3. Linear Regression ที่มีหลายตัวแปร
(Multivariate Linear Regression)

3.1 Multiple Features

Krittameth Teachasrisaksakul

Feature เดียว (1 feature) [Recap]

Linear regression: version เดิม (1 ตัวแปร)

ขนาดพื้นที่บ้าน Size (feet²) x	ราคาบ้าน Price ($1000) y
2104	460
1416	232
1534	315
852	178
…	…

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

หลาย Feature (variable / ตัวแปร)

ขนาดพื้นที่บ้าน (ตร. ฟุต)	จำนวนห้องนอน	จำนวนชั้น	อายุบ้าน (ปี)	ราคาบ้าน
x_1	x_2	x_3	x_4	y
Size (feet²)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

สัญลักษณ์

- n : จำนวน features (คุณลักษณะ)
- $x^{(i)}$: input (features) ของ training example ตัวที่ i
- $x^{(i)}_j$: ค่าของ feature j ใน training example ตัวที่ i

หลาย Feature (variable / ตัวแปร)

ขนาดพื้นที่บ้าน (ตร.ฟุต) x_1 Size (feet^2)	จำนวนห้องนอน x_2 Number of bedrooms	จำนวนชั้น x_3 Number of floors	อายุบ้าน (ปี) x_4 Age of home (years)	ราคาบ้าน y Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

สัญลักษณ์

- n : จำนวน features (คุณลักษณะ)
- $x^{(i)}$: input (features) ของ training example ตัวที่ i
- $x^{(i)}_j$: ค่าของ feature j ใน training example ตัวที่ i

$n = 4$

$x^{(2)}_3 = 2$

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

คำถาม

ขนาดพื้นที่บ้าน (ตร.ฟุต) Size ($feet^2$)	จำนวนห้องนอน Number of	จำนวนชั้น Number of	อายุบ้าน (ปี) Age of home	ราคาบ้าน Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178

จากชุดข้อมูล training set ด้านบน ค่าของ $x^{(4)}_1$ เท่ากับเท่าไร

(i) ขนาดบ้าน (ตารางฟุต) ของบ้านหลังที่ 1 ในชุดข้อมูล training set

(ii) อายุบ้าน (ปี) ของบ้านหลังที่ 1 ในชุดข้อมูล training set

(iii) ขนาดบ้าน (ตารางฟุต) ของบ้านหลังที่ 4 ในชุดข้อมูล training set

(iv) อายุบ้าน (ปี) ของบ้านหลังที่ 4 ในชุดข้อมูล training set

คำถาม

ขนาดพื้นที่บ้าน (ตร.ฟุต) Size (feet^2)	จำนวนห้องนอน Number of	จำนวนชั้น Number of	อายุบ้าน (ปี) Age of home	ราคาบ้าน Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178

จากชุดข้อมูล training set ด้านบน ค่าของ $x^{(4)}_1$ เท่ากับเท่าไร

(i) ขนาดบ้าน (ตารางฟุต) ของบ้านหลังที่ 1 ในชุดข้อมูล training set

(ii) อายุบ้าน (ปี) ของบ้านหลังที่ 1 ในชุดข้อมูล training set

(iii) ขนาดบ้าน (ตารางฟุต) ของบ้านหลังที่ 4 ในชุดข้อมูล training set

(iv) อายุบ้าน (ปี) ของบ้านหลังที่ 4 ในชุดข้อมูล training set

Hypothesis function

Hypothesis function เดิม (1 ตัวแปร)

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

Hypothesis function

Hypothesis function เดิม (1 ตัวแปร)

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

ในกรณีมีหลายตัวแปร

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

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

Hypothesis function (หลาย feature)

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

เพื่อความสะดวกในการเขียนสัญลักษณ์ นิยามให้ $x_0 = 1$

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

Hypothesis function (หลาย feature)

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

ทำให้ θ กับ x มีจำนวนสมาชิกเท่ากัน คือ $n+1$ ตัว
เพื่อทำให้สามารถทำ matrix operations ด้วย θ กับ x

เพื่อความสะดวกในการเขียนสัญลักษณ์ นิยามให้ $x_0 = 1$

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

สำหรับ $i \in 1, \ldots, m$

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

[เขียนแทน feature ด้วย column vector $\boldsymbol{X}$]

Hypothesis function (หลาย feature)

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

แล้วจะได้

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

[เขียนแทน parameter ด้วย column vector $\boldsymbol{\theta}$]

Hypothesis function (หลาย feature)

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

แล้วจะได้

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

[เขียนแทน parameter ด้วย column vector $\boldsymbol{\theta}$]

ดังนั้น เขียน hypothesis function ใหม่ ได้เป็น

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

จัดรูป hypothesis function เป็น vector (vectorization) สำหรับ 1 training example

Hypothesis function (หลาย feature)

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

แล้วจะได้

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

[เขียนแทน parameter ด้วย column vector $\boldsymbol{\theta}$]

ดังนั้น เขียน hypothesis function ใหม่ ได้เป็น

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

Hypothesis function (หลาย feature)

เราจะได้ Linear Regression ที่มีหลายตัวแปร (Multivariate Linear Regression)

$$
\begin{aligned}
h_\theta(x) &= \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n \\
&= \begin{bmatrix} \theta_0 & \theta_1 & \ldots & \theta_n \end{bmatrix} \times \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \\
&= \boldsymbol{\theta}^T \boldsymbol{x}
\end{aligned}
$$

3. Linear Regression ที่มีหลายตัวแปร
(Multivariate Linear Regression)

3.2 Gradient Descent
สำหรับ Multiple Features

Krittameth Teachasrisaksakul

เรามี ...

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

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

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

ทำให้เป็น vector (vectorize)

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

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

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

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

คำถาม

เมื่อมี features n ตัว เราจะนิยาม cost function เป็น

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

ในกรณี linear regression สมการใดเป็นนิยามที่ถูกต้อง และเทียบเท่ากันกับสมการด้านบนของ $J(\theta)$?

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

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

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

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

คำถาม

เมื่อมี features n ตัว เราจะนิยาม cost function เป็น

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

ในกรณี linear regression สมการใดเป็นนิยามที่ถูกต้อง และเทียบเท่ากันกับสมการด้านบนของ $J(\theta)$?

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

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

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

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

ทำให้ Gradient Descent เป็น vector

Gradient Descent:

$$
\begin{aligned}
&\text{Repeat } \{ \\
&\qquad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \ldots, \theta_n) \\
&\}
\end{aligned}
$$

(ปรับค่า parameter θ_j ทุกตัว พร้อมๆกัน สำหรับ $j = 0, 1, 2, \ldots, n$)
(simultaneously update for every $j = 0, 1, 2, \ldots, n$)

ทำให้ Gradient Descent เป็น vector

Gradient Descent:

$$\text{Repeat } \{$$

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

$$\}$$

(ปรับค่า parameter θ_j ทุกตัว พร้อมๆกัน สำหรับ $j = 0, 1, 2, \ldots, n$)
(simultaneously update for every $j = 0, 1, 2, \ldots, n$)

Feature เดียว [Recap / ทบทวน]

Feature เดียว ($n = 1$)

Repeat {

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

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

}

(ปรับค่า parameter θ_0, θ_1 พร้อมๆกัน)

Feature เดียว [Recap / ทบทวน]

Feature เดียว ($n = 1$)

Repeat {

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

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

}

(ปรับค่า parameter θ_0, θ_1 พร้อมๆกัน)

Feature เดียว vs. หลาย feature (Single vs. multiple features)

Feature เดียว ($n = 1$)

Repeat {

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

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

} (ปรับค่า parameter θ_0, θ_1 พร้อมๆกัน)

Feature เดียว vs. หลาย feature (Single vs. multiple features)

Feature เดียว ($n = 1$)

Repeat {

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

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

}

(ปรับค่า parameter θ_0, θ_1 พร้อมๆกัน)

หลาย Feature ($n \geq 1$)

Repeat {

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

}

(ปรับค่า parameter θ_j ทุกตัว พร้อมๆกัน สำหรับ $j = 0, 1, 2, \ldots, n$)

ใช้สมการ gradient descent รูปแบบเดิม
แค่ทำมันซ้ำ สำหรับ n features

ความหมายของสมการแบบหลาย feature (Multiple features)

หลาย Feature ($n \geq 1$)

Repeat {

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

}

(ปรับค่า parameter θ_j ทุกตัว พร้อมๆกัน สำหรับ $j = 0, 1, 2, \ldots, n$)

ก็คือ (เขียนสมการเดิม แต่เปลี่ยนค่า j)

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

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

$$\theta_2 := \theta_2 - \alpha \frac{1}{m} \Sigma_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_2^{(i)}$$

3. Linear Regression ที่มีหลายตัวแปร
(Multivariate Linear Regression)

3.3 Gradient Descent ในทางปฏิบัติ (1) - Feature Scaling

Krittameth Teachasrisaksakul

Feature Scaling: ความเข้าใจพื้นฐาน

แนวคิด: ทำให้แน่ใจว่า feature หลายๆตัว อยู่ใน scale (มาตราส่วน) เดียวกัน

เช่น X_1 = ขนาดพื้นที่บ้าน (0 - 2,000 ตร.ฟุต)

X_2 = จำนวนห้องนอน (1 - 5)

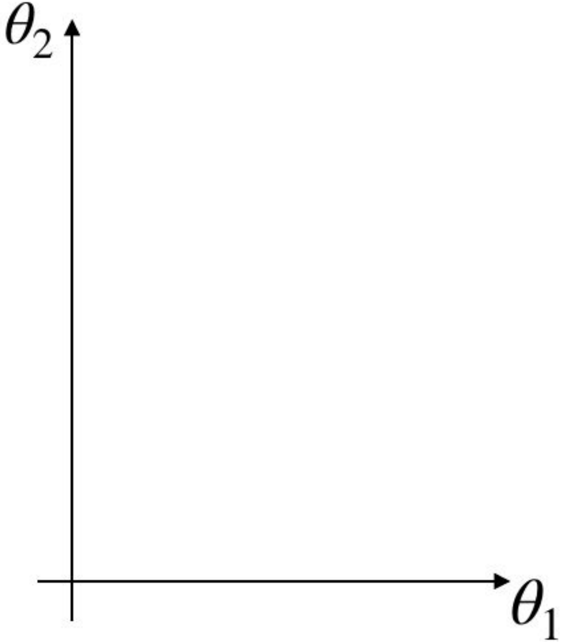


Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 2.2]

Feature Scaling: ความเข้าใจพื้นฐาน

แนวคิด: ทำให้แน่ใจว่า feature หลายๆตัว อยู่ใน scale (มาตราส่วน) เดียวกัน

เช่น X_1 = ขนาดพื้นที่บ้าน (0 - 2,000 ตร.ฟุต)

X_2 = จำนวนห้องนอน (1 - 5)

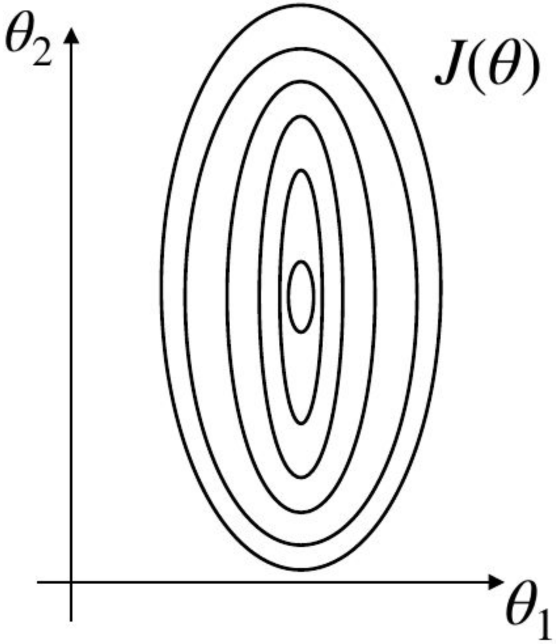


Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 2.2]

Feature Scaling: ความเข้าใจพื้นฐาน

แนวคิด: ทำให้แน่ใจว่า feature หลายๆตัว อยู่ใน scale (มาตราส่วน) เดียวกัน

เช่น x_1 = ขนาดพื้นที่บ้าน (0 - 2,000 ตร.ฟุต)

x_2 = จำนวนห้องนอน (1 - 5)

ถ้า **feature** ไม่สม่ำเสมอมากๆ / มี **range ต่างกันมากๆ** ⟶

- **gradient descent อาจ converge ช้า**
- เพราะ θ จะลดลงอย่างช้าๆ และจะแกว่งแบบไม่มีประสิทธิภาพลงสู่จุด optimum (ค่าที่เหมาะสม)

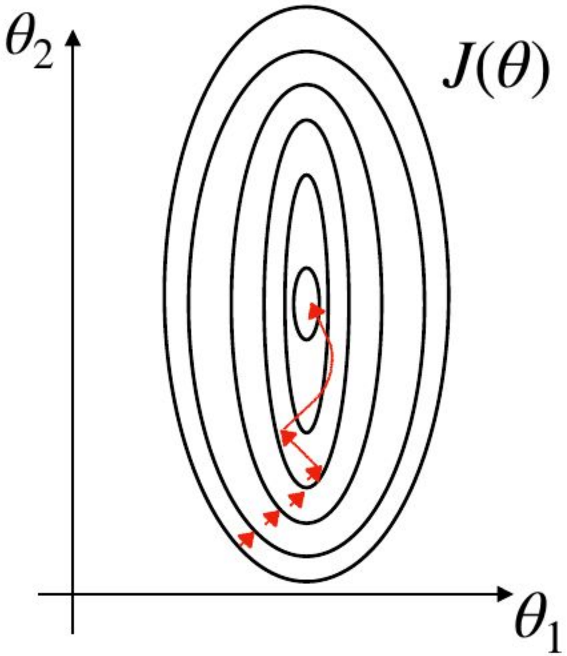


Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 2.2]

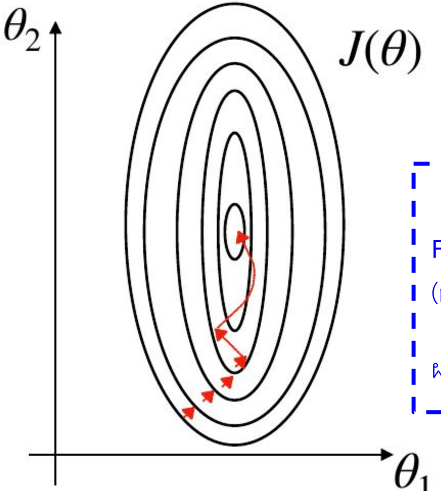
Feature Scaling: ความเข้าใจพื้นฐาน

Figure Source: Teeradaj Racharak; AI Practical Development Bootcamp [L 2.2]

แนวคิด: ทำให้แน่ใจว่า feature หลายๆตัว อยู่ใน scale (มาตราส่วน) เดียวกัน

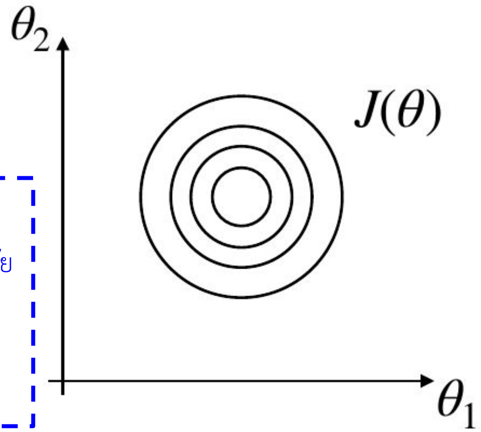
เช่น X_1 = ขนาดพื้นที่บ้าน (0 - 2,000 ตร.ฟุต)

X_2 = จำนวนห้องนอน (1 - 5)



X_1 = ขนาดพื้นที่ (ตร.ฟุต) / 2000

X_2 = #ห้องนอน / 5



Feature scaling คือ หารค่า input ด้วย ค่าพิสัย (range = ค่า max – ค่า min) ของ ตัวแปร input

ผลที่ได้ คือ range ใหม่ = 1

Feature Scaling: ความเข้าใจพื้นฐาน

Figure Source: Teeradaj Racharak; AI Practical Development Bootcamp [L 2.2]

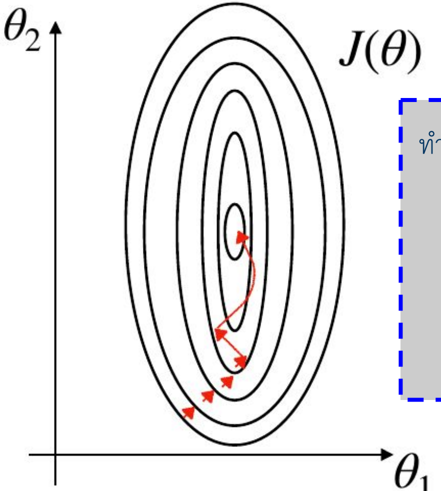
แนวคิด: ทำให้แน่ใจว่า feature หลายๆตัว อยู่ใน scale (มาตราส่วน) เดียวกัน

เช่น X_1 = ขนาดพื้นที่บ้าน (0 - 2,000 ตร.ฟุต)

X_2 = จำนวนห้องนอน (1 - 5)

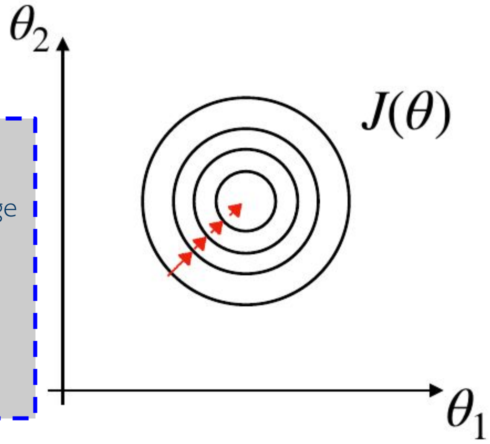
X_1 = ขนาดพื้นที่ (ตร.ฟุต) / 2000 $\quad 0 \le x_1 \le 1$

X_2 = #ห้องนอน / 5 $\quad 0 \le x_2 \le 1$



ทำให้ gradient descent converge เร็วขึ้นได้

- โดย ทำให้ค่า input แต่ละตัวอยู่ใน range เดียวกัน
- เพราะ θ จะลดลงอย่างเร็วใน range ที่เล็ก



Feature Scaling: ในทางปฏิบัติ

ทำให้ทุก feature มีค่าอยู่ในช่วง $-1 \leq x_i \leq 1$ โดยประมาณ

จริงๆแล้ว ไม่จำเป็นนักที่จะต้องเป็น ช่วง [-1, 1] เป๊ะๆ เช่น

$0 \leq x_1 \leq 3$ (ได้)

$-2 \leq x_2 \leq 0.5$ (ได้)

$-100 \leq x_3 \leq 100$ (มากไป)

$-0.0001 \leq x_4 \leq 0.0001$ (น้อยไป)

Feature Scaling: ในทางปฏิบัติ

Mean Normalization

ให้นิยามใหม่ x_i เป็น $\boxed{(x_i - \mu_i) / S_i}$

เมื่อ

- $\mu_i =$ ค่าเฉลี่ยของค่าทุกค่าของ feature ที่ i
- $S_i =$ range (พิสัย) ของค่า input ($= \max - \min$) หรือ standard deviation

Mean normalization คือ ลบตัวแปร input ด้วย ค่าเฉลี่ย (average value) ของมัน

ผลที่ได้ คือ ค่าเฉลี่ยใหม่ของตัวแปร input = 0

เพื่อทำให้ features มีค่าเฉลี่ยประมาณ 0

(ไม่ใช้ $x_0 = 1$)

เช่น $x_1 = (\text{size} - 1000) / 2000$ (เมื่อ $\mu_1 = 1000$)

$x_2 = (\#\text{bedrooms} - 2) / 4$ (เมื่อ $\mu_1 = 2$)

คำถาม

สมมติเราใช้ learning algorithm เพื่อประมาณค่าราคาบ้านในเมือง เราอยากให้หนึ่งใน features x_i เป็นอายุของบ้าน ในชุดข้อมูล training set บ้านทุกบ้านมีอายุระหว่าง 30 และ 50 ปี โดยมีอายุเฉลี่ยเป็น 38 ปี ค่าใดต่อไปนี้ที่เราจะใช้เป็น features ถ้าเราใช้วิธี feature scaling และ mean normalization

(i) $x_i =$ อายุบ้าน

(ii) $x_i =$ อายุบ้าน $/ 50$

(iii) $x_i = ($อายุบ้าน $- 38) / 50$

(iv) $x_i = ($อายุบ้าน $- 38) / 20$

คำถาม

สมมติเราใช้ learning algorithm เพื่อประมาณค่าราคาบ้านในเมือง เราอยากให้หนึ่งใน features x_i เป็นอายุของบ้าน ในชุดข้อมูล training set บ้านทุกบ้านมีอายุระหว่าง 30 และ 50 ปี โดยมีอายุเฉลี่ยเป็น 38 ปี ค่าใดต่อไปนี้ที่เราจะใช้เป็น features ถ้าเราใช้วิธี feature scaling และ mean normalization

(i) $x_i =$ อายุบ้าน

(ii) $x_i =$ อายุบ้าน $/ 50$

(iii) $x_i =$ (อายุบ้าน $- 38) / 50$

(iv) $x_i =$ (อายุบ้าน $- 38) / 20$

3. Linear Regression ที่มีหลายตัวแปร (Multivariate Linear Regression)

3.4 Gradient Descent ในทางปฏิบัติ (2) - Learning Rate

Krittameth Teachasrisaksakul

Gradient Descent (Recap)

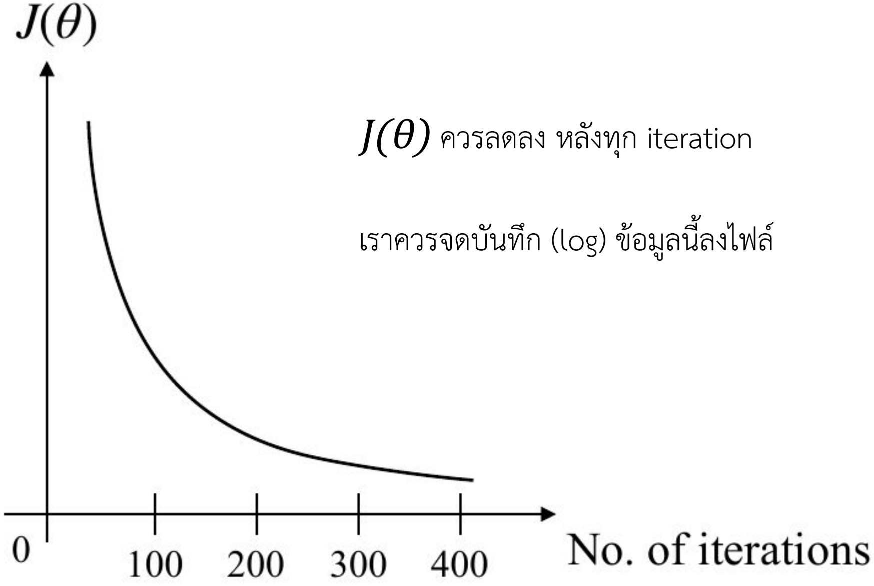
Gradient Descent Update Rule (กฎการปรับค่า parameter ของ Gradient Descent)

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

- ‘Debugging’ คือ ทำยังไง จึงจะมั่นใจว่า gradient descent ทำงานอย่างถูกต้อง
- เลือก learning rate อย่างไร

Gradient descent ทำงานอย่างถูกต้อง หรือไม่?

เคล็ดลับเพื่อทำให้แน่ใจว่า gradient descent ทำงานอย่างถูกต้อง



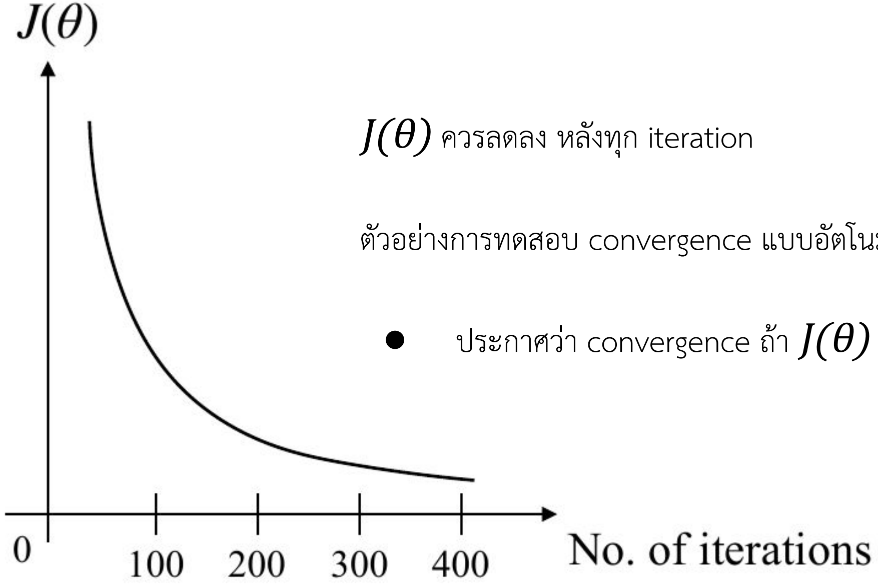
$J(\theta)$ ควรลดลง หลังทุก iteration

เราควรจดบันทึก (log) ข้อมูลนี้ลงไฟล์

Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 2.2]

Gradient descent ทำงานอย่างถูกต้อง หรือไม่?

เคล็ดลับเพื่อทำให้แน่ใจว่า gradient descent ทำงานอย่างถูกต้อง



$J(\theta)$ ควรลดลง หลังทุก iteration

ตัวอย่างการทดสอบ convergence แบบอัตโนมัติ

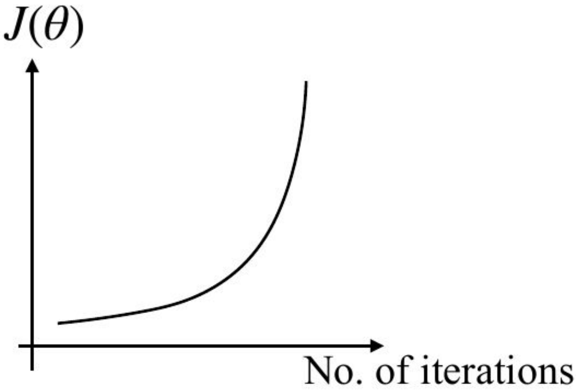
- ประกาศว่า convergence ถ้า $J(\theta)$ ลดลงน้อยกว่า 10^{-3} ใน 1 iteration

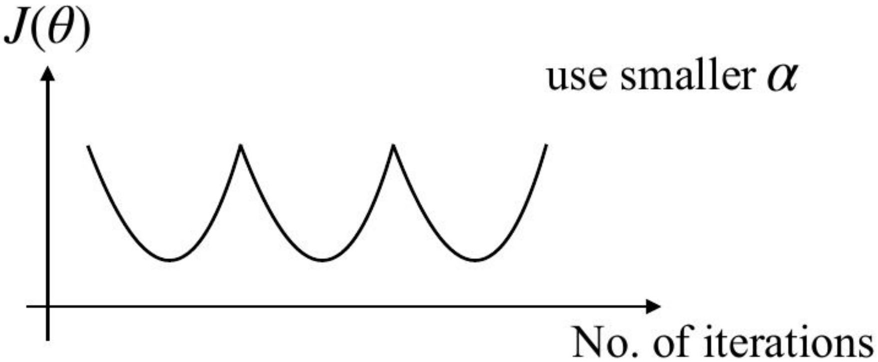
ε

(threshold)

Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 2.2]

Gradient descent ทำงานอย่างถูกต้อง หรือไม่?





- ถ้า α ที่น้อยเพียงพอ → $J(\theta)$ ควรลดลง ทุก iteration
- แต่ถ้า α น้อยเกินไป → gradient descent อาจ converge ช้า

Figure Source: *Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 2.2]*

คำถาม

สมมติเพื่อน run gradient descent 3 ครั้ง
ด้วยค่า $\alpha = 0.01, \alpha = 0.1, \alpha = 1$
และได้ plot 3 อัน (A, B, C) จงบอกค่า α ของแต่ละ plot

(i) A คือ $\alpha = 0.01$, B คือ $\alpha = 0.1$, C คือ $\alpha = 1$

(ii) A คือ $\alpha = 0.1$, B คือ $\alpha = 0.01$, C คือ $\alpha = 1$

(iii) A คือ $\alpha = 1$, B คือ $\alpha = 0.01$, C คือ $\alpha = 0.1$

(iv) A คือ $\alpha = 1$, B คือ $\alpha = 0.1$, C คือ $\alpha = 0.01$

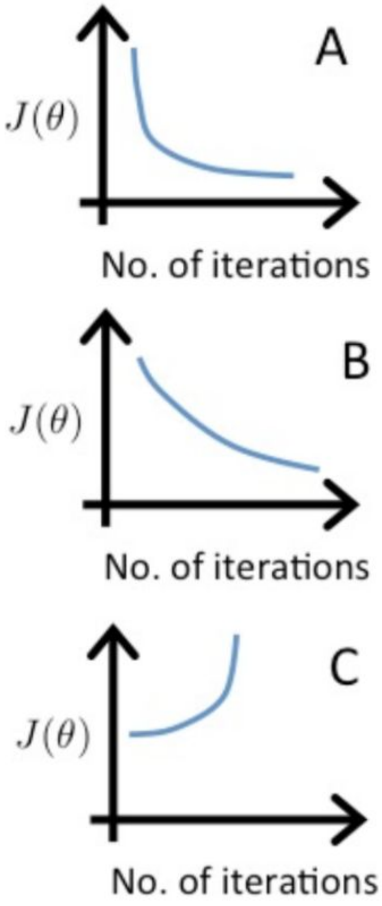


Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 2.2]

คำถาม

สมมติเพื่อน run gradient descent 3 ครั้ง

ด้วยค่า $\alpha = 0.01, \alpha = 0.1, \alpha = 1$

และได้ plot 3 อัน (A, B, C) จงบอกค่า α ของแต่ละ plot

(i) A คือ $\alpha = 0.01$, B คือ $\alpha = 0.1$, C คือ $\alpha = 1$

(ii) A คือ $\alpha = 0.1$, B คือ $\alpha = 0.01$, C คือ $\alpha = 1$

(iii) A คือ $\alpha = 1$, B คือ $\alpha = 0.01$, C คือ $\alpha = 0.1$

(iv) A คือ $\alpha = 1$, B คือ $\alpha = 0.1$, C คือ $\alpha = 0.01$

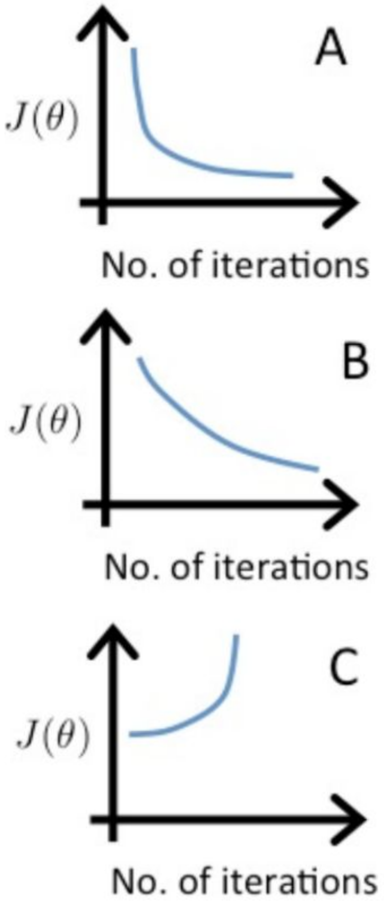


Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 2.2]

สรุป

ถ้า α น้อยเกินไป $\longrightarrow$ convergence จะเกิดช้า

ถ้า α มากเกินไป $\longrightarrow J(\theta)$ อาจไม่ลดลง ทุก iteration และอาจไม่ converge

เพื่อเลือกค่า α : ลองค่า α ต่อไปนี้

..., 0.001, ,0.01, ,0.1, ,1, ...

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

3. Linear Regression ที่มีหลายตัวแปร
(Multivariate Linear Regression)

3.5 Features & Polynomial Regression

Krittameth Teachasrisaksakul

การเลือก features : ความเข้าใจพื้นฐาน

การทำนายราคาบ้าน

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

- frontage = ด้านหน้า
- depth = ความลึก

การเลือก features : ความเข้าใจพื้นฐาน

การทำนายราคาบ้าน

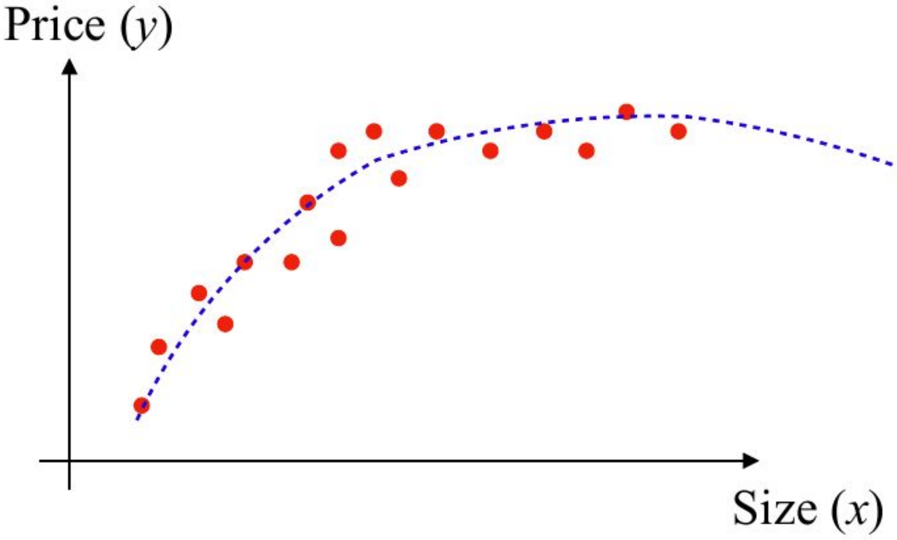
$$h_\theta(x) = \theta_0 + \theta_1 \times \underbrace{\textbf{frontage}}_{x_1} + \theta_2 \times \underbrace{\textbf{depth}}_{x_2}$$

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

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

- frontage = ด้านหน้า
- depth = ความลึก

Polynomial Regression (Regression ที่ใช้ฟังก์ชั่นพหุนาม)



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

Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 2.2]

Polynomial Regression

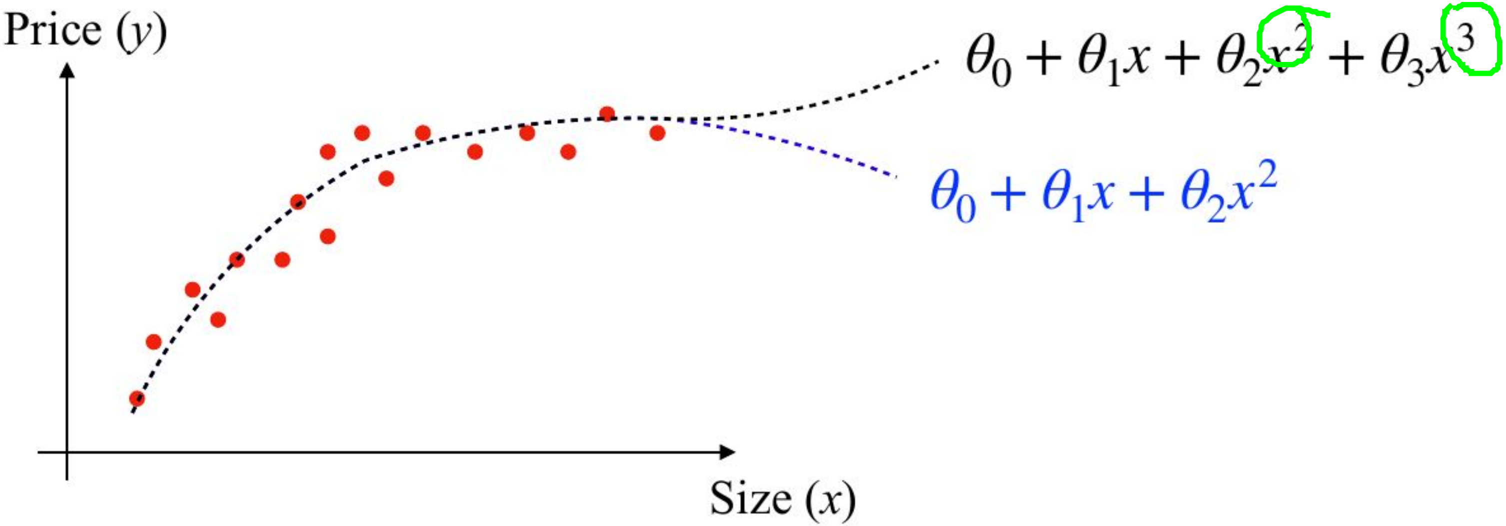
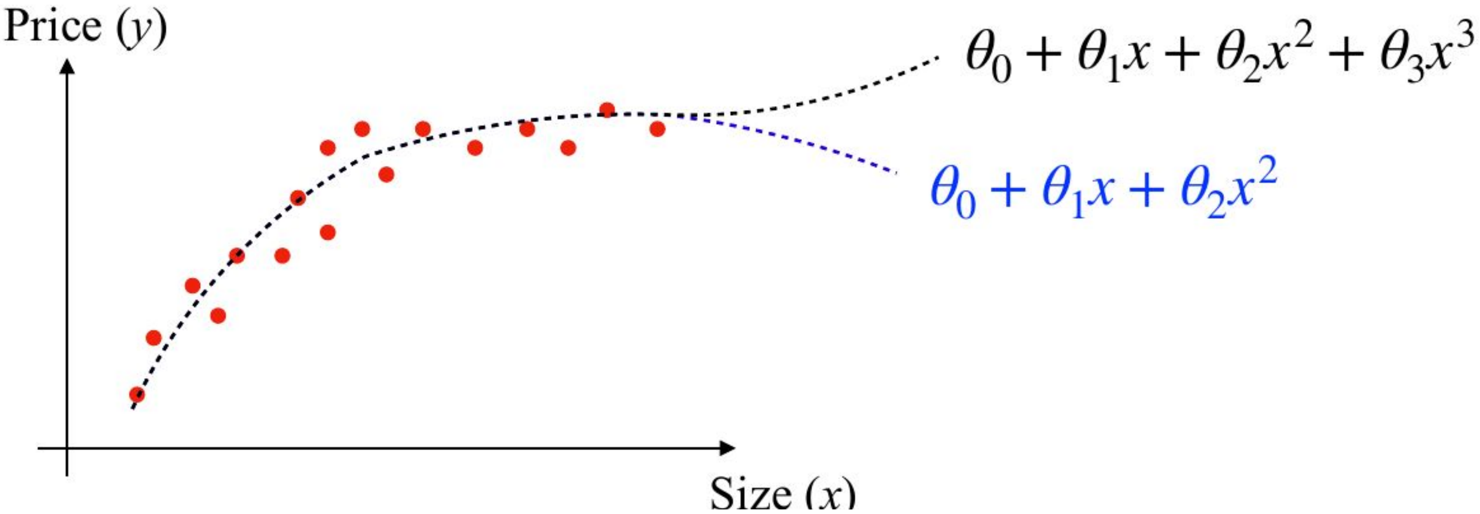


Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 2.2]

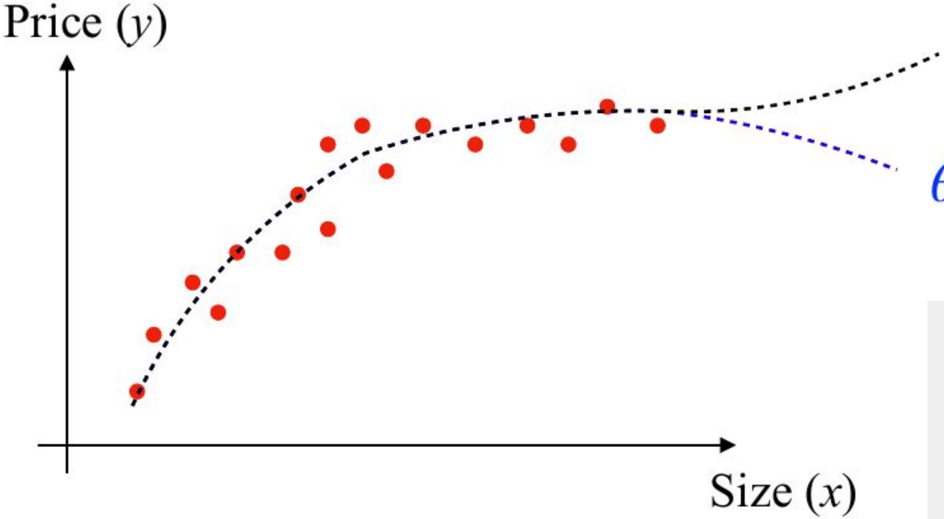
Polynomial Regression



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

Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 2.2]

Polynomial Regression



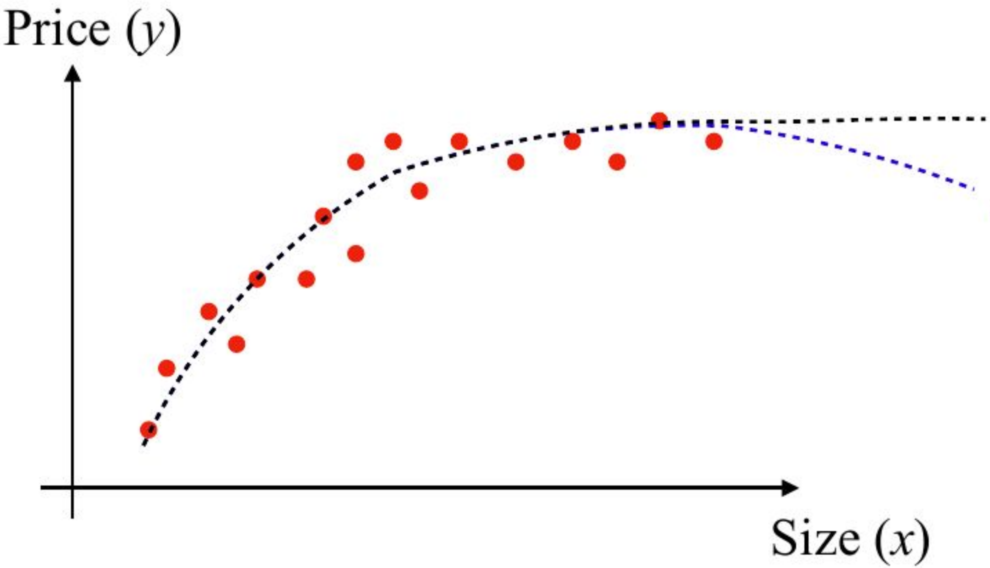
$$\therefore h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$
$$= \theta_0 + \theta_1(\textbf{size}) + \theta_2(\textbf{size}^2) + \theta_3(\textbf{size}^3)$$

Feature scaling สำคัญ

- size : 1 – 1,000
- size^2 : 1 – 1,000,000
- size^3 : 1 – 1,000,000,000

Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 2.2]

Hypothesis function อื่น



1/2

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

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

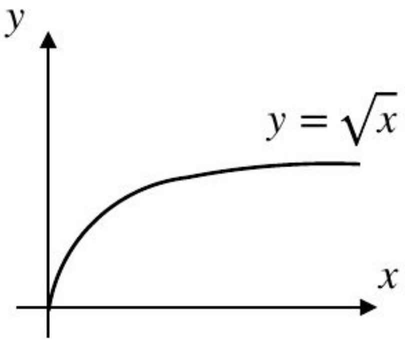


Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 2.2]

คำถาม

สมมติ เราอยากทำนายราคาบ้านเป็น function ของขนาดพื้นที่บ้าน Model ของเรา คือ: $h_\theta(x) = \theta_0 + \theta_1(\mathbf{size}) + \theta_2\sqrt{(\mathbf{size})}$ สมมติ size อยู่ในช่วงตั้งแต่ 1 ถึง 1000 (ตารางฟุต) เราจะ implement model นี้ โดยใช้ model $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$ สมมติ เราอยากใช้ feature scaling (โดยไม่ใช้ mean normalization) เราควรใช้ค่า x_1 และ x_2 เป็นเท่าไร ? (หมายเหตุ: $\sqrt{1000} \approx 32$)

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

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

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

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

คำถาม

สมมติ เราอยากทำนายราคาบ้านเป็น function ของขนาดพื้นที่บ้าน Model ของเรา คือ: $h_\theta(x) = \theta_0 + \theta_1(\textbf{size}) + \theta_2\sqrt{(\textbf{size})}$ สมมติ size อยู่ในช่วงตั้งแต่ 1 ถึง 1000 (ตารางฟุต) เราจะ implement model นี้ โดยใช้ model $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$ สมมติ เราอยากใช้ feature scaling (โดยไม่ใช้ mean normalization) เราควรใช้ค่า x_1 และ x_2 เป็นเท่าไร ? (หมายเหตุ: √1000 ≈ 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) $x_1 = \dfrac{\textbf{size}}{32}$, $x_2 = \sqrt{(\textbf{size})}$

สมมตินักเรียน $m = 4$ คนได้เข้าเรียนวิชาหนึ่ง และวิชานี้มีสอบกลางภาค และสอบปลายภาค เราได้เก็บข้อมูลคะแนนของนักเรียนจากการสอบ 2 ครั้งนี้ มีข้อมูล ดังนี้

คำถาม (2)

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

เราอยากใช้ polynomial regression เพื่อทำนายคะแนนสอบปลายภาคของนักเรียน จากคะแนนสอบกลางภาค สมมติ เราอยากสร้าง model ที่มีรูปแบบ คือ

$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$ เมื่อ x_1 เป็นคะแนนสอบกลางภาค หรือ $midterm\ score$ และ x_2 เป็น $(midterm\ score)^2$

นอกจากนี้ เรายังอยากใช้ feature scaling และ mean normalization normalized feature $x_2^{(4)}$ มีค่าเท่าไร ?

คำถาม (2)

สมมตินักเรียน $m = 4$ คนได้เข้าเรียนวิชาหนึ่ง และวิชานี้มีสอบกลางภาค และสอบปลายภาค เราได้เก็บข้อมูลคะแนนของนักเรียนจากการสอบ 2 ครั้งนี้ มีข้อมูล ดังนี้

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

เราอยากใช้ polynomial regression เพื่อทำนายคะแนนสอบปลายภาคของนักเรียน จากคะแนนสอบกลางภาค สมมติ เราอยากสร้าง model ที่มีรูปแบบ คือ

$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$ เมื่อ x_1 เป็นคะแนนสอบกลางภาค หรือ $midterm\ score$ และ x_2 เป็น $(midterm\ score)^2$

นอกจากนี้ เรายังอยากใช้ feature scaling และ mean normalization normalized feature $x_2^{(4)}$ มีค่าเท่าไร ? [ตอบ: **-0.47**]

3. Linear Regression ที่มีหลายตัวแปร
(Multivariate Linear Regression)

3.6 Normal Equations

Krittameth Teachasrisaksakul

Gradient Descent (Recap)

ขั้นตอนวิธีทำซ้ำ (Iterative algorithm) ที่ converge ที่ค่า local minimum (ค่าต่ำสุดสัมพัทธ์)

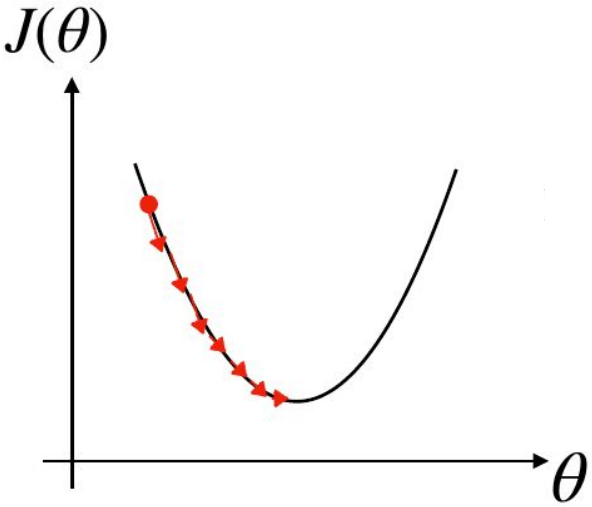
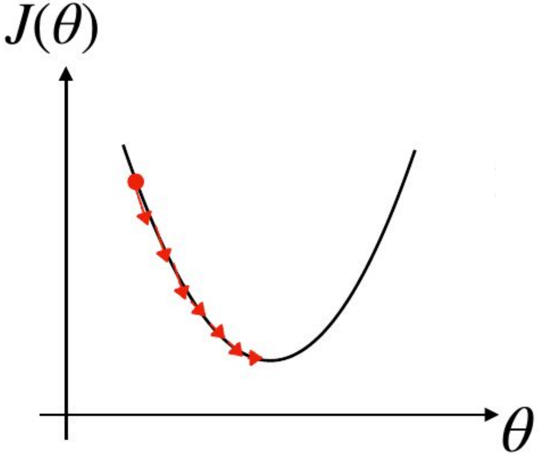


Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 2.2]

Gradient Descent vs. Normal Equation

Gradient Descent:

ขั้นตอนวิธีทำซ้ำ (Iterative algorithm) ที่ converge ที่ค่า local minimum (ค่าต่ำสุดสัมพัทธ์)



Normal equation:

วิธีแก้หาค่า θ ด้วยวิธีการทางพีชคณิต (analytically)

ก็คือ แก้หา local minimum ได้ใน 1 ขั้นตอน

Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 2.2]

Normal Equation : ความเข้าใจพื้นฐาน

$\theta \in \mathbb{R}$

ให้ $\boxed{J(\theta) = a\theta^2 + b\theta + c}$

เพื่อหาค่า optimum หา derivative (อนุพันธ์) และแก้หาค่า θ

ก็คือ

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

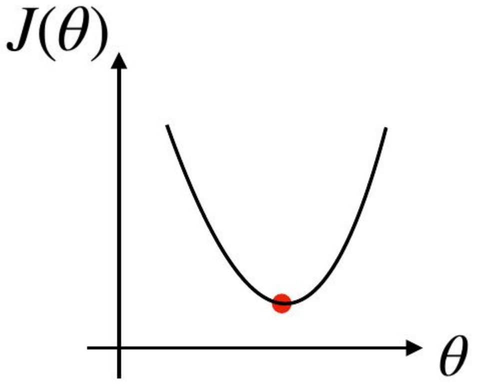


Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 2.2]

Normal Equation : ความเข้าใจพื้นฐาน

เพื่อหาค่า optimum หา derivative (อนุพันธ์) และแก้หาค่า θ

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

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

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

การแก้หาค่า $\theta_0, \theta_1, \ldots, \theta_n$ ทำให้ได้ค่า $\theta_0, \theta_1, \ldots, \theta_n$ ที่ minimize $J(\theta_0, \ldots, \theta_m)$

สังเกตว่าไม่จำเป็นต้องทำ feature scaling เมื่อใช้วิธี normal equation

Normal Equation : ความเข้าใจพื้นฐาน

หาค่า derivatives (อนุพันธ์) และให้เท่ากับ 0

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

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

$$\vdots$$

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

Normal Equation : ความเข้าใจพื้นฐาน

หาค่า derivatives (อนุพันธ์) และให้เท่ากับ 0

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

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

$$\vdots$$

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

Normal Equation

$$\Sigma cx = c \cdot \Sigma x$$

ถ้าจัดรูปเพิ่มเติม จะได้ normal equations

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

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

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

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

$$\vdots$$

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

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

Normal Equation

$x_j^{(i)}$ และ $y^{(i)}$ เป็นค่าคงที่ทั้งหมด เพราะมันเป็นชุดข้อมูล training data sets

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

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

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

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

$$\vdots$$

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

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

Normal Equation

เขียนสมการ โดยใช้ matrix และ vector:

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

(Design Matrix)

เราควรสามารถตรวจสอบได้ว่า ระบบสมการเชิงเส้น (linear equations) กลายเป็น:

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

สมมติว่า X^TX invertible (หาตัวผกผัน หรือ inverse ได้) แล้วจะได้ว่า:

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

คำถาม

สมมติเรามีข้อมูล training แบบในตารางด้านล่าง:

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

เราอยากทำนายน้ำหนัก (weight) ของเด็กเป็นฟังก์ชั่นของอายุ (age) และความสูง (height) ของเด็ก โดยใช้ model: $\text{weight} = \theta_0 + \theta_1 \text{age} + \theta_2 \text{height}$

X และ y มีค่าเท่าไร ?

คำถาม

ตอบ

$$X = \begin{bmatrix} 1 & 4 & 89 \\ 1 & 9 & 124 \\ 1 & 5 & 103 \end{bmatrix} \qquad y = \begin{bmatrix} 16 \\ 28 \\ 20 \end{bmatrix}$$

สมมติเรามีข้อมูล training แบบในตารางด้านล่าง:

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

เราอยากทำนายน้ำหนัก (weight) ของเด็กเป็นฟังก์ชั่นของอายุ (age) และความสูง (height) ของเด็ก โดยใช้ model: $\text{weight} = \theta_0 + \theta_1 \text{age} + \theta_2 \text{height}$

X และ y มีค่าเท่าไร ?

Normal Equation

สมมติว่า $X^T X$ invertible (หาตัวผกผัน หรือ inverse ได้) แล้วจะได้ว่า: $\theta = (X^T X)^{-1} X^T y$

คำถาม: เมื่อใดที่ matrix เป็น non-invertible (หรือ singular / degenerate ก็คือ ไม่สามารถหาตัวผกผัน หรือ inverse ได้) ?

Normal Equation

สมมติว่า $X^T X$ invertible (หาตัวผกผัน หรือ inverse ได้) แล้วจะได้ว่า: $\theta = (X^T X)^{-1} X^T y$

คำถาม: เมื่อใดที่ matrix เป็น non-invertible (หรือ singular / degenerate ก็คือ ไม่สามารถหาตัวผกผัน หรือ inverse ได้) ?

กรณีนี้เกิดขึ้น เมื่อเรามี ...

- **redundant features (features ที่ซ้ำซ้อน)** (ก็คือ column บาง column มีความสัมพันธ์เชิงเส้น : linearly dependent)
  - เช่น x_1 = ขนาด ในหน่วย ตารางฟุต, x_2 = ขนาด ในหน่วย ตารางเมตร
  - เพราะ 1 m (เมตร) ≈ 3.28 feet (ฟุต) ⟶ ดังนั้น $x_1 = (3.28)^2 x_2$
- **จำนวน features มากเกินไป** ($m \leq n$)
  - วิธีแก้ไข: (1) ลบ feature บางตัวออก หรือ
  - (2) ใช้ regularization (อธิบายในบทต่อๆไป)

Gradient Descent vs. Normal Equation

ถ้าเรามี $\boldsymbol{m}$ training examples, $\boldsymbol{n}$ features

Gradient Descent:

- ต้องเลือกค่า α
- ต้องทำหลาย iteration (การวนซ้ำ)
- ทำงานได้ดี แม้ $\boldsymbol{n}$ จะมาก $O(kn^2)$

Normal equation:

- ไม่ต้องเลือกค่า α
- ไม่ต้องวนซ้ำ (iterate)
- ต้องคำนวณ $(X^TX)^{-1}$
- ช้า ถ้า $\boldsymbol{n}$ เยอะมากๆ $O(n^3)$

($\boldsymbol{n}$ มีค่ามาก เมื่อ $n \geq 10^4$)

Computational Complexity : ความซับซ้อนในการคำนวณ

คือ การวิเคราะห์เวลาและเนื้อที่ที่ใช้สำหรับแก้ปัญหาหนึ่งๆ

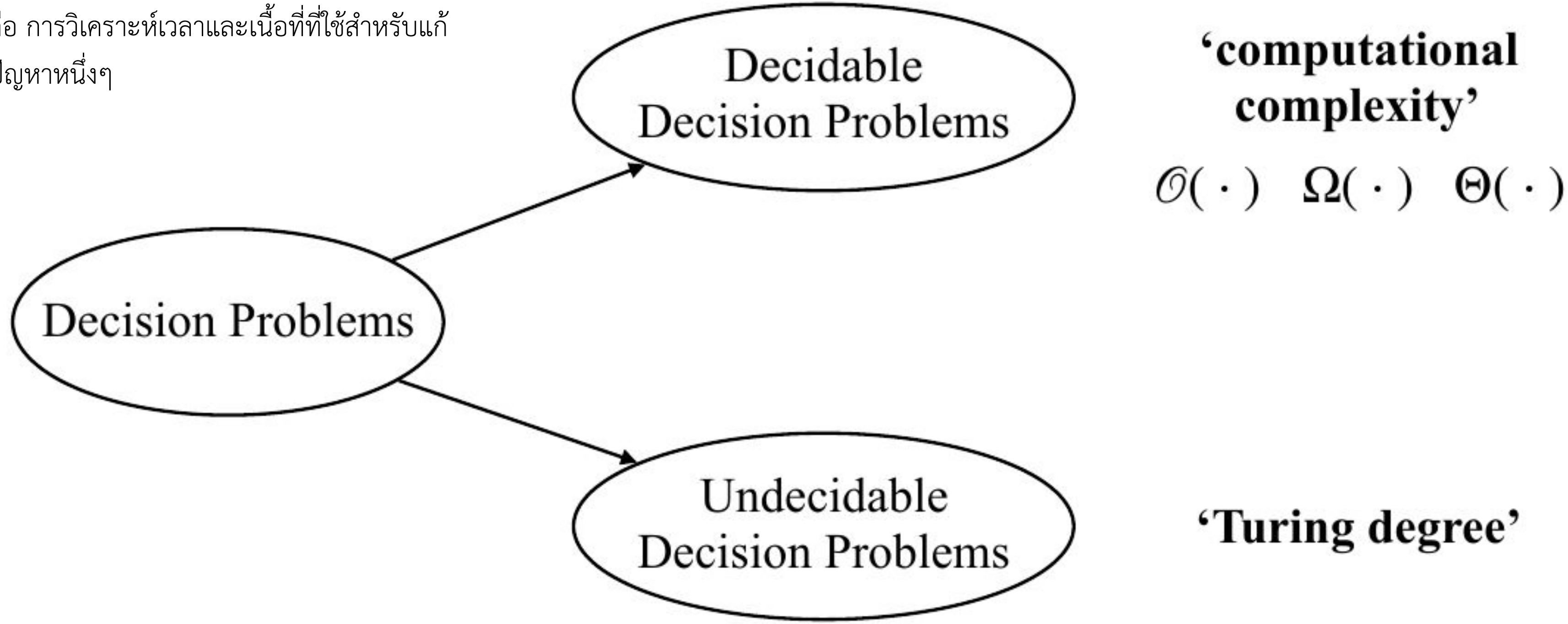


Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 2.2]

คำถาม

สมมติ เรามีชุดข้อมูลที่มี $m = 1{,}000{,}0000$ examples และ $n = 200{,}000$ features สำหรับแต่ละ example เราอยากใช้ multivariate linear regression เพื่อหา parameters θ ที่เหมาะกับข้อมูลของเรา เราควรใช้วิธี gradient descent หรือ the normal equation?

(i) Normal equation เพราะ gradient descent อาจไม่สามารถหาค่า θ ที่เหมาะสม (optimal)

(ii) Gradient descent เพราะถ้าใช้วิธี normal equation การคำนวณ $(X^TX)^{-1}$ จะช้ามาก

(iii) Normal equation เพราะเป็นวิธีที่มีประสิทธิภาพ ที่จะหาคำตอบโดยตรง

(iv) Gradient descent เพราะมันจะ converge ที่ค่า θ ที่เหมาะสม (optimal) เสมอ

คำถาม

สมมติ เรามีชุดข้อมูลที่มี $m = 1,000,0000$ examples และ $n = 200,000$ features สำหรับแต่ละ example เราอยากใช้ multivariate linear regression เพื่อหา parameters θ ที่เหมาะกับข้อมูลของเรา เราควรใช้วิธี gradient descent หรือ the normal equation?

(i) Normal equation เพราะ gradient descent อาจไม่สามารถหาค่า θ ที่เหมาะสม (optimal)

(ii) Gradient descent เพราะถ้าใช้วิธี normal equation การคำนวณ $(X^TX)^{-1}$ จะช้ามาก

(iii) Normal equation เพราะเป็นวิธีที่มีประสิทธิภาพ ที่จะหาคำตอบโดยตรง

(iv) Gradient descent เพราะมันจะ converge ที่ค่า θ ที่เหมาะสม (optimal) เสมอ

สรุป: 2 วิธีที่ใช้ minimize $J(\theta)$ [ทำให้ $J(\theta)$ น้อยที่สุด]

1. แก้สมการ **normal equations**

2. ใช้**วิธีที่วนซ้ำ (iterative methods)** มีหลายวิธีให้เลือกใช้

    a) Batch gradient descent (GD)

    b) Stochastic gradient descent (SGD)

    c) Mini-batch gradient descent (จุดกึ่งกลางระหว่าง GD และ SGD)

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.

References

1. Andrew Ng, Machine Learning, Coursera.
2. Teeradaj Racharak, AI Practical Development Bootcamp.
3. What is Machine Learning?, https://www.digitalskill.org/contents/5