Logistic Regression

อ. ปรัชญ์ ปิยะวงศ์วิศาล

Pratch Piyawongwisal

Today

- ~~Midterm Solution~~
- Recap - Linear Regression
  - Minimizing cost function
  - Solutions
    - Normal Equation
    - (Stochastic) Gradient Descent
  - Polynomial Regression
  - Regularization -> solves overfitting
- Logistic Regression

Recap: Supervised Learning

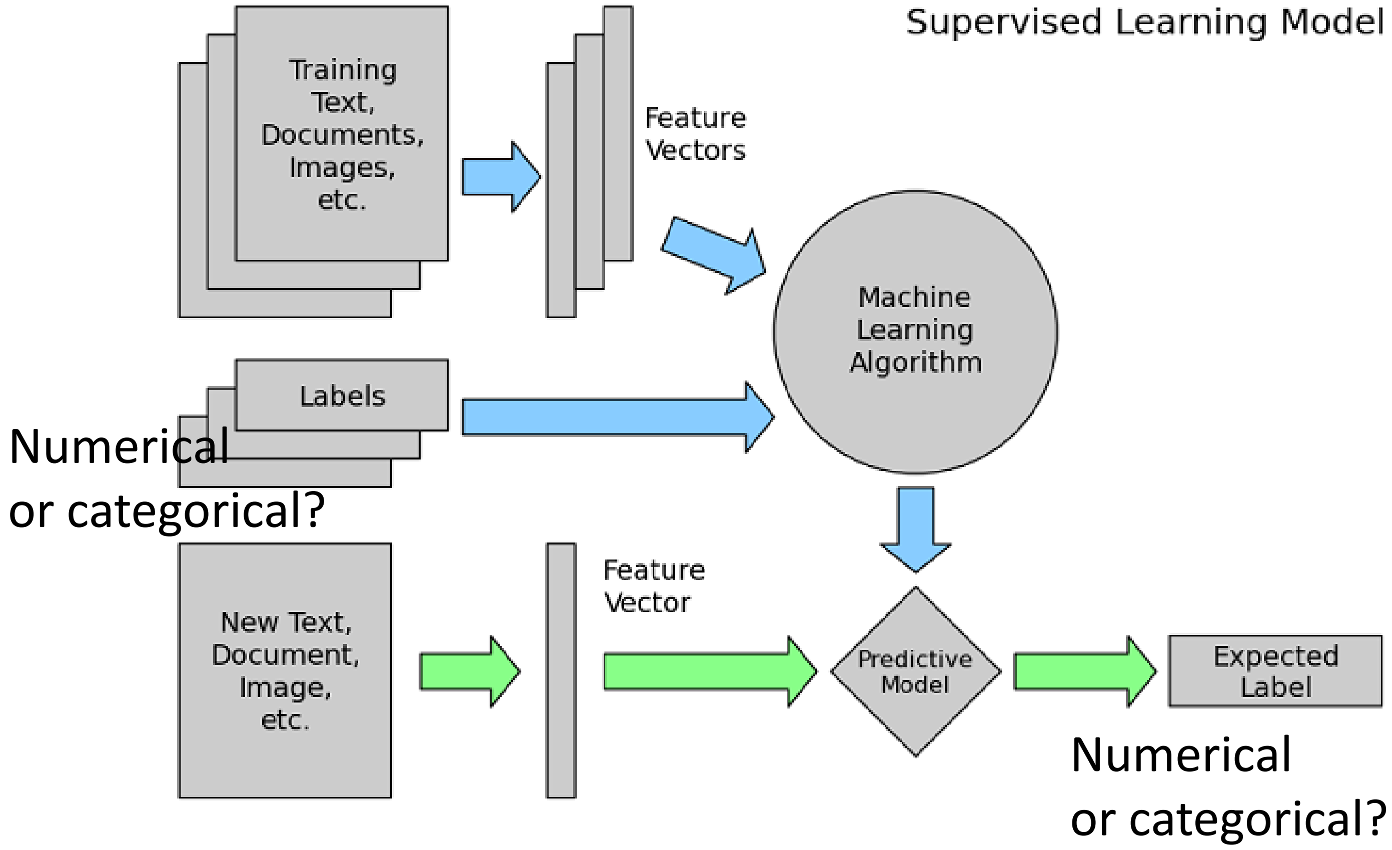
kNN

Logistic Regression

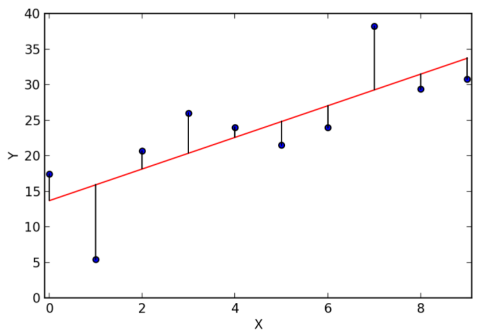
- Classification
  - Predicts class labels/categories
  - ทำนายค่าที่เป็นหมวดหมู่ = จำแนกประเภท
  - อาจมองเป็นการหา boundary ที่แบ่งข้อมูลในแต่ละหมวดหมู่ ออกจากกัน

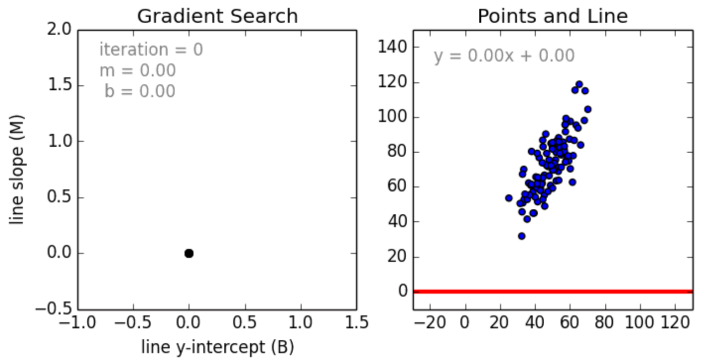
Linear Regression

- Regression
  - Predicts continuous values
  - ทำนายค่าที่เป็นจำนวนจริง
  - อาจมองเป็นการหา hyperplane ที่ fit กับข้อมูลที่มีมากที่สุด



Linear Regression - Summary



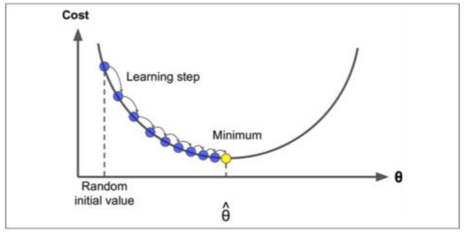


- Regression: หาโมเดลที่ fit กับข้อมูลได้ดีที่สุดได้อย่างไร?
- โมเดล: $\hat{y} = h_\theta(x) = \boldsymbol{\theta}^T \boldsymbol{x}$
- cost function ของโมเดล: mean-square error (MSE)

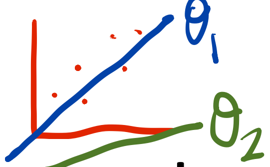
$$J(\theta) = MSE(\theta) = \frac{1}{M}\sum_{i=1}^{M}(\theta^T x^{(i)} - y^{(i)})^2$$

- การ train โมเดล คือ การหาค่าของ $\hat{\theta}_{MSE} = \underset{\theta}{\operatorname{argmin}}\, J(\theta)$
- Solution 1: แก้สมการตรง ๆ จะได้ Normal equation: $\hat{\theta}_{MSE} = (\mathbf{X}^T.\mathbf{X})^{-1}.\mathbf{X}^T.\mathbf{y}$
- Solution 2: หรือใช้วิธี Gradient Descend โดยค่อยๆ update θ:

$$\theta^{(\textbf{next step})} = \theta - \boxed{\eta}\nabla_\theta MSE(\theta)$$



ภูเขา MSE

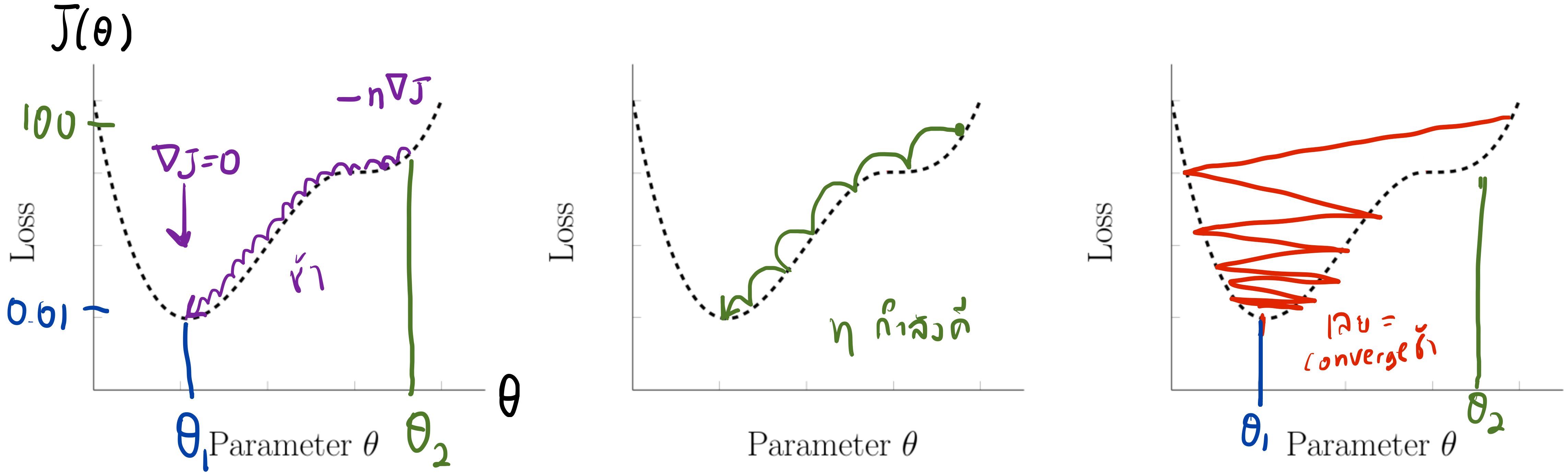


$J(\theta_1) = MSE(\theta_1) = 0.01$

$J(\theta_2) = MSE(\theta_2) = 100$

hyperparam

How to choose the learning rate η ?



low $\eta = 0.1$

high $\eta = 10$

$\theta_{\text{ใหม่}}$ $\quad$ $\theta_{\text{เก่า}}$ $\quad$ $\boxed{-\nabla_\theta J}$

$$\theta^{(\textbf{next step})} = \theta - \boldsymbol{\eta} \nabla_\theta MSE(\theta)$$

learning rate

Exercise: Housing Price Prediction

- ข้อมูล training

ขนาด (Sq. ft.)	จำนวนห้องนอน	ระยะทางไปห้างสรรพสินค้า	ราคา (ล้านบาท)
1000	2	5	9.5
1500	3	30	8.0
2000	5	40	12.5
1700	1	5	9.0
1200	2	30	5.5

โมเดล Linear Regression (ราคาสำหรับ 1 หลัง) คือ

$$h_\theta(x) = \theta_0 + \theta_1 \mathrm{x}_1 + \theta_2 \mathrm{x}_2 + \theta_3 \mathrm{x}_3$$

เราสามารถเขียนข้อมูลในรูป matrix (สำหรับ m หลัง) ได้ดังนี้

$$\boldsymbol{X} = \begin{bmatrix} \mathbf{1} & \mathbf{1000} & \mathbf{2} & \mathbf{5} \\ \mathbf{1} & \mathbf{1500} & \mathbf{3} & \mathbf{30} \\ \mathbf{1} & \mathbf{2000} & \mathbf{5} & \mathbf{40} \\ \mathbf{1} & \mathbf{1700} & \mathbf{1} & \mathbf{5} \\ \mathbf{1} & \mathbf{1200} & \mathbf{30} & \mathbf{5.5} \end{bmatrix} \qquad \boldsymbol{y} = \begin{bmatrix} \mathbf{9.5} \\ \mathbf{8.0} \\ \mathbf{12.5} \\ \mathbf{9.0} \\ \mathbf{5.5} \end{bmatrix}$$

โมเดลในรูป matrix: $\hat{\mathbf{y}} = \boldsymbol{X\theta}$

จากนั้นคำนวณหา $\hat{\theta}_{MSE}$ โดยใช้ normal equation:

$$\hat{\theta}_{MSE} = \left(\mathbf{X}^{\mathbf{T}}.\mathbf{X}\right)^{-\mathbf{1}}.\mathbf{X}^{\mathbf{T}}.\mathbf{y}$$

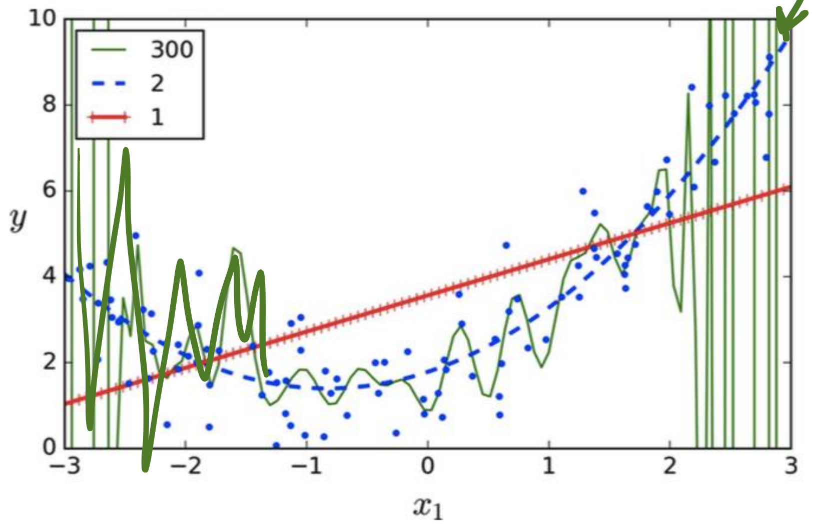
Polynomial Regression, Regularization

- สามารถทำ Polynomial Regression โดยเพิ่มมิติข้อมูล training เช่น
  - x_i เดิม = [น้ำหนัก,ส่วนสูง,อายุ] = [70, 150, 30]
  - x_i ใหม่ = [70, 150, 30, 70^2, 150^2, 30^2]
- แต่ถ้า poly degree สูงไปอาจทำให้ overfit ☹
- Solution: เพิ่มพจน์ regularization ใน cost function

$$J(\theta) = \mathrm{MSE}(\theta) + \boxed{\alpha \frac{1}{2} \sum_{i=1}^{n} \theta_i^2}$$

Ridge (L2) regularization

x100



ช่วยลดอิทธิพลของ polynomial degree สูงๆ อย่าง x^4, x^5 = ลด overfitting

Logistic Regression

How to apply?

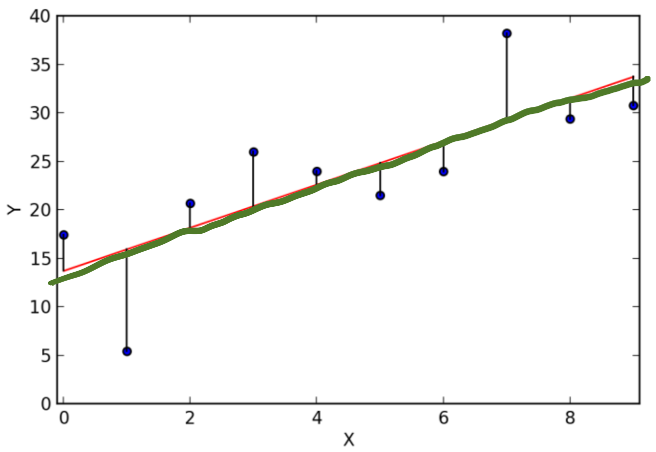
Linear Regression ➔ (Binary) Classification

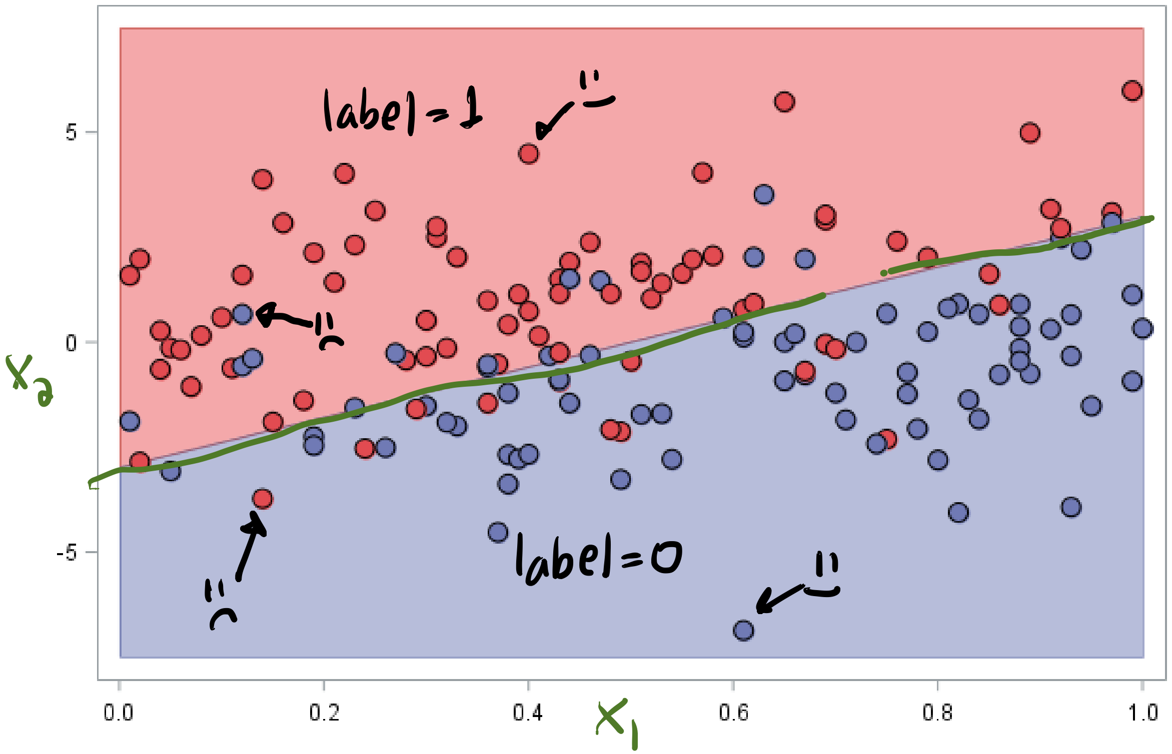
Linear Regression

- Input: $\boldsymbol{x}$ คือ input features
- Output: $\hat{y}$ เป็นจำนวนจริง

โมเดล: $\hat{y} = h_\theta(x) = \boldsymbol{\theta}^T \boldsymbol{x}$

- ความสัมพันธ์แบบ linear



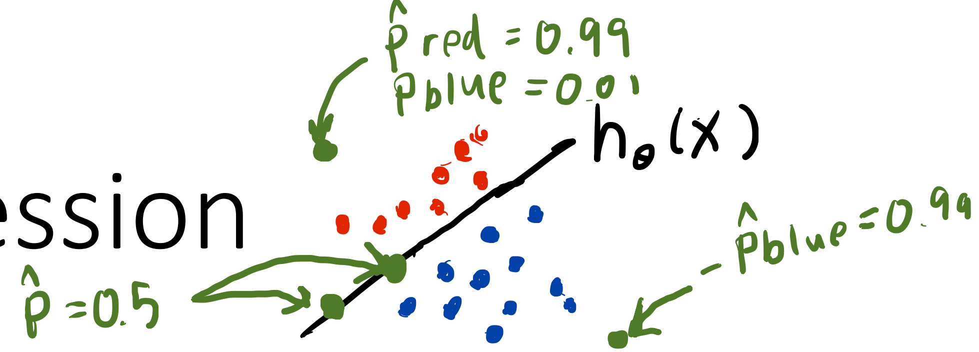


พิจารณาปัญหา binary classification เช่น

- อีเมลเป็น spam (label=1) หรือไม่เป็นสแปม (label=0)
- คนไข้เป็นมะเร็ง (label=1) หรือไม่เป็นมะเร็ง (label=0)

* $\hat{p}$ คือค่าเป็นความน่าจะเป็น

Logistic Regression



- **Key Idea:** เราต้องการให้โมเดล $\underline{h_\theta(x)}$ ทำนายค่า
  - $\hat{p}(y = 1|x;\theta)$ **ความน่าจะเป็นที่ label จะเป็น 1** red
  - $\hat{p}(y = 0|x;\theta)$ **ความน่าจะเป็นที่ label จะเป็น 0** blue
- จากนั้นเราสามารถใช้ค่าของ $\hat{p}$ ในการ classify ข้อมูลด้วยกฎง่ายๆ นี้ได้

$$\hat{y} = \begin{cases} 0 & \text{if } \hat{p} < 0.5, \\ 1 & \text{if } \hat{p} \geq 0.5. \end{cases}$$

classification output

เช่น หากพบว่า

$$\boxed{\begin{array}{l} \hat{p}(y = red \mid x) = 0.9 \\ \hat{p}(y = blue \mid x) = 0.1 \end{array}}$$

มั่นใจมาก

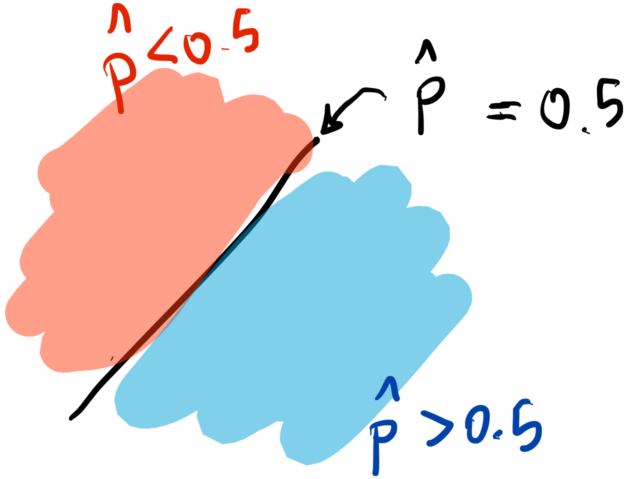
decision $\Downarrow$

$\hat{p}_{red} = 0.9 > 0.5$

ดังนั้นจึง classify ให้ x มี label เป็น $\hat{y} = red$

Logistic Regression

- **Key Idea:** เราต้องการให้โมเดล $h_\theta(x)$ ทำนายค่า
  - $\hat{p}(y = 1|x; \theta)$ **ความน่าจะเป็นที่ label จะเป็น 1**
  - $\hat{p}(y = 0|x; \theta)$ **ความน่าจะเป็นที่ label จะเป็น 0**



- Q: ใช้ $h_\theta(x)$ เดียวกันกับ Linear Regression เลยได้ไหม?

$$\hat{p}(y = 1|x; \theta) = h_\theta(x) = \underbrace{\boldsymbol{\theta}^T \boldsymbol{x}}$$

Linear Regression
ok?

Logistic Regression

- **Key Idea:** เราต้องการให้โมเดล $h_\theta(x)$ ทำนายค่า
  - $\hat{p}(y = 1|x;\theta)$ **ความน่าจะเป็นที่ label จะเป็น 1**
  - $\hat{p}(y = 0|x;\theta)$ **ความน่าจะเป็นที่ label จะเป็น 0**

- Q: ใช้ $h_\theta(x)$ เดียวกันกับ Linear Regression เลยได้ไหม?

$$\hat{p}(y = 1|x;\theta) = h_\theta(x) = \boldsymbol{\theta}^T\boldsymbol{x}$$

- **ปัญหา:** <u>ความน่าจะเป็น</u>ต้องมีค่าในช่วง 0-1 เท่านั้น ในขณะที่ $\boldsymbol{\theta}^T\boldsymbol{x}$ มีค่าเป็นเท่าใดก็ได้ ☹

Sigmoid Function $\sigma(x)$

ค่าเยอะๆ บีบ ค่าในช่วง

$-100,000$ – $100,000$ $\rightarrow \boxed{\sigma} \rightarrow$ $0-1$

- Solution: เปลี่ยน model เป็น

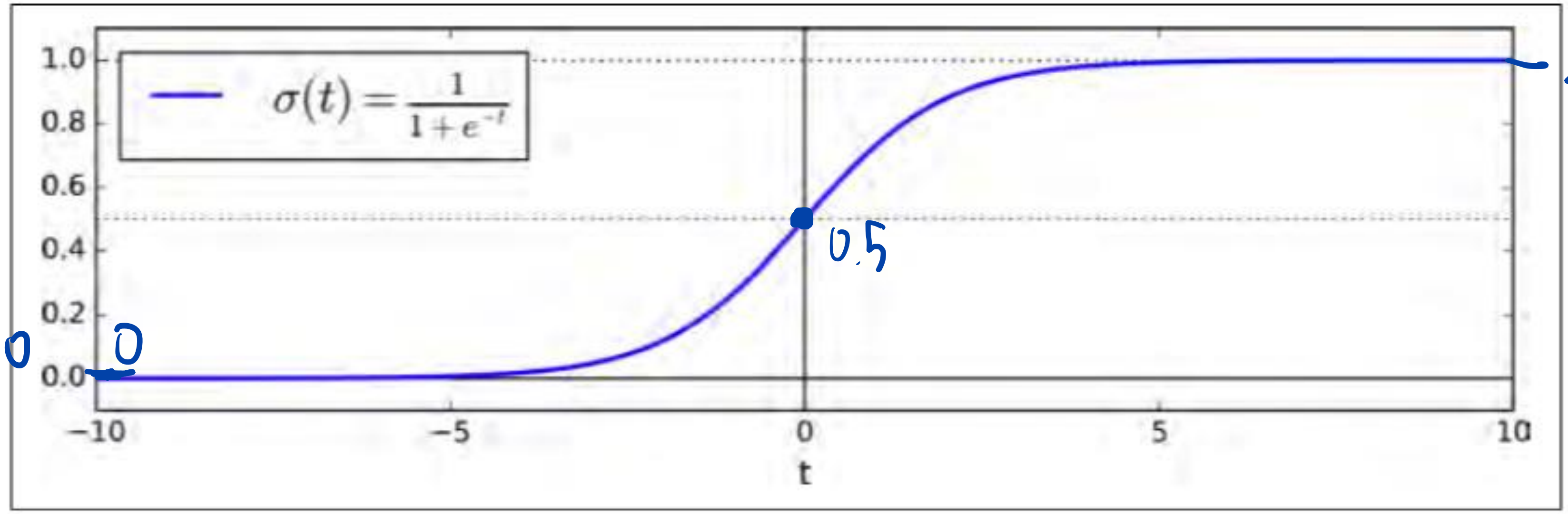
$$\hat{p} = h_\theta(x) = \sigma(\theta^T x)$$

- โดยที่ $\sigma(t)$ คือ logistic (sigmoid) function

$$\sigma(t) = \frac{1}{1 + \exp(-t)}$$

$$\frac{1}{1+e^0} = \frac{1}{2}$$

$$\frac{1}{1+e^{-\infty}} = \frac{1}{1+0} = 1$$



$$\frac{1}{1+e^{-(-\infty)}} = \frac{1}{e^\infty} = 0$$

0.5

1

0

binary
โจทย์ classification $\hat{y} = 0, 1$

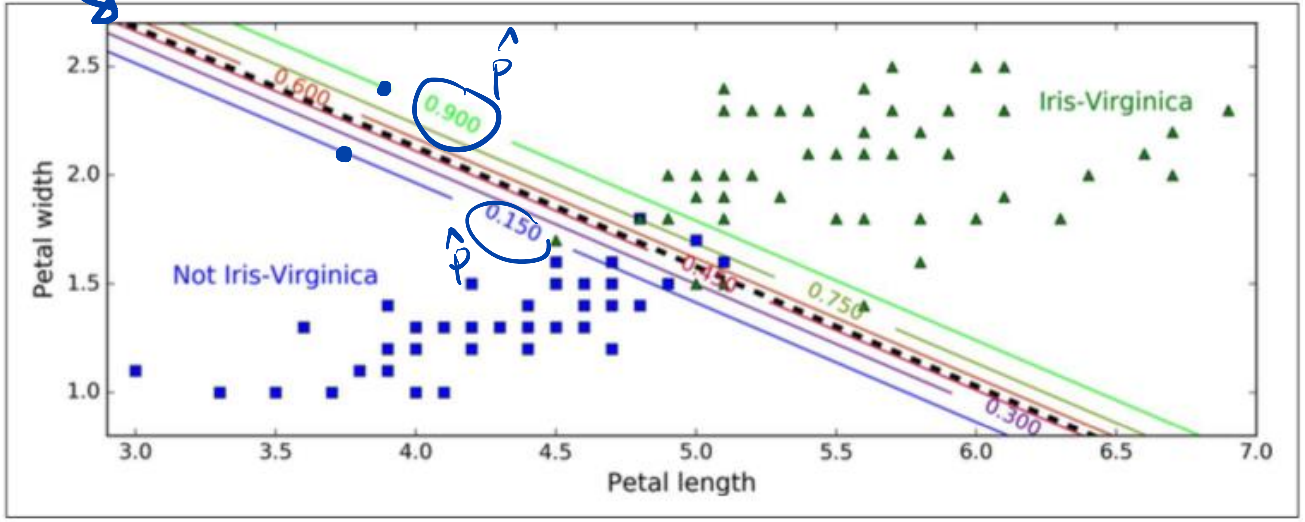
Logistic Regression Model

- โมเดล: $\hat{p}(y = 1|x; \theta) = h_\theta\ (x) = \sigma(\theta^T x)$ เช่น $\hat{p} = 0.9$
- กฎการตัดสินใจ decision rule

$$\hat{y} = \begin{cases} 0 & \text{if } \hat{p} < 0.5, \\ 1 & \text{if } \hat{p} \geq 0.5. \end{cases}$$

- โดยที่
  - Input: $\boldsymbol{x}$ คือ input features
  - Output: $\hat{y}$ เป็น binary (0/1)

$\hat{p} = 0.5$



ตัวอย่าง decision boundary จากการใช้วิธี logistic regression

ใช้ MSE ไม่ได้ $C_{เดิม}(\theta) = (h_\theta(x) - y)^2$ เพราะใช้กับงาน regression เท่านั้น

Cost Function

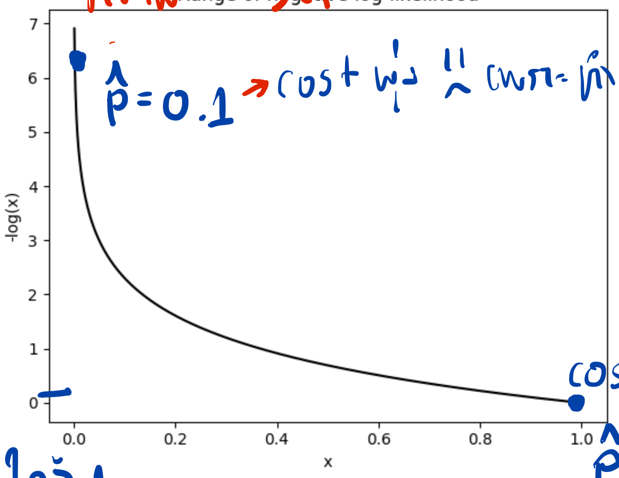
บ่งบอกถึงความแย่ของ θ (โมเดล)

- cost function จะต้องเปลี่ยนไปด้วย
- cost สำหรับแต่ละ instance ข้อมูลคือ

label เฉลย 1 → $\hat{p}$ ก็ควรใกล้ 1

$$c(\theta) = \begin{cases} -\log(\hat{p}) & \text{if } y = 1, \\ -\log(1-\hat{p}) & \text{if } y = 0. \end{cases}$$

กราฟ $-\log(\hat{p})$ สำหรับเฉลย $y = 1$



$\hat{p}=0.1$ → cost พุ่ง !! เพราะทำนายไม่ตรงเฉลย

cost ต่ำ :) เพราะทำนายถูก

$\hat{p}=1$ ตรงเฉลย $y=1$

ไอเดียคือ ถ้าเฉลยเป็นมะเร็ง (y=1)

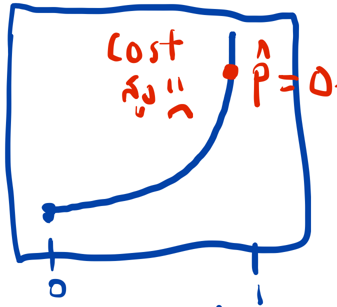
แต่ $\hat{p}$ ทำนายว่าใกล้ 0 ค่า cost = -log(~0) จะใหญ่มาก

ถ้า $\hat{p}$ ทำนายว่าใกล้ 1 ค่า cost จะเป็น 0

สูตรนี้ ทำให้ cost สูง เวลาทำนาย $\hat{p}$ ได้ไม่ดี เช่น เฉลย $y=0$ แต่ดันให้ $\hat{p}$ เป็น 0.97

- cost สำหรับทั้งชุดข้อมูลจึงเป็นดังสมการ

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(\hat{p}^{(i)}\right) + \left(1-y^{(i)}\right)\log\left(1-\hat{p}^{(i)}\right)\right]$$



$-\log(1-\hat{p})$ for $y=0$

เขียนในบรรทัดเดียวได้ $c(\theta) = -y\log\hat{p} - (1-y)\log(1-\hat{p})$

How to train?

- จะ train โมเดลนี้อย่างไร?   (minimize cost อย่างไร?)

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}log\left(\hat{p}^{(i)}\right) + \left(1 - y^{(i)}\right)log\left(1 - \hat{p}^{(i)}\right)\right]$$

- ไม่มี closed-form solution แบบ Normal Eq ☹

How to train?

- จะ train โมเดลนี้อย่างไร? (minimize cost อย่างไร?)

$\sigma(\theta^T x)$

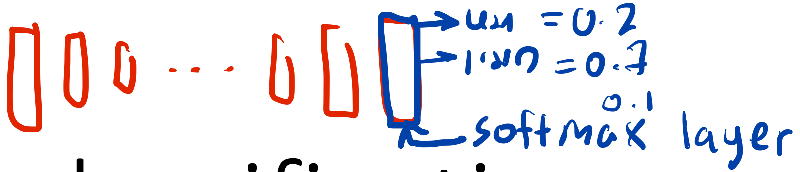
$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}log\left(\hat{p}^{(i)}\right) + \left(1 - y^{(i)}\right)log\left(1 - \hat{p}^{(i)}\right)\right]$$

- ไม่มี closed-form solution แบบ Normal Eq ☹
- แต่สามารถหา gradient/partial derivative ได้ ☺

$$\frac{\partial}{\partial \theta_j}J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(\sigma\left(\theta^T \cdot \mathbf{x}^{(i)}\right) - y^{(i)}\right)x_j^{(i)} \Longrightarrow \nabla_\theta J(\theta)$$

- ดังนั้น ใช้วิธี Stochastic Gradient Descent แบบเดิมได้

update rule $\theta_{new} = \theta - \eta \nabla_\theta J(\theta)$



From Binary to Multi-class classification

- ข้อจำกัดของ Logistic Regression คือ สามารถใช้กับข้อมูลที่มี label เพียง 2 คลาสเท่านั้น
- หากต้องการ generalize ไปใช้กับ multi-class สามารถเปลี่ยนไปใช้ $\boxed{\text{softmax}}$ function

เดิม $\hat{p} = \sigma(\theta^T x)$ ⇒ ใหม่

$$\underbrace{\hat{p}_k}_{\text{คลาสที่ k}} = \sigma(\mathbf{s}(\mathbf{x}))_k = \frac{\exp\left(s_k(\mathbf{x})\right)}{\underbrace{\sum_{j=1}^{K}}_{\text{ทุกคลาส}} \exp\left(s_j(\mathbf{x})\right)}$$

เช่น

$$\hat{p}_{หมา} = \frac{e^{s_{หมา}}}{e^{s_{หมา}} + e^{s_{แมว}} + e^{s_{นก}}}$$

$$\hat{p}_{แมว} = \frac{e^{s_{แมว}}}{\cdots}$$

$$\hat{p}_{นก} = \frac{e^{s_{นก}}}{\cdots}$$

- K is the number of classes.
- $\mathbf{s}(\mathbf{x})$ is a vector containing the scores of each class for the instance $\mathbf{x}$.
- $\sigma(\mathbf{s}(\mathbf{x}))_k$ is the estimated probability that the instance $\mathbf{x}$ belongs to class k given the scores of each class for that instance.

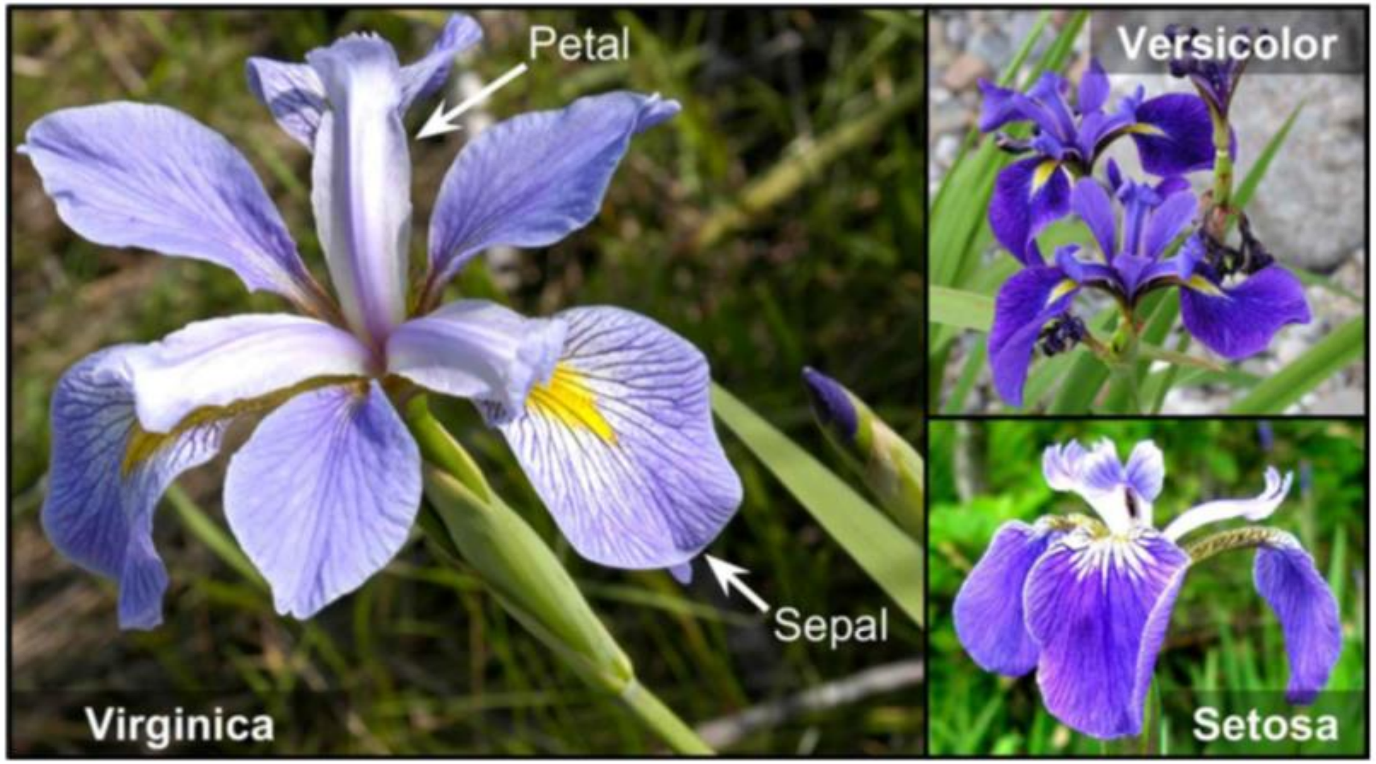
- และ cost function จะเปลี่ยนเป็น Cross-Entropy ซึ่งจะเหมือน LR cost หาก k=2

2 คลาส = binary

$$J(\Theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log\left(\hat{p}_k^{(i)}\right)$$

Lab: ใช้ Logistic Regression จำแนกพันธุ์ดอกกล้วยไม้

```
from sklearn.linear_model import LogisticRegression
```



เป้าหมาย: จำแนก Iris-Virginica ออกจากชนิดอื่น โดยใช้ขนาดของกลีบ Sepal/Petal - width/length

Next week

- Support Vector Machine (SVM)