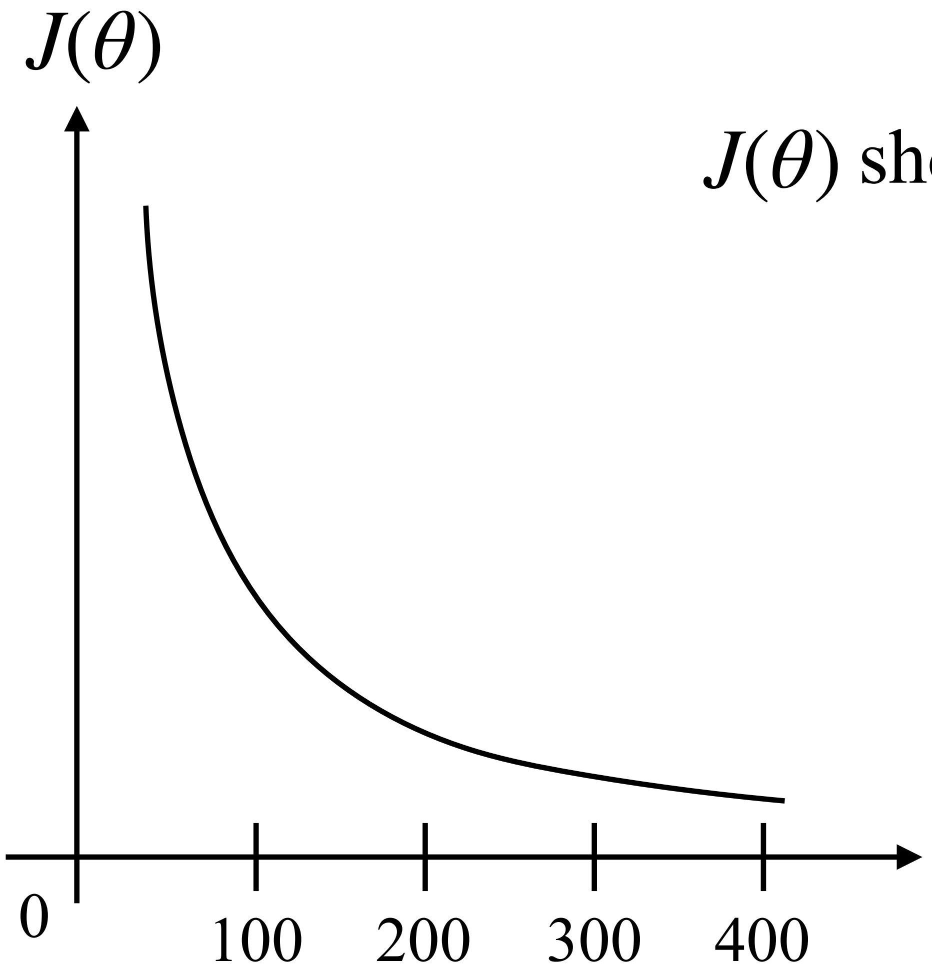
Some tips to make sure that gradient descent is working correctly.



$J(\theta)$ should decrease after every iteration

Is descent working correctly?

Some tips to make sure that gradient descent is working correctly.

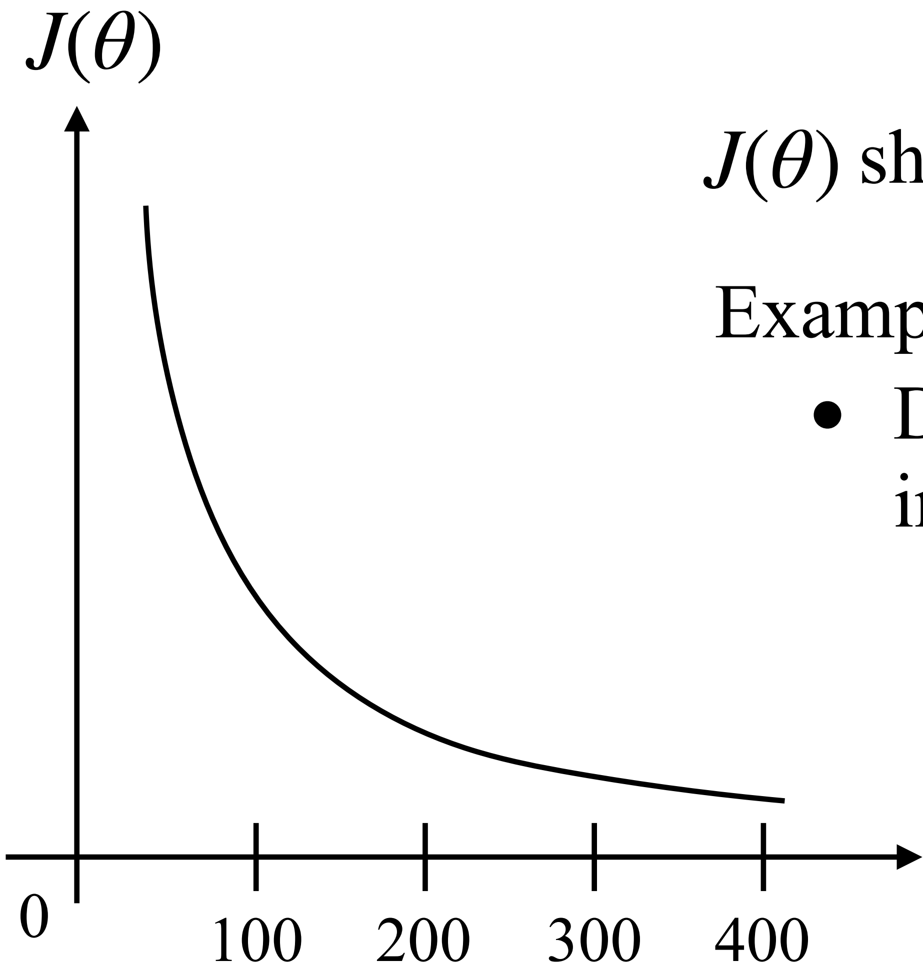


$J(\theta)$ should decrease after every iteration

Should we log this information in TensorBoard ?

Is descent working correctly?

Some tips to make sure that gradient descent is working correctly.



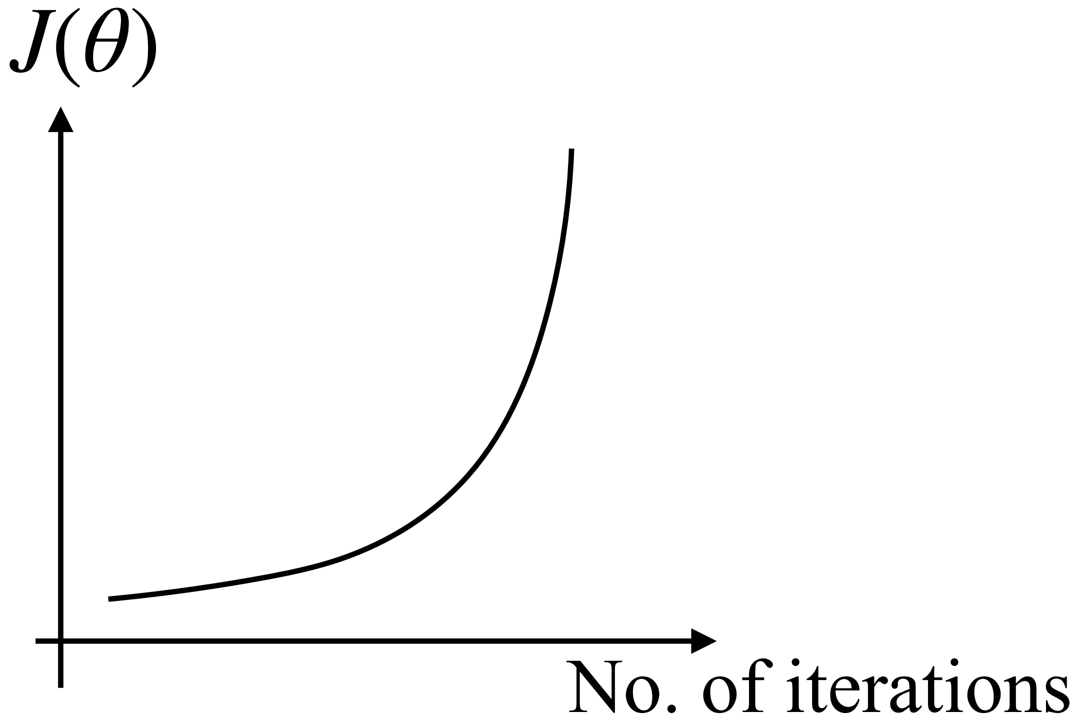
$J(\theta)$ should decrease after every iteration

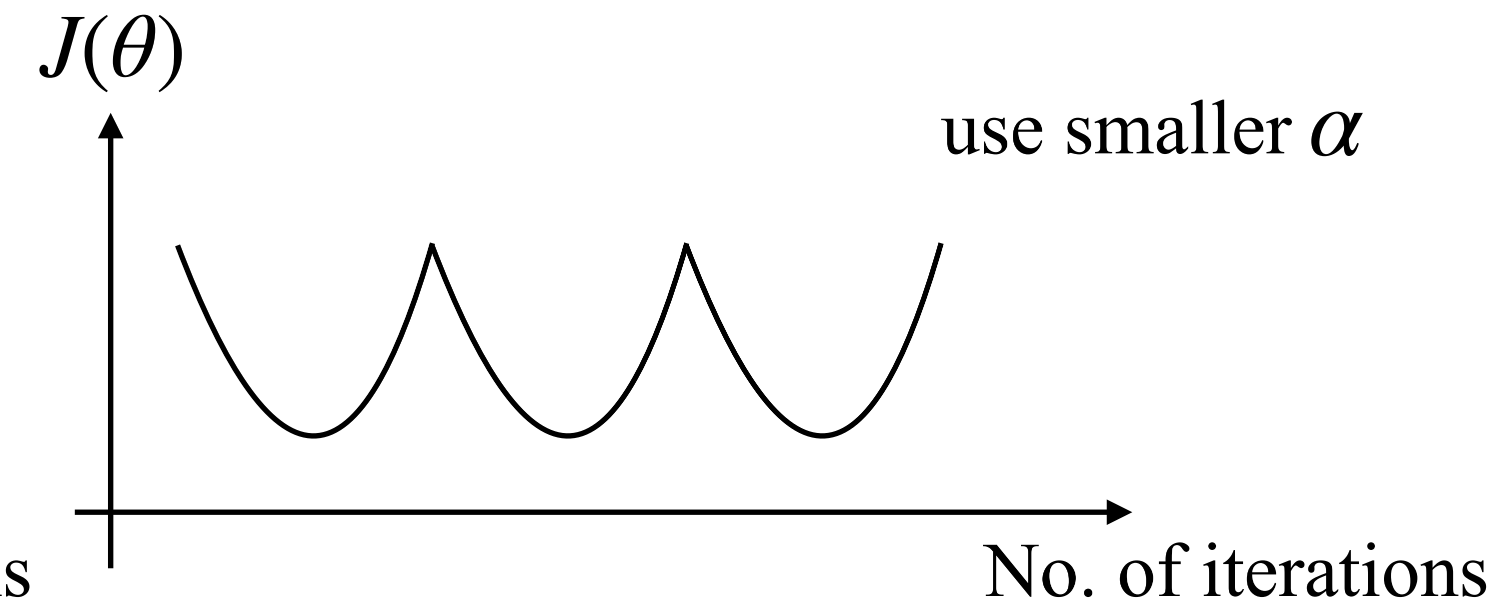
Example of automatic convergence test

- Declare convergence if decreases by less than 10^{-3} in one iteration.

ϵ
(a threshold)

Is descent working correctly?



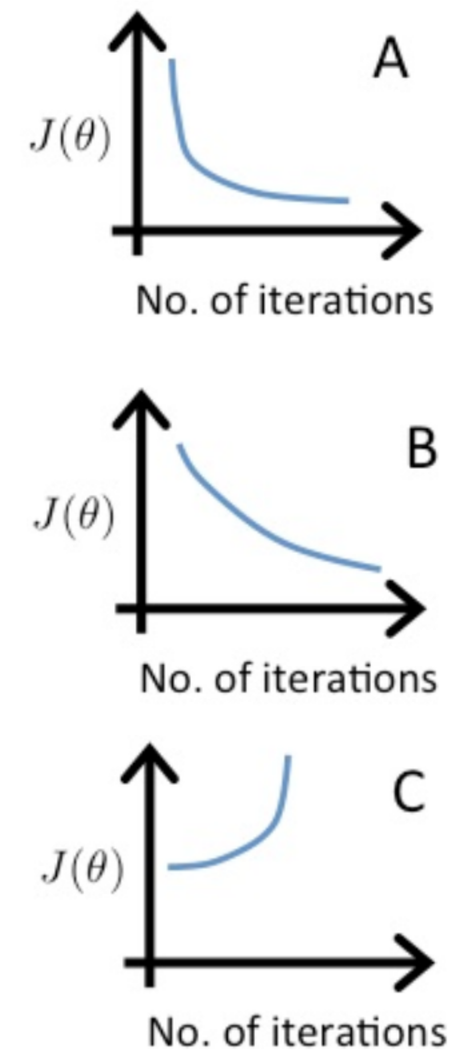


- For sufficiently small α, $J(\theta)$ should decrease on every iteration.
- But if α is too small, gradient descent can be slow to converge.

Question

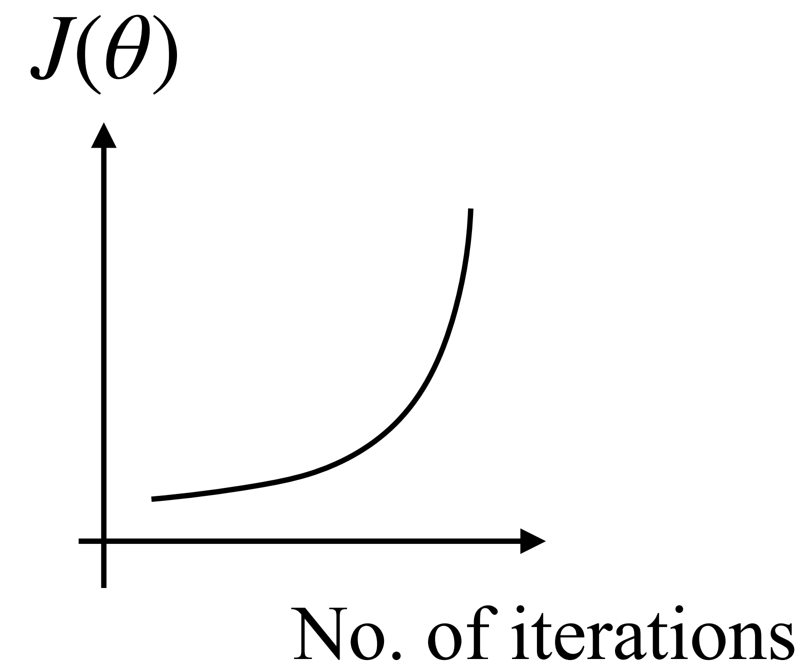
- Suppose a friend ran gradient descent three times, with $\alpha = 0.01$, $\alpha = 0.1$, and $\alpha = 1$, and got the following three plots (labeled A, B, and C). Which plots corresponds to which values of α?

(i) A is $\alpha = 0.01$, B is $\alpha = 0.1$, C is $\alpha = 1$
(ii) A is $\alpha = 0.1$, B is $\alpha = 0.01$, C is $\alpha = 1$
(iii) A is $\alpha = 1$, B is $\alpha = 0.01$, C is $\alpha = 0.1$
(iv) A is $\alpha = 1$, B is $\alpha = 0.1$, C is $\alpha = 0.01$



Summary

- If α is too small, then convergence is slow.
- If α is too large, then $J(\theta)$ may not decrease on every iteration; may not converge.



- To choose α, try

…,0.001, ,0.01, ,0.1, ,1,…

…,0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1,…

Features and Polynomial Regression

Selecting Features (Intuitively)

Housing prices prediction

$$h_\theta(x) = \theta_0 + \theta_1 \times \textbf{frontage} + \theta_2 \times \textbf{depth}$$

Selecting Features (Intuitively)

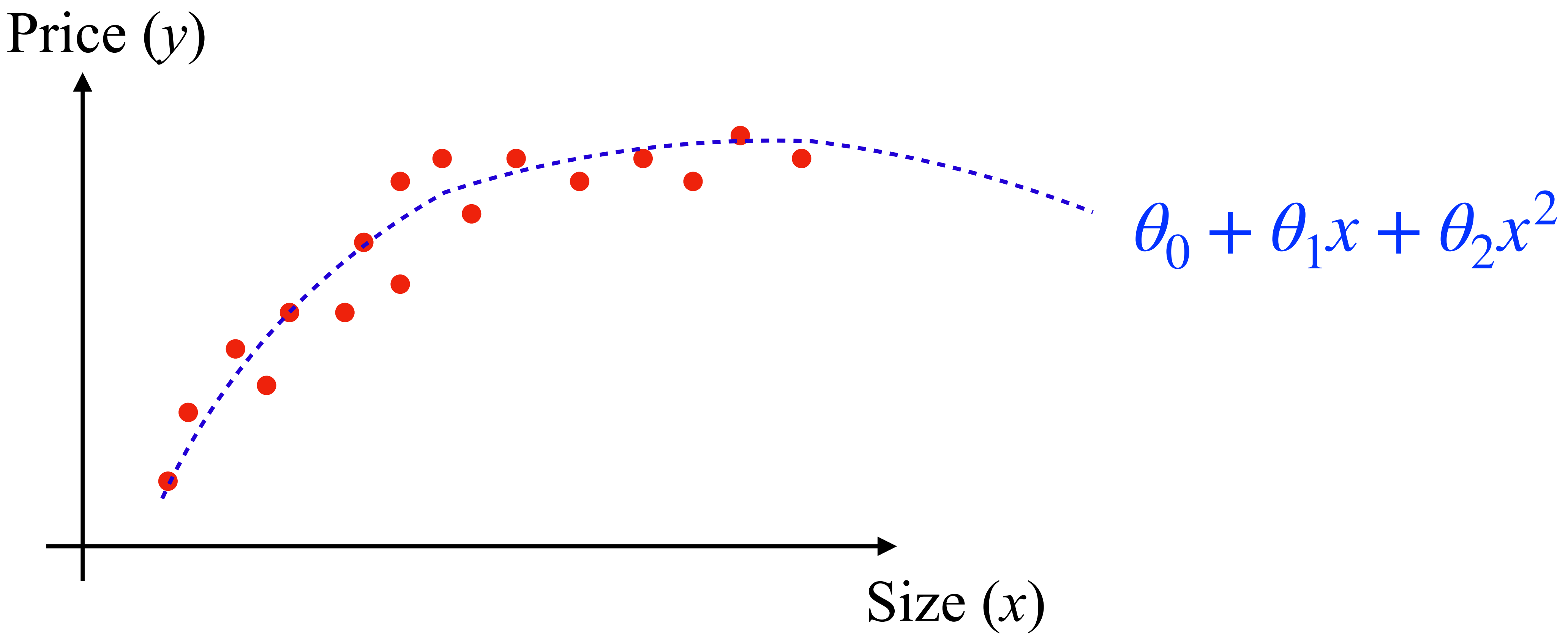
Housing prices prediction

$$h_\theta(x) = \theta_0 + \theta_1 \times \underbrace{\textbf{frontage}}_{x_1} + \theta_2 \times \underbrace{\textbf{depth}}_{x_2}$$

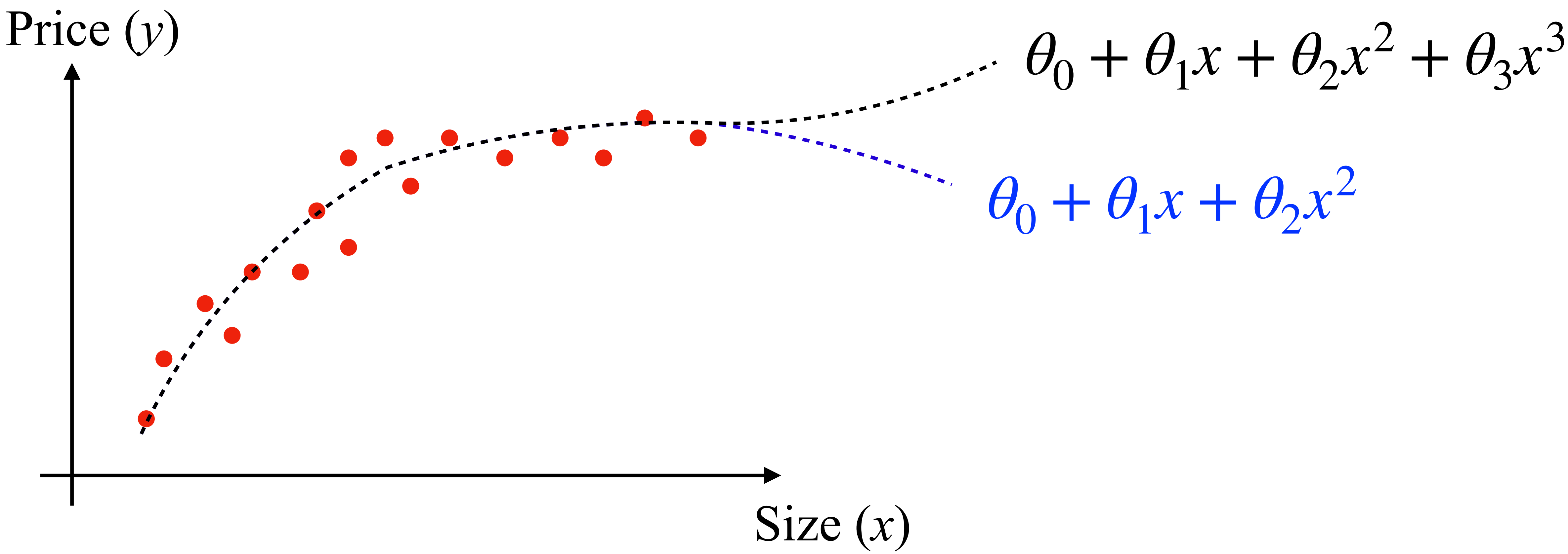
$$\because \textbf{area}(x) = \textbf{frontage} \times \textbf{depth}$$

$$\therefore h_\theta(x) = \theta_0 + \theta_1 x$$

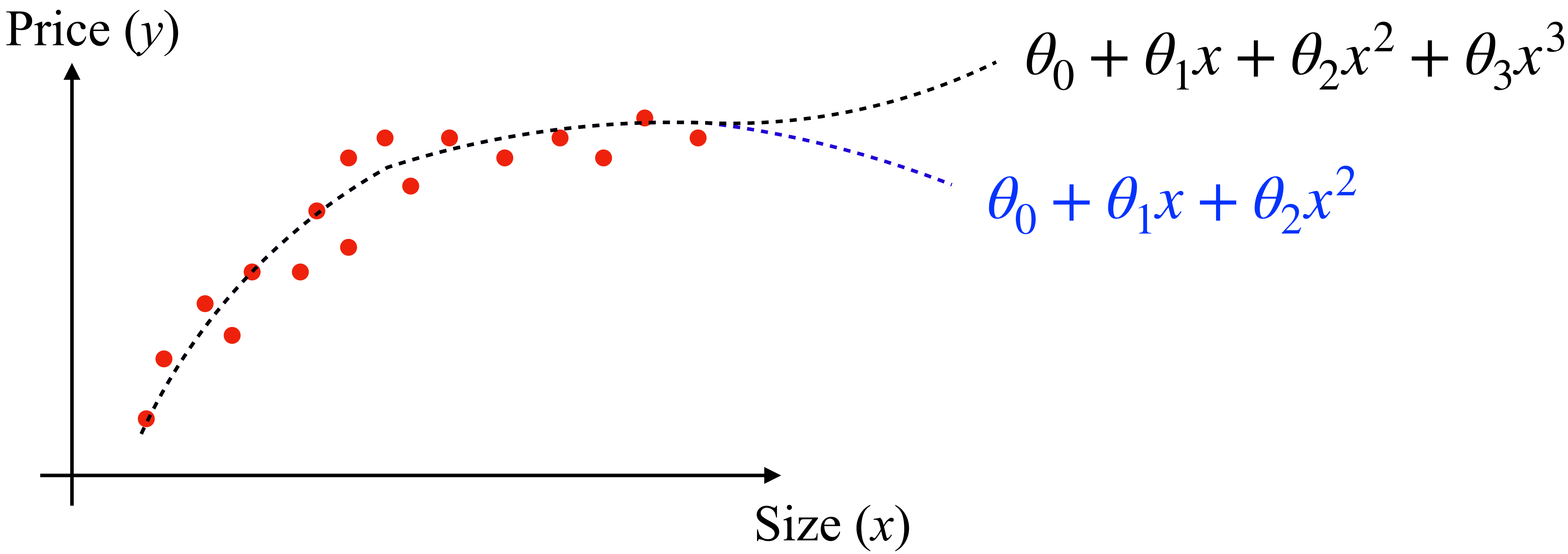
Polynomial Regression



Polynomial Regression

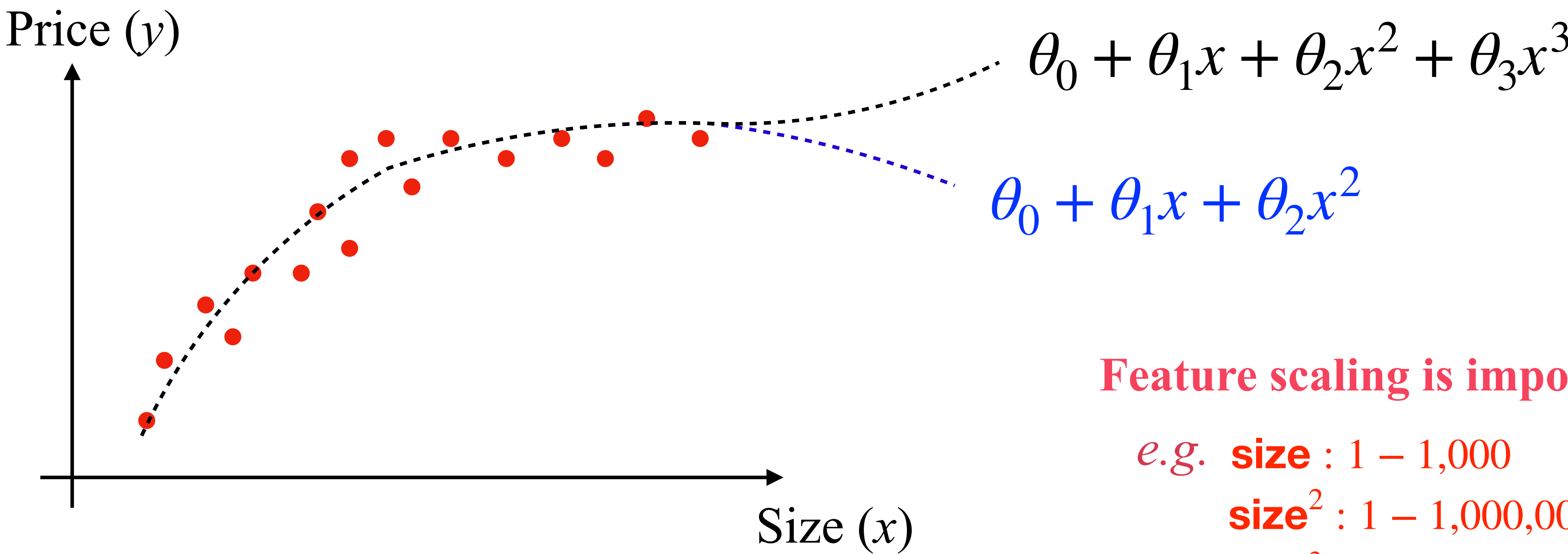


Polynomial Regression



$$\therefore h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$
$$= \theta_0 + \theta_1(\textbf{size}) + \theta_2(\textbf{size}^2) + \theta_3(\textbf{size}^3)$$

Polynomial Regression



Feature scaling is important !

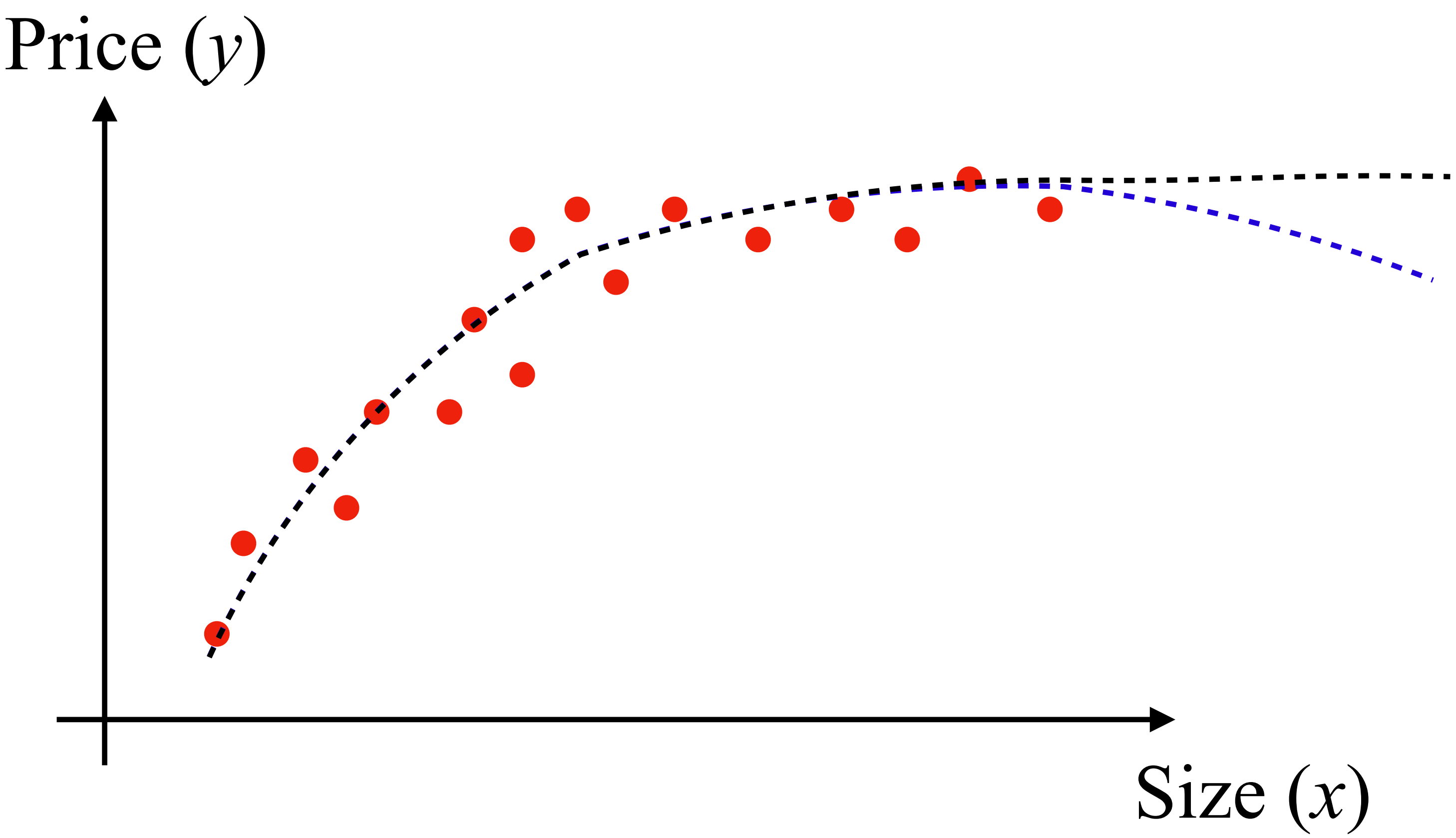
e.g. **size** : 1 – 1,000

size^2 : 1 – 1,000,000

size^3 : 1 – 1,000,000,000

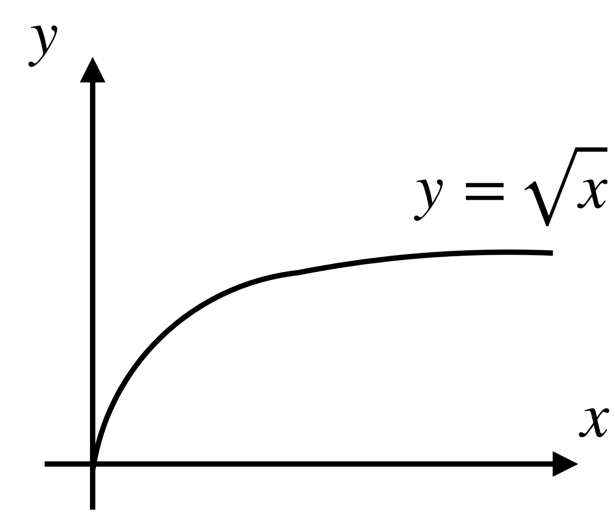
$$\therefore h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$
$$= \theta_0 + \theta_1(\textbf{size}) + \theta_2(\textbf{size}^2) + \theta_3(\textbf{size}^3)$$

Other Hypothesis Functions



$$h_\theta(x) = \theta_0 + \theta_1(\textbf{size}) + \theta_2\sqrt{(\textbf{size})}$$

$$h_\theta(x) = \theta_0 + \theta_1(\textbf{size}) + \theta_2(\textbf{size})^2$$



Question

- Suppose you want to predict a house's price as a function of its size. Your model is: $h_\theta(x) = \theta_0 + \theta_1(\textbf{size}) + \theta_2\sqrt{(\textbf{size})}$. Suppose size ranges from 1 to 1000 (feet^2). You will implement this by fitting a model $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$. Finally, suppose you want to use feature scaling (without mean normalization). Which of the following choices for x_1 and x_2 should you use ? (Note: $\sqrt{1000} \approx 32$)

(i) $x_1 = \textbf{size}$, $x_2 = 32\sqrt{(\textbf{size})}$

(ii) $x_1 = 32(\textbf{size})$, $x_2 = \sqrt{(\textbf{size})}$

(iii) $x_1 = \dfrac{\textbf{size}}{1000}$, $x_2 = \dfrac{\sqrt{(\textbf{size})}}{32}$

(iv) $x_1 = \dfrac{\textbf{size}}{32}$, $x_2 = \sqrt{(\textbf{size})}$

Question

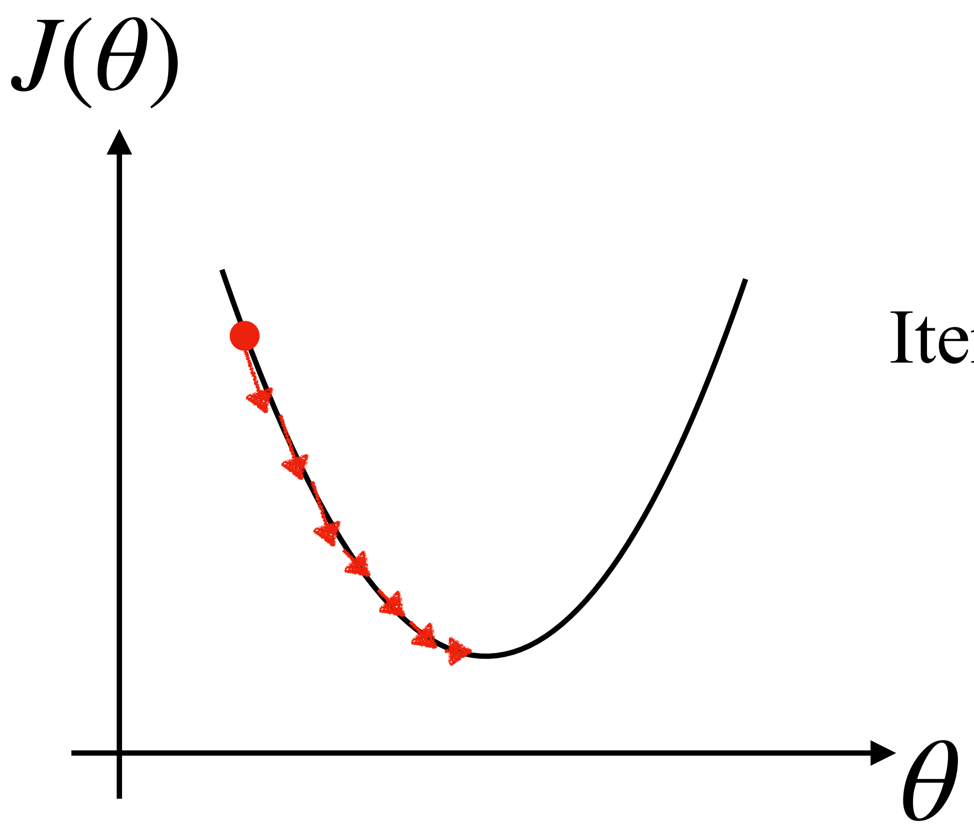
- Suppose $m = 4$ students have taken some class, and the class had a midterm exam and a final exam. You have collected a data set of their scores on the two exams, which is as follows:

midterm exam	(midterm exam)^2	final exam
89	7921	96
72	5184	74
94	8836	87
69	4761	78

You'd like to use polynomial regression to predict a student's final exam score from their midterm exam score. Concretely, suppose you want to fit a model of the form: $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$, where x_1 is the midterm score and x_2 is (midterm score)^2. Further, you plan to use both feature scaling and mean normalization. What is the normalized feature $x_2^{(4)}$?

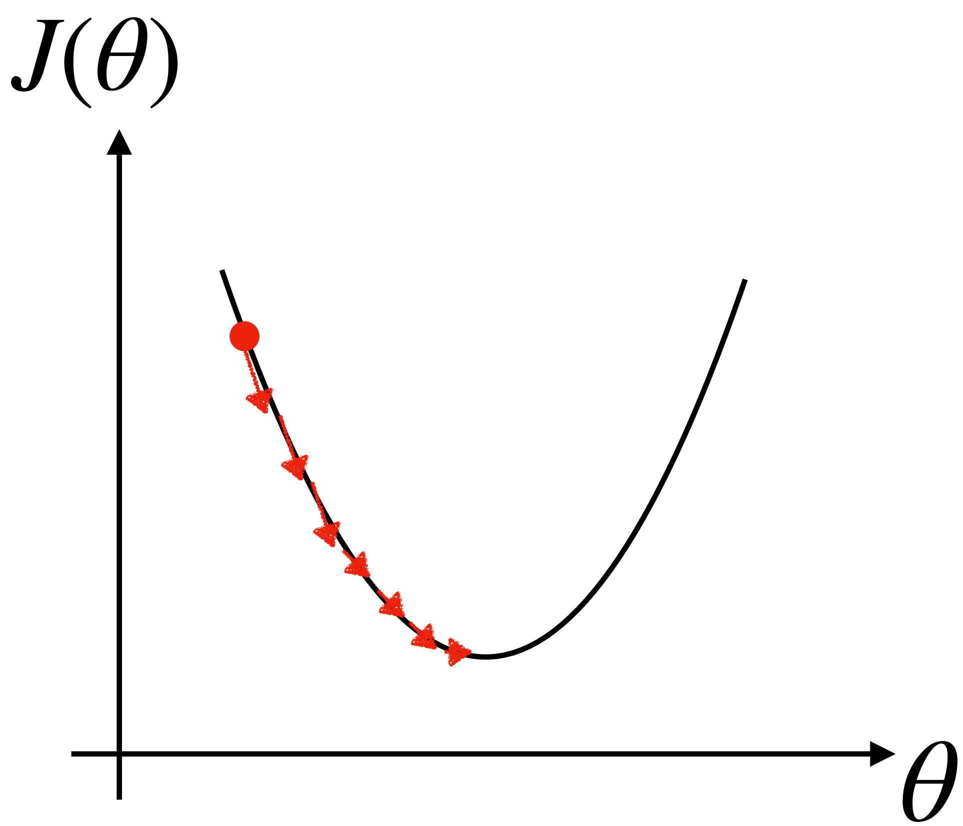
Normal Equation

Gradient Descent (Recap)



Iterative algorithm to converge the local minimum

Gradient Descent *vs.* Normal Equation



Iterative algorithm to converge the local minimum

<u>Normal equation:</u>

Method to solve for θ analytically

i.e. we can solve for the local minimum within just one step

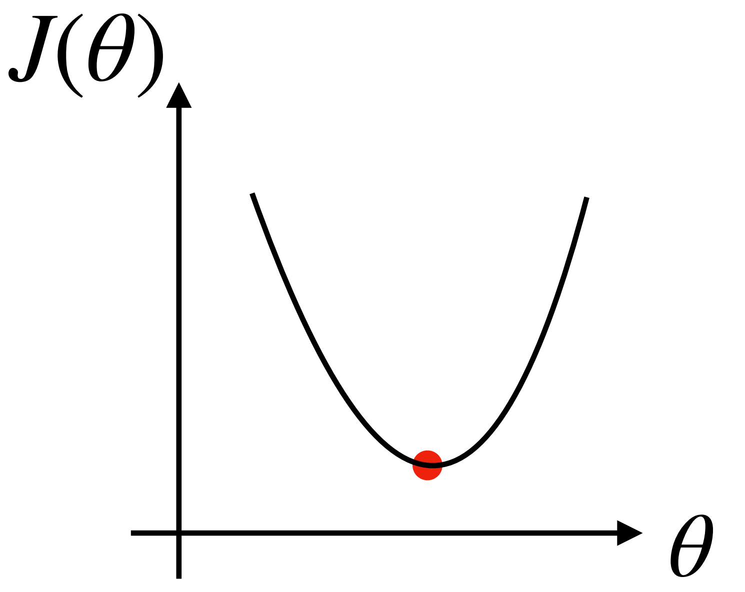
Normal Equation (Intuitively)

Intuition : $(\theta \in \mathbb{R})$

Let $J(\theta) = a\theta^2 + b\theta + c$

To solve for the optimum,
we take the derivative, set to 0 and solve for θ

i.e. $\frac{\partial}{\partial\theta}J(\theta) = 2a\theta + b = 0$



Normal Equation (Intuitively)

To solve for the optimum,
we take the derivative, set to 0 and solve for θ

i.e. $\frac{\partial}{\partial \theta} J(\theta) = 2a\theta + b = 0$

$$\theta \in \mathbb{R}^{n+1} \implies J(\theta_0, \theta_1, \ldots, \theta_m) = \frac{1}{2m} \Sigma_{i=1}^{m} (h_\theta(x^i) - y^{(i)})^2$$

$$\frac{\partial}{\partial \theta_j} J(\theta) = \ldots = 0 \quad \text{(for every } j\text{)}$$

Solving for $\theta_0, \theta_1, \ldots, \theta_n$ yields $\theta_0, \theta_1, \ldots, \theta_n$ that minimizes $J(\theta_0, \ldots, \theta_m)$

Notice that feature scaling is not necessary for normal equation !

Normal Equation

Since we have to take the derivatives and set them to 0 *i.e.*

$$\frac{\partial J}{\partial \theta_0} = \frac{1}{m}\Sigma_{i=1}^{m}(h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) - y^{(i)}) = 0$$

$$\frac{\partial J}{\partial \theta_1} = \frac{1}{m}\Sigma_{i=1}^{m}(h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) - y^{(i)})x_1^{(i)} = 0$$

$$\vdots$$

$$\frac{\partial J}{\partial \theta_n} = \frac{1}{m}\Sigma_{i=1}^{m}(h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) - y^{(i)})x_n^{(i)} = 0$$

Normal Equation

Since we have to take the derivatives and set them to 0 *i.e.*

$$\frac{1}{m}\Sigma_{i=1}^{m}(h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) - y^{(i)}) = 0 \iff \Sigma_{i=1}^{m}(\theta_0 + \theta_1 x_1^{(i)} + \ldots + \theta_n x_n^{(i)} - y^{(i)}) = 0$$

$$\frac{1}{m}\Sigma_{i=1}^{m}(h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) - y^{(i)})x_1^{(i)} = 0 \iff \Sigma_{i=1}^{m}(\theta_0 x_1^{(i)} + \theta_1 x_1^{(i)} x_1^{(i)} + \ldots + \theta_n x_n^{(i)} x_1^{(i)} - y^{(i)} x_1^{(i)}) = 0$$

$$\vdots$$

$$\frac{1}{m}\Sigma_{i=1}^{m}(h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) - y^{(i)})x_n^{(i)} = 0 \iff \Sigma_{i=1}^{m}(\theta_0 x_n^{(i)} + \theta_1 x_1^{(i)} x_n^{(i)} + \ldots + \theta_n x_n^{(i)} x_n^{(i)} - y^{(i)} x_n^{(i)}) = 0$$

Normal Equation

With a bit more manipulation, we get the normal equations *i.e.*

$$\Sigma_{i=1}^{m}(\theta_0 + \theta_1 x_1^{(i)} + \ldots + \theta_n x_n^{(i)} - y^{(i)}) = 0$$

$$\iff \theta_0 m + \theta_1 \Sigma_{i=1}^{m} x_1^{(i)} + \ldots + \theta_n \Sigma_{i=1}^{m} x_n^{(i)} = \Sigma_{i=1}^{m} y^{(i)}$$

$$\Sigma_{i=1}^{m}(\theta_0 x_1^{(i)} + \theta_1 x_1^{(i)} x_1^{(i)} + \ldots + \theta_n x_n^{(i)} x_1^{(i)} - y^{(i)} x_1^{(i)}) = 0$$

$$\iff \theta_0 \Sigma_{i=1}^{m} x_1^{(i)} + \theta_1 \Sigma_{i=1}^{m} x_1^{(i)} x_1^{(i)} + \ldots + \theta_n \Sigma_{i=1}^{m} x_n^{(i)} x_1^{(i)} = \Sigma_{i=1}^{m} y^{(i)} x_1^{(i)}$$

$$\vdots$$

$$\Sigma_{i=1}^{m}(\theta_0 x_n^{(i)} + \theta_1 x_1^{(i)} x_n^{(i)} + \ldots + \theta_n x_n^{(i)} x_n^{(i)} - y^{(i)} x_n^{(i)}) = 0$$

$$\iff \theta_0 \Sigma_{i=1}^{m} x_n^{(i)} + \theta_1 \Sigma_{i=1}^{m} x_1^{(i)} x_n^{(i)} + \ldots + \theta_n \Sigma_{i=1}^{m} x_n^{(i)} x_n^{(i)} = \Sigma_{i=1}^{m} y^{(i)} x_n^{(i)}$$

Normal Equation

Remember that $x_j^{(i)}$ and $y^{(i)}$ are all constant (they are the training datasets).

$$\Sigma_{i=1}^m(\theta_0 + \theta_1 x_1^{(i)} + \ldots + \theta_n x_n^{(i)} - y^{(i)}) = 0$$

$$\iff \theta_0 m + \theta_1 \Sigma_{i=1}^m x_1^{(i)} + \ldots + \theta_n \Sigma_{i=1}^m x_n^{(i)} = \Sigma_{i=1}^m y^{(i)}$$

$$\Sigma_{i=1}^m(\theta_0 x_1^{(i)} + \theta_1 x_1^{(i)} x_1^{(i)} + \ldots + \theta_n x_n^{(i)} x_1^{(i)} - y^{(i)} x_1^{(i)}) = 0$$

$$\iff \theta_0 \Sigma_{i=1}^m x_1^{(i)} + \theta_1 \Sigma_{i=1}^m x_1^{(i)} x_1^{(i)} + \ldots + \theta_n \Sigma_{i=1}^m x_n^{(i)} x_1^{(i)} = \Sigma_{i=1}^m y^{(i)} x_1^{(i)}$$

$$\vdots$$

$$\Sigma_{i=1}^m(\theta_0 x_n^{(i)} + \theta_1 x_1^{(i)} x_n^{(i)} + \ldots + \theta_n x_n^{(i)} x_n^{(i)} - y^{(i)} x_n^{(i)}) = 0$$

$$\iff \theta_0 \Sigma_{i=1}^m x_n^{(i)} + \theta_1 \Sigma_{i=1}^m x_1^{(i)} x_n^{(i)} + \ldots + \theta_n \Sigma_{i=1}^m x_n^{(i)} x_n^{(i)} = \Sigma_{i=1}^m y^{(i)} x_n^{(i)}$$

Normal Equation

Let's try to write our equations in matrix and vector forms as follows:

$$\underset{\text{(Design Matrix)}}{\boldsymbol{X}} = \begin{bmatrix} 1 & x_1^{(1)} & \dots & x_n^{(1)} \\ 1 & x_1^{(2)} & \dots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & \dots & x_n^{(m)} \end{bmatrix} \quad \text{and} \quad \boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

You should be able to verify that our system of linear equations become:

$$\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\theta} = \boldsymbol{X}^T\boldsymbol{y}$$

Assuming that $\boldsymbol{X}^T\boldsymbol{X}$ is invertible, then: $\boldsymbol{\theta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$

Question

- Suppose you have the training in the table below:

age (x_1)	height in cm (x_2)	weight in kg (y)
4	89	16
9	124	28
5	103	20

You would like to predict a child's weight as a function of his age and height with the model: $\textbf{weight} = \theta_0 + \theta_1 \textbf{age} + \theta_2 \textbf{height}$
What are X and y?

Normal Equation

Assuming that $\boldsymbol{X}^T\boldsymbol{X}$ is invertible, then: $\boldsymbol{\theta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$

Question: when a matrix is non-invertible (*aka.* singular/degenerate) ?

Normal Equation

Assuming that $\boldsymbol{X}^T\boldsymbol{X}$ is invertible, then: $\boldsymbol{\theta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$

Question: when $\boldsymbol{X}^T\boldsymbol{X}$ is non-invertible (*aka.* singular/degenerate) ?

This case happens when we have:

- redundant features (*i.e.* some columns are linearly dependent) *e.g.*
  - x_1 = size in feet^2
  - x_2 = size in m^2

1 **m** ≈ 3.28 **feet** $\quad x_1 = (3.28)^2 x_2$

- too many features ($m \leq n$)
  - delete some features, or
  - use regularization (we'll talk about this later)

Gradient Descent *vs.* Normal Equation

***m* training examples, *n* features**

<u>Gradient Descent</u>

- Need to choose α
- Needs many iterations

- Works well even when n is large $\mathcal{O}(kn^2)$

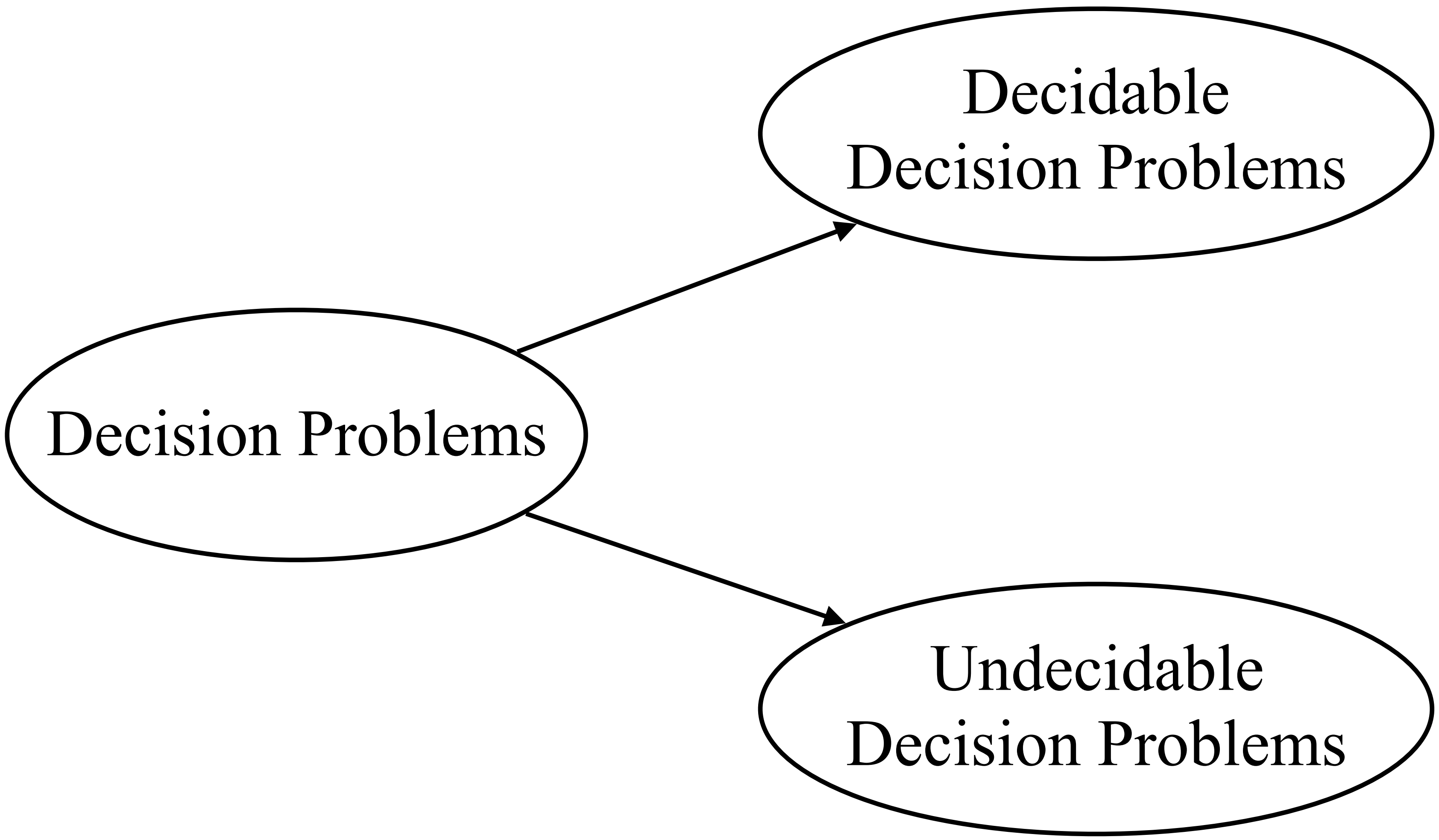
<u>Normal Equation</u>

- No need to choose α
- Don't need to iterate

- Need to compute $(X^T X)^{-1}$
- Slow if n is very large $\mathcal{O}(n^3)$

(Large if $n \geq 10^4$)

Computational Complexity (Review for non-CS students)



'computational complexity'

$\mathcal{O}(\,\cdot\,)\quad \Omega(\,\cdot\,)\quad \Theta(\,\cdot\,)$

'Turing degree'

Question

- Suppose you have a dataset with $m = 1{,}000{,}0000$ examples and $n = 200{,}000$ features for each example. You want to use multivariate linear regression to fit the parameters θ to our data.
Should you prefer gradient descent or the normal equation?

(i) The normal equation, since gradient descent might be unable to find the optimal θ
(ii) Gradient descent, since $(X^T X)^{-1}$ will be very slow to compute in the normal equation
(iii) The normal equation, since it provides an efficient way to directly find the solution.
(iv) Gradient descent, since it will always converge to the optimal θ

Last Question

Why least square? Why not some other cost function?

The least square method comes naturally from a probabilistic interpretation of the problem.

Let's suppose that $(\boldsymbol{x}^{(i)}, y^{(i)})$ were generated according to:

$$y^{(i)} = \boldsymbol{\theta}^T \boldsymbol{x}^{(i)} + \epsilon^{(i)},$$

where $\epsilon^{(i)}$ is an error term representing noise and whatever effects our linear model doesn't capture.

Last Question

We might assume that the noise term $\epsilon^{(i)}$ were sampled independently and identically distributed from some distribution $p(\epsilon^{(i)})$.

Is independently and identically distributed a reasonable assumption?

In many applications, a natural choice of $p(\epsilon^{(i)})$ is a Gaussian (*aka.* normal distribution), which we write:

$$\epsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$$

Last Question

The univariate zero-mean Gaussian distribution is written:

$$p(\epsilon^{(i)}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{(-\frac{(\epsilon^{(i)})^2}{2\sigma^2})}$$

Now, we would like to know $p(y^{(i)} | \boldsymbol{x}^{(i)}; \boldsymbol{\theta})$.

Suppose $y^{(i)} \sim \mathcal{N}(\theta^T \boldsymbol{x}^{(i)}, \sigma^2)$, then:

$$p(y^{(i)} | \boldsymbol{x}^{(i)}; \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{(-\frac{(y^{(i)} - \boldsymbol{\theta}^T \boldsymbol{x}^{(i)})^2}{2\sigma^2})}$$

Last Question

Now, consider the distribution of the vector of targets $\boldsymbol{y}$ given the design matrix $\boldsymbol{X}$. We write this:

$$p(\boldsymbol{y} \mid \boldsymbol{X}; \boldsymbol{\theta})$$

Since the elements of $\boldsymbol{y}$ are independent, we can write:

$$p(\boldsymbol{y} \mid \boldsymbol{X}; \boldsymbol{\theta}) = \Pi_{i=1}^{m} p(y^{(i)} \mid \boldsymbol{x}^{(i)}; \boldsymbol{\theta})$$

The Gaussians are centered at our predictions of the $y^{(i)}$'s, so the conditional probability $p(\boldsymbol{y} \mid \boldsymbol{X}; \boldsymbol{\theta}) = \Pi_{i=1}^{m} p(y^{(i)} \mid \boldsymbol{x}^{(i)}; \boldsymbol{\theta})$ will be maximized when our predictions of the $y^{(i)}$'s are perfect.

This means choosing θ to maximize $p(\boldsymbol{y} \mid \boldsymbol{X}; \boldsymbol{\theta})$ would be good !

Last Question

Now, we would like to choose θ that maximize $p(\boldsymbol{y} \mid \boldsymbol{X}; \boldsymbol{\theta})$.

First, let's write this distribution as a function of the parameter vector θ:

$$L(\boldsymbol{\theta}) = L(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{y}) = p(\boldsymbol{y} \mid \boldsymbol{X}; \boldsymbol{\theta})$$

$L(\boldsymbol{\theta})$ is called the likelihood function.

The maximum likelihood principle states that we should choose the θ that makes the likelihood $L(\boldsymbol{\theta})$ as large as possible:

$$\boldsymbol{\theta}^* = \max_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$$

Last Question

OK ! Let's try to maximize $L(\boldsymbol{\theta})$.

$$\because L(\boldsymbol{\theta}) = \Pi_{i=1}^{m} p(y^{(i)} | \boldsymbol{x}^{(i)}; \boldsymbol{\theta})$$

$$= \Pi_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} \mathbf{exp}\left(-\frac{(y^{(i)} - \boldsymbol{\theta}^T \boldsymbol{x}^{(i)})^2}{2\sigma^2}\right)$$

This looks so difficult to maximize directly (try to evaluate $\frac{\partial L}{\partial \theta}$) !

However, we can equivalently maximize any function that is strictly increasing in $L(\boldsymbol{\theta})$.

Last Question

It will be more convenient to maximize the log likelihood $l(\boldsymbol{\theta})$:

$$
\begin{aligned}
l(\boldsymbol{\theta}) &= \log L(\boldsymbol{\theta}) \\
&= \log \Pi_{i=1}^{m} p(y^{(i)} \,|\, \boldsymbol{x}^{(i)}; \boldsymbol{\theta}) \\
&= \Sigma_{i=1}^{m} \log \frac{1}{\sqrt{2\pi\sigma^2}} \textbf{exp}(-\frac{(y^{(i)} - \boldsymbol{\theta}^T \boldsymbol{x}^{(i)})^2}{2\sigma^2}) \\
&= m \log \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2} \Sigma_{i=1}^{m} (y^{(i)} - \boldsymbol{\theta}^T \boldsymbol{x}^{(i)})^2
\end{aligned}
$$

Last Question

Clearly, any $\boldsymbol{\theta}$ maximizing

$$m \log \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}\Sigma_{i=1}^{m}(y^{(i)} - \boldsymbol{\theta}^T\boldsymbol{x}^{(i)})^2$$

will also maximize

$$-\frac{1}{2m}\Sigma_{i=1}^{m}(y^{(i)} - \boldsymbol{\theta}^T\boldsymbol{x}^{(i)})^2$$

Amazingly, maximizing $L(\boldsymbol{\theta})$ or $l(\boldsymbol{\theta})$ gives us the same $\boldsymbol{\theta}$ we get by minimizing

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

Last Question

We now see that, for linear regression, least squares and maximum likelihood are equivalent.

The equivalence comes from the assumption of independently and identically distributed Gaussian errors in our samples $y^{(i)}$.

For other problems, we may not be able to formulate a least squares cost function, but we will be able to use the more general principle of maximum likelihood.

Summary

Now, we have two choices for minimizing $J(\theta)$ as follows:

1. Solve the normal equations

2. Use the iterative methods:
   a) Batch gradient descent
   b) Stochastic gradient descent
   c) Mini-batch gradient descent
   (compromise between GD and SGD)

Now, you are ready for Assignment 2 !