

ML Application

Data Mining (Web click data, biology, engineering)

Application can't program by hand

(NLP, CV, handwriting recognition)

Self-customizing programs

ML Learning definition

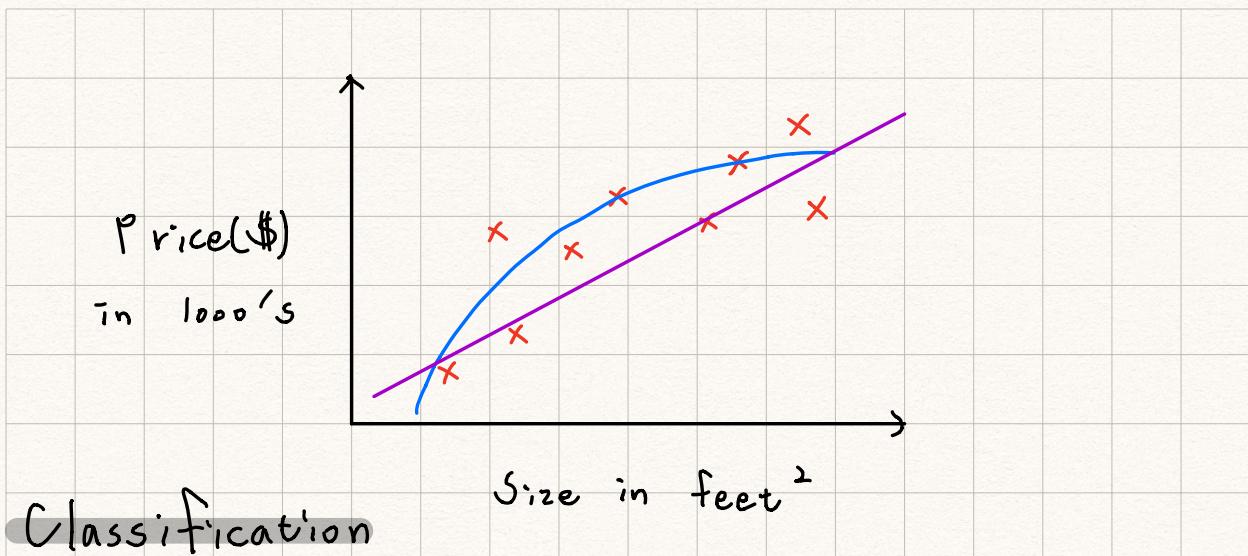
A computer program is said to learn from experience E with respect to some task T and some performance measure P , if its performance on T , as measured by P , improves with experience E .

Supervise Learning

Regression

Predict continuous valued output (price)

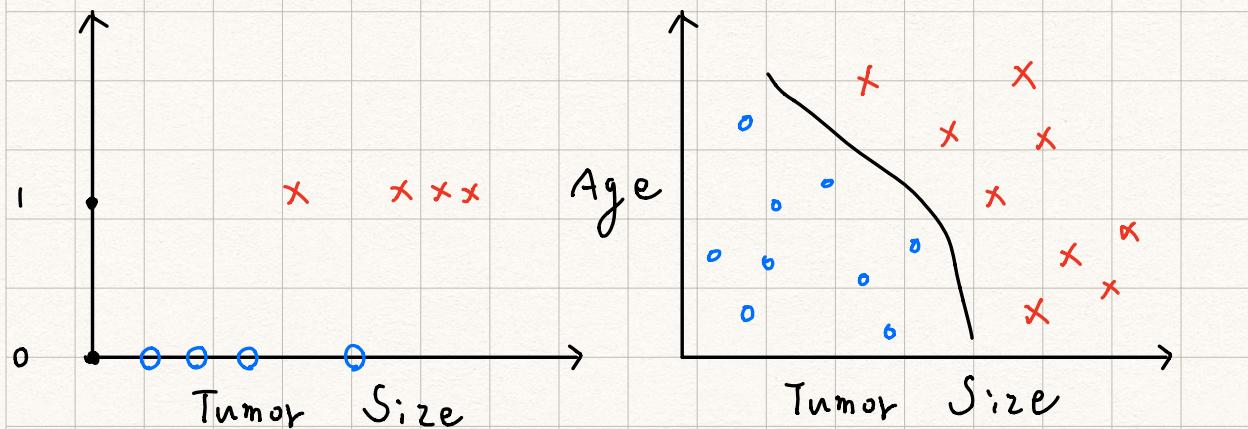
房地產預測



Discrete valued output (0 or 1)

可能不只 0.1 的分类, 0. 1. 2. 3

健康 痘1 痘2 痘3



one feature

two features

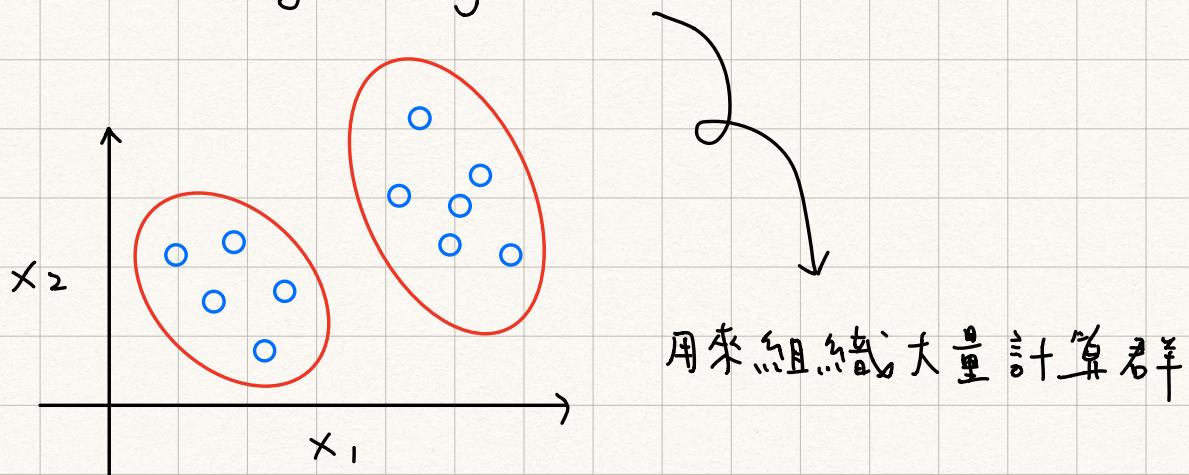
∞ 個 features?

支持向量机

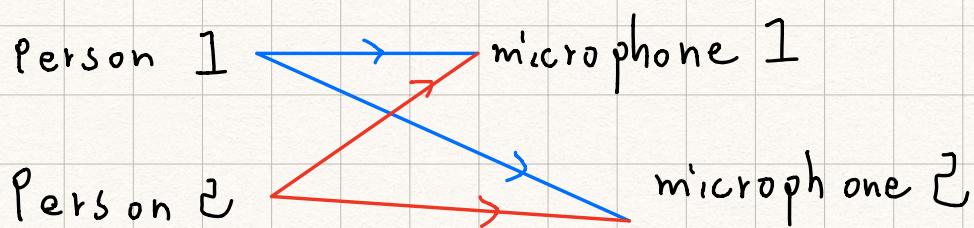
Unsupervised Learning

資料沒有標籤 or 屬性
沒有 Test data

Clustering Algorithm



Cocktail Party Algorithm



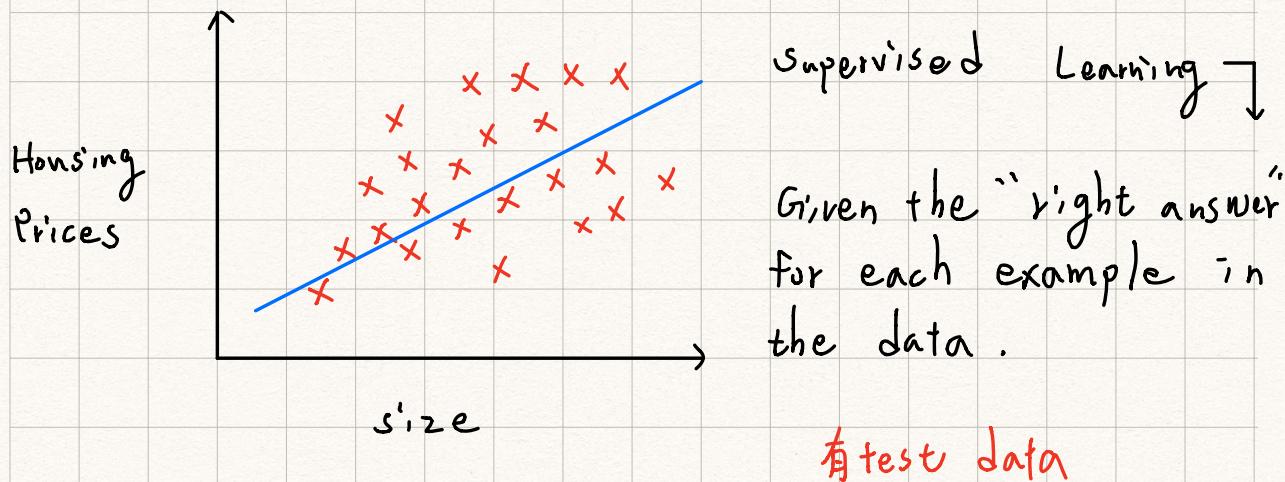
利用算法剔除其中一種聲源

Octave 確認算法可行後，

再到 C++ . Java 環境重新建置

Model Representation

Linear Regression

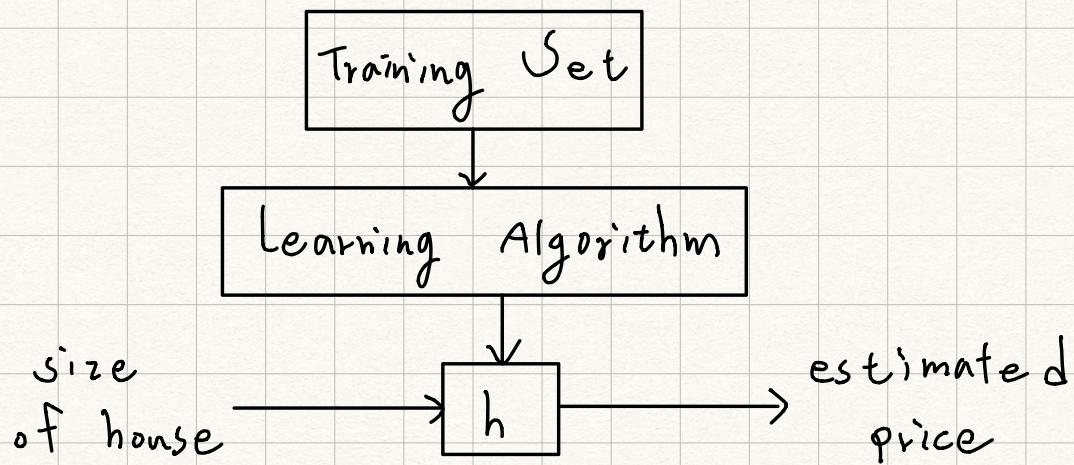


本課的符號說明：

m : Number of training data

x 's : "input" variable / features

y 's : "output" variable / "target" variable



Cost function

$$h(\text{hypothesis}): h_{\theta}(x) = \theta_0 + \theta_1 x$$

θ_0, θ_1 被稱為 模型參數

$$\text{minimize } \theta_0, \theta_1 \rightarrow \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

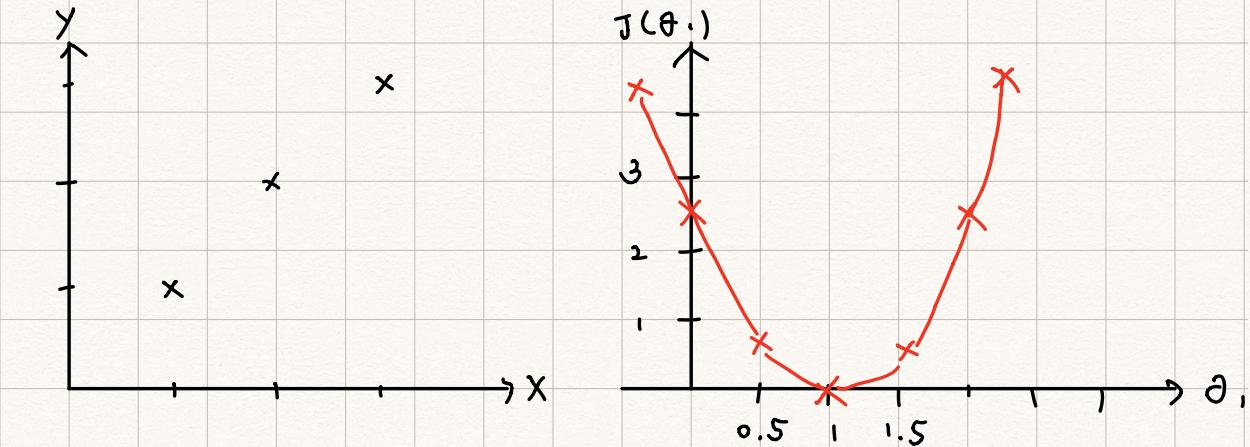
$$h_{\theta}(x^{(i)}) = \theta_0 + \theta_1 x^{(i)}$$

squared error function = $J(\theta_0, \theta_1)$

對於大部分回歸問題皆適用

Figure Comparison I

hypothesis: $h_{\theta}(x)$ $\theta_0 = 0$, cost function
 $\therefore J(\theta_1)$



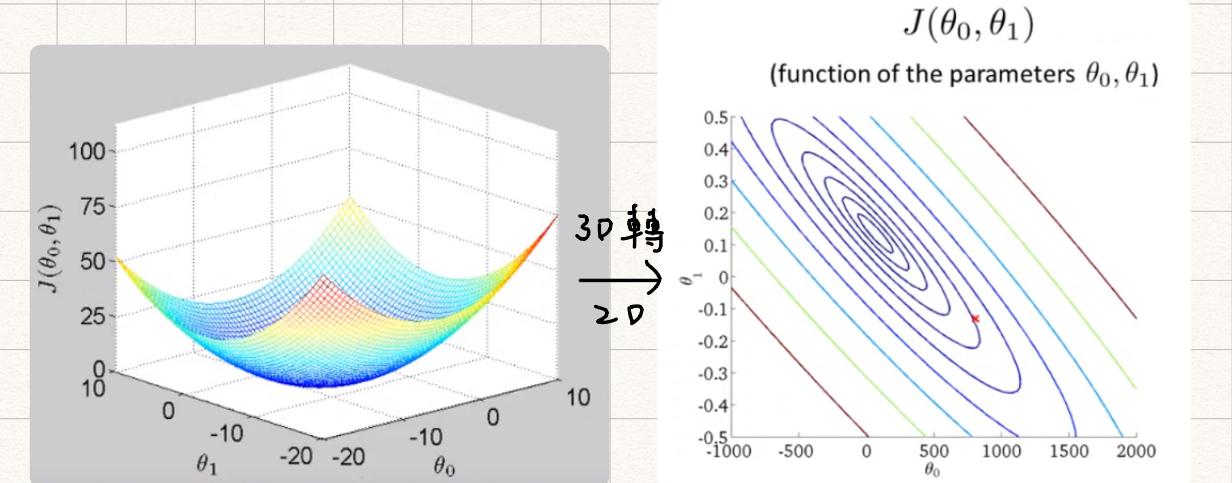
當 θ_1 變更, cost function 值也不一樣

最大小化 $J(\theta)$, 找到最佳點

Figure Comparison II

$\theta_0, \theta_1 \neq 0$,
則根據訓練集樣本，會
長得类似此圖

輪廓圖
(contour plot)



每圈的 cost function 值是
一樣的！

Gradient Descent

$J(\theta_0, \theta_1)$, want $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

$J(\theta_0, \theta_1, \theta_2, \theta_3, \dots, \theta_n)$ 的題也可用此法解

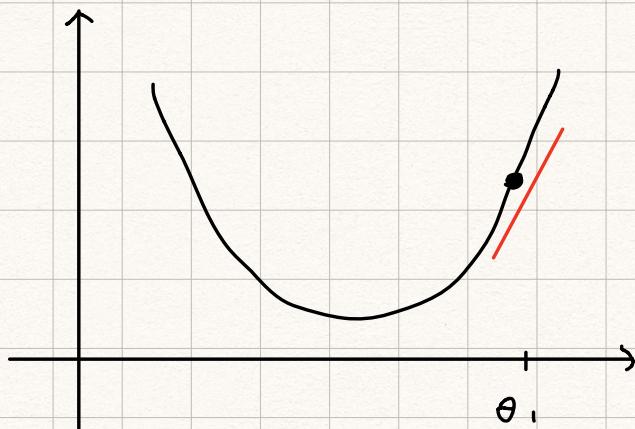
山坡 → 山底

repeat until convergence

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \quad (\text{for } j=0 \text{ and } j=1)$$

assignment learning rate simultaneously update

Intuition



$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$

≥ 0

$$\theta_1 := \theta_1 - \alpha \quad (\text{positive num})$$

α 太大，導致收斂或發散

Gradient descent can converge to a local minimum, even with the learning rate α fixed.

gradient descent can automatically take smaller steps.

因此 α 值不必特意再調整
越近 Local minimum, 斜率越小
每次更新值隨之越小

Gradient Descent For Linear Regression

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \quad (\text{for } j=0 \text{ and } j=1)$$

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

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

$$\begin{aligned} \rightarrow \frac{\partial}{\partial \theta_j} &= \frac{\partial}{\partial \theta_j} \cdot \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 \\ &= \frac{\partial}{\partial \theta_j} \cdot \frac{1}{2m} \sum_{i=1}^m (\underline{\theta_0 + \theta_1 x^{(i)}} - y^{(i)})^2 \end{aligned}$$

$$\theta_0 \quad j=0: \quad \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})$$

$$\theta_1 \quad j=1: \quad \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) \cdot x^{(i)}$$

$$\left\{ \begin{array}{l} \theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) \\ \theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) \cdot x^{(i)} \end{array} \right.$$

*

Linear Regression 的 loss function $\frac{1}{2} \sum (y_i - \hat{y}_i)^2$ 為凸函數

為 Convex Function (凸函數), 無局部極小值

Linear Algebra Review

Matrix : Rectangular array of numbers

$$A = \begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix}_{x \times y} \quad x=3, \quad y=2$$

$A_{11} = a, \quad A_{22} = d.$

Dimension of matrix :

number of rows \times number of columns

Vector : An $n \times 1$ matrix

1-indexed vs 0-indexed

$$y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

$$y = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

程式語言通常是以
1-indexed 為主

Matrix - vector multiplication

practical problem : $h_\theta(x) = -40 + 0.25x$

House sizes : 2104, 1416, 1534, 852

$$\begin{bmatrix} 1 & 2104 \\ 1 & 1416 \\ 1 & 1534 \\ 1 & 852 \end{bmatrix} \times \begin{bmatrix} -40 \\ 0.25 \end{bmatrix} = \begin{bmatrix} -40 \times 1 + 0.25 \times 2104 \\ -40 \times 1 + 0.25 \times 1416 \\ \vdots \\ -40 \times 1 + 0.25 \times 1534 \end{bmatrix}$$

可以矩陣運算特性計算題目，如此一來
便不須使用 loop

Matlab 對 Array Calculation 有優化！

以上是 $h_\theta(x) = -40 + 0.25x$

$h_\theta(x) = 200 + 0.1x$

$h_\theta(x) = -150 + 0.4x$

$$\begin{bmatrix} 1 & 2104 \\ 1 & 1416 \\ 1 & 1534 \\ 1 & 852 \end{bmatrix}_{4 \times 2} \times \begin{bmatrix} -40 & 200 & -150 \\ 0.25 & 0.1 & 0.4 \end{bmatrix}_{2 \times 3} = \begin{bmatrix} \dots \\ \dots \end{bmatrix}_{4 \times 3}$$

$$A \times B \neq B \times A$$

I: identity matrix $\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$

$$A \times I = I \times A$$

Inverse and transpose

$$A(A^{-1}) = A^{-1}A = I \quad A^{-1}: \text{inverse}$$

Matrices that don't have an inverse
are "singular" or "degenerate"

$$A = \begin{bmatrix} 1 & 2 & 0 \\ 3 & 5 & 9 \end{bmatrix} \quad B = A^T = \begin{bmatrix} 1 & 3 \\ 2 & 5 \\ 0 & 9 \end{bmatrix}$$

$$B_{ij} = A_{ji}$$