

多媒體技術與應用

Spring 2021

Instructor : Yen-Lin Chen(陳彥霖), Ph.D.

Professor

Dept. Computer Science and Information Engineering

National Taipei University of Technology



Lecture 4

機器學習概論



人工智慧與機器學習概述

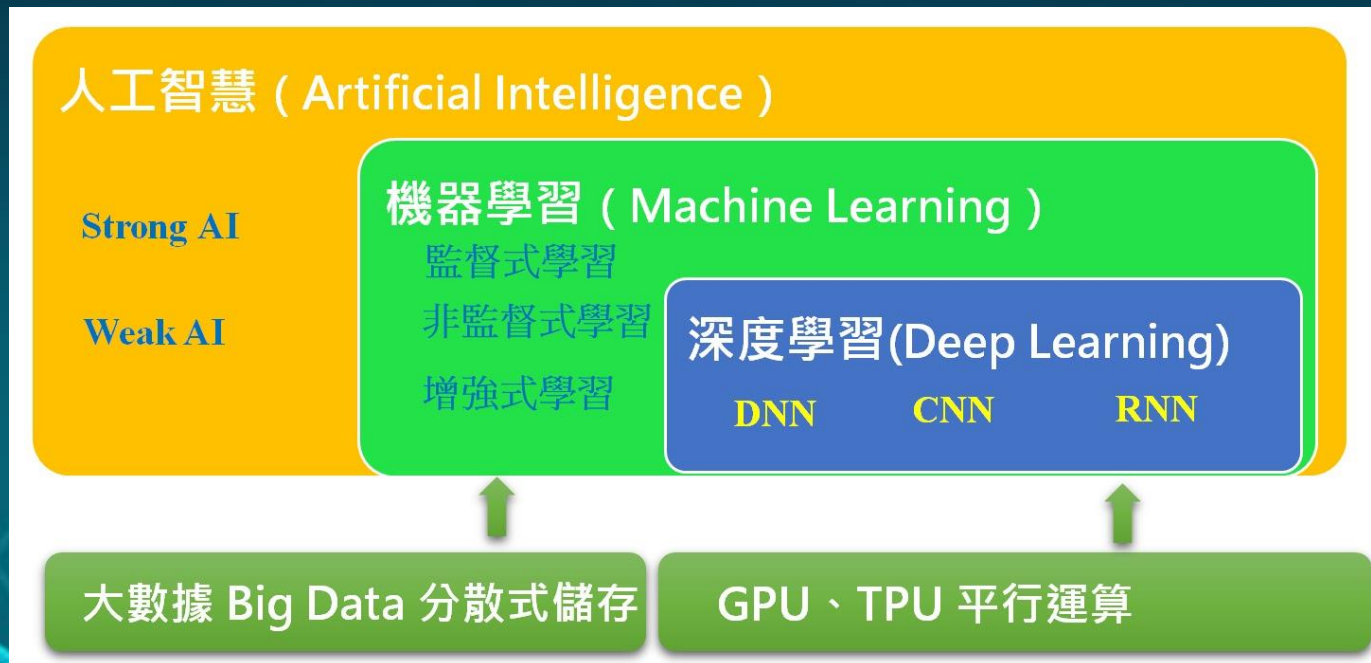


人工智慧(Artificial Intelligence)與 機器學習(Machine Learning)概述

- 人工智慧(Artificial Intelligence, AI) 顧名思義就是要如何有智慧。簡單來說人工智慧主要是在研究，如何使用電腦的功能，來做一些本來必須由人類執行才能完成的工作；簡單的說，就是藉由電腦來執行人類智慧的過程，機器在經過程式設計之後，能表現出與人類類似的智慧。

人工智慧(Artificial Intelligence)與機器學習(Machine Learning)概述

- 在人工智慧的領域中，又可以向下細分為機器學習(Machine Learning, ML)、深度學習(Deep Learning, DL)等。
- 機器學習領域中，常用來將大數據資料進行預測分析，而深度學習則常用來進行影像辨識。





人工智慧(Artificial Intelligence)-實例

- 2016年，Google DeepMind 的 AlphaGo 圍棋軟體與世界棋王李世乜對弈，五局對戰以4:1戰勝李世乜，一戰成名，引起國際間對於人工智慧的廣泛討論。當時各家媒體廣泛使用「人工智慧」、「機器學習」與「深度學習」這幾個名詞來描述 DeepMind 的致勝關鍵，可見人工智慧、機器學習、深度學習等議題的火熱程度。



機器學習簡介與應用



機器學習(Machine Learning)-簡介

- 機器學習(Machine Learning, ML)是透過演算法將收集到的資料進行分類或預測模型訓練，在未來中，當得到新的資料時，可以透過訓練出的模型進行預測，如果這些效能評估可以透過利用過往資料來提升的話，就叫機器學習。
- ML應用非常的廣泛，例如；推薦引擎、天氣預測、人臉辨識、指紋辨識、車牌辨識、醫學診斷輔助、測謊、證卷分析、股票預測、語音處理...等。



機器學習(Machine Learning)-簡介

- 機器學習依其學習方式，主要可以分為以下四種：
 - 監督式學習 (Supervised Learning)
 - 半監督式學習 (Semi-supervised learning)
 - 非監督式學習 (Unsupervised Learning)
 - 增強學習 (reinforcement learning)



機器學習(Machine Learning)-簡介

- 監督式學習 (Supervised Learning)

- 在訓練的過程中提供物件 (向量) 和預期輸出，可以是「有標籤」的分類資料或是一個連續的值 (迴歸分析)，例如輸入了大量的已標示清楚標籤的腳踏車和機車給機器後，讓機器分辨尚無標籤的照片是機車還是腳踏車。類似於動物和人類的認知感知中的「概念學習」 (concept learning)。

- 半監督式學習 (Semi-supervised learning)

- 介於監督學習與非監督學習之間。這樣的學習方式會先將「有標籤」的資料和「無標籤」的資料切出一條分界線，再將「無標籤」的資料依據標籤的整體分佈，調整出兩大類別的新分界。不需要百分之百的資料，讓半監督學習同時能降低成本又具有非監督式學習的高自動化的優點。



機器學習(Machine Learning)-簡介

- 非監督式學習 (Unsupervised Learning)
 - 這樣的機器學習方式不需要人力事前的輸入標籤，僅僅提供了輸入範例，便直接以沒有標準答案的資料來訓練機器，在學習時機器會自動找出潛在分類類別的規則，並且反覆以經過測試後的學習結果應用到新的案例上。
- 增強學習 (reinforcement learning)
 - 源自於心理中行為主義理論的學習方法，即如何在環境給予的獎懲刺激下，一步步形成對於這些刺激的預期，來產生能夠獲得最大利益的習慣性行為，強調的是透過環境而行動，並會隨時根據輸入的資料逐步修正。這個方法具有普適性，因此在其他許多領域，如博弈論、統計學及遺傳演算法等都有相關研究。



機器學習(Machine Learning)-簡介

- 機器學習的流程主要可以分為以下七個步驟：
 1. 收集資料 (Gathering data)
 2. 準備數據 (Preparing that data)
 3. 選擇模型 (Choosing a model)
 4. 訓練機器 (Training)
 5. 評估分析 (Evaluation)
 6. 調整參數 (Hyperparameter tuning)
 7. 預測推論 (Prediction)

常用機器學習架構介紹

SVM分類器



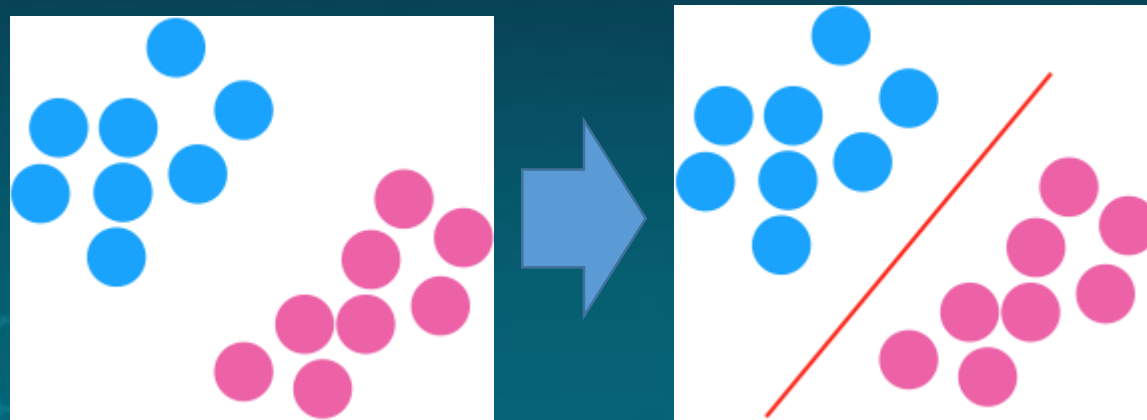
SVM(Support Vector Machine)-簡介

- 在機器學習中，SVM(support vector machine, 支援向量機)是在分類與迴歸分析中分析資料的監督式學習模型與相關的學習演算法。給定一組訓練實例，每個訓練實例被標記為屬於兩個類別中的一個或另一個，SVM訓練演算法建立一個將新的實例分配給兩個類別之一的模型，使其成為非概率二元線性分類器。



SVM(Support Vector Machine)-簡介

- SVM是一種監督式的學習方法，用統計風險最小化的原則來估計一個分類的超平面(hyperplane)，其基礎的概念非常簡單，就是找到一個決策邊界(decision boundary)讓兩類之間的邊界(margins)最大化，使其可以完美區隔開來。

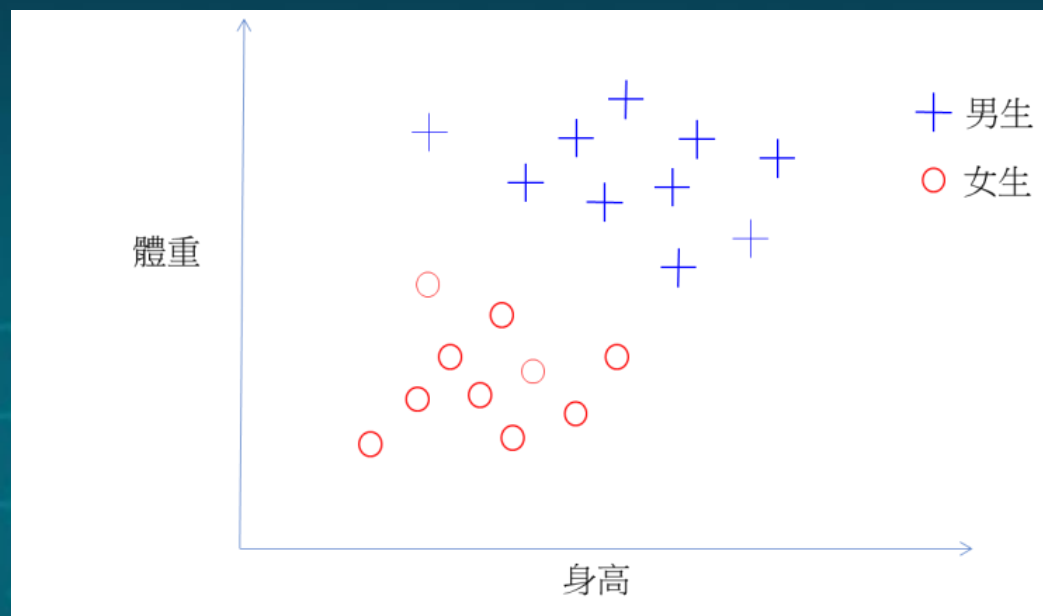


(所有分類的問題都是在找右圖中紅色那條分類的線)



SVM(Support Vector Machine)-簡介

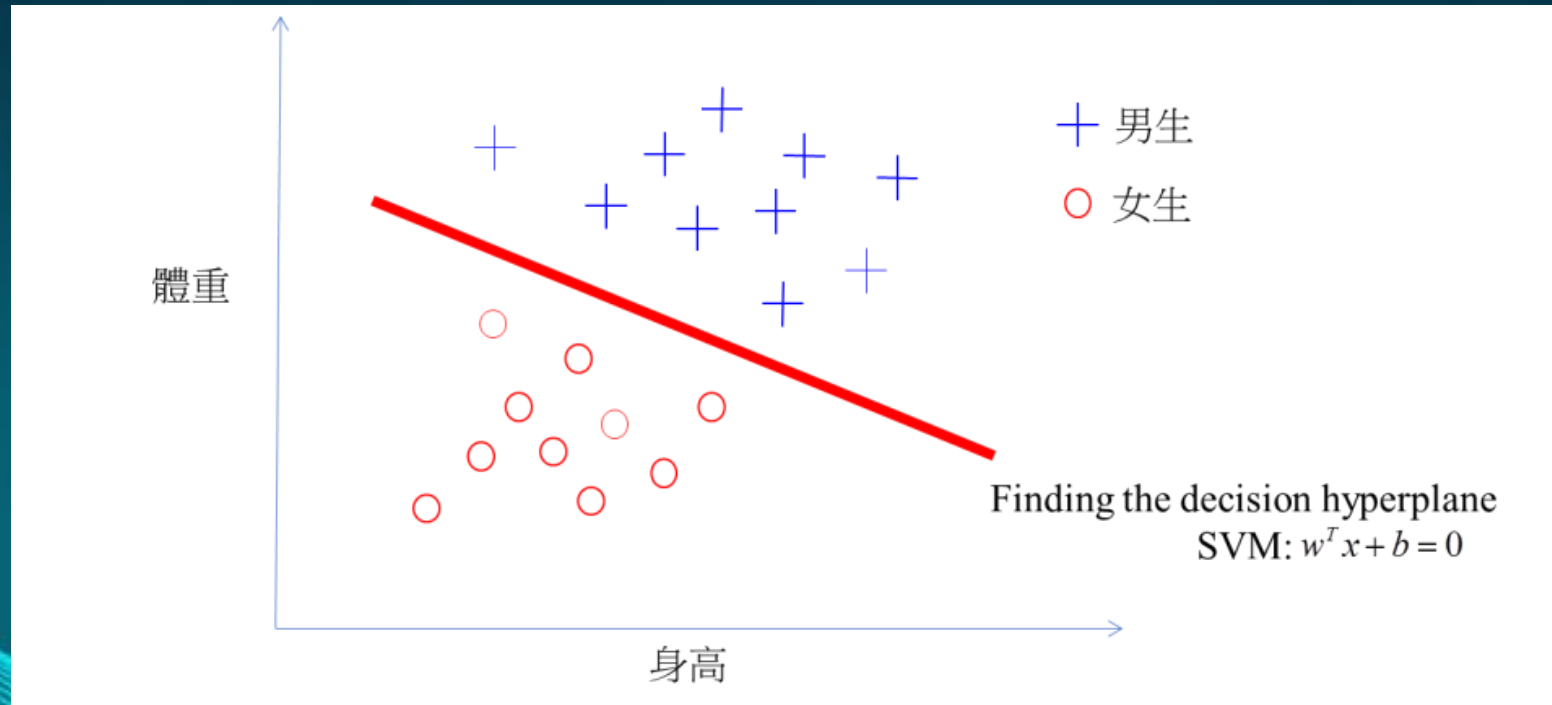
- 以下用一個例子來說明要「如何只用身高體重就來判斷是男生還是女生」。e.g. 分類男生和女生兩類，特徵資料只有「身高」和「體重」。
- 這邊我先隨便舉男生有10組資料，女生也是10組資料。





SVM(Support Vector Machine)-簡介

- 所有分類的問題都是在找下圖紅色那條分類的線
- 不一定是直線，有可能是曲線，本範例只會提到直線部分，不同的演算法都是在不同的假設或是條件下去找那條分類的線。





SVM(Support Vector Machine)-簡介

- 在各式分類器中，比如說高斯分類器，就是在利用兩組資料的高斯機率分布/高斯概似函數(Gaussian likelihood function)，去判斷誰的後驗機率(posteriori probability)/概似函數值較大，就判給哪一類別。
- SVM則是去假設有一個hyperplane($wTx+b=0$)可以完美分割兩組資料，所以SVM就是在找參數(w 和 b)讓兩組之間的距離最大化。



SVM-數學式

- 假設訓練資料為：

$$\{(\mathbf{x}_i, y_i)\}, \quad \forall i = 1, \dots, n, \quad \mathbf{x}_i \in R^d, y_i \in \{\pm 1\}$$

- 以剛剛範例為例，這邊的：

$$\mathbf{x}_i = \begin{bmatrix} \text{身高} \\ \text{體重} \end{bmatrix}, y_i = \begin{cases} +1 & \text{if 男生} \\ -1 & \text{if 女生} \end{cases}$$

- 在此我們先不考慮有「資料混在一起」的問題，也就是所以所有的資料都可以被完美分類(hard-margin SVM)。
- 因此數學將此必須滿足條件寫出來則是：

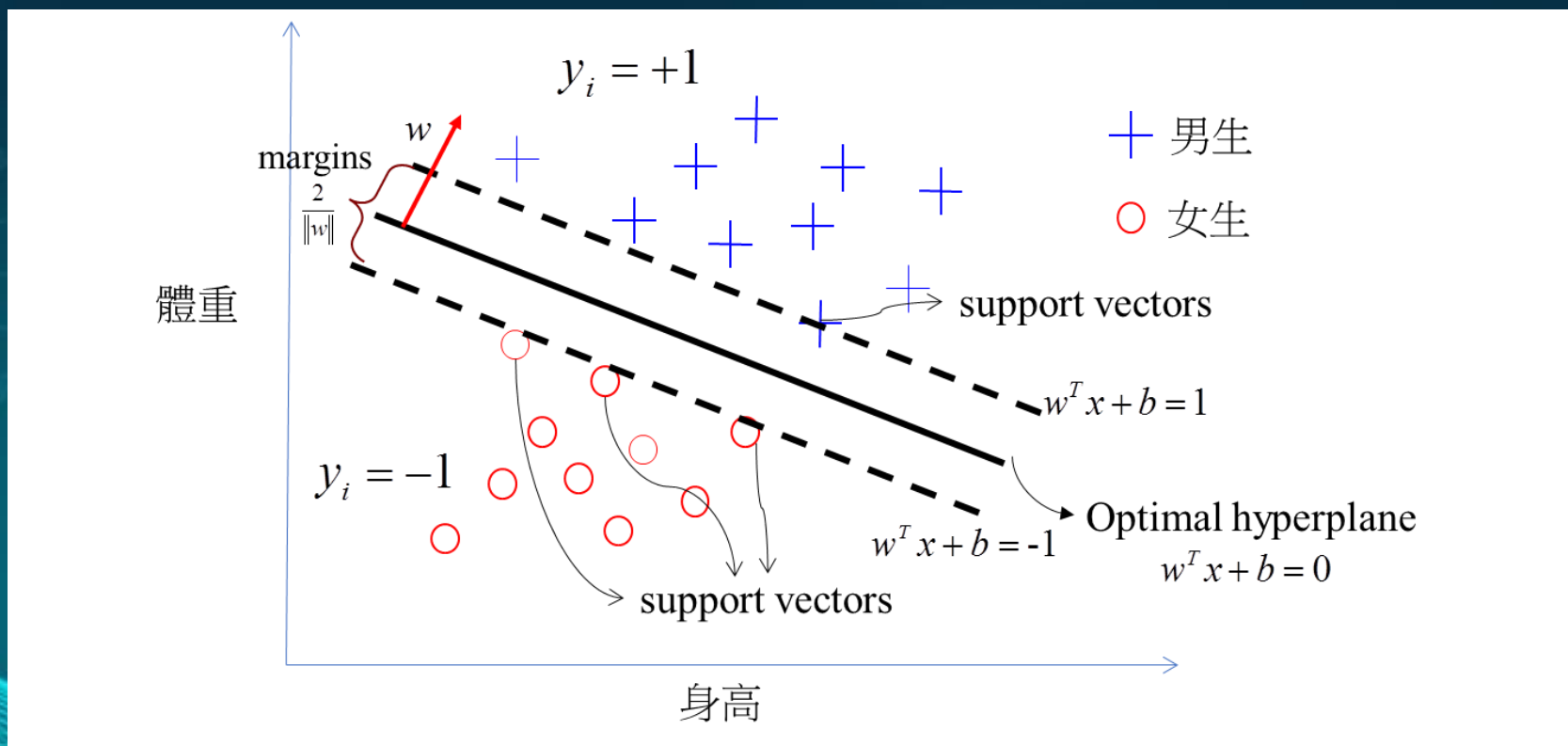
男生($y_i = +1$)是 $w^T \mathbf{x} + b \geq 1$ ，女生($y_i = -1$)是 $w^T \mathbf{x} + b \leq -1$ ，將兩類整理成下公式：

$$y_i(w^T \mathbf{x}_i + b) \geq 1, \quad \forall i = 1, \dots, n$$



SVM-數學式

- 所以SVM在找Optimal hyperplane就是希望區隔兩類之間的邊界 ($2/|w|$) 可以越大越好。





SVM-數學式

- SVM的求解本身就是一個簡單的最佳化問題(在一些條件下，希望兩類的邊界距離越大越好)，轉換成數學公式如下：

$$\begin{aligned} \max_w \{2/\|w\|\} &\rightarrow \min_w \frac{1}{2} w^T w \\ \text{subject to } &y_i(w^T x_i + b) \geq 1, \quad \forall i = 1, \dots, n \end{aligned}$$



SVM-Python程式碼實作

```
1 from sklearn import svm
2 import numpy as np
3 from sklearn import datasets
4 from sklearn.model_selection import train_test_split
5
6
7 iris=datasets.load_iris()
8 X=iris.data
9 y=iris.target
10
11 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
12 #0.setosa
13 #1.versicolor
14 #2.virginica
```

- 事前作業：SVM為scikit-learn函式庫中所提供的函式。 `pip install scikit-learn`
- 範例程式：
 - [1~5]引入函式庫
 - [7~9]讀取scikit-learn中所提供的鳶尾花資料集X為data(input),y為target(output)
 - [11]指定將資料及切割為train set(訓練集)與test set(測試集)兩部分。
*data,target均可以直接以np.array的形式指定，不一定要使用資料集。
 - train_test_split(data,target,test_size,random_state)
 - test_size為train set(訓練集)與test set(測試集)分割比例，random_state用於打亂資料集。



SVM-Python程式碼實作

- 範例程式：
 - [16]指定SVM的參數。
 - `svm.SVC(kernel, C, gamma)`
 - `kernel`為kernel function，定義了該節點在給定的輸入或輸入的集合下的輸出。
 - `C`為懲罰係數，`C`愈大代表錯誤的容忍程度愈低，在訓練集樣本中會區分的愈精細，設定太大容易造成過度擬和的問題，預設為1。
 - `gamma`決定支援向量多寡，影響訓練速度與預測速度，預設為scale。
 - [17]將data和target送入SVM進行訓練。
 - [20~21]輸出預測結果，`clf.predict(data)`，data輸入模型後輸出的target。
 - [24~25]輸出準確率，`clf.score(data, target)`，輸出以data進行predict後的結果與target進行比對計算準確率。

```
15
16 clf=svm.SVC(kernel='linear',C=1,gamma='auto')
17 clf.fit(X_train,y_train)
18
19 print("predict")
20 print(clf.predict(X_train)) #target = y_train
21 print(clf.predict(X_test))  #target = y_test
22
23 print("Accuracy")
24 print(clf.score(X_train,y_train))
25 print(clf.score(X_test, y_test))
26
```

gamma='scale':

$$\text{gamma 值} = \frac{1}{(n_{\text{features}} \times X.\text{var}())}$$

gamma='auto':

$$\text{gamma 值} = \frac{1}{n_{\text{features}}}$$



SVM-Python程式碼實作

- 執行結果：

```
clf.predict(X_train) predict
[2 1 0 2 2 1 0 2 1 1 2 0 2 0 0 1 2 2 2 2 1 2 1 1 2 2 2 2 1 2 1 0 2 1 1 1 1
 2 0 0 2 1 0 0 1 0 2 1 0 1 2 1 0 2 2 2 2 0 0 2 2 0 2 0 2 2 0 0 2 0 0 0 1 2
 2 0 0 0 1 1 0 0 1 0 2 1 2 1 0 2 0 2 0 0 2 0 2 1 1 1 2 2 2 1 0 1 2 2 0 1 1
 2 1 0 0 0 2 1 2 0]
clf.predict(X_test) [2 1 0 2 0 2 0 1 1 1 2 1 1 1 1 0 1 1 0 0 2 1 0 0 2 0 0 1 1 0]
Accuracy
clf.score 0.975
1.0
```



SVM-Python程式碼實作

- 常用kernel function :

- RBF (高斯, 此為SVM預設的kernel function)

$$k(x, y) = \exp \left(-\frac{\|x - y\|^2}{2\sigma^2} \right)$$

- Linear (線性) $k(x, y) = \langle x, y \rangle$

- Poly (多項式) $k(x, y) = (\langle x, y \rangle + c)^d$

- Sigmoid $k(x, y) = \tanh(\alpha x^T y + c)$

- Kernel主要用途在於將資料投影到更高維度的空間，並進行進一步的操作(分類、降維等等)。



SVM-Python程式碼實作

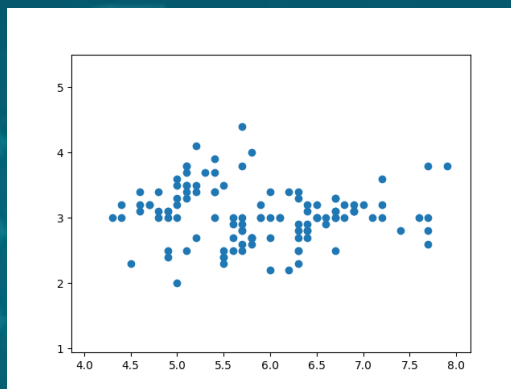
- 簡易的資料可視化：

- (1)安裝matplotlib函式庫

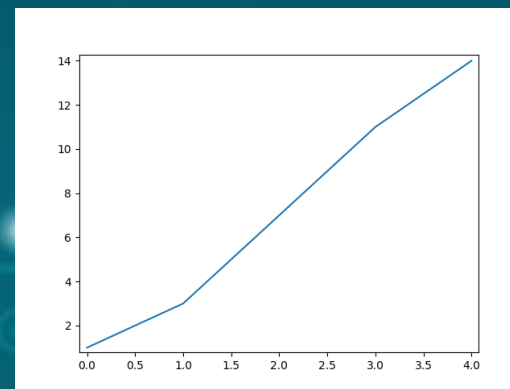
```
pip install matplotlib
```

- (2)以前面程式碼為例

- [5]引入函式庫
 - [28~29]畫出分布圖
 - plt.scatter(x,y)：點出所有在x，y陣列中的點座標。
 - plt.plot(x,y)：將x，y陣列中的所有點連成一條線(若僅輸入一維陣列，則預設x為0,1,2,3.....)。
 - [30]顯示圖片：



```
1 from sklearn import svm
2 import numpy as np
3 from sklearn import datasets
4 from sklearn.model_selection import train_test_split
5 import matplotlib.pyplot as plt
6
7 iris=datasets.load_iris()
8 X=iris.data
9 y=iris.target
10
11 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
12 #0.setosa
13 #1.versicolor
14 #2.virginica
15
16 clf=svm.SVC(kernel='linear', C=1, gamma='auto')
17 clf.fit(X_train, y_train)
18
19 print("predict")
20 print(clf.predict(X_train)) #target = y_train
21 print(clf.predict(X_test)) #target = y_test
22
23 print("Accuracy")
24 print(clf.score(X_train, y_train))
25 print(clf.score(X_test, y_test))
26
27
28 plt.scatter(X_train[:,0], X_train[:,1])
29 plt.plot([0,1,3,4], [1,3,11,14])
30 plt.show()
```





SVM-總結

- SVM的特性：
 1. 切出來的hyperplane(超平面)很漂亮，擁有最大margin(邊界)的特性
 2. 可以很容易透過更換Kernel，做出非線性的線(非線性的決策邊界)



KNN分類器



KNN分類

- KNN分類是由其鄰居的「多數決」來決定它是哪一類。
- K是一個常數，KNN就是看離你最近的K個點，然後看哪個類別的點最多就把自己也當成那個類別。
- 如果看 $K=3$ ，就是看3個點也就是3NN。



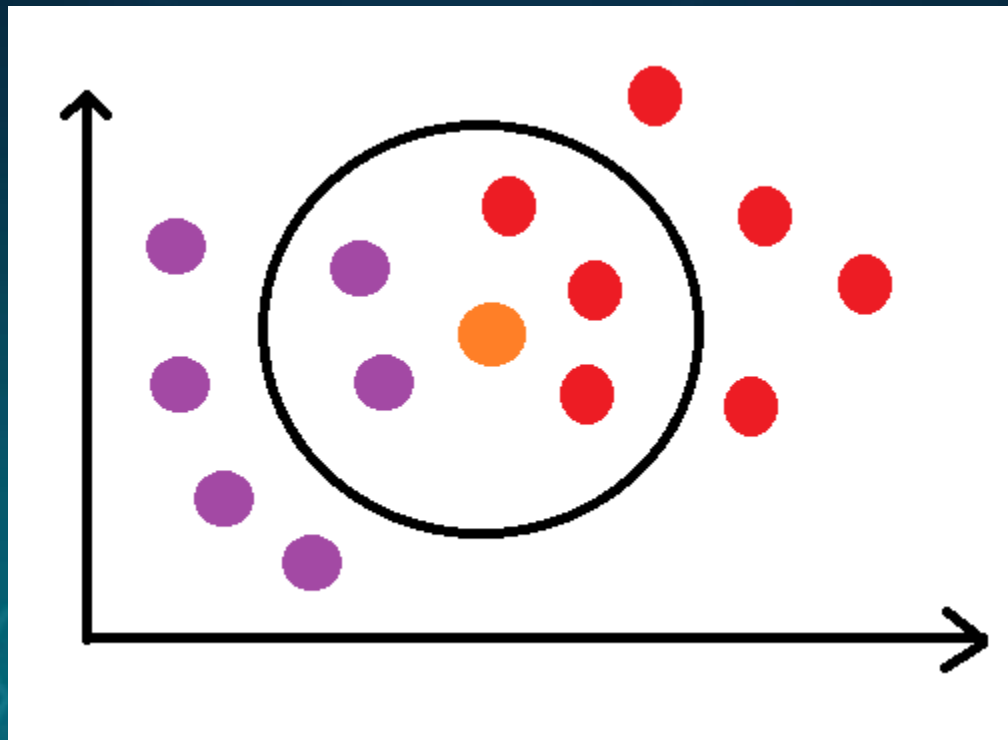
KNN分類

- KNN分類是由其鄰居的「多數決」來決定它是哪一類。
- K是一個常數，KNN就是看離你最近的K個點，然後看哪個類別的點最多就把自己也當成那個類別。
- 如果看 $K=3$ ，就是看3個點也就是3NN。



KNN分類

- 如圖設K等於5，紅3紫2，
因此橘色被歸類為紅色。



Cascade 分類器

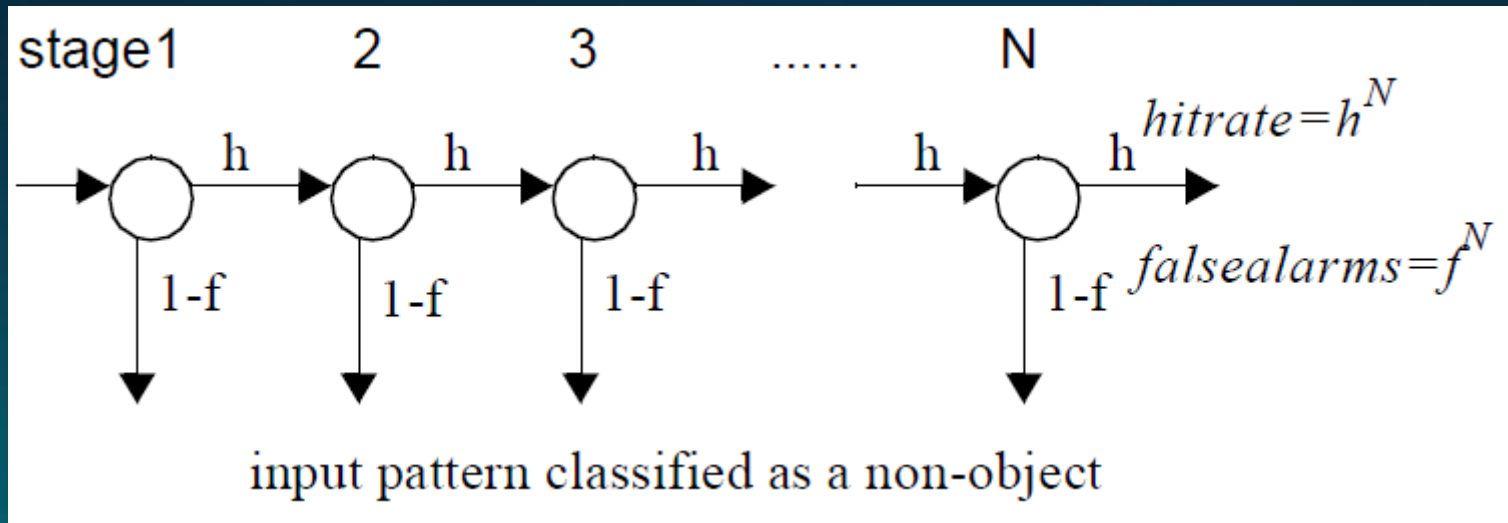


Cascade classifiers

- 在Cascade Classifier分類器，分為好幾個stage。
- 每一個stage為一個Classifier，越前面的Classifier辨識率越低，每一個Classifier有hitrate(正確辨識率)，及falsealarms(容錯率)。
- 每一個stage篩選出的目標影像會進入下一個stage並再次進行篩選，因此下一個stage的輸入影像也隨之越來越少，但同時能把大量非目標影像去除，因此越多的stage篩選的結果會越準確，同時也會降低誤判率。



Cascade classifiers





機器學習於影像上的應用

- 可以將影像作為機器學習(ex.SVM)的輸入進行訓練，以得到一個足以辨識、分類影像類別的模型，最基本即為將影像資訊視為一個二維陣列輸入，但為了更有效將影像的特徵表現出來，傳統上會利用一些專門的方法，本周課程先讓同學對機器學習有初步的認識，下周上課會再進一步說明如何將原始影像的特徵萃取出來再進行分類。