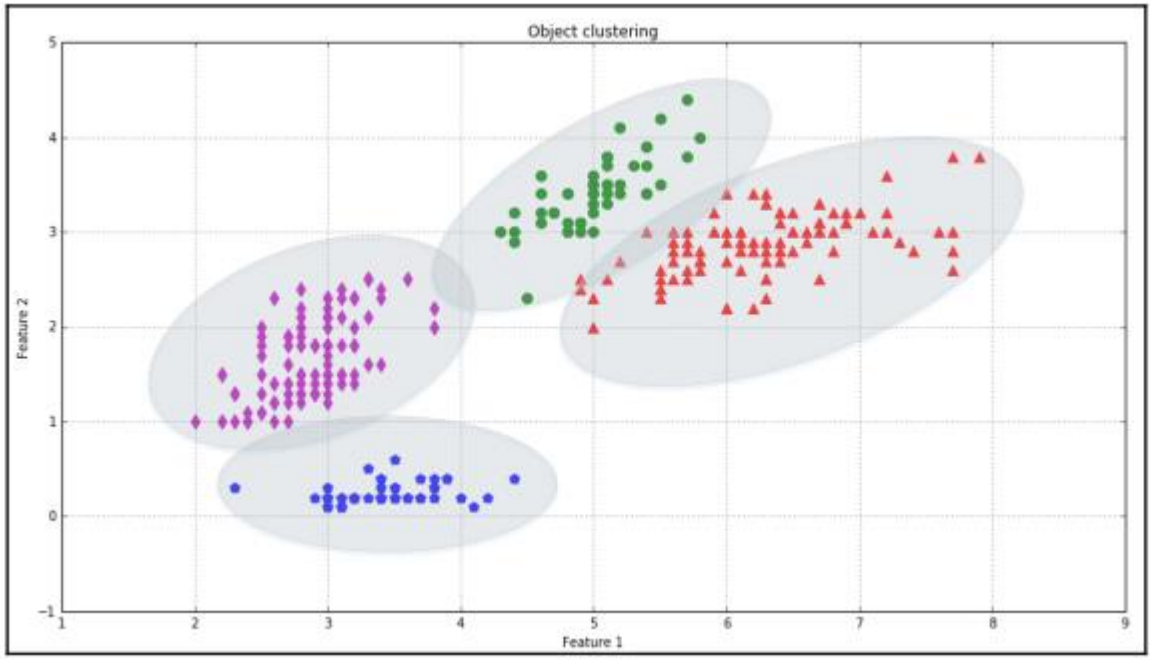
非監督式機器學習演算法
集群分析法(Cluster Analysis)

- 群集分析的應用

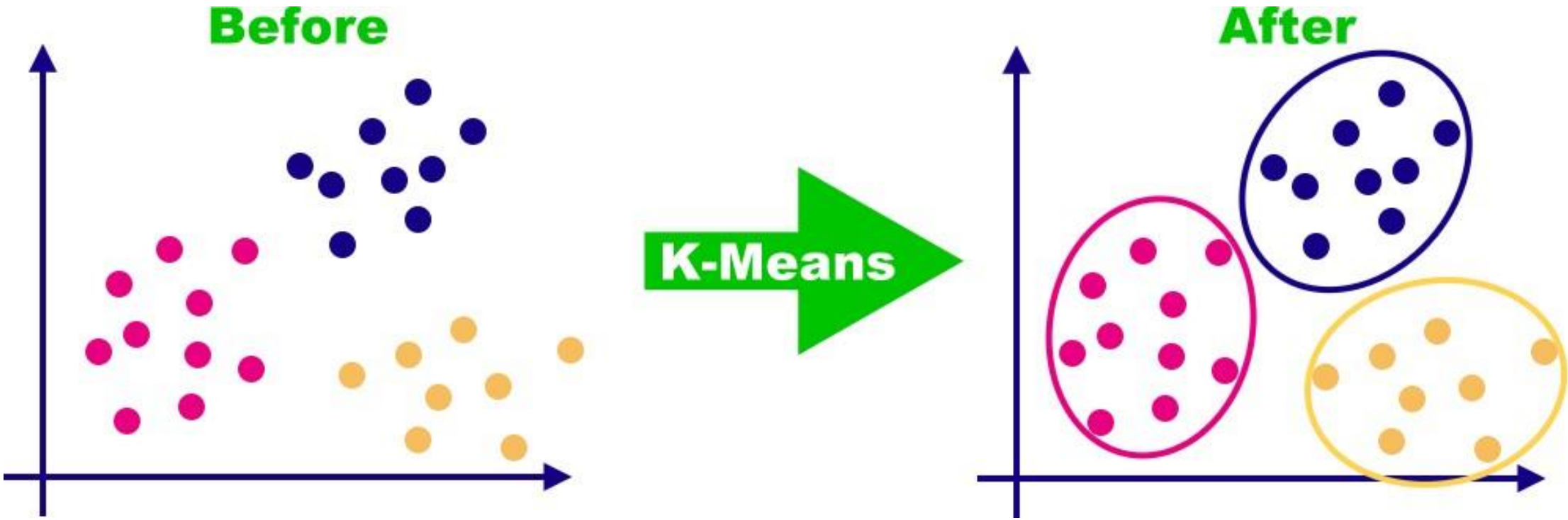
1. 市場區隔
2. 產品組合
3. 文字探勘



Cluster Analysis

問題:

- 分類個數?
- 在分類個數確定下，每一個點應該屬於哪一類?



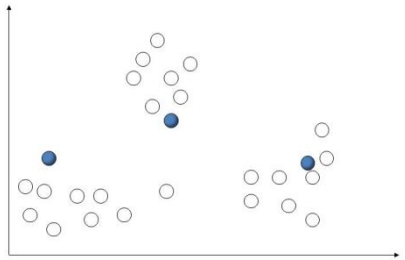
K-Means

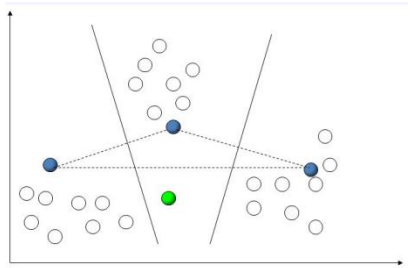
K-Means 是 JMacQueen 於1967年所提出的分群演算法，其演算法需事前設定群集的數量 k，然後找尋下列公式的極小值，以達到分群的最佳化之目的。

$$argmin \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - \mu_i\|^2$$

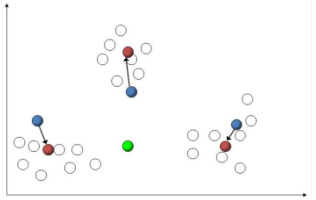
Step:

1.隨機指派群集中心, 例如設定k=3





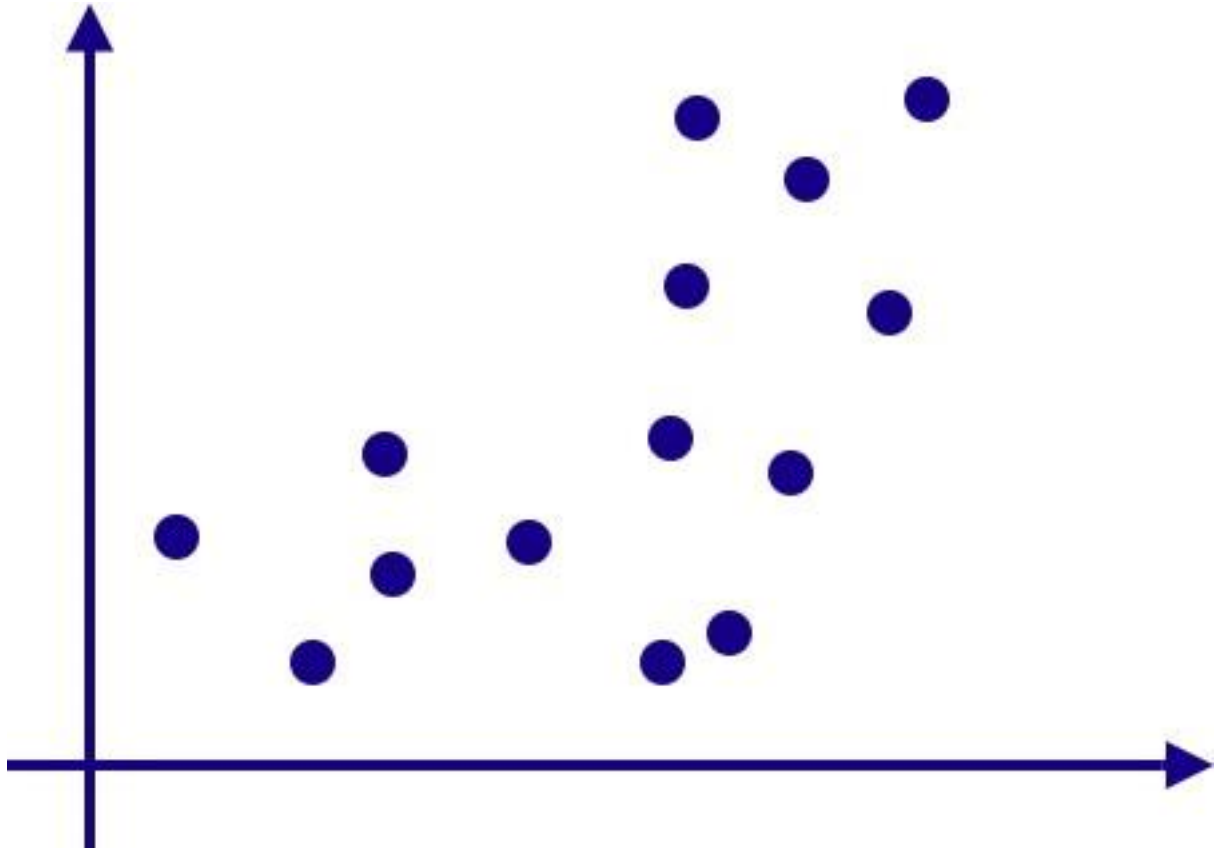
2.將群集中心附近的點根據與這3個中心點的距離分配到這3群(產生初始群集)，並重新計算中心點，並重覆此步驟



3.收斂(不再變動)：得到與各集群中心點距離和最小值 $\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2$

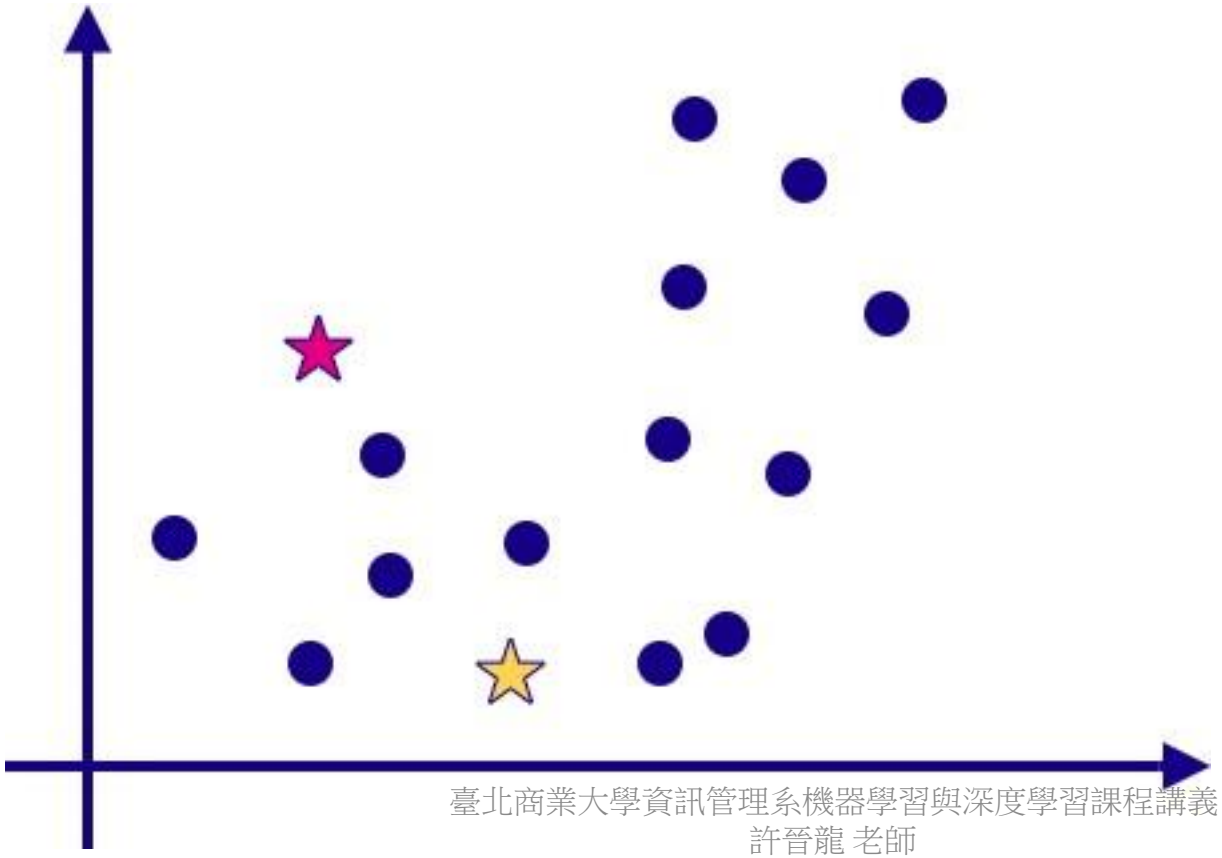
K-Means algorithm

- STEP 1： 選擇集群數量k，例如K=2



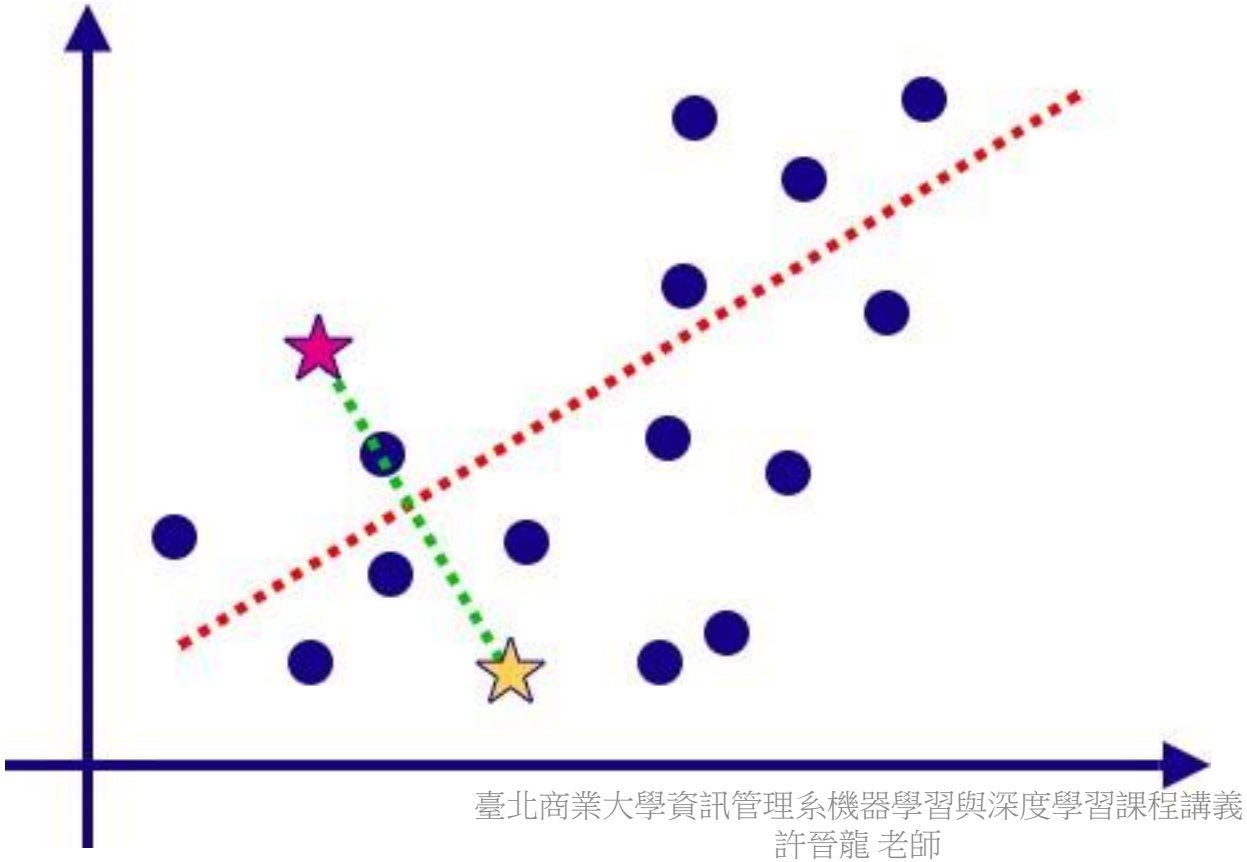
K-Means algorithm

- STEP 2： 在數據當中，隨機選擇k個點，這些點當做初始化中心點(centroid)



K-Means algorithm

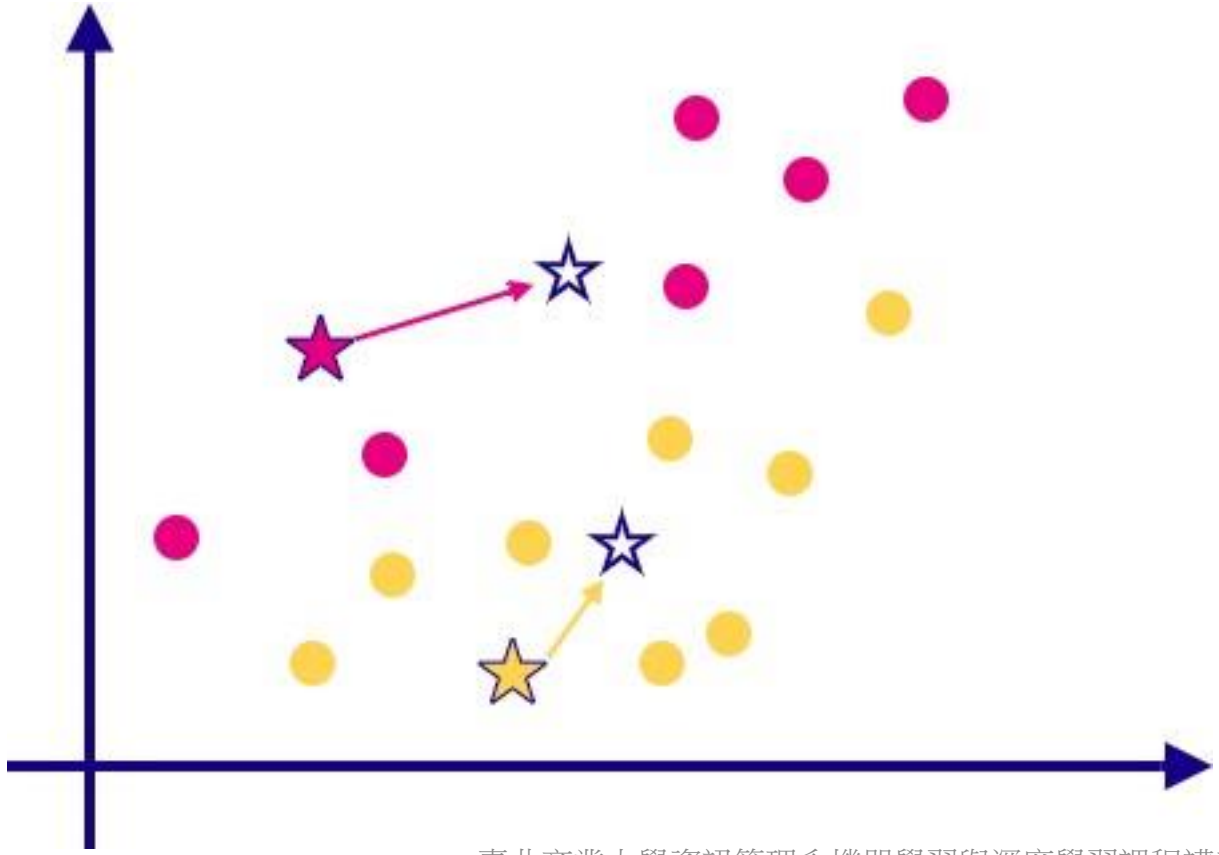
- STEP 3： 決定所有數據資料點是哪一類，與初始化點距離愈接近就是那一類 (形成K clusters)



歐幾里得距離

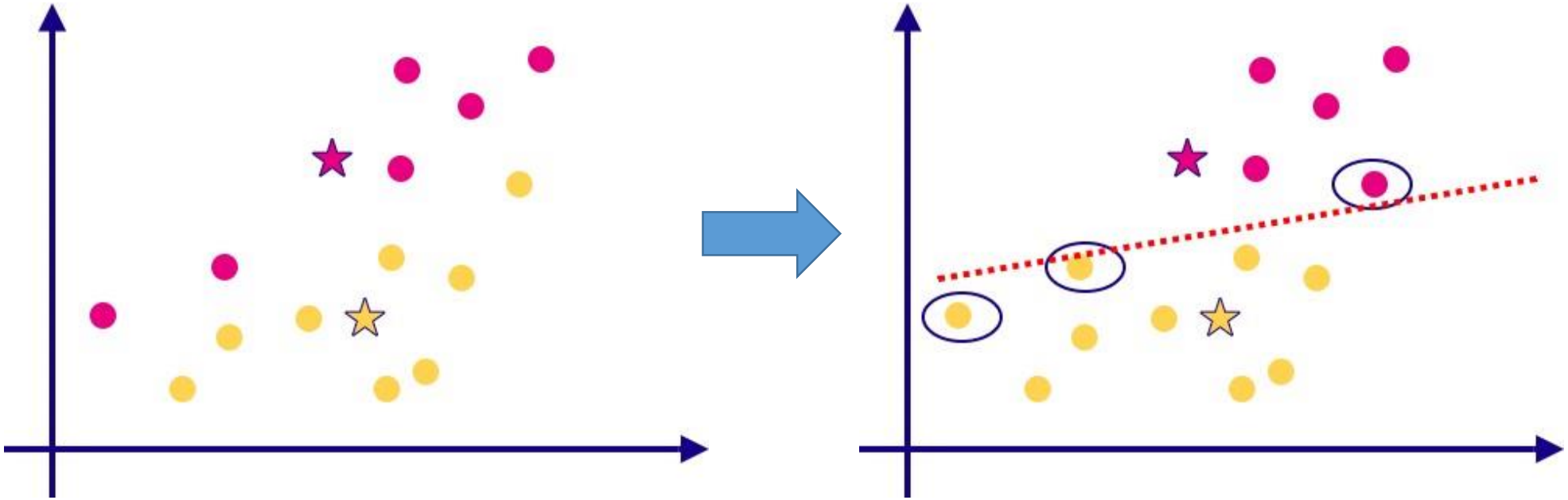
K-Means algorithm

- STEP 4： 更新每一類的中心點



K-Means algorithm

- STEP 5： 每一個點依據新的中心點再次重新分類



K-Means algorithm

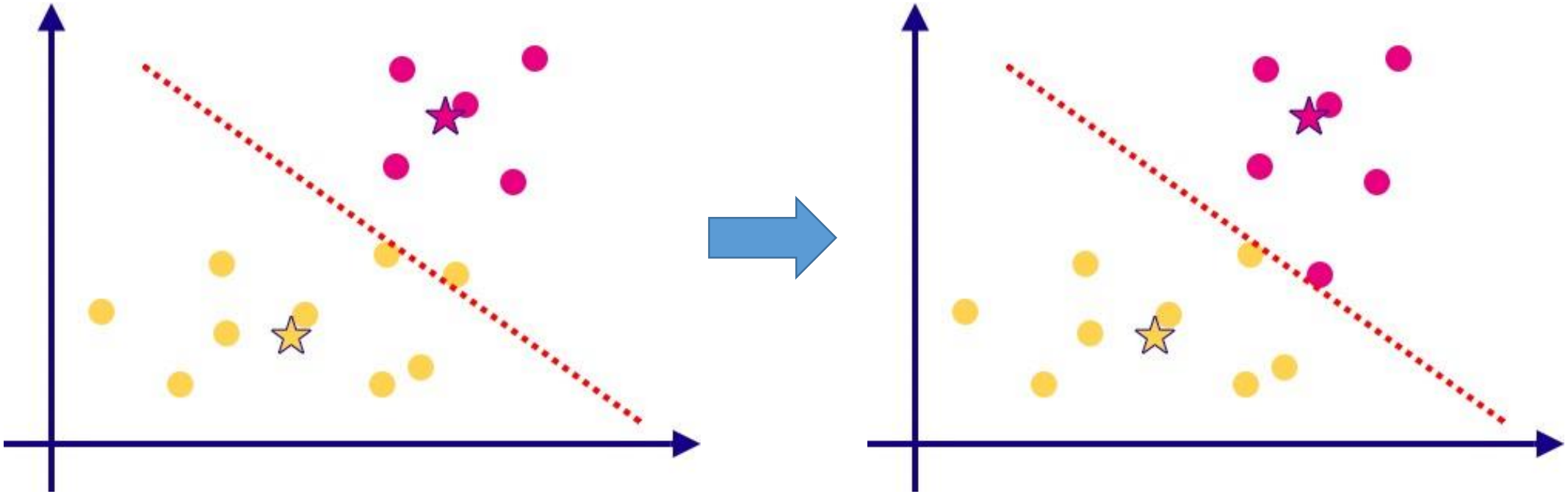
重覆STEP 4, STEP 5, 直到結果不再改變

K-Means algorithm

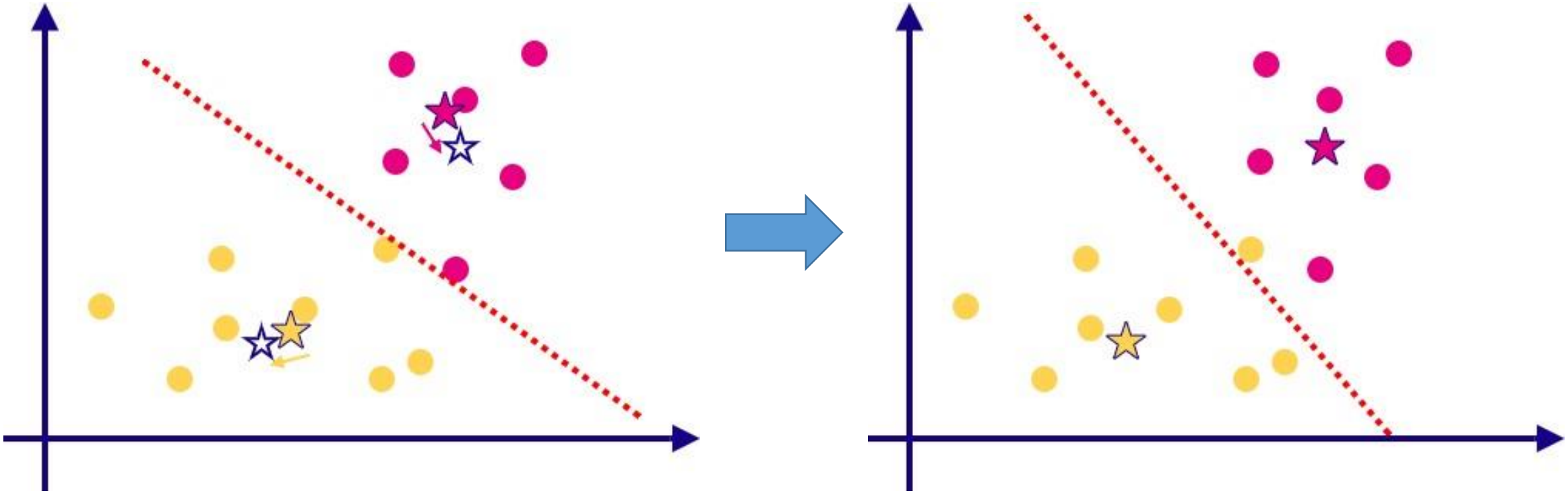
- STEP 4：再次更新每一類的中心點

K-Means algorithm

- STEP 5： 每一個點依據新的中心點再一次重新分類

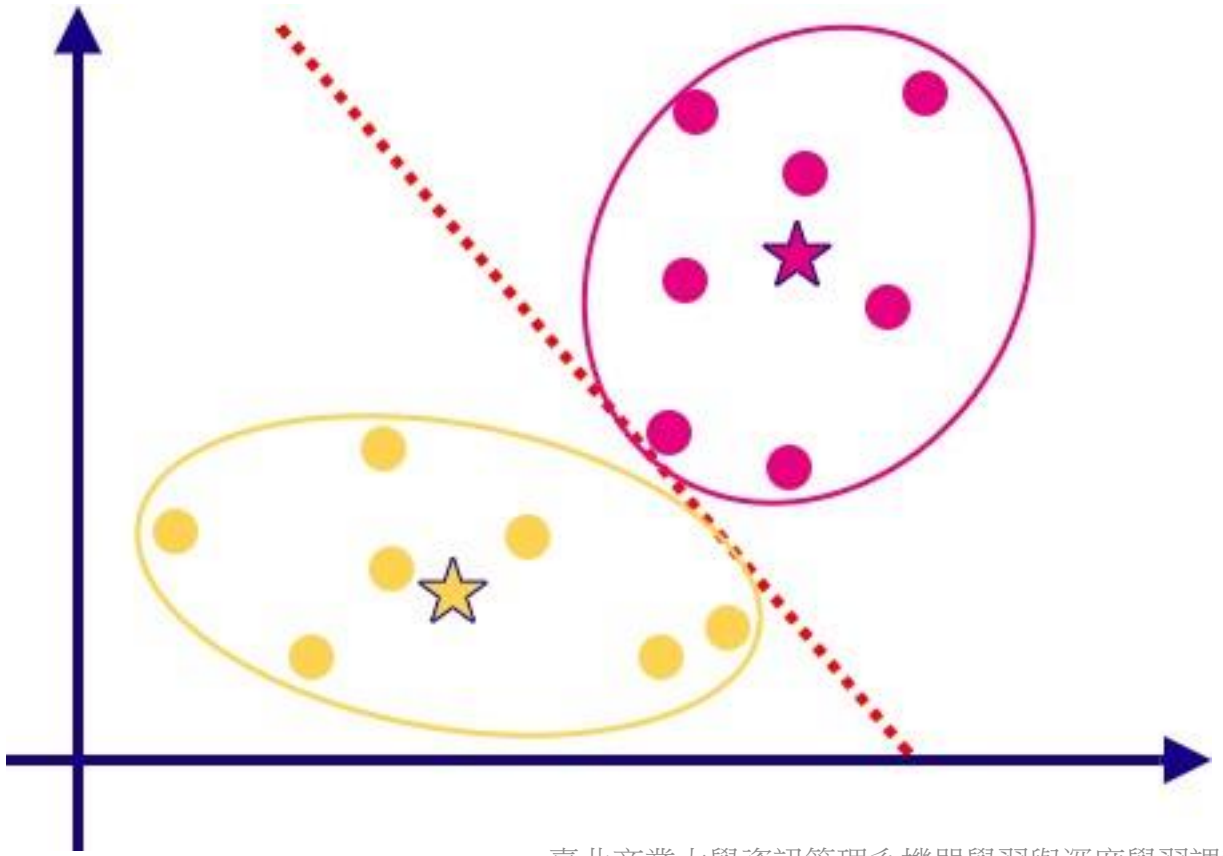


K-Means algorithm

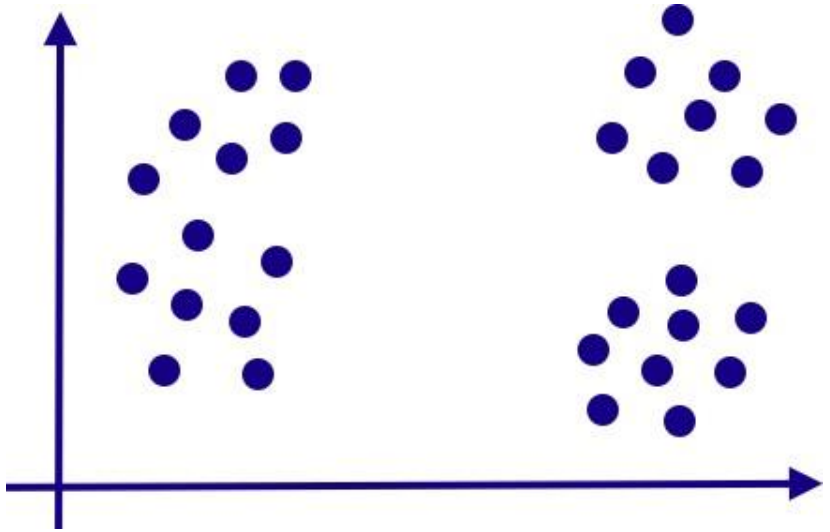


K-Means algorithm

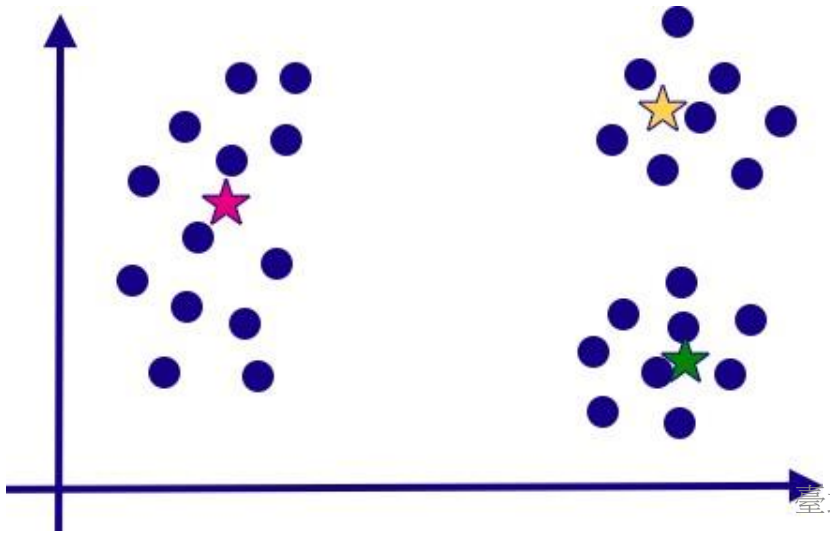
- 結束:找到最佳集群分類結果



初始中心點？

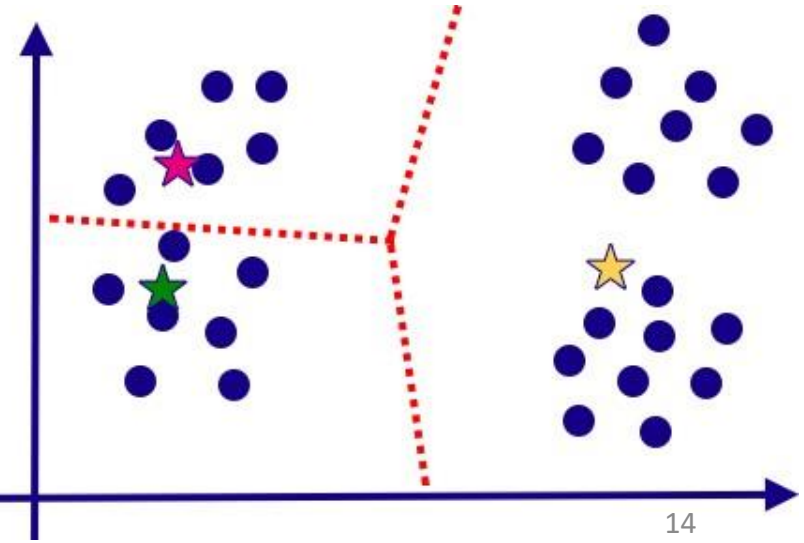


假使將k設定為 3

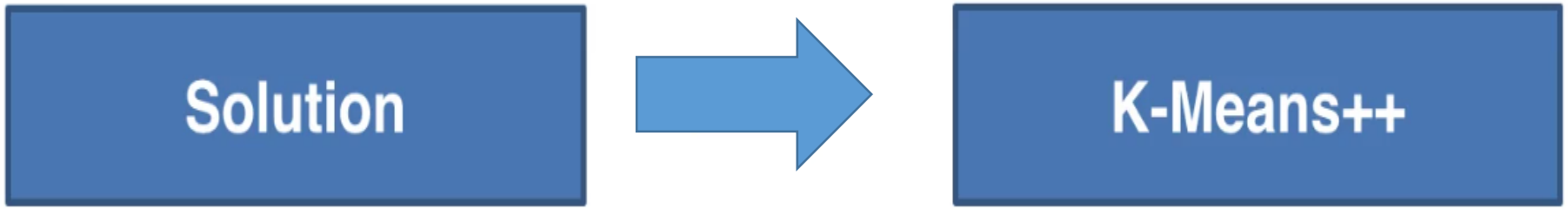


初始中心點對於集群結果有最決定性影響

到底哪一種較好？



解決初始化問題 K-Means++



進階優化 K-means++

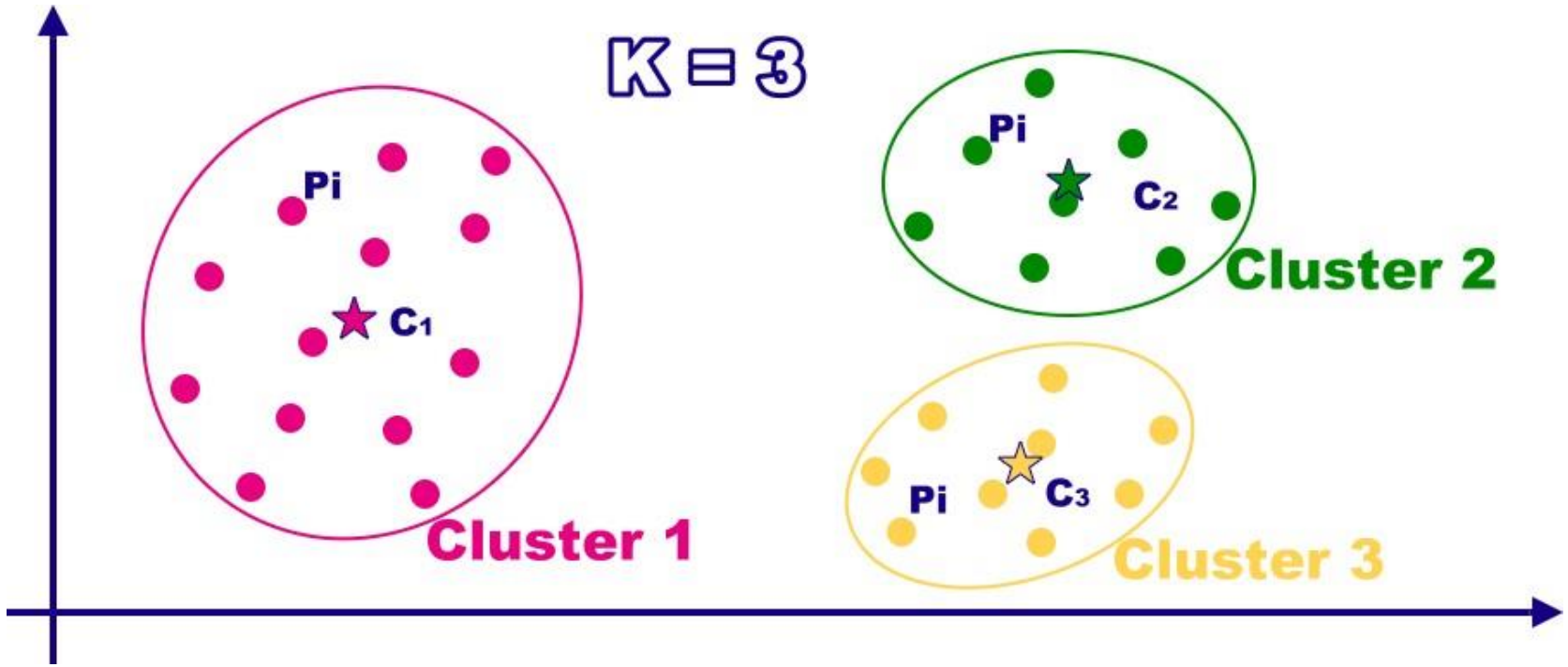
- K-means缺點是依賴隨機初始點, 可能會造成更長時間來收斂
- K-means++
  - 初始的群集中心之間的相互距離要儘可能的遠。

```
class sklearn.cluster. KMeans (n_clusters=8, init='k-means++', n_init=10, max_iter=300, tol=0.0001,
precompute_distances='auto', verbose=0, random_state=None, copy_x=True, n_jobs=1, algorithm='auto') ¶
```

http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

https://rpubs.com/skydome20/R-Note9-Clustering

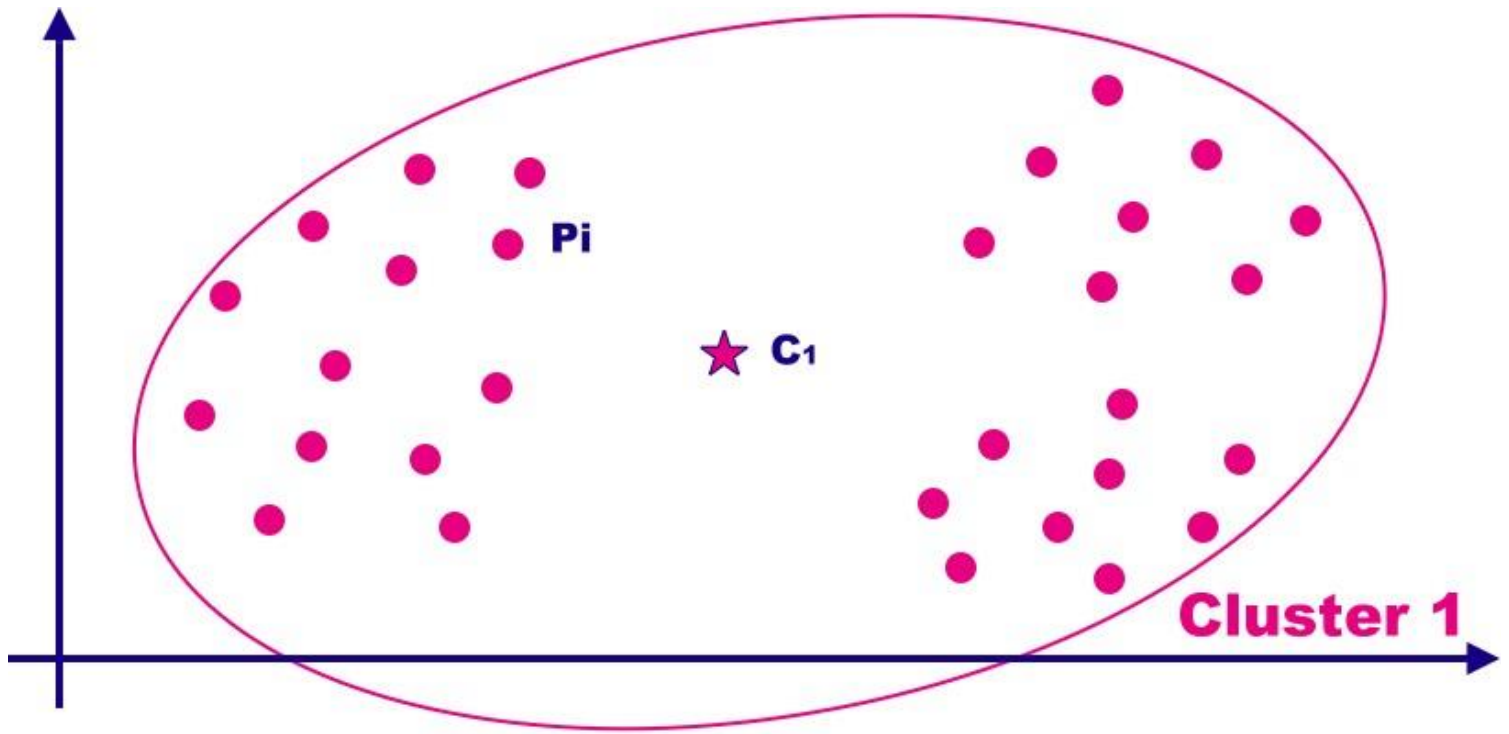
如何選擇一個好的k值來作K-means集群分析？



组内平方和

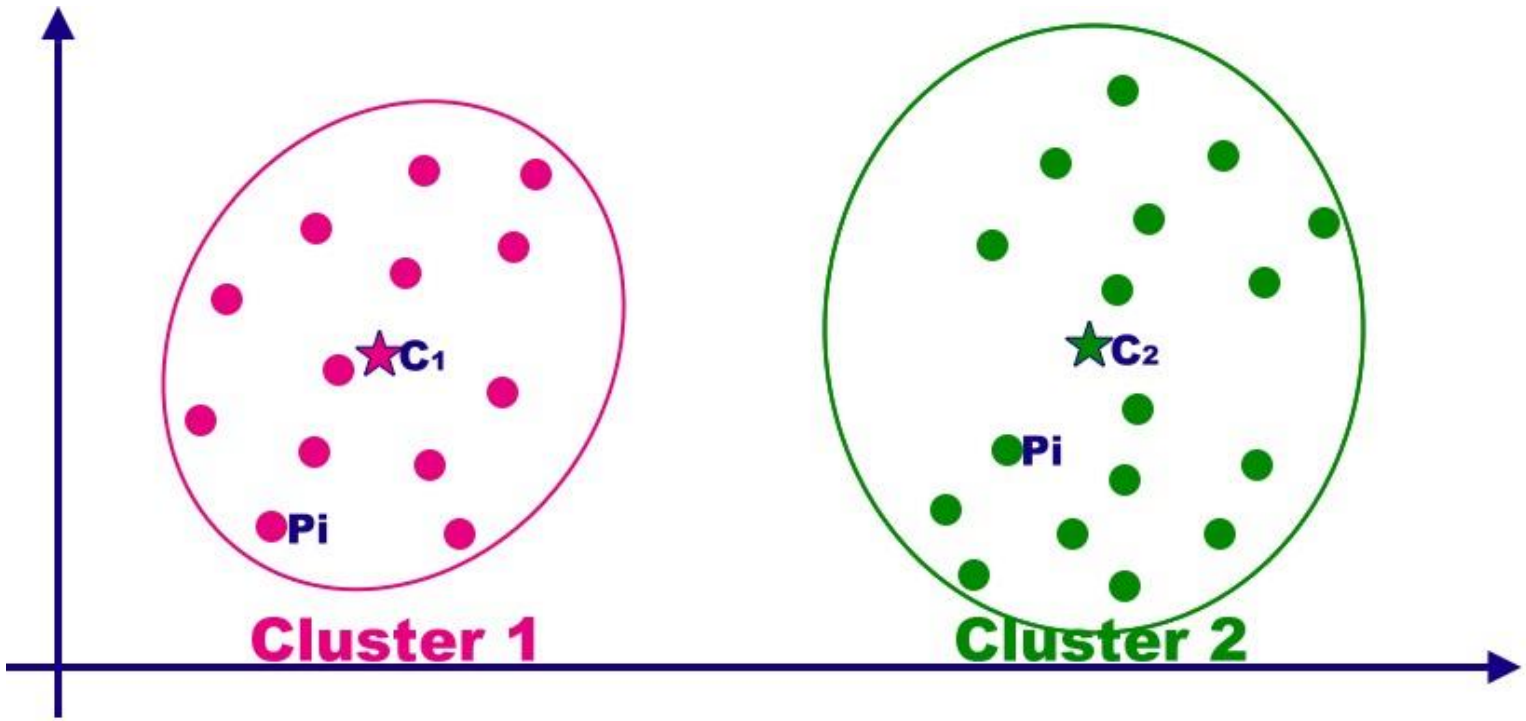
$$\text{WCSS} = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2 + \sum_{P_i \text{ in Cluster 3}} \text{distance}(P_i, C_3)^2$$

K = 1



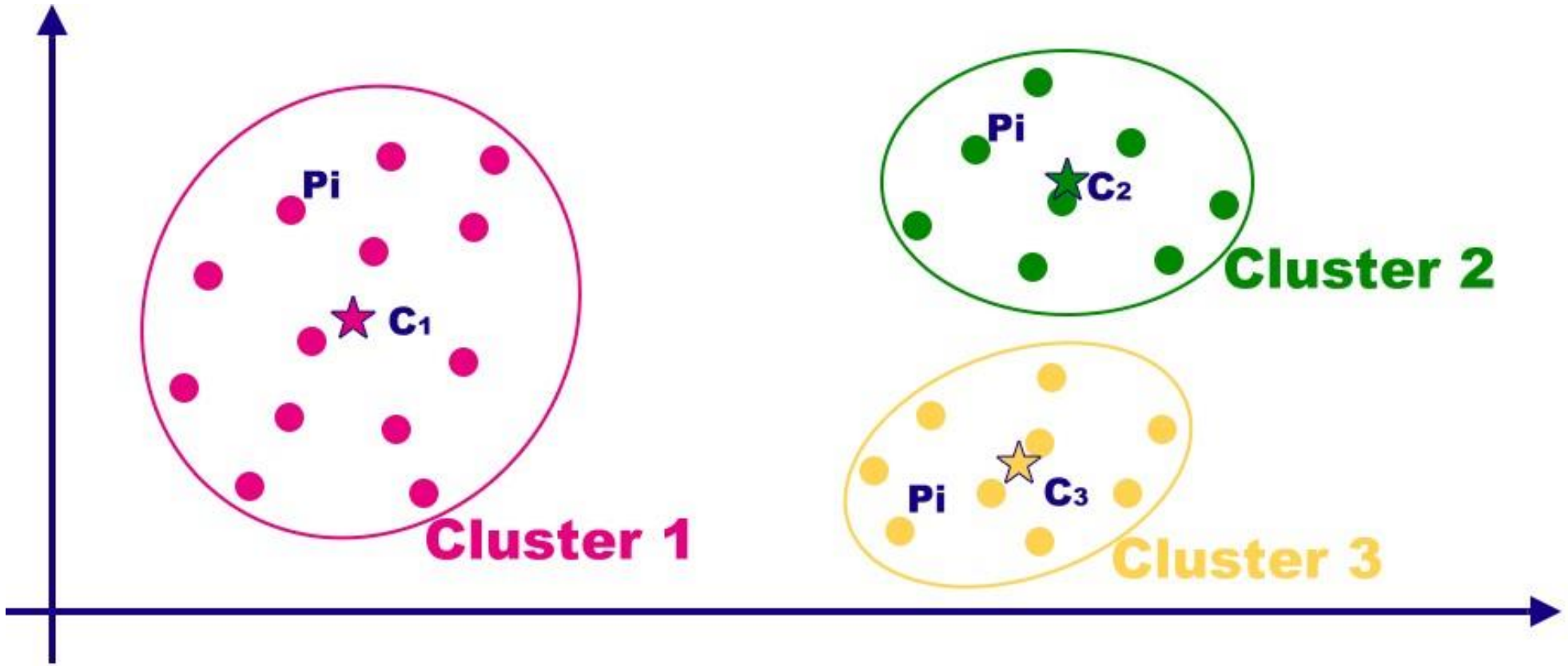
$$\text{WCSS} = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2$$

K = 2



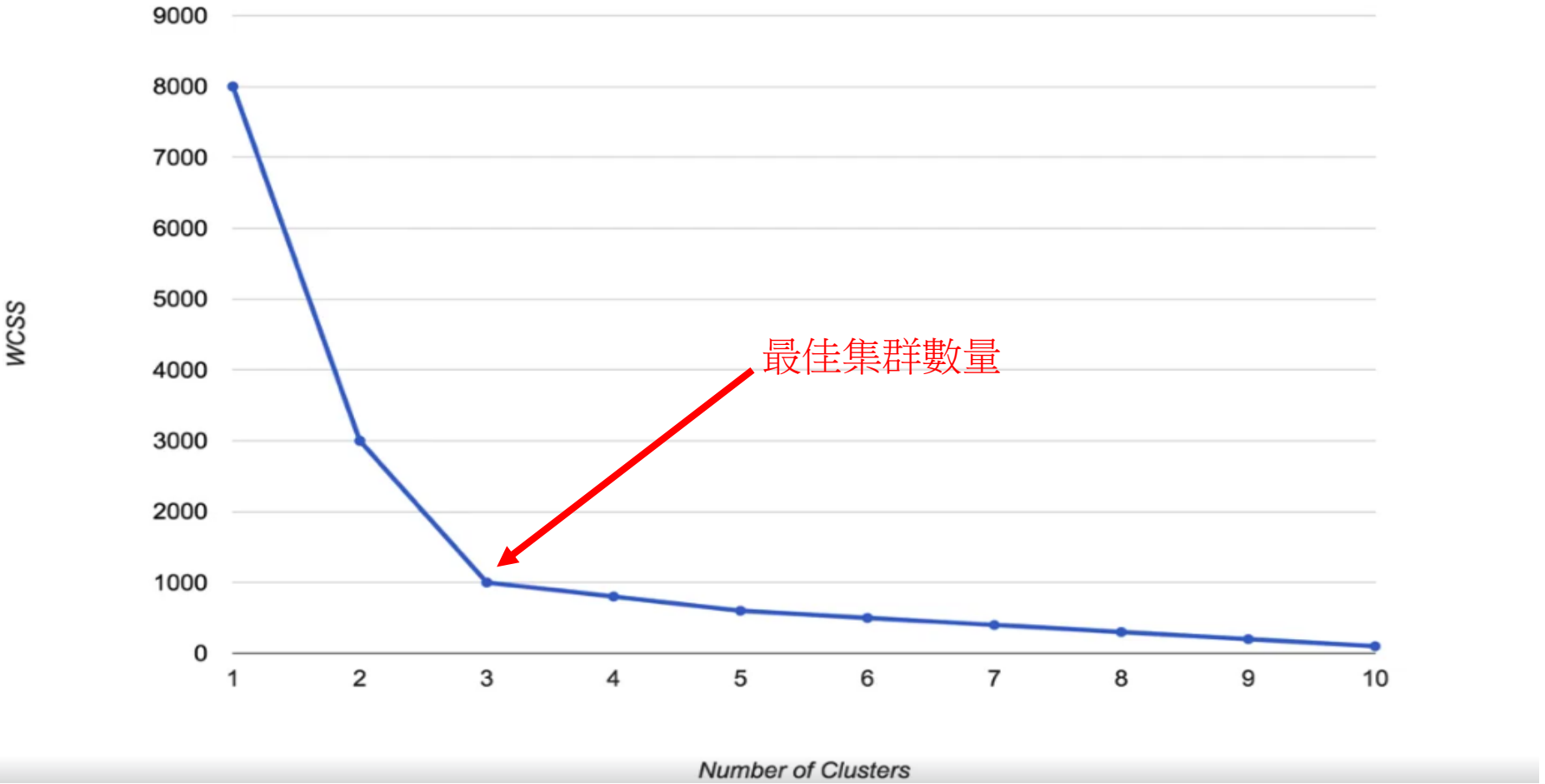
$$\text{WCSS} = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2$$

K = 3



$$\text{WCSS} = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2 + \sum_{P_i \text{ in Cluster 3}} \text{distance}(P_i, C_3)^2$$

The Elbow Method (手肘方法)



How can we choose a "good" K for K-means clustering?

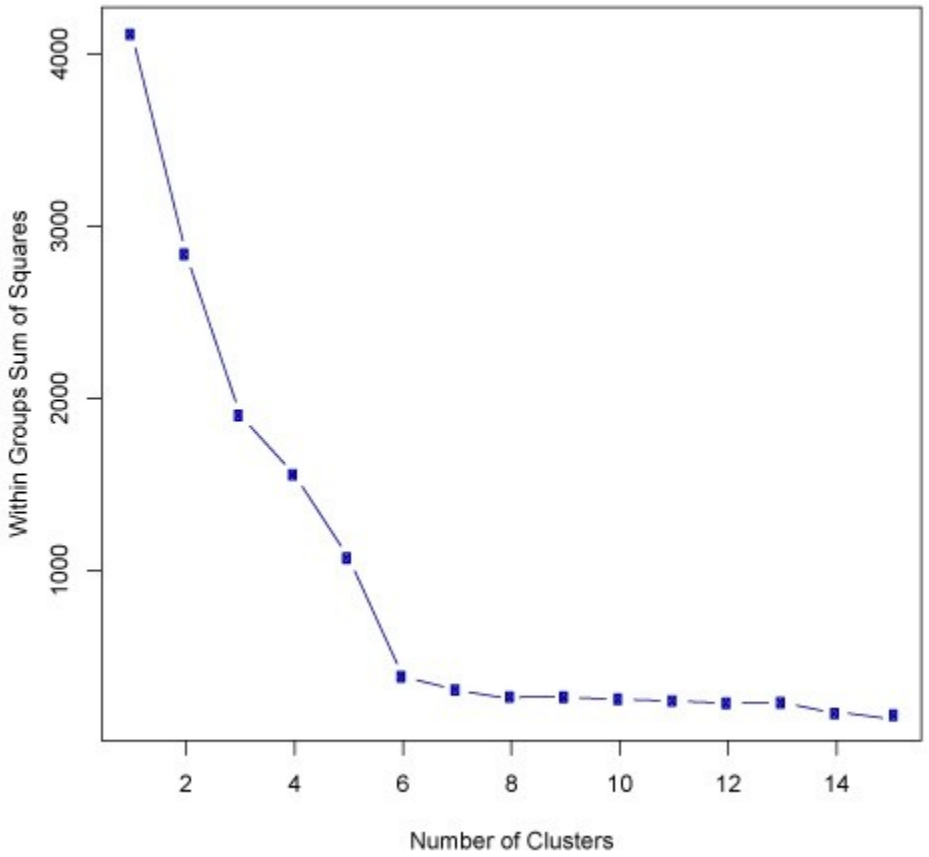
1

指定k值(e.g. L、M、S)

2

不指定K值

使用不同k值，計算點和中心的距離總和，圖顯示k=6是較好的k值



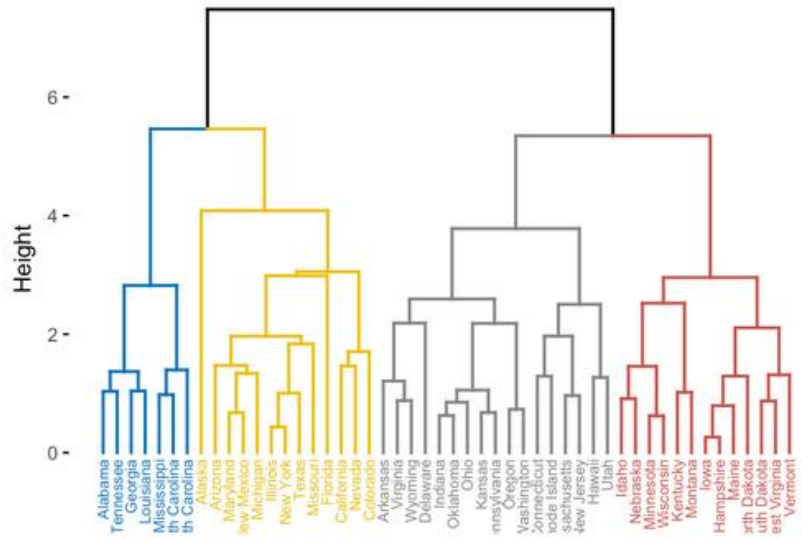
https://www.quora.com/How-can-we-choose-a-good-K-for-K-means-clustering

階層式分群法 (Hierarchical Clustering)

- 聚合式階層分群法 (Bottom-up, agglomerative) : 較常使用
  - 採用聚合的方式，階層式分群法可由樹狀結構的底部開始，將資料或群聚逐次合併。
  - 群由小變大(合併)
- 分裂式階層分群法 (Top-down, divisible) :如同決策樹
  - 採用分裂的方式，則由樹狀結構的頂端開始，將群聚逐次分裂。
  - 群由大變小(分裂)

聚合式階層分群法 (Bottom-up, Agglomerative Clustering)

- 方法
  - 每個資料都視為一群, 再開始計算距離, 較接近的合併成一群, 以此類推.



http://www.sthda.com/english/articles/28-hierarchical-clustering-essentials/94-divisive-hierarchical-clustering-essentials/

距離計算方法

Ward: (預設)
集群聚合後的變異/距離平方和為**最小**

Average
集群中各點與各點資料點的平均距離

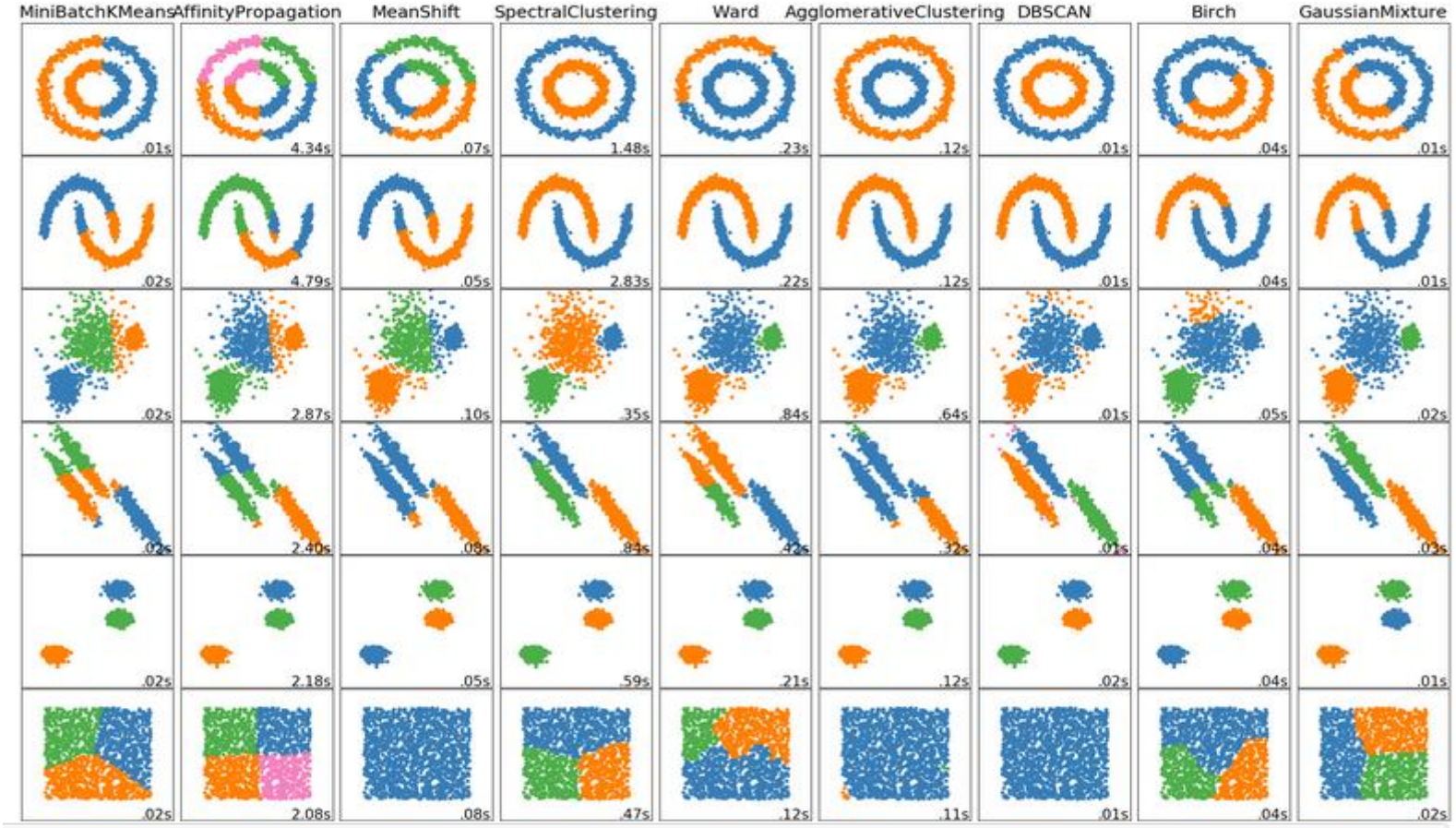
Complete
集群中兩點最遠的距離

Agglomerative Clustering with sklearn

```
class sklearn.cluster.AgglomerativeClustering(n_clusters=2, affinity='euclidean',
memory=None, connectivity=None, compute_full_tree='auto', linkage='ward',
pooling_func='deprecated') ¶
```

http://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html

clustering algorithms

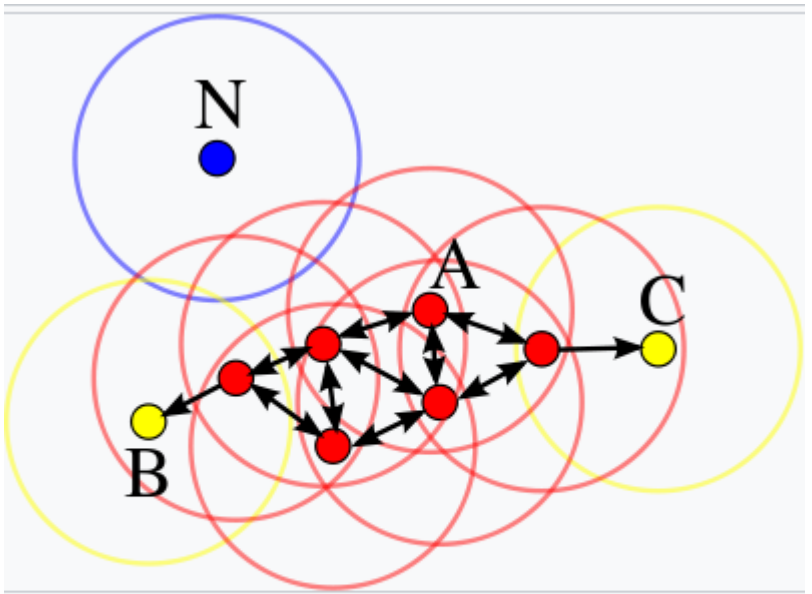


http://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

DBSCAN
(Density-based Spatial Clustering of Applications with Noise)

- 基於密度來做分群
  - 核心點(A)
  - 邊緣點(B, C)
  - 雜訊點(N)

- 集群方法
  - 二個核心點距離小於半徑, 合併成一個新的集群,而邊緣點被併入新的集群中



在這幅圖裡，minPts = 4，點 A 和其他紅色點是核心點，因為它們的 ε-鄰域（圖中紅色圓圈）裡包含最少 4 個點（包括自己），由於它們之間相互相可達，它們形成了一個聚類。點 B 和點 C 不是核心點，但它們可由 A 經其他核心點可達，所以也屬於同一個聚類。點 N 是局外點，它既不是核心點，又不由其他點可達。

DBSCAN

- 優點
  - 不受雜訊點影響
  - 不需事先設定集群數, DBSCAN會自行判斷
  - 有利於有一定密度, 但有特殊形狀來做集群

```
class sklearn.cluster. DBSCAN (eps=0.5, min_samples=5, metric='euclidean',
metric_params=None, algorithm='auto', leaf_size=30, p=None, n_jobs=None) ¶
```

http://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

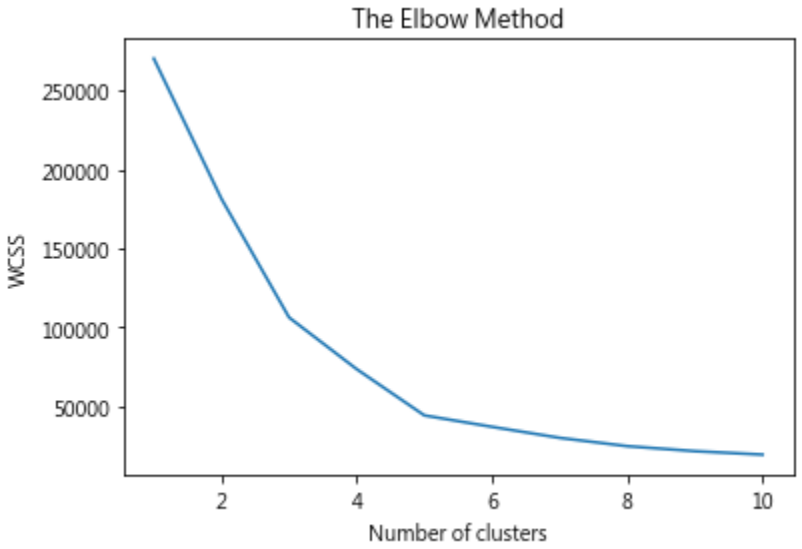
Conclusion

- 群集分析乃是一種有效、非監督式的學習技術，可運用在許多商業狀況中，將資料區隔為有意義的小群組。
- K平均演算法是一種用於反覆區隔資料的簡單統計技術。不過，只有一種試探技術可用來選擇合適的群集數量。

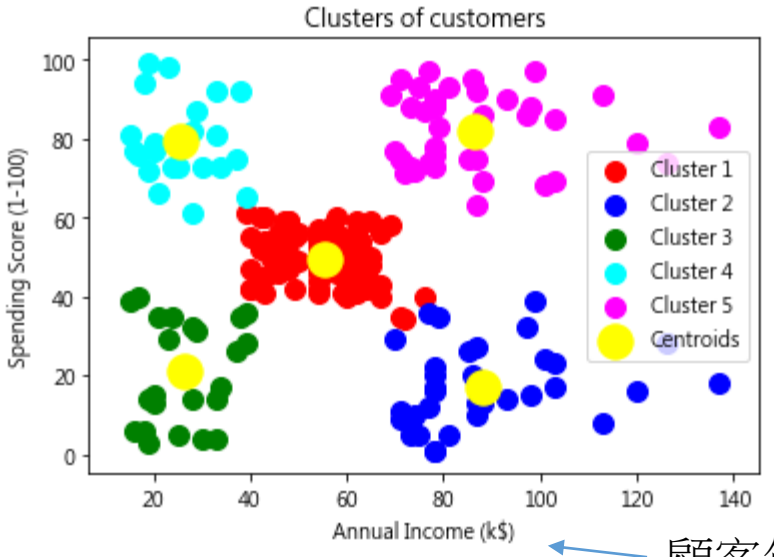
實作時間

- 程式
  - 購物中心顧客分群.ipynb
  - 人口密度.ipynb
- 資料集
  - Mall_Customers
  - Land

購物中心顧客分群



來店花費分數

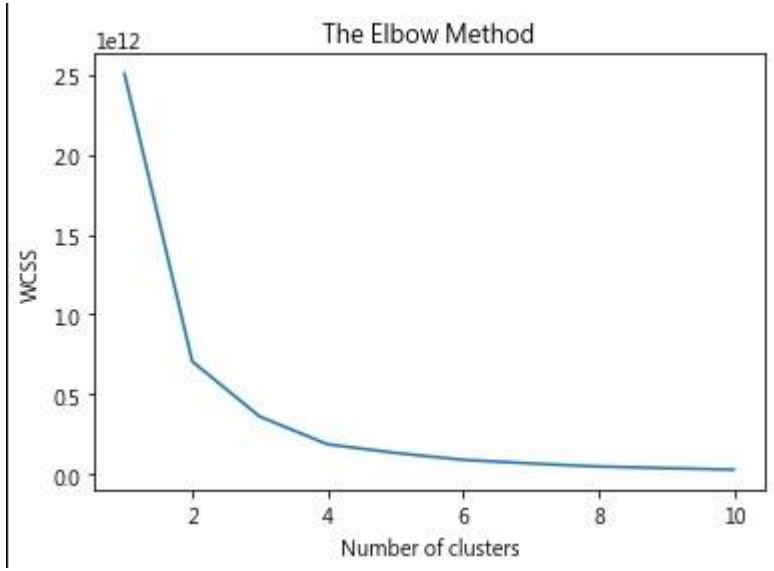


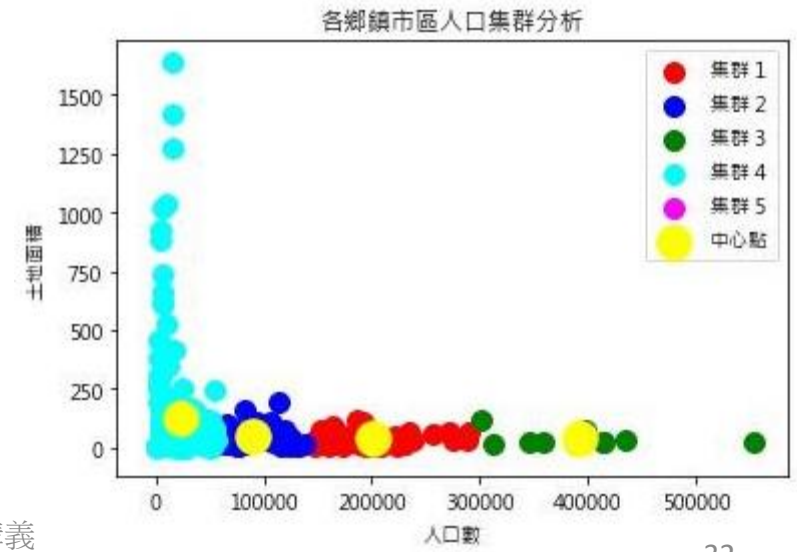
顧客年收入

為了學習非監督式機器學習演算法，下載民國105年各鄉鎮市區人口密度資料集，並採用集群分析法(Cluster Analysis)，分析國內各鄉鎮105年底人口數與土地面積二個變數，是否會形成集群情況。經資料清理，資料共計369筆。研究採用K-means法，首先求算出k值(k=4)，再進行分析。

研究結果顯示國內土地最大的的區域，呈現出人口數較少情況(集群4)，集群3則顯示出近40萬的人口鄉鎮(超過40萬人口有新北市板橋區、桃園市桃園區、新北市中和區、新北市新莊區)，卻僅使用不到250平方公里的土地。

資料集來源：http://data.gov.tw/node/8410





版權聲明

- 本講義所使用之圖片, 表格, 文字, 內容, 書籍資料, 引用統計資料與程式碼及數據集資料等, 除自製外，其智慧財產權為原網站, 作者, 公司所擁有。

- 講義投影片, 程式碼與數據集僅供教學使用, 請同學勿將課程所使用之講義投影片, 程式碼與數據集放在網路上供人下載及分享, 也請勿做商業用途。