Anomaly detection (異常偵測)

Problem motivation

Anomaly detection example I

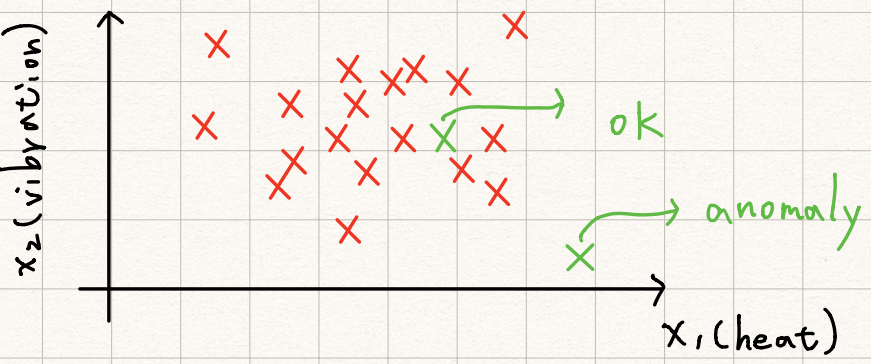
Aircraft engine features

x_1 = heat generated

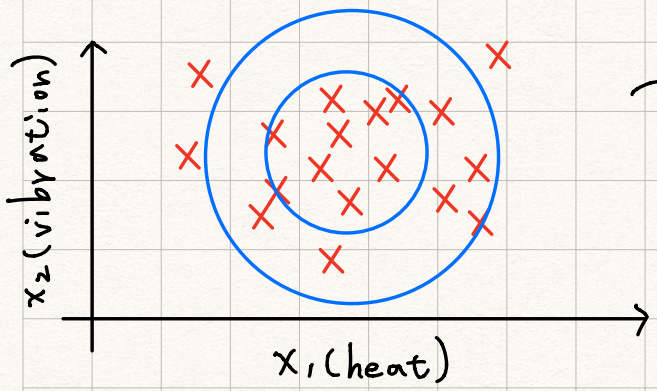
x_2 = vibration intensity

Dataset: $\{x^{(1)}, x^{(2)}, \cdots, x^{(m)}\}$

New engine: x_{test}



Density estimation



→ 建立 model $p(x)$

$p(x_{test}) < \varepsilon \rightarrow$ flag anomaly

$p(x_{test}) \geq \varepsilon \rightarrow$ OK

Anomaly detection example II

- Fraud detection:
  $x^{(i)}$ = 使用者的活動特徵
  Model $p(x)$ from data
  Identify unusual users by checking which have $p(x) < \varepsilon$

- Manufacturing

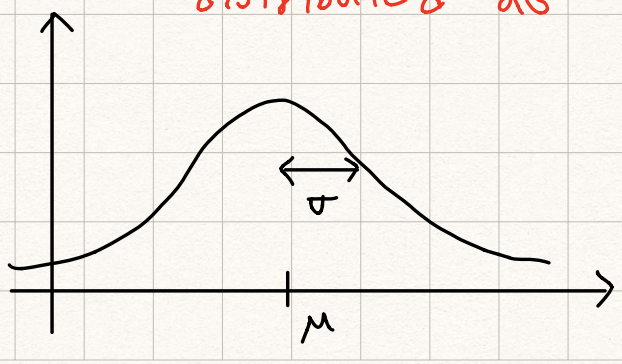
- Monitoring computers in a data center

Gaussian distribution

Say $x \in \mathbb{R}$. If x is a distributed Gaussian with mean μ, variance σ^2 (standard deviation σ)

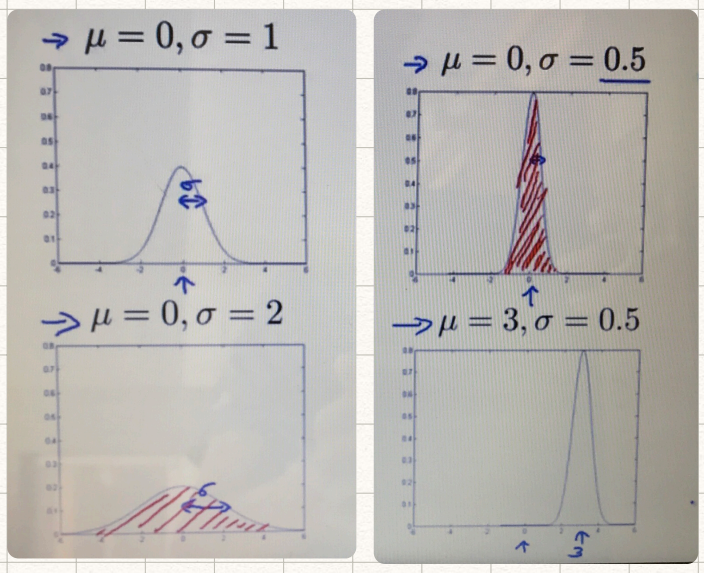
$x \sim N(\mu, \sigma^2)$

↑ "distributed as"



$$p(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\cdot\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Gaussian distribution example



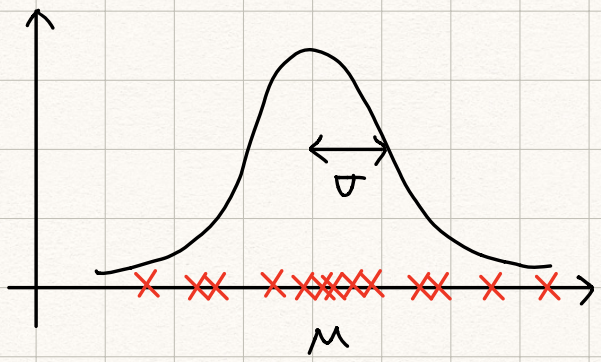
σ 越小，
$\wedge$ 越細(瘦)

越大
則越粗(胖)

Parameter estimation

Dataset: $\{x^{(1)}, x^{(2)}, \cdots x^{(m)}\}$

$x^{(i)} \in \mathbb{R}$.



$x^{(i)} \sim N(\mu, \sigma^2)$

$$\mu = \frac{1}{m} \sum_{i=1}^{m} x^{(i)}$$

$$\sigma^2 = \frac{1}{m} \sum_{i=1}^{m} (x^{(i)} - \mu)^2$$

$m-1$ $\quad \frac{1}{m-1}$?

(ML中，較常用 $\frac{1}{m}$)

Algorithm

Density estimation

Training set : $\{x^{(1)}, \dots x^{(m)}\}$, $x \in R^n$

$$p(x) = p(x_1; \mu_1, \sigma_1^2) \times p(x_2; \mu_2, \sigma_2^2) \times \cdots \times p(x_n; \mu_n, \sigma_n^2)$$

$$= \prod_{j=1}^{n} p(x_j; \mu_j, \sigma_j^2)$$

各機率 $p(x_1) \sim p(x_n)$ 獨立

$\sum_{i=1}^{n} i = 1+2+3+\cdots+n$

$\prod_{i=1}^{n} i = 1 \times 2 \times 3 \times \cdots \times n$

Anomaly detection algorithm

1. 選擇特徵 x_i，且這些特徵為異常例子的指標

2. Fit parameters $\mu_1, \dots, \mu_n$
$\sigma_1^2, \dots, \sigma_n^2$

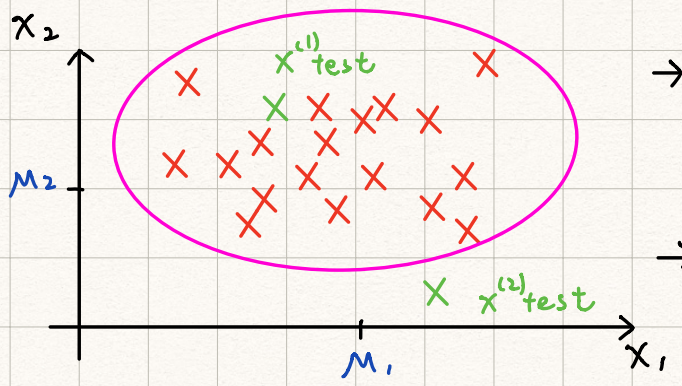
$$\mu_j = \frac{1}{m} \sum_{i=1}^{m} x_j^{(i)} \quad , \quad \sigma_j^2 = \frac{1}{m} \sum_{i=1}^{m} (x_j^{(i)} - \mu_j)^2$$

3. Given new example x, compute $p(x)$

$$p(x) = \prod_{j=1}^{n} p(x_j; \mu_j, \sigma_j^2) = \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi} \cdot \sigma_j} \exp\left(-\frac{(x_j - \mu_j)^2}{2\sigma_j^2}\right)$$

Anomaly if $p(x) < \varepsilon$

Anomaly detection example



→ 找出 μ_1, μ_2 的高斯分布

→ 算出 $p(x)$ 並畫出三維圖形 $(x_1, x_2, p(x))$

$\varepsilon = 0.02$,

$p(x^{(1)}_{test}) = 0.0426 \geq \varepsilon$

$p(x^{(2)}_{test}) = 0.021 < \varepsilon$

Building an Anomaly Detection System

Developing and Evaluating an Anomaly Detection System

The importance of real-number evaluation

when developing a learning algorithm, making decisions is much easier if we have a way of evaluating our learning algorithm.

assume we have some labeled data, of anomalous and non-anomalous examples.

Aircraft engines motivating example

10000 good (normal) engines ⟶ 儘管裡頭有些異常樣本也ok

20 flawed engines (anomalous)

Train: 6000 good engines (unlabeled)

CV: 2000 good engines ($y=0$), 10 anomalous ($y=1$)

Test: 2000 good engines ($y=0$), 10 anomalous ($y=1$)

Algorithm evaluation

- Fit model $p(x)$ on training set
- On a cv / test example x, predict

$$y = \begin{cases} 1 & \text{if } p(x) < \varepsilon \text{ (anomaly)} \\ 0 & \text{if } p(x) \geq \varepsilon \text{ (normal)} \end{cases}$$

- Possible evaluation metrics:

Precision / Recall / F_1-score

挑選不同的 ε，並在驗証集上確認哪個 ε 有最好的 ε

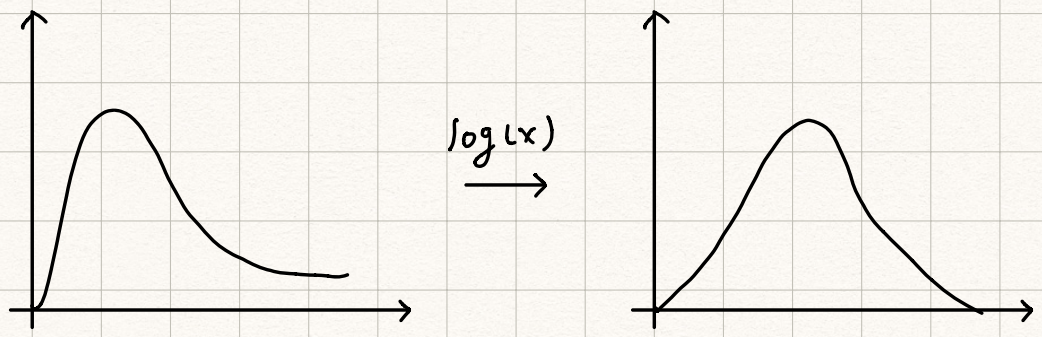
Anomaly Detection v.s. Suppervised learning

Anomaly detection v.s. Suppervised learning

• 十分少的正樣本數量 (y=1) (0-20 is common) • 大數量的負樣本數量 (y=0) • 存在許多不同種類負的異常狀況，未來遇到的異常可能是以前從沒見過的	• 大數量的正負樣本 • 足夠數量的正樣本讓算法了解正樣本長得如何
Fraud detection Manufacturing Monitoring machines in a data center	Email spam classification Weather prediction Cancer classification

Choosing what features to use

Non-gaussian features



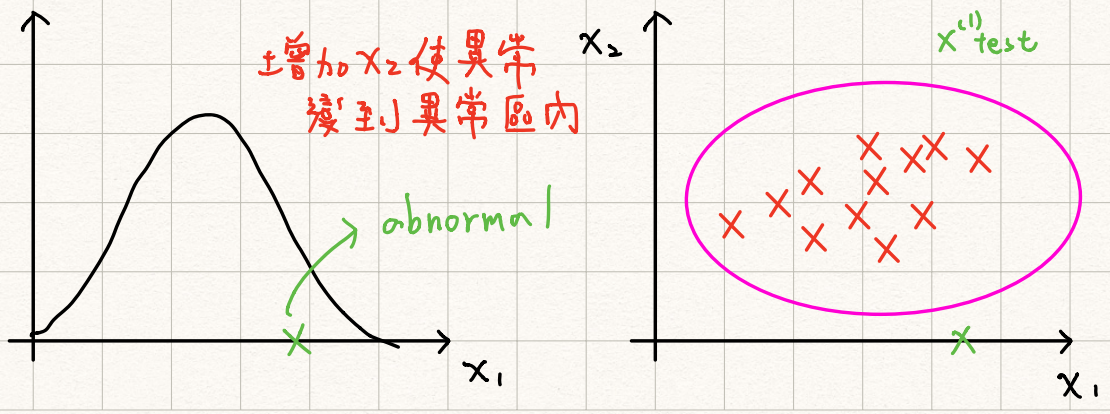
如果某特徵分佈得不像高斯函數，可對其取 $\log(x)$, $\log(x+c)$, x^c

調整c值→hist→觀察

使特徵為高斯分佈故可讓算法工作良好

Error analysis for anomaly detection

Want $p(x)$ large for normal examples x
$p(x)$ small for anomalous examples x



Monitoring computers in a data center

Choosing features that might take on unusually large or small values in the event of an anomaly.

x_1 = memory use of computer
x_2 = number of disk accesses / sec
x_3 = CPU load
x_4 = network traffic

電腦陷入死迴循，x_3 大 but x_4 小，故設計 x_5、x_6 feature

x_5 = CPU / network traffic
x_6 = (CPU load)^2 / network traffic .....

Recommender Systems

reason: 許多矽谷工程師最想改進的便是推薦系統
(Netflex, Amazon, 電商都有在用)

在學術界，推薦系統的討論卻滿少的

Predicting Movie Ratings

Problem Formulation

Example: Predicting movie rating

User rates movies using zero to five stars

Movie	People 1	2	3	4	x_1	x_2
movie 1	5	5	0	0	0.9	0
2	5	?	?	0	1	0.01
3	?	4	0	?	0.99	0
4	0	0	5	4	0.1	1.0
5	0	0	5	?	0	0.9

根據上述 data，預測 ? 處是多少

Content Based Recommendations

設定特徵 x_1 (romance) . x_2 (action)

$x_0 = 1$

$x^{(1)} = \begin{bmatrix} 1 \\ 0.9 \\ 0 \end{bmatrix}$

0.9	0
1.0	0.01
0.99	0
0.1	1.0
0	0.9

For each user j, learn a parameter $\theta^{(j)} \in \mathbb{R}^3$.
Predict user j as rating movie i with $(\theta^{(j)})^T x^{(i)}$ stars.

$x^{(3)} = \begin{bmatrix} 1 \\ 0.99 \\ 0 \end{bmatrix} \leftrightarrow \theta^{(1)} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}$, $(\theta^{(1)})^T x^{(3)} = 4.95$

對每個用戶應用不同的線性迴歸模型
→ 每個人有自己有一個 θ

Problem formulation

- n_u = no. users
- n_m = no. movies
- $m^{(j)}$ = no. of movies rated by user j
- $r(i,j) = 1$, if user j has rated movie i
- $y(i,j)$ = rating given by user j to movie i (defined only if $r(i,j) = 1$)

- $\theta^{(j)}$ = parameter vector for user j
- $x^{(i)}$ = feature vector for movie i

For user j, movie i, predicted rating: $(\theta^{(j)})^T (x^{(i)})$

Optimization objective

To learn $\theta^{(j)}$. $\theta^{(j)} \in \mathbb{R}^{n+1}$:

$$\min_{\theta^{(j)}} \frac{1}{2m^{(j)}} \sum_{i:r(i,j)=1} \left((\theta^{(j)})^T (x^{(i)}) - y^{(i,j)} \right)^2 + \frac{\lambda}{2m^{(j)}} \sum_{k=1}^{n} (\theta_k^{(j)})^2$$

To learn $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(n_u)}$: $J(\theta^{(1)} + \ldots + \theta^{(n_u)})$

$$\min_{\theta^{(1)}, \ldots \theta^{(n_u)}} \frac{1}{2} \sum_{j=1}^{n_u} \sum_{i:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} (\theta_k^{(j)})^2$$

Gradient descent update

$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha \sum_{i:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right) x_k^{(i)} \quad \text{for } k = 0$$

$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha \left(\sum_{i:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right) x_k^{(i)} + \lambda \theta_k^{(j)} \right) \quad \text{for } k \neq 0$$

如果對 x_1, x_2 特徵又沒有掌握怎麼辦？

Collaborative Filtering (協同過濾)

能自動替人學習到優良的特徵

Problem motivation

x_1 (romance)	x_2 (action)	$x_0 = 1$
?	?	
?	?	
?	?	採訪 user，問他們喜歡 romance or action 的程度，藉此得到 θ 值
?	?	
?	?	

user 1 ~ 4

$$\theta^{(1)} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix} \quad \theta^{(2)} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix} \quad \theta^{(3)} = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix} \quad \theta^{(4)} = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix}$$

由 θ 和 user 對產品的評分

推測該產品的特徵（如 movie 屬 x_1 v x_2 的程度）

$$(\theta^{(1)})^T x^{(1)} \simeq 5 \quad , \quad (\theta^{(2)})^T x^{(1)} \simeq 5$$
$$(\theta^{(3)})^T x^{(1)} \simeq 0 \quad , \quad (\theta^{(4)})^T x^{(1)} \simeq 0$$

5. 5. 0. 0. 為評分

Optimization algorithm

Given $\theta^{(1)}, \dots, \theta^{(n_u)}$, to learn $x^{(i)}$:

$$\min_{x^{(i)}} \frac{1}{2} \sum_{j:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{k=1}^{n} \left(x_k^{(i)} \right)^2$$

Given $\theta^{(1)}, \dots, \theta^{(n_u)}$, to learn $x^{(1)}, \dots, x^{(n_m)}$:

$$\min_{x^{(1)}, \dots, x^{(n_m)}} \frac{1}{2} \sum_{i=1}^{n_m} \sum_{j:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^{n} \left(x_k^{(i)} \right)^2$$

Collaborative filtering

Given $x^{(1)}, \dots x^{(n_m)}$ (and movie ratings)
can estimate $\theta^{(1)}, \dots, \theta^{(n_u)}$

Given $\theta^{(1)}, \dots, \theta^{(n_u)}$
can estimate $x^{(1)}, \dots, x^{(n_m)}$

Guess $\theta \to x \to \theta \to x \to \theta \to x \to \dots$

Collaborative Filtering Algorithm

Given $x^{(1)} \sim x^{(n_m)}$, estimate $\theta^{(1)} \sim \theta^{(n_u)}$

$$\min_{\theta^{(1)},\dots\theta^{(n_u)}} \frac{1}{2}\sum_{j=1}^{n_u}\sum_{i:r(i,j)=1}\left((\theta^{(j)})^T x^{(i)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}\left(\theta_k^{(j)}\right)^2$$

Given $\theta^{(1)} \sim \theta^{(n_u)}$, estimate $x^{(1)} \sim x^{(n_m)}$

$$\min_{x^{(1)},\dots,x^{(n_m)}} \frac{1}{2}\sum_{i=1}^{n_m}\sum_{j:r(i,j)=1}\left((\theta^{(j)})^T x^{(i)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2}\sum_{i=1}^{n_m}\sum_{k=1}^{n}\left(x_k^{(i)}\right)^2$$

合併以上兩種算法，使不需 $x \to \theta \to x \to \theta \dots$

minimizing $x^{(1)}, \dots, x^{(n_m)}$ and $\theta^{(1)}, \dots, \theta^{(n_u)}$ simultaneously

$$J(x^{(1)},\dots x^{(n_m)}, \theta^{(1)},\dots,\theta^{(n_u)}) = \frac{1}{2}\sum_{(i,j):r(i,j)=1}\left((\theta^{(j)})^T x^{(i)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2}\sum_{i=1}^{n_m}\sum_{k=1}^{n}\left(x_k^{(i)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}\left(\theta_k^{(j)}\right)^2$$

$$\min_{\substack{x^{(1)}\dots x^{(n_m)} \\ \theta^{(1)}\dots\theta^{(n_u)}}} J\left(x^{(1)},\dots,x^{(n_m)},\theta^{(1)},\dots\theta^{(n_u)}\right)$$

$x \in \mathbb{R}^n$, $\theta \in \mathbb{R}^n$ ~~$x_0 = 1$~~

Collaborative filtering algorithm

1. Initialize $x^{(1)}, \dots, x^{(n_m)}, \theta^{(1)}, \dots, \theta^{(n_u)}$ to small random values

2. Minimize $J(x^{(1)}, \dots, x^{(n_m)}, \theta^{(1)}, \dots, \theta^{(n_u)})$ using gradient descent (or an advanced optimization algorithm). E.g. for every $j = 1, \dots, n_u$, $i = 1, \dots, n_m$:

$$x_k^{(i)} := x_k^{(i)} - \alpha \left(\sum_{j : r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)}) \theta_k^{(j)} + \lambda x_k^{(i)} \right)$$

$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha \left(\sum_{i : r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)}) x_k^{(i)} + \lambda \theta_k^{(j)} \right)$$

3. For a user with parameters θ and a movie with (learned) feature x, predict a star rating of $\theta^T x$.

Low Rank Metrix Factorization

Vectorization

$n_m = 5$, $n_u = 4$ $\longrightarrow$

$$Y = \begin{pmatrix} 5 & 5 & 0 & 0 \\ 5 & ? & ? & 0 \\ ? & 4 & 0 & ? \\ 0 & 0 & 5 & 4 \\ 0 & 0 & 5 & 0 \end{pmatrix}$$

$$X = \begin{bmatrix} \cdots (x^{(1)})^T \cdots \\ \cdots (x^{(2)})^T \cdots \\ \vdots \\ \cdots (x^{(n_m)})^T \cdots \end{bmatrix} \qquad \Theta = \begin{bmatrix} \cdots (\theta^{(1)})^T \cdots \\ \cdots (\theta^{(2)})^T \cdots \\ \vdots \\ \cdots (\theta^{(n_u)})^T \cdots \end{bmatrix}$$

Predicted ratings: $X\Theta^T$ ※

- 將 X. Θ. Y 向量化

Finding related movies

For each product i, we learn a feature vector $x^{(i)} \in \mathbb{R}^n$.

x_1 = romance, x_2 = action, x_3 = comedy, x_4 = ...

How to find movie j related to movie i?

small $||x^{(i)} - x^{(j)}||$ → movie j and i are similar

根據用戶看過電影 i 的特徵 x，來找尋是否有其他雷同的 x

5 most similar movies to movie i:

Find the 5 movies j with smallest $||x^{(i)} - x^{(j)}||$

Implementational Detail: Mean Normalization

能讓演算法運作的更好

$$Y = \begin{bmatrix} 5 & 5 & 0 & 0 & ? \\ 5 & ? & ? & 0 & ? \\ ? & 4 & 0 & ? & ? \\ 0 & 0 & 5 & 4 & ? \\ 0 & 0 & 5 & 0 & ? \end{bmatrix} \quad \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix}$$

見此行，根據J( )公式，由於不存在任何 $r(i,j)=1$，故只剩下黑色框起來的這項，

$$J = \frac{\lambda}{2}\left[(\theta_1^{(5)})^2 + (\theta_2^{(5)})^2\right]$$

$$\theta^{(5)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$J(x^{(1)}, \dots x^{(n_m)}, \theta^{(1)}, \dots, \theta^{(n_u)}) = \frac{1}{2} \sum_{(i,j):r(i,j)=1} ((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^{n} (x_k^{(i)})^2 + \boxed{\frac{\lambda}{2} \sum_{j=1}^{n_u} \sum_{k=1}^{n} (\theta_k^{(j)})^2}$$

Mean Normalization

$$\mu_Y = \begin{bmatrix} 2.5 \\ 2.5 \\ 2 \\ 2.5 \\ 1.25 \end{bmatrix} \rightarrow Y - \mu = \begin{bmatrix} 2.5 & 2.5 & -2.5 & -2.5 & ? \\ 2.5 & ? & ? & -2.5 & ? \\ ? & 2 & -2 & ? & ? \\ -2.5 & -2.25 & 2.75 & 1.75 & ? \\ -2.5 & -1.25 & 3.75 & -1.25 & ? \end{bmatrix}$$

For user j, on movie i predict:

$(\theta^{(j)})^T (x^{(i)}) + \mu_i$

User 5:

$\theta^{(5)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$\underbrace{(\theta^{(5)})^T (x^{(i)})}_{=0} + \boxed{\mu_i}$

$?_i = \mu_i$