6. Regularization

6.1 ปัญหา Overfitting

Krittameth Teachasrisaksakul

Summary about supervised learning

- If you have continuous $\mathscr{X}$ and continuous $\mathscr{Y}$, your first go-to model should be linear regression. Also, consider non-linear transformation of the inputs.
- If you have continuous $\mathscr{X}$ and discrete $\mathscr{Y}$ but don't know much about $p(\boldsymbol{x} \mid y)$, your first go-to model should be logistic or softmax regression, or may come up with a new GLM from scratch.
- If you have continuous $\mathscr{X}$ and discrete $\mathscr{Y}$ and know something about $p(\boldsymbol{x} \mid y)$, you should model the distribution accurately, as a Gaussian (GDA) or build a new generative model from scratch.
- If you have discrete $\mathscr{X}$ and $\mathscr{Y}$, you should probably start with naive Bayes and build up from there.

ตัวอย่าง: Linear regression (ราคาบ้าน / housing prices)

$$\theta_0 + \theta_1 x$$

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

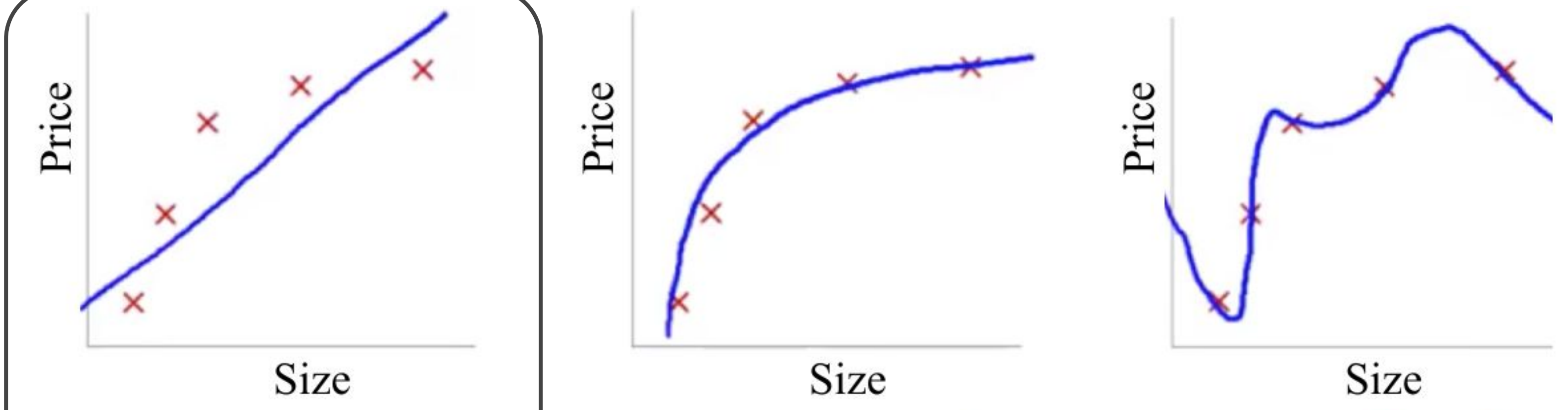
$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 5.1]

ตัวอย่าง: Linear regression (ราคาบ้าน / housing prices)

(1) ใช้โมเดลเส้นตรง ⟶ จุดข้อมูลไม่อยู่บนเส้นตรง ⟶ เส้นตรง ไม่เหมาะกับ ข้อมูลชุดนี้

ทำนายค่า y จาก $x \in \mathbb{R}$:



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

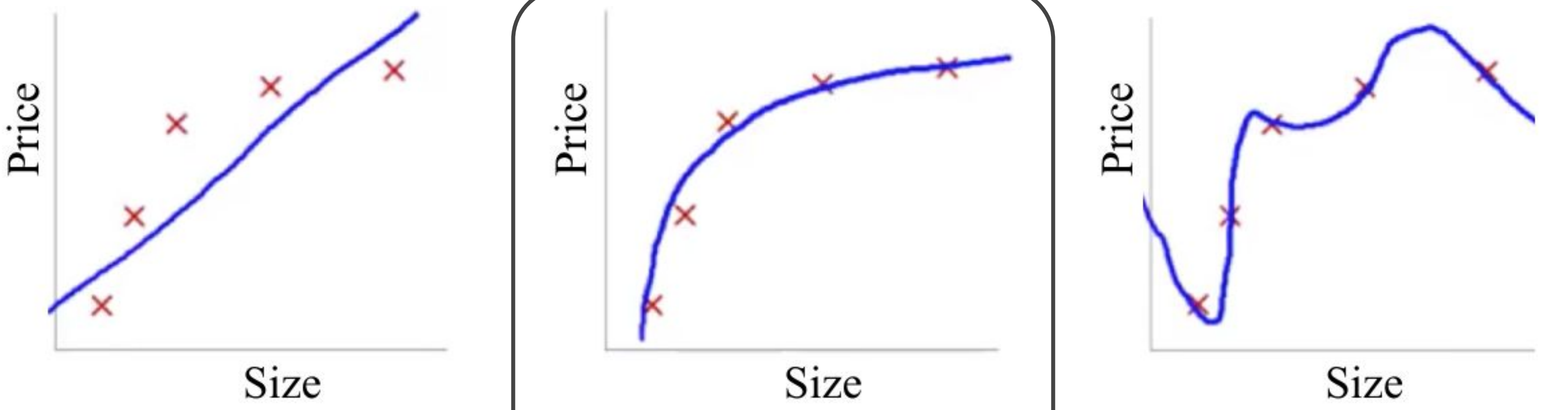
'Underfit' ⟶ hypothesis function h เข้ากับข้อมูลได้ไม่ดี

'High bias' ⟶ **สาเหตุ:** function เรียบง่าย (simple) เกินไป หรือ ใช้ feature จำนวนน้อยเกินไป

Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 5.1]

ตัวอย่าง: Linear regression (ราคาบ้าน / housing prices)

ทำนายค่า y จาก $x \in \mathbb{R}$:



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

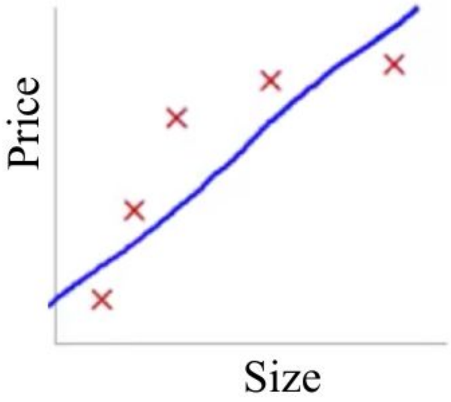
(2) เพิ่ม feature 1 ตัว (x^2) $\longrightarrow$ ได้โมเดล ที่เข้ากับข้อมูล มากขึ้นเล็กน้อย $\longrightarrow$ เหมือนว่า ยิ่งมี feature มาก ยิ่งได้ผลดีขึ้น

Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 5.1]

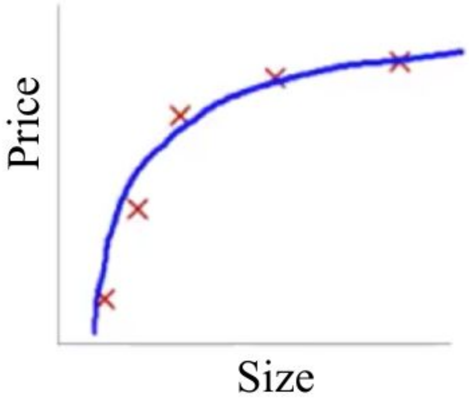
ตัวอย่าง: Linear regression (ราคาบ้าน / housing prices)

ทำนายค่า y จาก $x \in \mathbb{R}$:

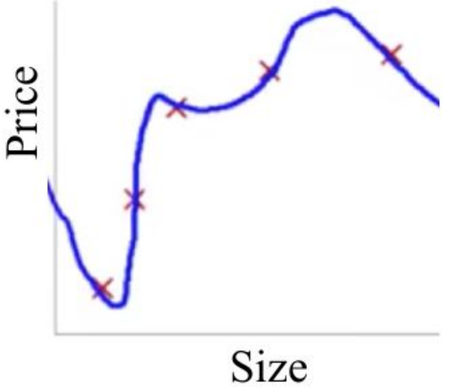
(3) ใช้ฟังก์ชั่นพหุนาม (5^{th} order polynomial) ⟶ โมเดล / เส้นโค้ง จะลากผ่านจุดข้อมูลทุกจุดอย่าง perfect ⟶ แต่โมเดลนี้ อาจไม่ได้เป็น ตัวทำนายที่ดีของ ราคา (y) ของบ้าน (x) ต่างๆ



$$\theta_0 + \theta_1 x$$



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



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

สาเหตุ: ใช้ feature จำนวนมากเกินไป หรือ function ที่ซับซ้อน จะมีส่วนโค้งและมุม ที่ไม่สอดคล้อง (unrelated) กับข้อมูล

‘Overfit’

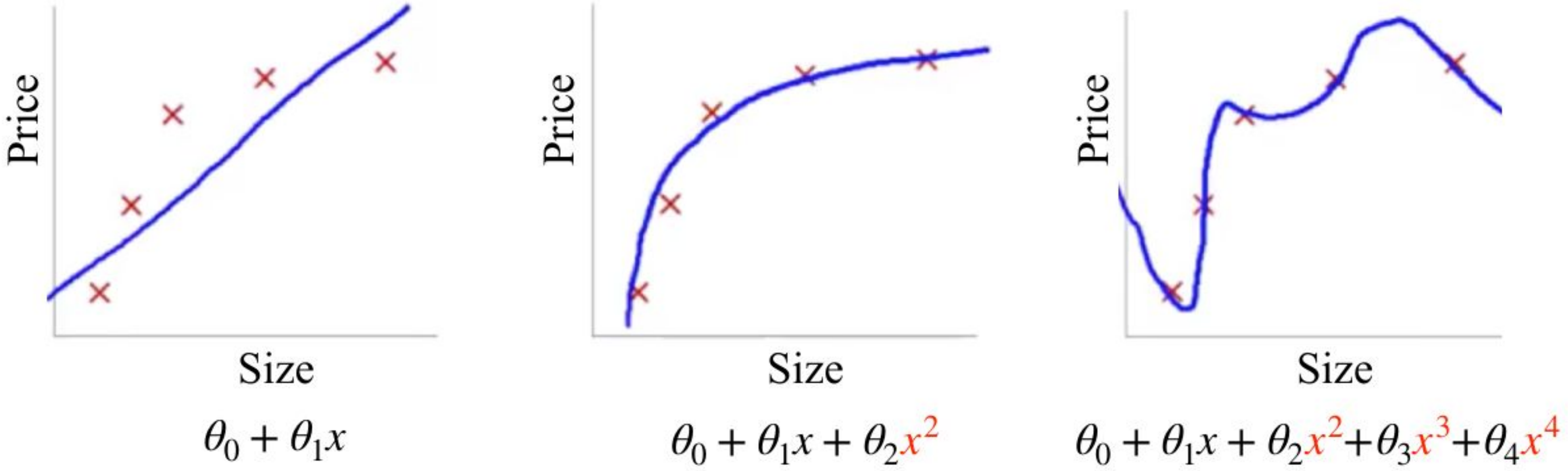
‘High variance’

Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 5.1]

Overfitting คืออะไร?

$$(J(\theta) = \frac{1}{2m}\Sigma_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 \approx 0)$$

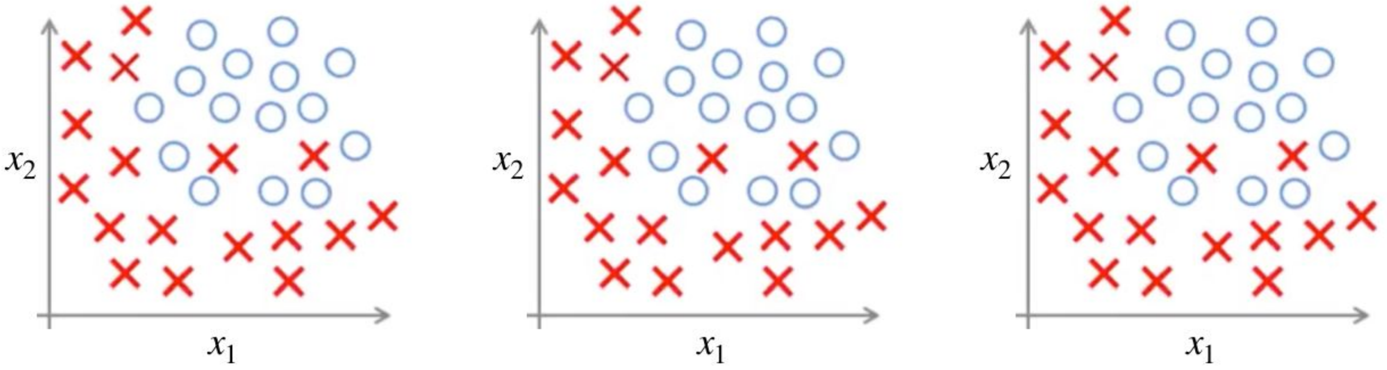
ตัวอย่าง: Linear regression (ราคาบ้าน / housing prices)



Overfitting: ผล ⟶ hypothesis ที่เรียนรู้ อาจประมาณค่า ชุดข้อมูล training set ได้ดีมากๆ แต่ไม่สามารถประมาณค่า (generalize) ข้อมูลใหม่ ที่ไม่เคยเจอมาก่อน (ก็คือ ทำนายราคาของตัวอย่างใหม่)

Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 5.1]

Overfitting ใน Logistic Regression



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

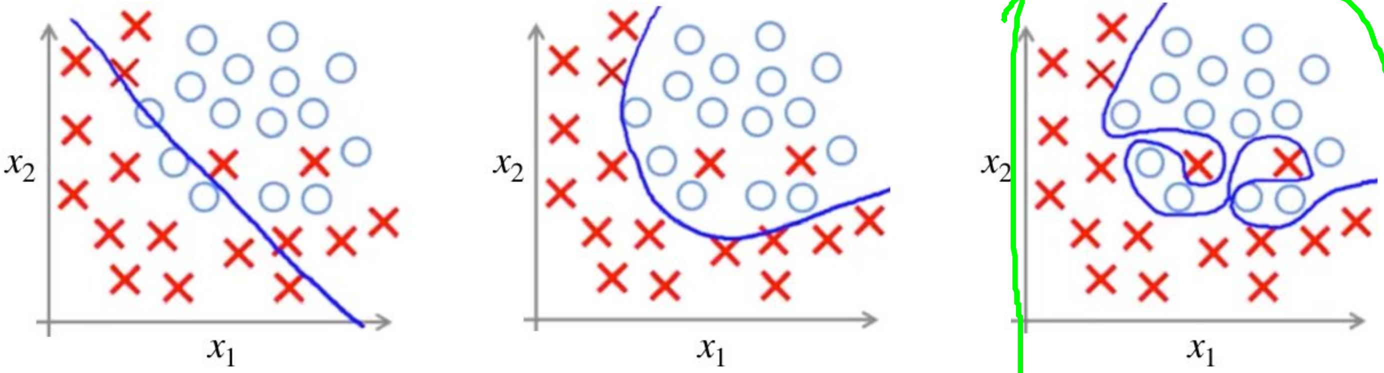
(g is a sigmoid function)

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

$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \ldots)$$

Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 5.1]

Overfitting ใน Logistic Regression



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

(g is a sigmoid function)

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

$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \ldots)$

'Underfit'

'High bias'

'Overfit'

'High variance'

Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 5.1]

คำถาม

พิจารณาปัญหาการวินิจฉัยโรค (medical diagnosis) ที่แบ่งประเภทเนื้องอกเป็น ร้าย (malignant) หรือ ไม่ร้าย (benign) ถ้า hypothesis $h_\theta(x)$ overfit ชุดข้อมูล training set หมายความว่าอะไร ?

(i) มันทำนายตัวอย่างใน training set ได้ แม่นยำ และ generalize ได้ดี ทำให้ทำนายตัวอย่างใหม่ ที่ไม่เคยเจอ ได้แม่นยำด้วย

(ii) มันทำนายตัวอย่างใน training set ได้ **ไม่**แม่นยำ และ generalize ได้ดี ทำให้ทำนายตัวอย่างใหม่ ที่ไม่เคยเจอ ได้แม่นยำด้วย

(iii) มันทำนายตัวอย่างใน training set ได้ แม่นยำ และ generalize ได้**ไม่**ดี ทำให้ทำนายตัวอย่างใหม่ ที่ไม่เคยเจอ ได้**ไม่**แม่นยำ

(iv) มันทำนายตัวอย่างใน training set ได้ **ไม่**แม่นยำ และ generalize ได้**ไม่**ดี ทำให้ทำนายตัวอย่างใหม่ ที่ไม่เคยเจอ ได้**ไม่**แม่นยำ

คำถาม

พิจารณาปัญหาการวินิจฉัยโรค (medical diagnosis) ที่แบ่งประเภทเนื้องอกเป็น ร้าย (malignant) หรือ ไม่ร้าย (benign) ถ้า hypothesis $h_\theta(x)$ overfit ชุดข้อมูล training set หมายความว่าอะไร ?

(i) มันทำนายตัวอย่างใน training set ได้ แม่นยำ และ generalize ได้ดี ทำให้ทำนายตัวอย่างใหม่ ที่ไม่เคยเจอ ได้แม่นยำด้วย

(ii) มันทำนายตัวอย่างใน training set ได้ **ไม่**แม่นยำ และ generalize ได้ดี ทำให้ทำนายตัวอย่างใหม่ ที่ไม่เคยเจอ ได้แม่นยำด้วย

(iii) มันทำนายตัวอย่างใน training set ได้ แม่นยำ และ generalize ได้**ไม่**ดี ทำให้ทำนายตัวอย่างใหม่ ที่ไม่เคยเจอ ได้**ไม่**แม่นยำ

(iv) มันทำนายตัวอย่างใน training set ได้ **ไม่**แม่นยำ และ generalize ได้**ไม่**ดี ทำให้ทำนายตัวอย่างใหม่ ที่ไม่เคยเจอ ได้**ไม่**แม่นยำ

จัดการปัญหา Overfitting

ใช้ features จำนวนมากเกินไป อาจทำให้เกิด overfitting

- x_1 = ขนาดพื้นที่บ้าน
- x_2 = จำนวน ห้องนอน
- x_3 = จำนวน ชั้น
- x_4 = อายุบ้าน
- x_5 = รายได้เฉลี่ยของบริเวณใกล้เคียง
- x_6 = ขนาดพื้นที่ห้องครัว

$\vdots$

- x_{100}

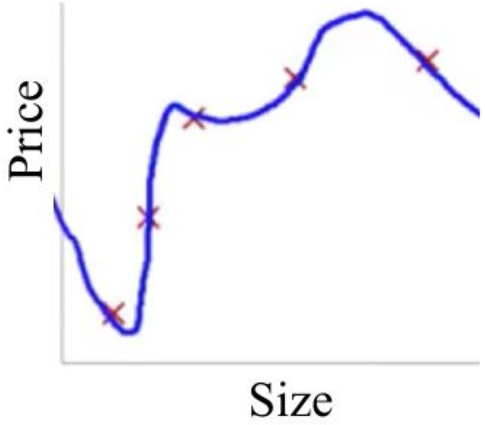


Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 5.1]

จัดการปัญหา Overfitting

1. ลดจำนวน features

- เลือก features ที่จะเก็บไว้ ด้วยมือ
- ใช้ algorithm ที่ทำการเลือก model (Model selection algorithm)

2. Regularization

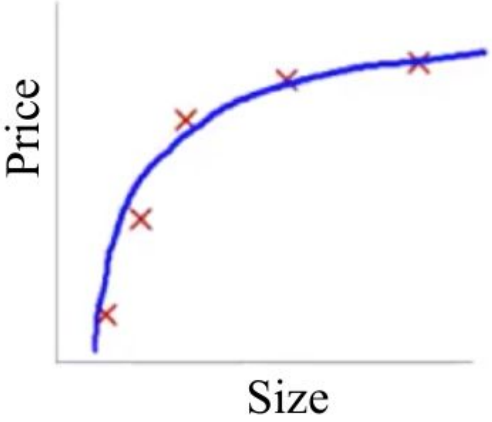
- เก็บ features ทั้งหมดไว้ แต่ลดขนาด (magnitude) หรือ ค่าของ parameters θ_j
- ทำงานได้ดีเมื่อเรามีจำนวน features มากๆ โดยที่ feature แต่ละตัวส่งผลให้ทำนาย y ได้

6. Regularization

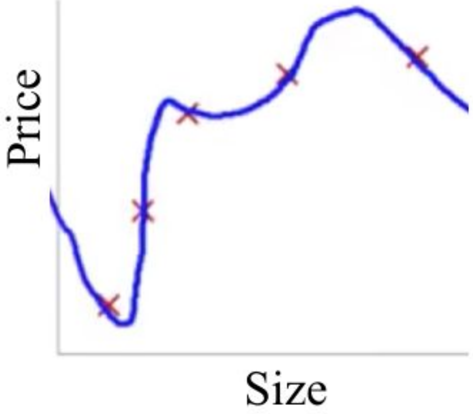
6.2 Cost Function

Krittameth Teachasrisaksakul

Regularization : ความเข้าใจพื้นฐาน



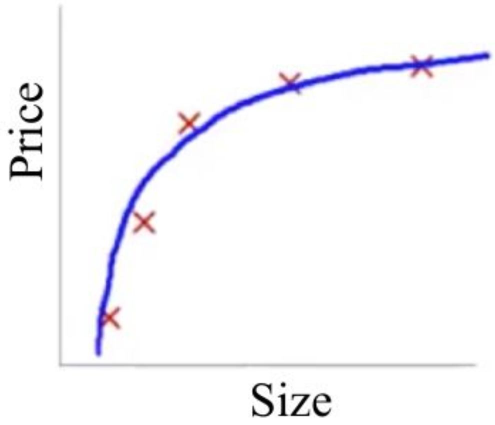
$$\theta_0 + \theta_1 x + \theta_2 x^2$$

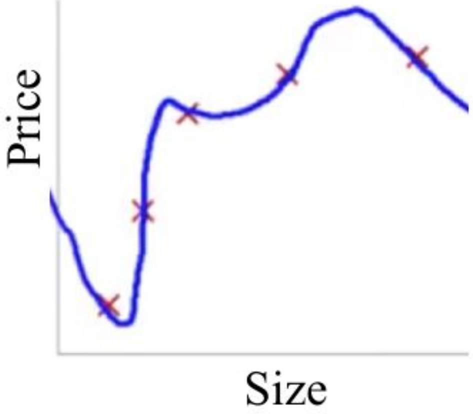


$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 5.1]

Regularization : ความเข้าใจพื้นฐาน





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

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

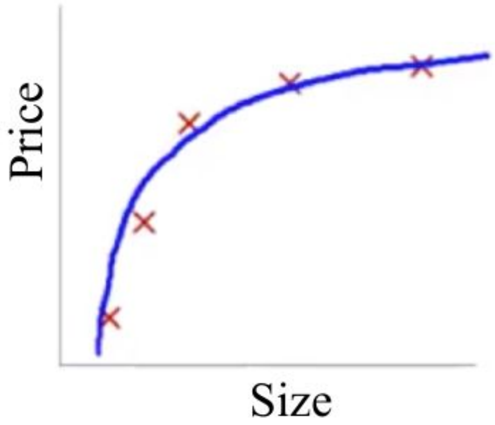
Optimization objective:

(เป้าหมายของ optimization หรือ การปรับค่า parameter เพื่อหาค่าที่เหมาะสม)

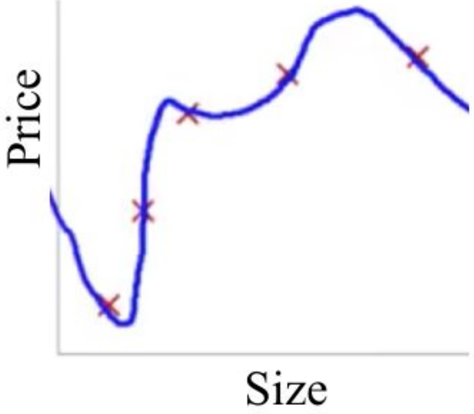
$$\min_{\theta} \frac{1}{2m} \Sigma_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 5.1]

Regularization : ความเข้าใจพื้นฐาน



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



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

สมมติเรา penalize (ทำโทษ) และทำให้ θ_3, θ_4 น้อยมากๆ (ก็คือ ไม่สนับสนุนให้ใช้ θ_3, θ_4)

ปรับ Optimization objective เป็น:

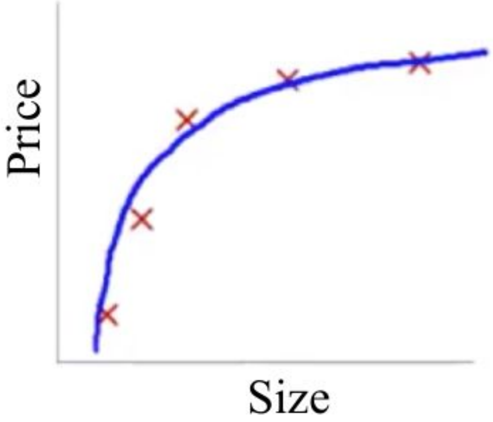
$$\min_{\theta} \frac{1}{2m} \Sigma_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + 1000\theta_3^2 + 1000\theta_4^2$$

$$\therefore \theta_3 \approx 0, \theta_4 \approx 0$$

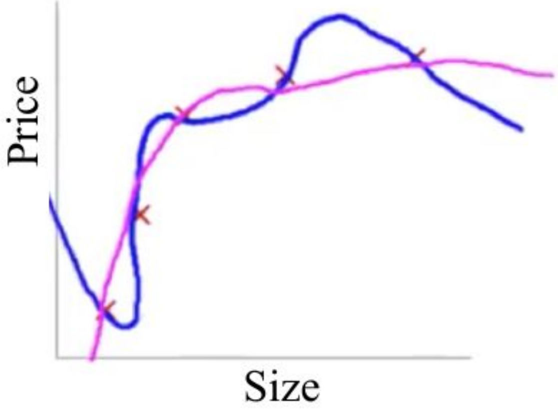
Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 5.1]

Regularization : ความเข้าใจพื้นฐาน

จากการปรับ optimization objective ทำให้ได้กราฟเส้นใหม่ (สีฟ้าอ่อน)



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



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

ถ้ามี hypothesis function ที่ overfitting เมื่อใช้กับข้อมูล

- **(1) เพิ่ม 2 พจน์ท้าย**
- **เพิ่ม cost** ของ θ_3, θ_4
- **ลดค่าน้ำหนัก (weight)** θ_3, θ_4 ของบางพจน์ใน function
- **(2)** ถ้าอยากให้ cost function เข้าใกล้ 0 ⟶ ต้องลดค่า θ_3, θ_4 ให้ใกล้ 0

สมมติเรา penalize (ทำโทษ) และ ทำให้ θ_3, θ_4 น้อยมากๆ (ก็คือ ไม่สนับสนุนให้ใช้ θ_3, θ_4)

ปรับ Optimization objective เป็น:

(1)

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

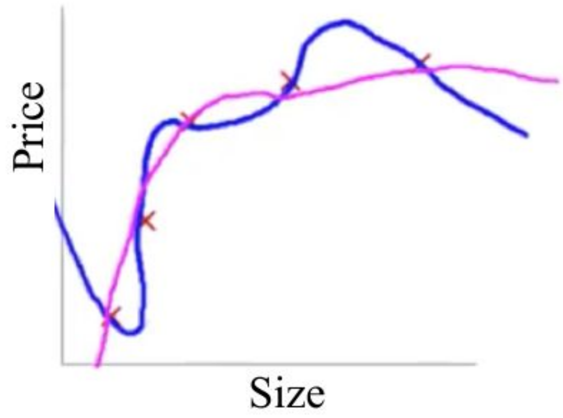
(2) $\boxed{\therefore \theta_3 \approx 0, \theta_4 \approx 0}$

Figure Source: Teeradaj Racharak; AI Practical Develop- ment Bootcamp [L 5.1]

Regularization : อธิบายแบบทางการ

ค่าที่น้อย ของ parameter $\theta_1, \theta_2, ..., \theta_n$ จะทำให้เกิด

- Hypothesis ที่ง่ายขึ้น (smooth มากขึ้น)
- มีแนวโน้ม overfitting น้อยลง



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

ตัวอย่าง การทำนายราคาบ้าน:

- Features: $x_1, x_2, ..., x_{100}$
- Parameters: $\theta_1, \theta_2, ..., \theta_{100}$

Cost function (ของ linear regression) :

$$J(\theta) = \frac{1}{2m}\left[\Sigma_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \boxed{\lambda \Sigma_{i=j}^{n} \theta_j^2}\right]$$

เพิ่มพจน์นี้ ⟶ ทำให้ค่า output ของ hypothesis function smooth ⟶ เพื่อลด overfitting

ถ้า λ มีค่ามากเกินไป ⟶ อาจ smooth out function มากเกินไป ⟶ ทำให้เกิด underfitting

Regularization : อธิบายแบบทางการ

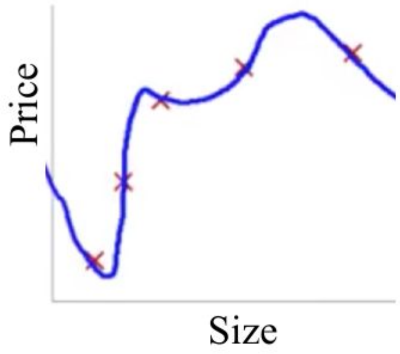
กำหนดว่า cost ของ parameter $\boldsymbol{\theta}$ ถูกเพิ่มขึ้นเท่าไร

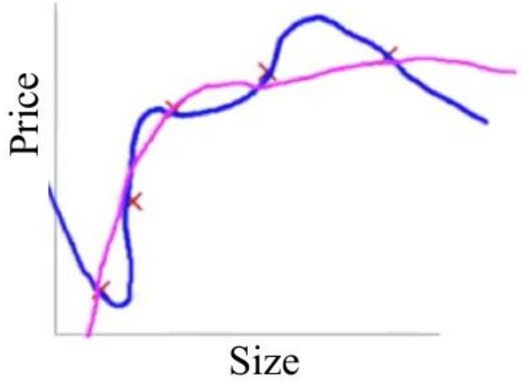
Regularized cost function:

$$J(\theta) = \frac{1}{2m}\left[\Sigma_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \overbrace{\lambda\Sigma_{j=1}^{n}\theta_j^2}^{\text{regularization term}}\right]$$

regularization parameter

Goal: $\min_{\theta} J(\theta)$





คำถาม

ใน regularized linear regression เราเลือกค่า θ เพื่อทำให้ $J(\theta)$ น้อยที่สุด

$$J(\theta) = \frac{1}{2m}\left[\Sigma_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\Sigma_{j=1}^{n}\theta_j^2\right]$$

ถ้า λ ถูกตั้งค่าเป็นค่าที่เยอะมากๆ (อาจมากเกินไปสำหรับปัญหาของเรา สมมติ $\lambda = 10^{10}$)

(i) Algorithm ทำงานได้ดี; ตั้งค่า λ เป็นค่าเยอะมาก ไม่มีผลอะไร

(ii) Algorithm ไม่สามารถแก้ปัญหา overfitting ได้

(iii) Algorithm ทำให้เกิด underfitting (ไม่สามารถหา parameter ของ training set ได้)

(iv) Gradient descent จะไม่ converge

คำถาม

ใน regularized linear regression เราเลือกค่า θ เพื่อทำให้ $J(\theta)$ น้อยที่สุด

$$J(\theta) = \frac{1}{2m}\left[\Sigma_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\Sigma_{j=1}^{n}\theta_j^2\right]$$

ถ้า λ ถูกตั้งค่าเป็นค่าที่เยอะมากๆ (อาจมากเกินไปสำหรับปัญหาของเรา สมมติ $\lambda = 10^{10}$)

(i) Algorithm ทำงานได้ดี; ตั้งค่า λ เป็นค่าเยอะมาก ไม่มีผลอะไร

(ii) Algorithm ไม่สามารถแก้ปัญหา overfitting ได้

(iii) Algorithm ทำให้เกิด underfitting (ไม่สามารถหา parameter ของ training set ได้)

(iv) Gradient descent จะไม่ converge

6. Regularization

ize

-ize
-ify
-ate

6.3 Regularized Linear Regression

Krittameth Teachasrisaksakul

- สามารถใช้ regularization ได้กับ linear regression และ logistic regression
- พิจารณา linear regression ก่อน

Cost Function (Recap / ทบทวน)

Cost function:

$$J(\theta) = \frac{1}{2m}\left[\Sigma_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\Sigma_{j=1}^{n}\theta_j^2\right]$$

Goal:

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

Gradient Descent (เดิม)

Repeat {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \Sigma_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \qquad (j = 0,1,2,\ldots,n)$$

}

Gradient Descent สำหรับ Regularized Linear Regression

- ปรับเปลี่ยน gradient descent function เพื่อแยก θ_0 ออกจาก parameter ตัวอื่นๆ
- เพราะเราไม่อยาก penalize θ_0 (ลงโทษ / ขัดขวางการใช้ θ_0)

Repeat {

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

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

ทำ regularization

$$(j = \cancel{0}, 1, 2, \ldots, n)$$

}

$$\frac{\partial}{\partial \theta_j} J(\theta)$$

Gradient Descent สำหรับ Regularized Linear Regression

Repeat {

$$\frac{\partial}{\partial \theta_0} J(\theta)$$

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

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

$$(j = \cancel{0}, 1, 2, \ldots, n)$$

}

$$\frac{\partial}{\partial \theta_j} J(\theta)$$

หลังการปรับปรุง กฎในการปรับค่า (update rule) ของ parameter θ จะเป็น:

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

Gradient Descent สำหรับ Regularized Linear Regression

Repeat {

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

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

$$(j = \cancel{0},1,2,\ldots,n)$$

}

หลังการปรับปรุง กฎในการปรับค่า (update rule) ของ parameter θ จะเป็น:

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

พจน์แรกมาจาก การดึงตัวร่วม θ_j

Gradient Descent สำหรับ Regularized Linear Regression

Repeat {

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

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

}

$$(j = \cancel{0},1,2,\ldots,n)$$

(1) พจน์แรก น้อยกว่า 1 เสมอ

พจน์แรก คือ ลดค่า $\boldsymbol{\theta_j}$ ลง ทุกครั้งที่ปรับค่ามัน

(2) พจน์ที่ 2 เหมือนกับสมการบน

หลังการปรับปรุง กฎในการปรับค่า (update rule) ของ parameter θ จะเป็น:

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

คำถาม

สมมติเรากำลังทำ gradient descent กับชุดข้อมูล training set ที่มีตัวอย่างจำนวน $m > 0$ ตัวอย่าง โดยใช้ learning rate ที่ค่อนข้างน้อย $\alpha > 0$ และ regularization parameter $\lambda > 0$ พิจารณา update rule (กฎการปรับค่า parameter)

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

ข้อใดเป็นจริงเกี่ยวกับพจน์

$$\left(1 - \alpha\frac{\lambda}{m}\right)$$

$1 - \alpha\frac{\lambda}{m} > 1$ | $1 - \alpha\frac{\lambda}{m} = 1$ | $1 - \alpha\frac{\lambda}{m} < 1$ | None of these

คำถาม

สมมติเรากำลังทำ gradient descent กับชุดข้อมูล training set ที่มีตัวอย่างจำนวน $m > 0$ ตัวอย่าง โดยใช้ learning rate ที่ค่อนข้างน้อย $\alpha > 0$ และ regularization parameter $\lambda > 0$ พิจารณา update rule (กฎการปรับค่า parameter)

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

ข้อใดเป็นจริงเกี่ยวกับพจน์

$$\left(1 - \alpha\frac{\lambda}{m}\right)$$

$1 - \alpha\frac{\lambda}{m} > 1$

$1 - \alpha\frac{\lambda}{m} = 1$

$\boxed{1 - \alpha\frac{\lambda}{m} < 1}$

None of these

Normal Equation (Recap)

$$X = \begin{bmatrix} (x^{(1)})^T \\ \vdots \\ (x^{(m)})^T \end{bmatrix}_{m \times (n+1)} \qquad y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix} \in \mathbb{R}^m$$

Goal: $\min_{\theta} J(\theta)$

Normal Equation (Recap)

$$X = \begin{bmatrix} (x^{(1)})^T \\ \vdots \\ (x^{(m)})^T \end{bmatrix}_{m \times (n+1)} \qquad y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix} \in \mathbb{R}^m$$

Goal: $\min_{\theta} J(\theta)$

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

Normal Equation (Recap)

ทำ regularization ด้วยวิธี Normal Equation: ใช้สมการเดิม + เพิ่ม 1 พจน์ในวงเล็บ (เพื่อทำ regularization)

$$X = \begin{bmatrix} (x^{(1)})^T \\ \vdots \\ (x^{(m)})^T \end{bmatrix}_{m \times (n+1)} \qquad y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix} \in \mathbb{R}^m$$

$$\frac{\partial}{\partial \theta_j} J(\theta) = \ldots = 0$$

Goal: $\min_{\theta} J(\theta)$

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

ปัญหา non-invertibility (การหา inverse matrix ไม่ได้)

- $m =$ จำนวน examples / ตัวอย่าง, $n =$ จำนวน features
- ถ้า $m < n$ $\longrightarrow$ แล้ว (X^TX) เป็น non-invertible / singular (หา inverse matrix ไม่ได้)
- ถ้า $m = n$ $\longrightarrow$ แล้ว (X^TX) อาจเป็น be non-invertible

อย่างไรก็ตาม regularization สามารถจัดการกับปัญหา non-invertibility ได้

ถ้า $\lambda > 0$ แล้ว :

$$\theta = (\underbrace{X^TX + \lambda \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{(n+1)\times(n+1)}}_{\text{invertible}})^{-1} X^T y$$

(หา inverse matrix ได้)

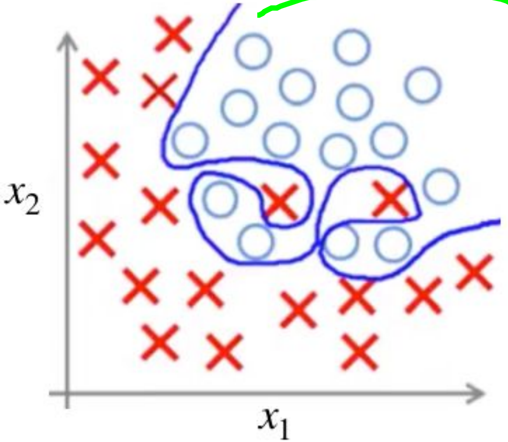
6. Regularization

6.4 Regularized Logistic Regression

Krittameth Teachasrisaksakul

เพื่อหลีกเลี่ยงปัญหา overfitting ⟶ ทำ regularization กับ logistic regression ด้วย วิธีคล้ายๆกับบทก่อนหน้า (ทำ regularization กับ linear regression)

Logistic Regression (Recap)

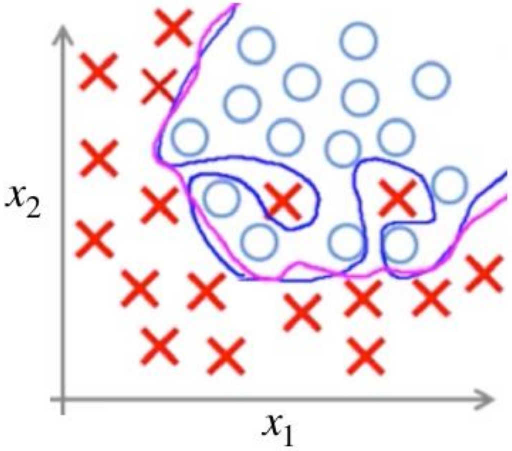


$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \ldots)$$

Cost function:

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

Regularized Logistic Regression



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \ldots)$$

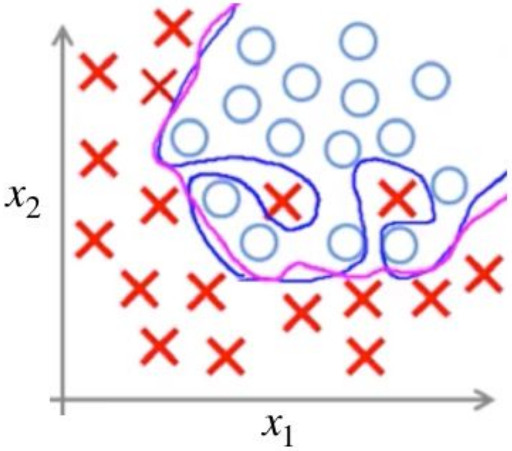
regularized function (เส้นสีชมพู) มีแนวโน้มที่จะ overfit น้อยกว่า non-regularized function (เส้นสีน้ำเงิน):

Cost function:

$$J(\theta) = -\left[\frac{1}{m}\Sigma_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log\left(1 - h_\theta(x^{(i)})\right)\right] + \frac{\lambda}{2m}\Sigma_{j=1}^{n}\theta_j^2$$

Figure source: Andrew Ng's Machine Learning course

Regularized Logistic Regression



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \ldots)$$

Regularize โดยเพิ่มพจน์ท้ายสุด

- vector θ มี index จาก 0 ถึง n (มีทั้งหมด $\mathrm{n}+1$ ตัว : θ_0 ถึง θ_n)
- ผลรวมข้าม θ_0 โดยให้ j เป็น 1 ถึง n (ข้าม 0)
- **ก็คือ แยก พจน์ bias term $\boldsymbol{\theta_0}$ ออก**

Cost function:

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

$$+\frac{\lambda}{2m}\Sigma_{j=1}^{n}\theta_j^2$$

Figure source: Andrew Ng's Machine Learning course

Regularized Logistic Regression

Repeat {

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

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

}

$$(j = \cancel{0}, 1, 2, \ldots, n)$$

นี่ **ไม่ใช่** algorithm เดียวกับ gradient descent สำหรับ regularized linear regression เพราะ ...

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

คำถาม

เมื่อใช้ regularized logistic regression วิธีใดเป็นวิธีที่ดีที่สุดที่จะสังเกตการณ์ว่า gradient descent ทำงานอย่างถูกต้อง ?

(i) Plot

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

เป็น functic

(ii) Plot

$$-[\frac{1}{m}\Sigma_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)})\log(1 - h_\theta(x^{(i)}))] + \frac{\lambda}{2m}\Sigma_{j=1}^{n}\theta_j^2$$

เป็น func

(iii) Plot $\Sigma_{j=1}^{n}\theta_j^2$ เป็น function ของจำนวน iterations และทำให้แน่ใจว่ามันลดลง

คำถาม

เมื่อใช้ regularized logistic regression วิธีใดเป็นวิธีที่ดีที่สุดที่จะสังเกตการณ์ว่า gradient descent ทำงานอย่างถูกต้อง ?

(i) Plot

เป็น functic $-[\frac{1}{m}\Sigma_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))]$

(ii) Plot

เป็น func $-[\frac{1}{m}\Sigma_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))] + \frac{\lambda}{2m}\Sigma_{j=1}^{n}\theta_j^2$

(iii) Plot $\Sigma_{j=1}^{n}\theta_j^2$ เป็น function ของจำนวน iterations และทำให้แน่ใจว่ามันลดลง

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