台灣人工智慧學校

機器學習基礎與演算法

Chapter 9　非監督式學習 (Unsupervised Learning)
Chapter 10 總結 (Summary)

講師投影片Chapter9
講師投影片Chapter10
資料與程式碼
課程投影片

課程內容

9.非監督式學習 (Unsupervised Learning)

-Introduction to unsupervised learning

-Dimension reduction

-Principal component analysis

-t-SNE

-K-means clustering

-Hierarchical clustering

[實作] pca

[實作] t-SNE

[實作] K-means

10. 總結 (Summary)

Code 放在Hub中的course內

- 為維護課程資料，courses中的檔案皆為read-only，如需修改請cp至自身環境中
- 打開terminal，輸入

```
cp -r courses-tpe/Machine_Learning <存放至本機的名稱>
```

Chapter 9 非監督式學習 (Unsupervised learning)

- 範例程式(example)的檔名會以**藍色字體**顯示且旁邊附上
- 練習(exercise)的檔案以**紅色字體**顯示且旁邊附上

09-1: Introduction to unsupervised learning

Unsupervised learning

- Extract patterns from only feature ***X*** (no target ***y***)
- Examples
  - Find few "representative" dimensions in ***X*** (dimension reduction)
  - Divide data points into groups based on input features (clustering)

09-2: Dimension reduction

Toy example

- Consider the following 3d points
  - (1,2,3), (2,4,6), (3,6,9), (4,8,12), (5,10,15), (6,12,18)
  - If each integer requires 1 byte, we need 1*3*6=18 bytes
- However, we may also store the first point (1,2,3) as the base, and store the multiplier of each point
  - One point (3 bytes) + multipliers (6 bytes)
- Reduced 50% of the storage

09-3: Principal component analysis

Finding $\boldsymbol{u}_1$ (cont')

- By Lagrange multiplier, we have

$$\mathcal{L} = \boldsymbol{u}_1^T \boldsymbol{\Sigma} \boldsymbol{u}_1 + \lambda(1 - \boldsymbol{u}_1^T \boldsymbol{u}_1)$$

- Take derivative and set to 0

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{u}_1} = 2\boldsymbol{\Sigma} \boldsymbol{u}_1 - 2\lambda \boldsymbol{u}_1 = 0 \Rightarrow \boldsymbol{\Sigma} \boldsymbol{u}_1 = \lambda \boldsymbol{u}_1$$

So, $\boldsymbol{u}_1$ is an eigenvector of $\boldsymbol{\Sigma}$ with eigenvalue λ

- Since we want to maximize $\boldsymbol{u}_1^T \boldsymbol{\Sigma} \boldsymbol{u}_1$, $\boldsymbol{u}_1$ must be the eigenvector with maximum eigenvalue of $\boldsymbol{\Sigma}$

09-4: t-SNE

KL-divergence and similarity measure in t-SNE

$$L = \sum_i KL[P(\cdot\,|\mathbf{x}_i)\,||\,Q(\cdot\,|\mathbf{z}_i)] = \sum_i \sum_j P(\mathbf{x}_j|\mathbf{x}_i) \log \frac{P(\mathbf{x}_j|\mathbf{x}_i)}{Q(\mathbf{z}_j|\mathbf{z}_i)}$$

- Similarity in the original dimension

$$S(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|_2\right)$$

- Similarity in the new (low) dimension

$$S'(\mathbf{z}_i, \mathbf{z}_j) = \frac{1}{1 + \|\mathbf{z}_i - \mathbf{z}_j\|_2}$$

Note:

$$P(\mathbf{x}_j|\mathbf{x}_i) = \frac{S(\mathbf{x}_i, \mathbf{x}_j)}{\sum_{k \neq i} S(\mathbf{x}_i, \mathbf{x}_k)}$$

$$Q(\mathbf{z}_j|\mathbf{z}_i) = \frac{S'(\mathbf{z}_i, \mathbf{z}_j)}{\sum_{k \neq i} S'(\mathbf{z}_i, \mathbf{z}_k)}$$

09-5: K-means clustering

K-means algorithm

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess *k* cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns...
5. ...and jumps there
6. ...Repeat until terminated!

09-6: Hierarchical clustering

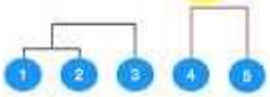
Agglomerative example

	1	2	3	4	5
1	0				
2	2	0			
3	6	3	0		
4	10	9	7	0	
5	9	8	5	4	0

	(1,2)	3	4	5
(1,2)	0			
3	3	0		
4	9	7	0	
5	8	5	4	0

	(1,2,3)	4	5
(1,2,3)	0		
4	7	0	
5	5	4	0

- $d_{(1,2,3),4} = \min\{d_{(1,2),4}, d_{3,4}\} = \min\{9,7\} = 7$
- $d_{(1,2,3),5} = \min\{d_{(1,2),5}, d_{3,5}\} = \min\{8,5\} = 5$



主成份分析 (PCA)

- 實務上我們經常遇到資料有非常多的 features，有些 features 可能高度相關，有什麼方法能夠把高度相關的 features 去除？
- PCA 透過計算 eigen-value, eigen-vector，可以將原本的 features 降維至特定的維度
    - 原本 Data 有 100 個 features，透過 PCA，可以將這 100 個 features 降成 2 個 feautres
    - 新 features 為舊 features 的線性組合

新 feaures 彼此不相關

The original variables is noted as $x_1, x_2, \ldots, x_n$, and the new variables can be represented as

$$\left.\begin{aligned}
z_1 &= a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\
z_2 &= a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\
&\vdots \\
z_n &= a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n
\end{aligned}\right\} \text{Uncorrelated}$$

PCA in Scikit-learn

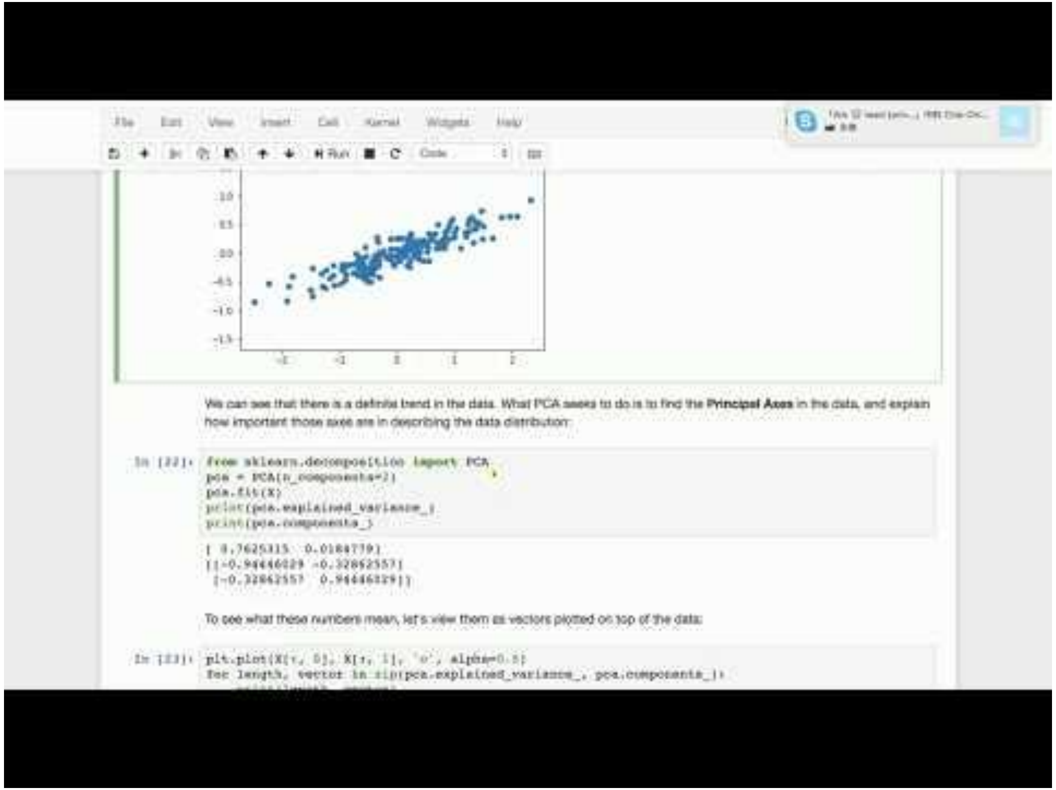
```python
from sklearn.decomposition import PCA

pca = PCA(n_componets=2)

X_reduct = pca.fit_transform(X) #X.shape=(200, 64)

print(X_reduct.shape)  #(200, 2)
```

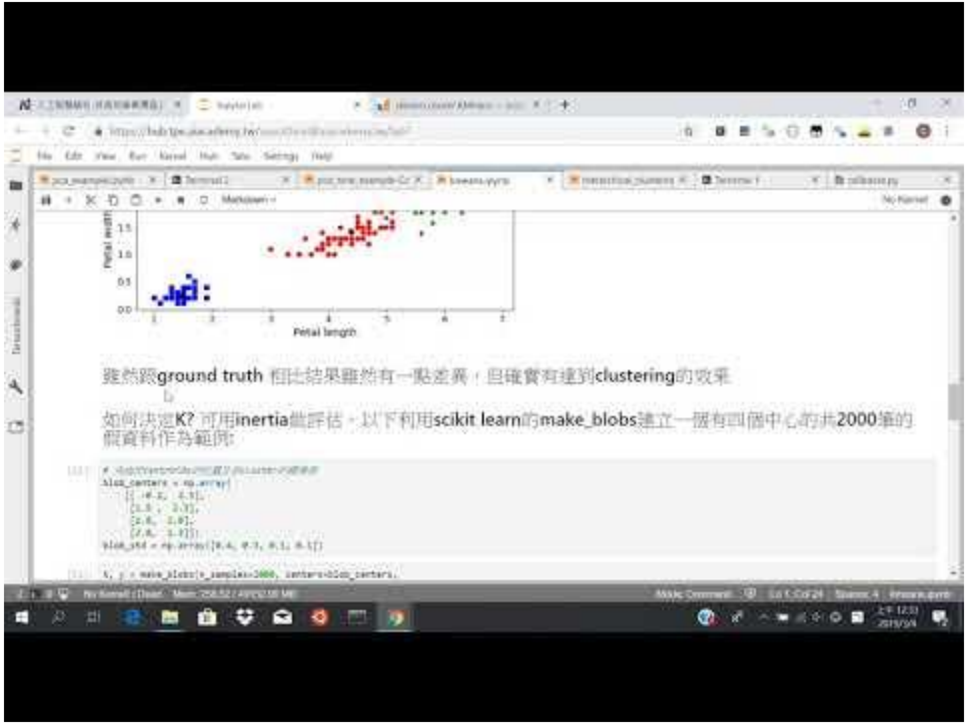
[實作課程] PCA 實戰



We can see that there is a definite trend in the data. What PCA seeks to do is to find the **Principal Axes** in the data, and explain how important those axes are in describing the data distribution:
In [22]:
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
pca.fit(X)
print(pca.explained_variance_)
print(pca.components_)
[0.7625315 0.0184779]
[[-0.94446029 -0.32862557]
 [-0.32862557 0.94446029]]
To see what these numbers mean, let's view them as vectors plotted on top of the data:
In [23]:
plt.plot(X[:, 0], X[:, 1], 'o', alpha=0.5)
for length, vector in zip(pca.explained_variance_, pca.components_):

[實作課程] t-SNE

[實作課程] K-means



Petal length
雖然跟ground truth 相比結果雖然有一點差異，但確實有達到clustering的效果
如何決定K? 可用inertia做評估，以下利用scikit learn的make_blobs建立一個有四個中心的共2000筆的假資料作為範例

練習

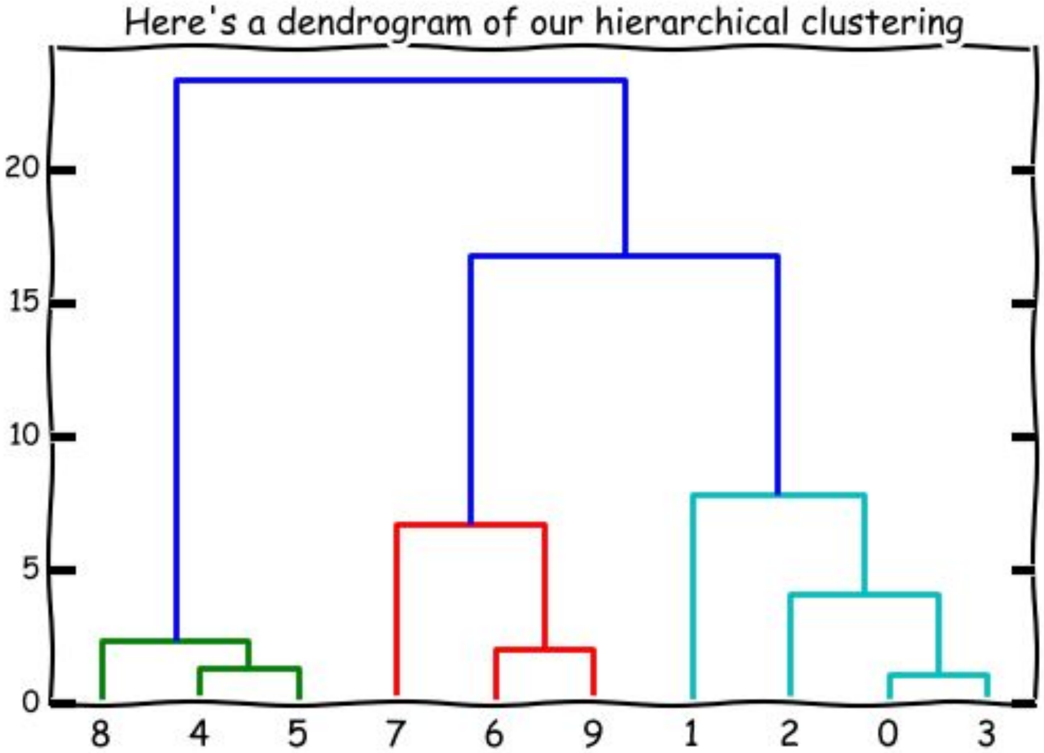
- 使用 digits dataset ，比較如果將資料降維之後再訓練模型，準確度是否會提升

階層式分析

- 不需指定分群的數量
  1. 每筆資料視為獨立一群
  2. 計算每兩群之間的**距離**
  3. 將最近的兩群合併成一群
  4. 重複 2,3 直到所有資料合併為同一群為止
- 計算距離的方式有
  - `'complete'`: cluster 中，最遠兩點的距離
  - `'single'`: cluster 中，最近兩點的距離
  - `'average'`: cluster 中，所有點的距離平均

階層分析後的樹狀圖 (dendrogram)

● 可定義 4, 5 是一群，或 8, 4 ,5 是一群，端看距離怎麼衡量



練習

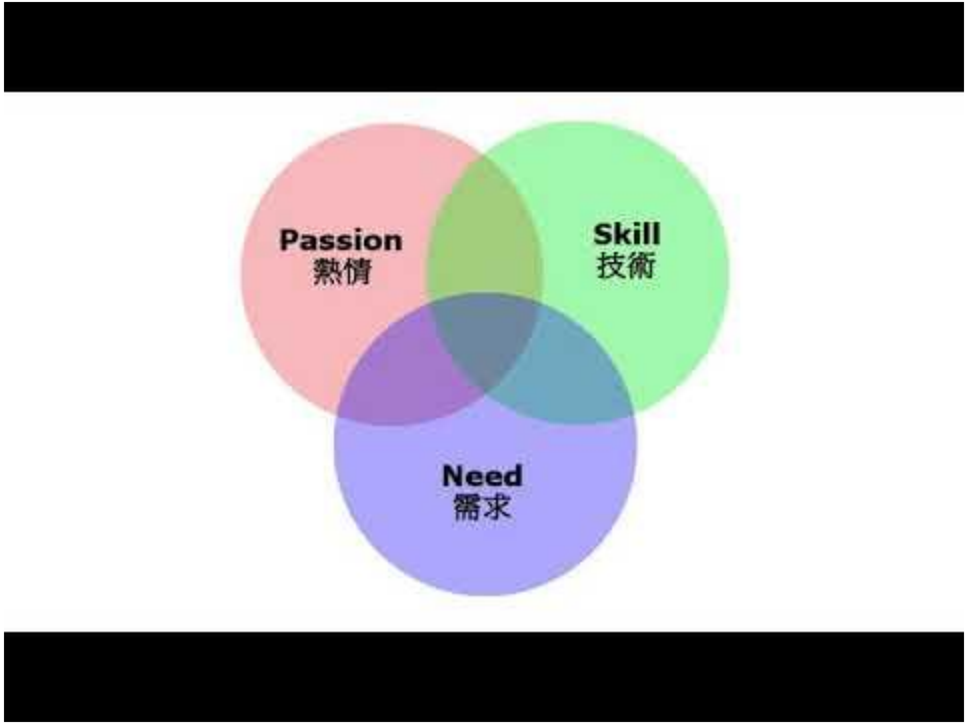
- 請參考 session3 中的 **hierarchical_clustering_example**，試著理解 code

補充閱讀

- [PCA]
- [Hierarchical]

Chapter 10 總結 (Summary)

10: Summary



課後問卷

親愛的學員您好:

　為了解課程內容的安排是否恰當，想請各位學員給我們一些回饋，各位寶貴的意見將能協助我們設計出更優質的課程！

[問卷連結]