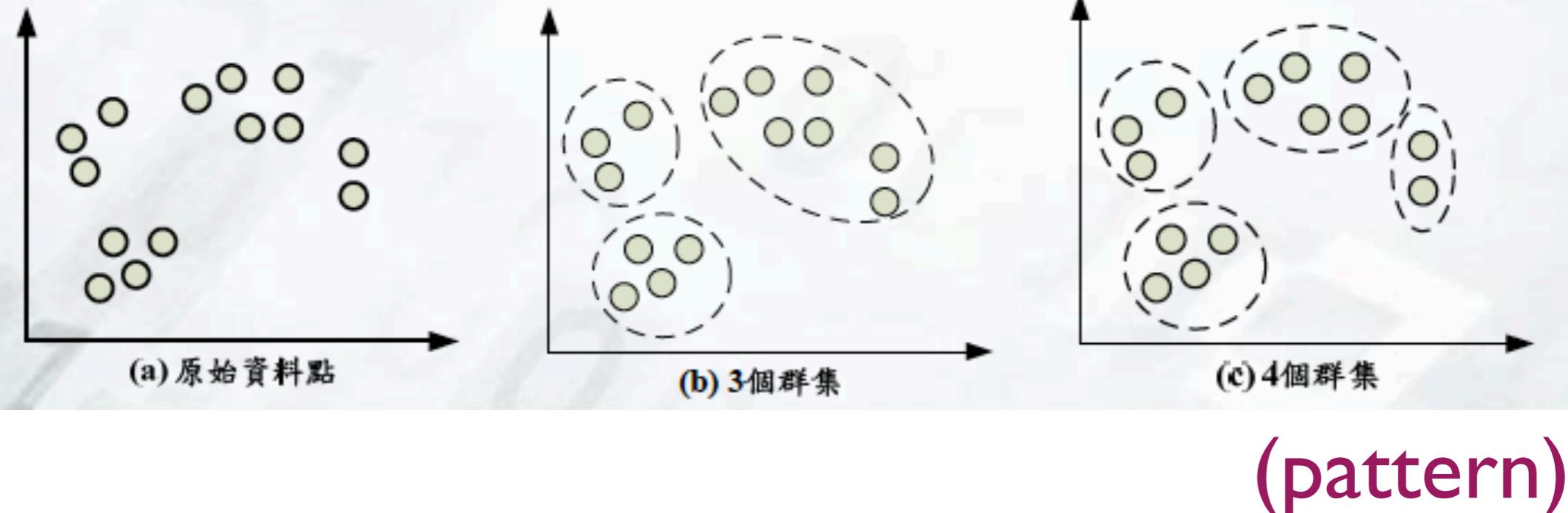


# Machine Learning

## Lecture 9 Clustering

# Unsupervised Learning - Clustering

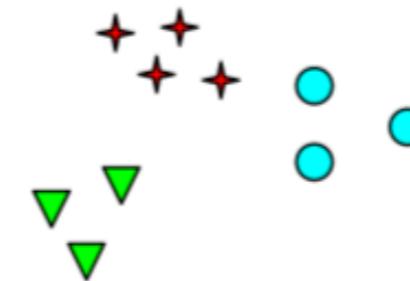
- 依據資料**相似度(similarity)**或**相異度(dissimilarity)**將資料分群歸屬到數個群集(clusters)
- 使同一群內的資料或個體相似程度大，各群間的相似程度小
- 事先並不知道群集數目，分群結果的特徵及其所代表的意義僅能事後加以解釋
- 群集分析為非監督式學習；分類方法為監督式學習



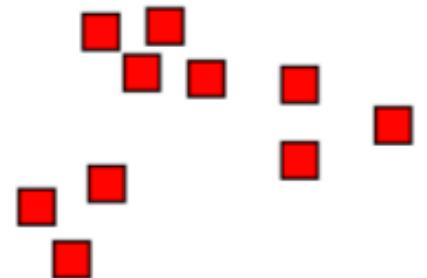
# Clustering



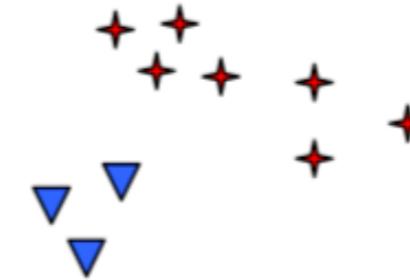
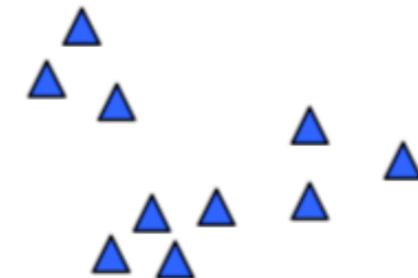
Original Data Set



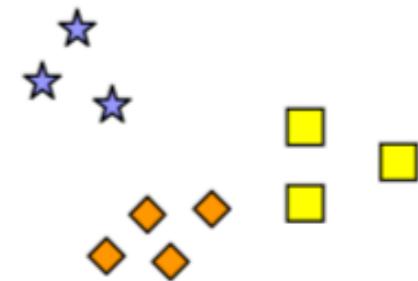
Six Clusters



Two Clusters



Four Clusters



聚類其實是模稜兩可的事情  
要定義分群取決於資料的特性與使用者的預設立場

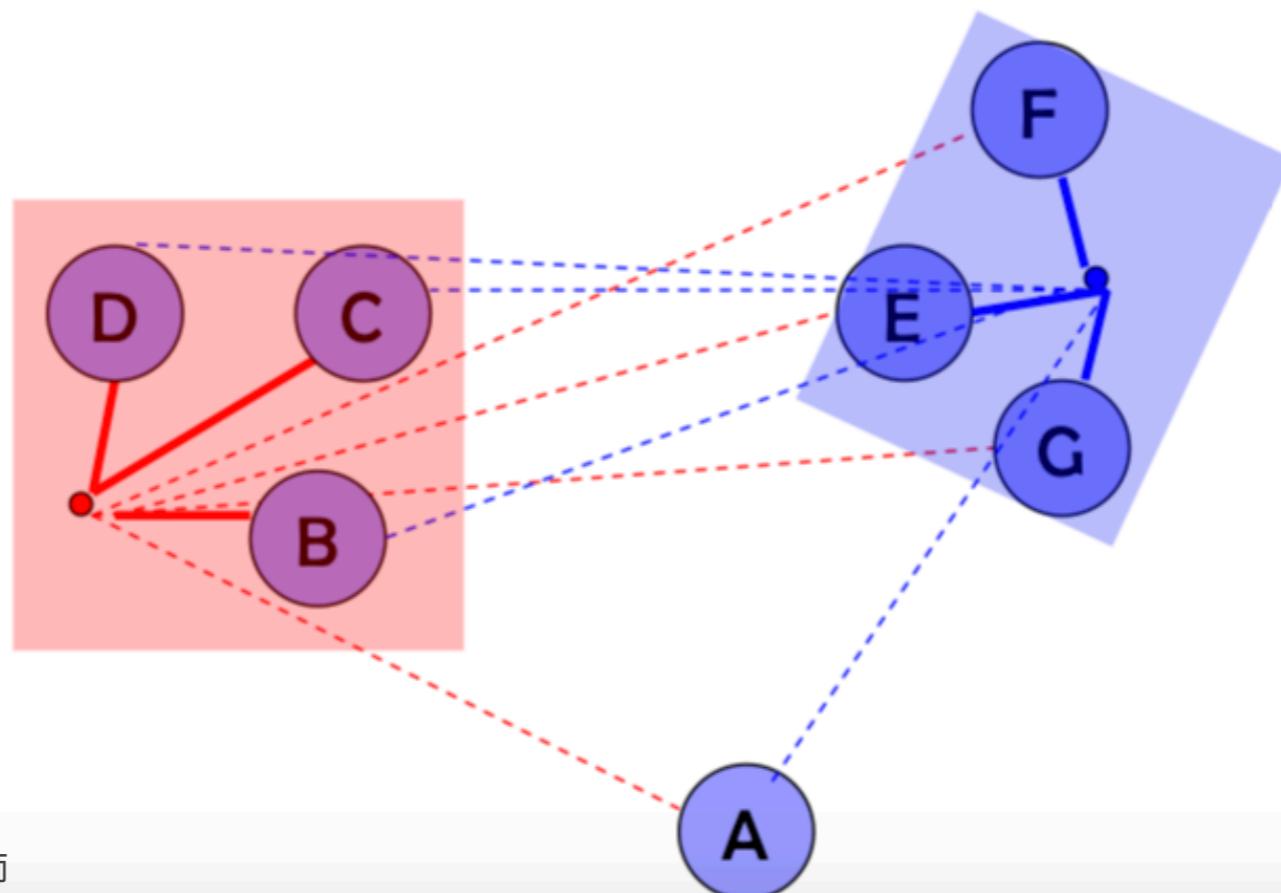
# Clustering

- 聚類分析演算法種類繁多，主要有以下幾類：
  - 分割方法 (Partitioning Methods)
  - 階層式的方法 (Hierarchical Methods)
  - 基於密度的方法(Density-based Methods)
  - 基於網格的方法(Grid-based Methods)
  - 基於模型的方法 (Model-based Methods)
  - ...
- 實際應用中的聚類演算法，往往是上述聚類方法中多種方法的整合

# 分割式聚類

## ■ 主要概念：

- 事先挑選**聚類核心**和訂定**臨界值**，所有Objects與該聚類核心之距離只要沒有超過臨界值，一律歸併入該聚類內，否則屬於其它聚類。
- 採**不重疊**的方式劃分，將原有的資料分到不同的聚類中



# 常用距離公式

- $i=(x_{i1}, x_{i2}, \dots, x_{ip})$  和  $j=(x_{j1}, x_{j2}, \dots, x_{jp})$  是兩個  $p$  維資料對象
- 歐幾里得(Euclidean)距離

$$d(i, j) = \sqrt{(|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \dots + |x_{ip} - x_{jp}|^2)}$$

- 曼哈頓(Manhattan)距離

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{ip} - x_{jp}|$$

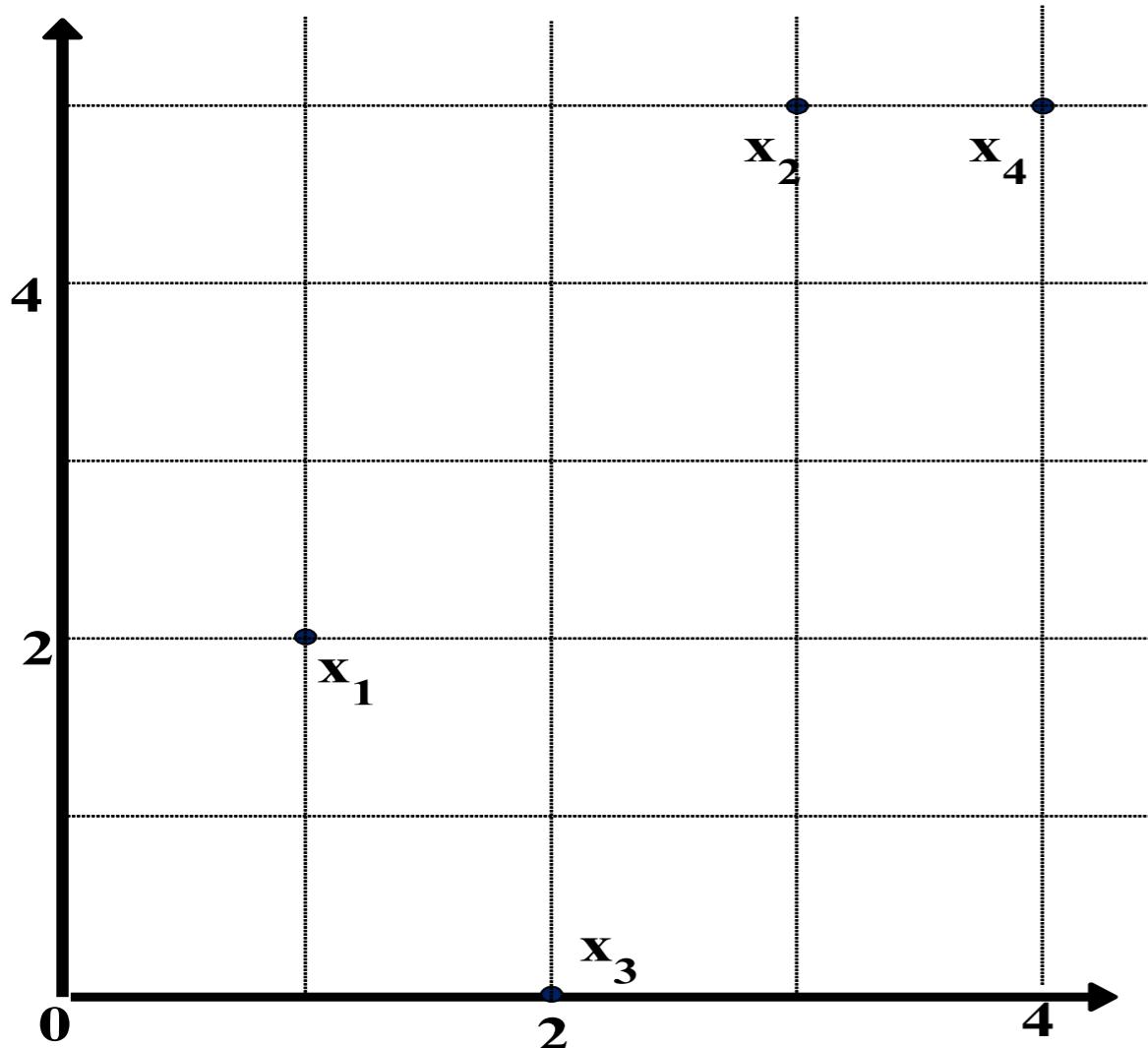
- 明可夫斯基(Minkowski)距離

$$d(i, j) = \sqrt[q]{(|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \dots + |x_{ip} - x_{jp}|^q)}$$

- 上式中， $q$ 為正整數，如果  $q=1$  則表示 Manhattan 距離，如果  $q=2$  則表示 Euclidean 距離

# Example

| point | attribute 1 | attribute 2 |
|-------|-------------|-------------|
| x1    | 1           | 2           |
| x2    | 3           | 5           |
| x3    | 2           | 0           |
| x4    | 4           | 5           |



相異度矩陣 (dissimilarity matrix)

Manhattan ( $L_1$ )

| L  | x1 | x2 | x3 | x4 |
|----|----|----|----|----|
| x1 | 0  |    |    |    |
| x2 | 5  | 0  |    |    |
| x3 | 3  | 6  | 0  |    |
| x4 | 6  | 1  | 7  | 0  |

Euclidean ( $L_2$ )

| L2 | x1   | x2  | x3   | x4 |
|----|------|-----|------|----|
| x1 | 0    |     |      |    |
| x2 | 3.61 | 0   |      |    |
| x3 | 2.24 | 5.1 | 0    |    |
| x4 | 4.24 | 1   | 5.39 | 0  |

Supremum ( $L_\infty$ )

| $L^\infty$ | x1 | x2 | x3 | x4 |
|------------|----|----|----|----|
| x1         | 0  |    |    |    |
| x2         | 3  | 0  |    |    |
| x3         | 2  | 5  | 0  |    |
| x4         | 3  | 1  | 5  | 0  |

# 分割式聚類

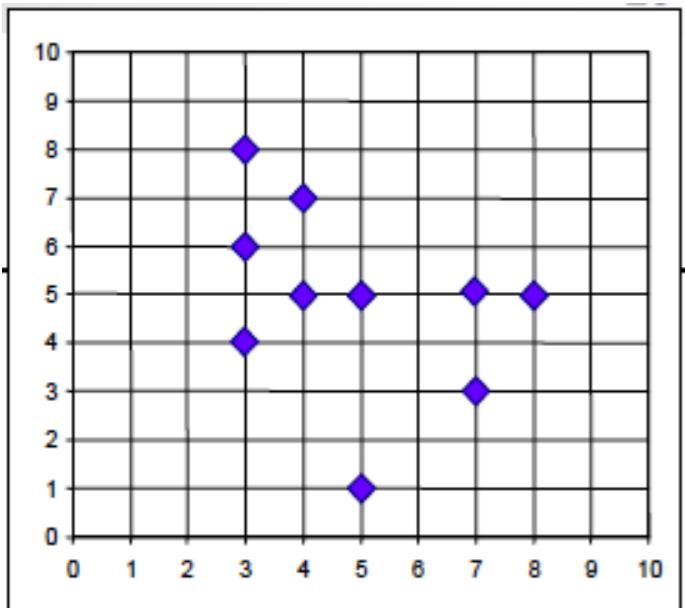
- 紿定一個具有 $n$ 個物件的資料庫，一個分割方法會構建資料的 $k$ 個分割區域，每個區域表示一個聚類，並且 $k \leq n$ 。
  - 每個聚類至少**包含一個物件**
  - 每個物件對象**屬於且僅屬於一個聚類**
- **分割準則**：同一個聚類中的對象儘可能的**接近或相關**，不同聚類中的對象儘可能的**遠離或不同**
- 分割式聚類方法的種類：
  - k-平均演算法 (k-Means; 又可稱為c-Means)
  - Fuzzy C-Means
  - ...

# K-Means

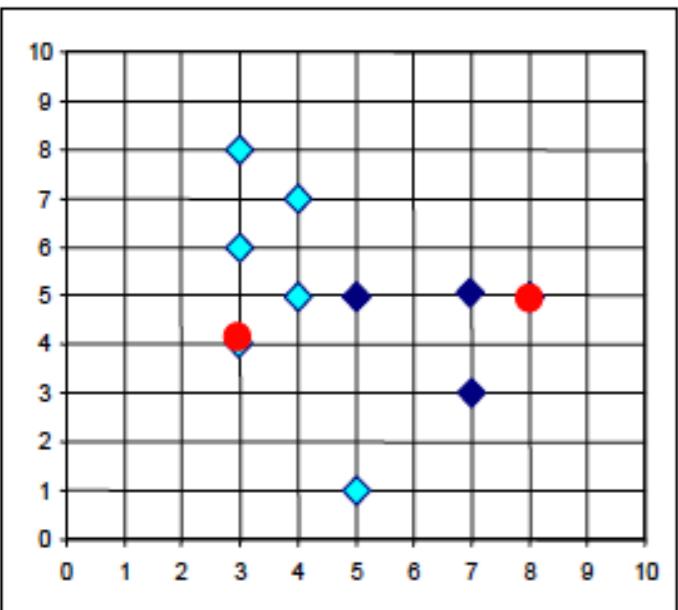
## ■ 運作概念簡介：

- 將n筆待分群資料依資料特性，**選出K個資料點**（假若資料可被分成K個聚類），而這K個資料點即為**K個聚類的中心點(Centroid；亦可稱為質心)**。
  - 中心點(質心)通常是**計算出來**的。不過在實作時，初始中心點可選用現有的資料點來表示，或是隨機挑選可行座標點出來。
- 將所有資料與此K個中心點做**距離運算**。若某資料j與K個中心點當中之一（例如：中心點  $C_i$ ）距離最近，則此資料j可被歸類於該中心點所表示之聚類 i。
- 若所有資料皆被歸類完畢，則**重新計算K個聚類的中心點**。
- 回到第二步驟重新執行，直到**滿足停止條件**為止。

# Example |

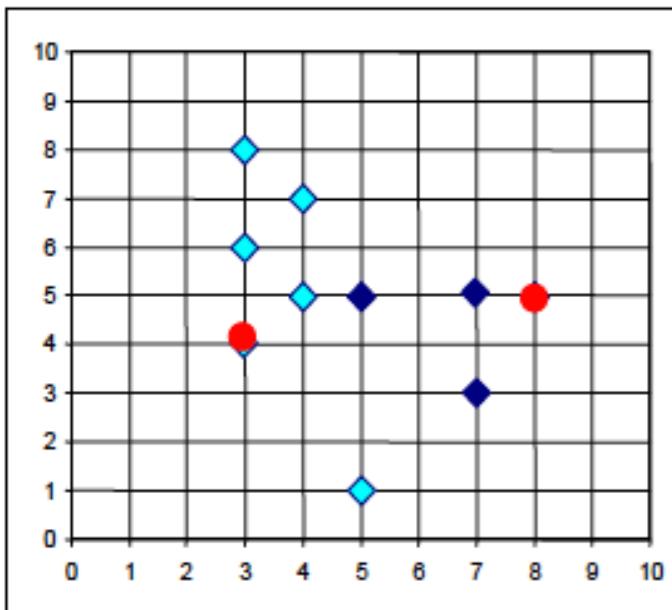


# Example |

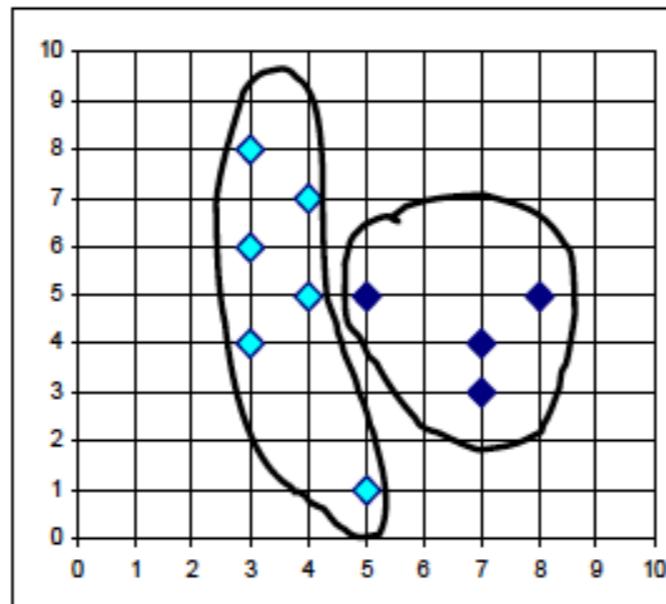


K=2  
↑  
任意選擇 K 個體當  
作起始群組中心

# Example |



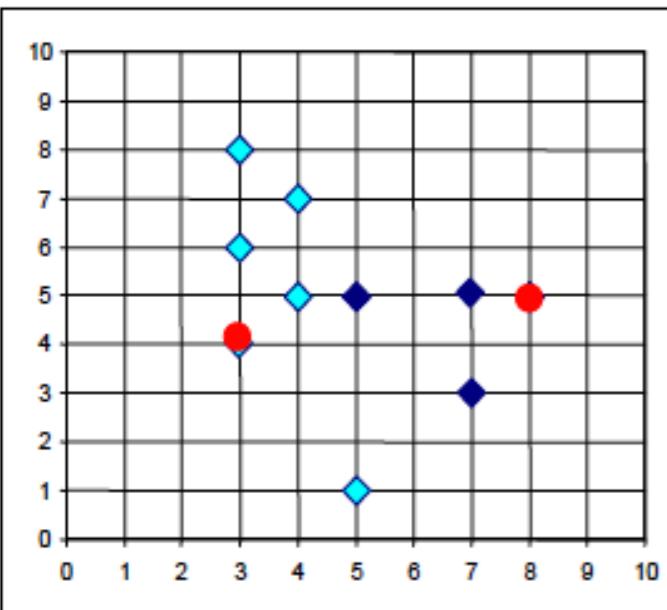
將每個  
個體分  
配至最  
接近中  
心



K=2

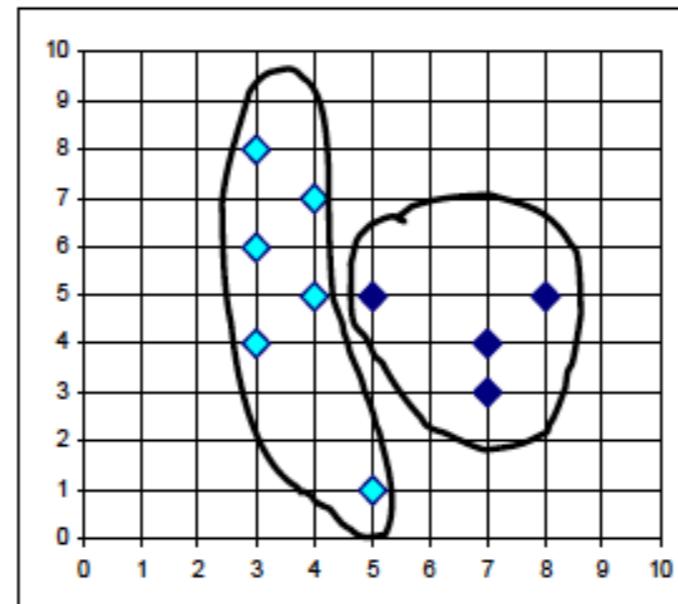
任意選擇 K 個體當  
作起始群組中心

# Example |

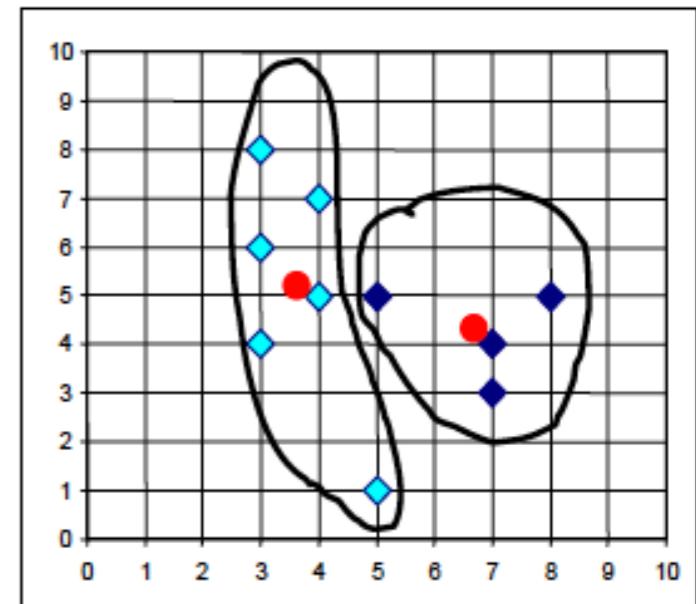


K=2  
任意選擇 K 個體當  
作起始群組中心

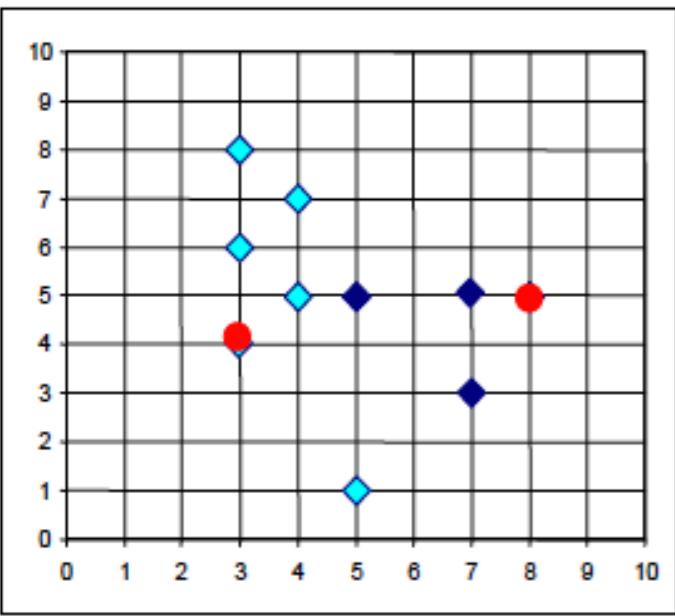
將每個  
個體分  
配至最  
接近中  
心



更新群  
組均值



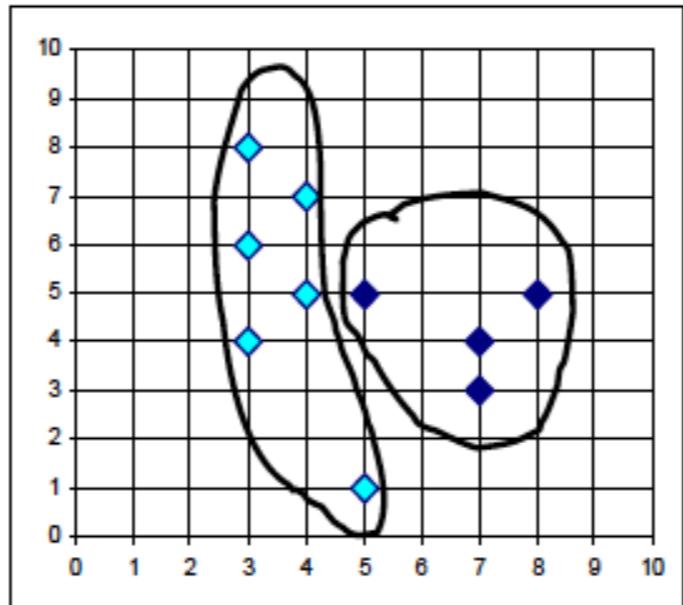
# Example |



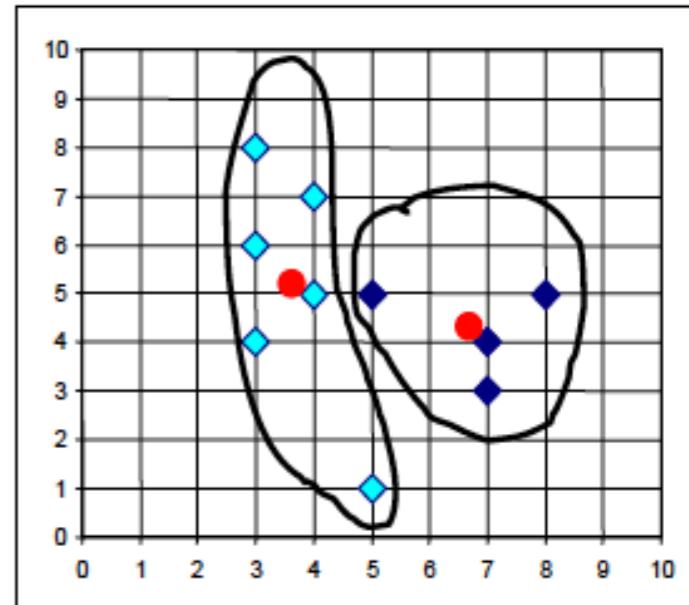
任意選擇  $K$  個體當  
作起始群組中心

$K=2$

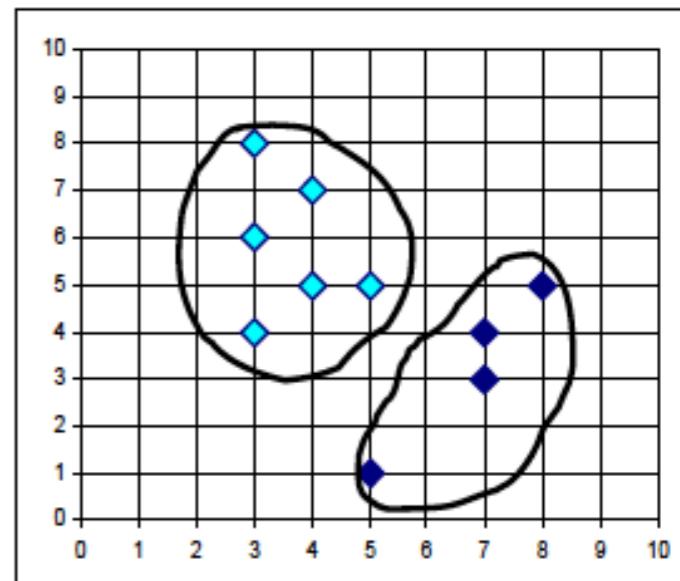
將每個  
個體分  
配至最  
接近中  
心



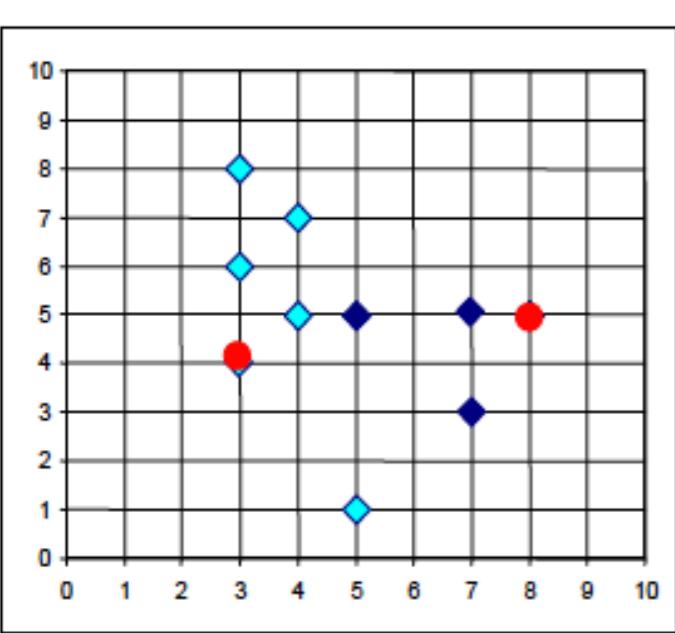
更新群  
組均值



針對所有資  
料重新分配



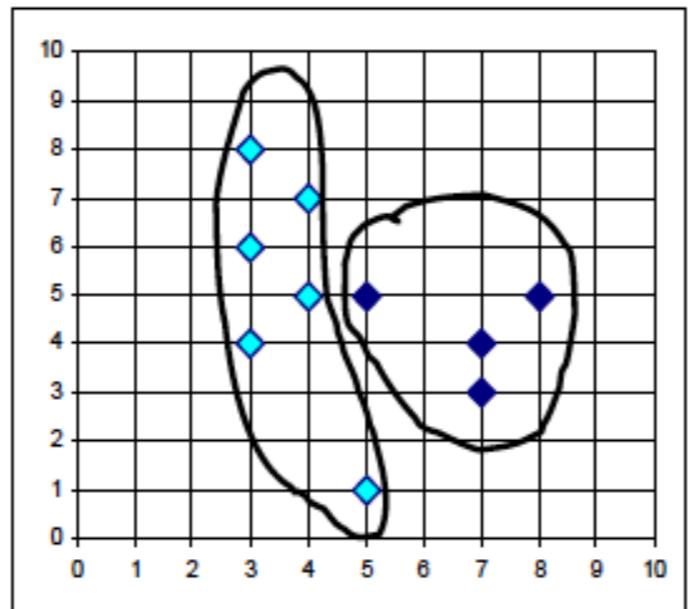
# Example |



任意選擇  $K$  個體當  
作起始群組中心

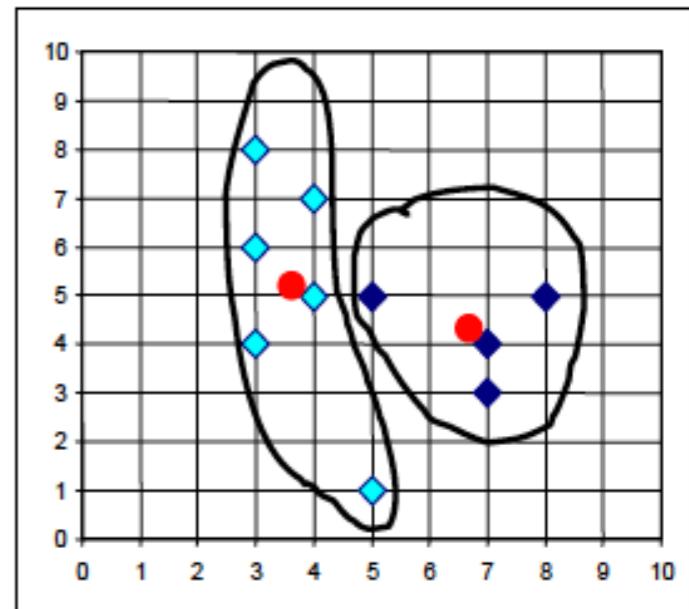
$K=2$

將每個  
個體分  
配至最  
接近中  
心



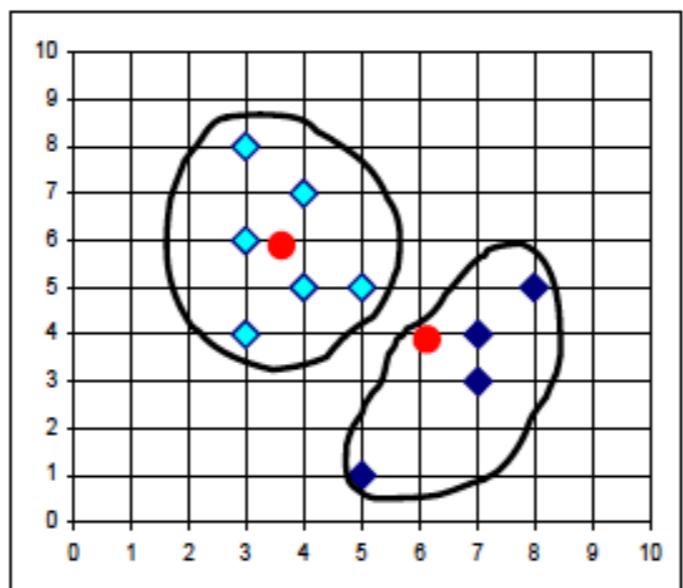
針對所有資  
料重新分配

更新群  
組均值



針對所有資  
料重新分配

更新群  
組均值



# Exercise

| i | X | Y |
|---|---|---|
| A | 1 | 1 |
| B | 1 | 0 |
| C | 0 | 3 |
| D | 2 | 4 |
| E | 3 | 2 |

# Exercise

|   | i | X | Y | d( i, A ) | d( i, C ) |
|---|---|---|---|-----------|-----------|
| 1 | A | 1 | 1 |           |           |
| 2 | B | 1 | 0 |           |           |
| 3 | C | 0 | 3 |           |           |
| 4 | D | 2 | 4 |           |           |
| 5 | E | 3 | 2 |           |           |

# Exercise

|   | i | X | Y | $d( i, A )$ | $d( i, C )$ | Cluster |
|---|---|---|---|-------------|-------------|---------|
| 1 | A | 1 | 1 | 0           | 2.24        | 1       |
|   | B | 1 | 0 | 1           | 3.16        | 1       |
| 2 | C | 0 | 3 | 2.24        | 0           | 2       |
|   | D | 2 | 4 | 3.16        | 2.24        | 2       |
|   | E | 3 | 2 | 2.24        | 3.16        | 1       |

# Exercise

| i | X | Y |
|---|---|---|
| A | 1 | 1 |
| B | 1 | 0 |
| C | 0 | 3 |
| D | 2 | 4 |
| E | 3 | 2 |

$$X_1 = \left( \frac{1+1+3}{3}, \frac{1+0+2}{3} \right) \\ = (1.67, 1)$$

$$X_2 = \left( \frac{0+2}{2}, \frac{3+4}{2} \right) \\ = (1, 3.5)$$

# Exercise

$$X_1 = (1.67, 1)$$

$$X_2 = (1, 3.5)$$

| i      | X | Y | d( i, X1 ) | d( i, X2 ) | Cluster |
|--------|---|---|------------|------------|---------|
| 1<br>A | 1 | 1 |            |            |         |
| 2<br>B | 1 | 0 |            |            |         |
| 3<br>C | 0 | 3 |            |            |         |
| 4<br>D | 2 | 4 |            |            |         |
| 5<br>E | 3 | 2 |            |            |         |

# Exercise

$$X_1 = (1.67, 1)$$

$$X_2 = (1, 3.5)$$

| i      | X | Y | d( i, X1 ) | d( i, X2 ) | Cluster |
|--------|---|---|------------|------------|---------|
| 1<br>A | 1 | 1 | 0.67       | 2.5        | 1       |
| 2<br>B | 1 | 0 | 1.2        | 3.5        | 1       |
| 2<br>C | 0 | 3 | 2.61       | 1.12       | 2       |
| 2<br>D | 2 | 4 | 3.02       | 1.12       | 2       |
| 2<br>E | 3 | 2 | 1.66       | 2.5        | 1       |

# K-Means 的缺點

- 無法直接處理類別型資料，可改用 **K眾數法 (K-mode)**

# K-mode

## ■ 衡量相似度：簡單匹配方法

$$d(i, j) = \frac{p - m}{p}$$

- **m: 匹配的數目，即對象i和j取值相同的變數的數目 (也可加上權重)**
- **p: 類別變數的個數**

# Example: 簡單匹配方法

- 有一組類別變數的資料表示如下：

| 個體編號 | 屬性1 | 屬性2 |
|------|-----|-----|
| 1    | 黃   | 圓領  |
| 2    | 綠   | 圓領  |
| 3    | 藍   | 高領  |
| 4    | 黃   | 尖領  |
| 5    | 綠   | 圓領  |

- 假設建構一個 $5 \times 5$ 相異矩陣(如下左)，利用簡單匹配方法可計算出該矩陣之所有值(如下右)：

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ d(4,1) & d(4,2) & d(4,3) & 0 & \\ d(5,1) & d(5,2) & d(5,3) & d(5,4) & 0 \end{bmatrix} \xrightarrow{\quad} \begin{bmatrix} 0 & & & & \\ 1/2 & 0 & & & \\ 1 & 1 & 0 & & \\ 1/2 & 1/2 & 1 & 0 & \\ 1/2 & 0 & 1 & 1 & 0 \end{bmatrix}$$

# K-mode

## ■ 衡量相似度：簡單匹配方法

$$d(i, j) = \frac{p - m}{p}$$

- m: 匹配的數目，即對象i和j取值相同的變數的數目 (也可加上權重)
- p: 類別變數的個數
- 群集中心：眾數
- 頻率為基礎(frequency-based)的方法

# K-Means 的缺點

- 無法直接處理類別型資料，可改用 **K眾數法 (K-mode)**
- 須事先決定群集數目
- 易受到離群值或雜訊影響，可改用 **K中心點法 (K-medoids)**

以群集中最接近中心位置的資料點作為群集中心

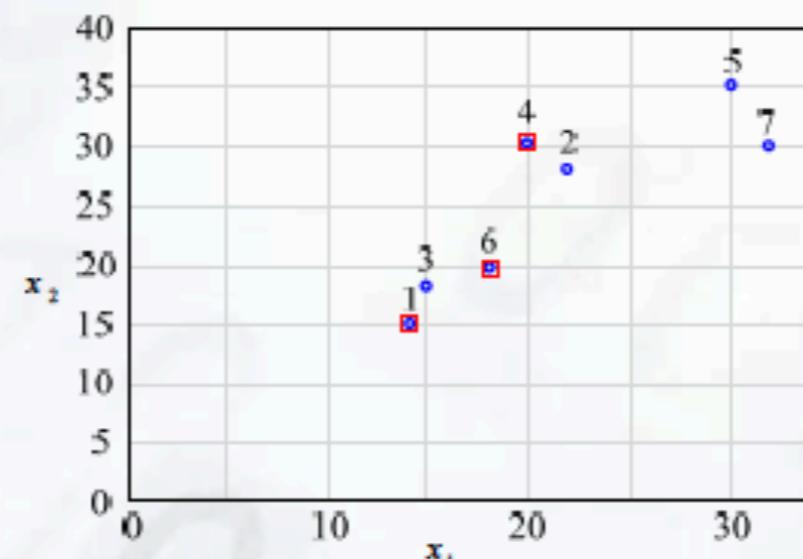
# Example

- 假設起始群集數  $k$  設為 3，以歐式距離平方作為衡量相似度的依據，先隨機選取資料 1、4、6 作為群集中心

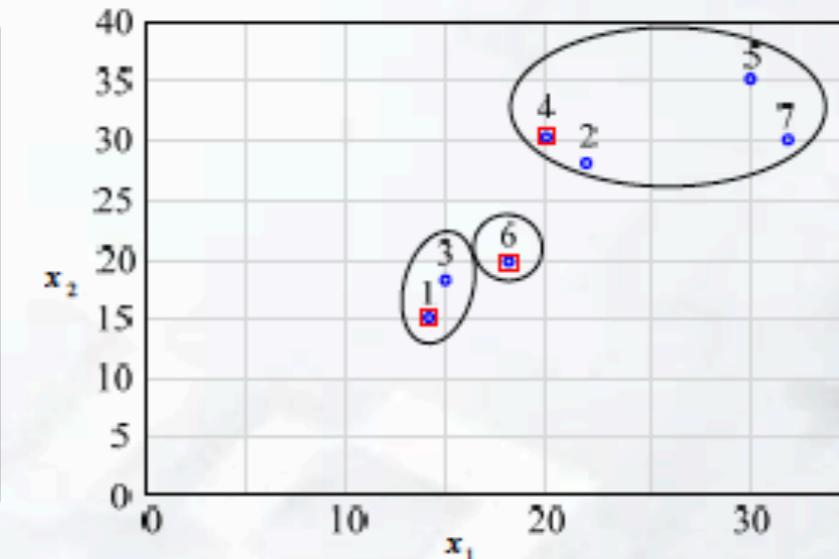
## ■ 步驟1

Sum = 287

| # | 與群集 A (1)<br>的相異度 | 與群集 B (4)<br>的相異度 | 與群集 C (6)<br>的相異度 | 最小相<br>異度 | 分配的<br>群集 |
|---|-------------------|-------------------|-------------------|-----------|-----------|
| 1 | 0                 | 261               | 41                | 0         | A         |
| 2 | 233               | 8                 | 80                | 8         | B         |
| 3 | 10                | 169               | 13                | 10        | A         |
| 4 | 261               | 0                 | 104               | 0         | B         |
| 5 | 656               | 125               | 369               | 125       | B         |
| 6 | 41                | 104               | 0                 | 0         | C         |
| 7 | 549               | 144               | 296               | 144       | B         |



(a) 選擇起始點

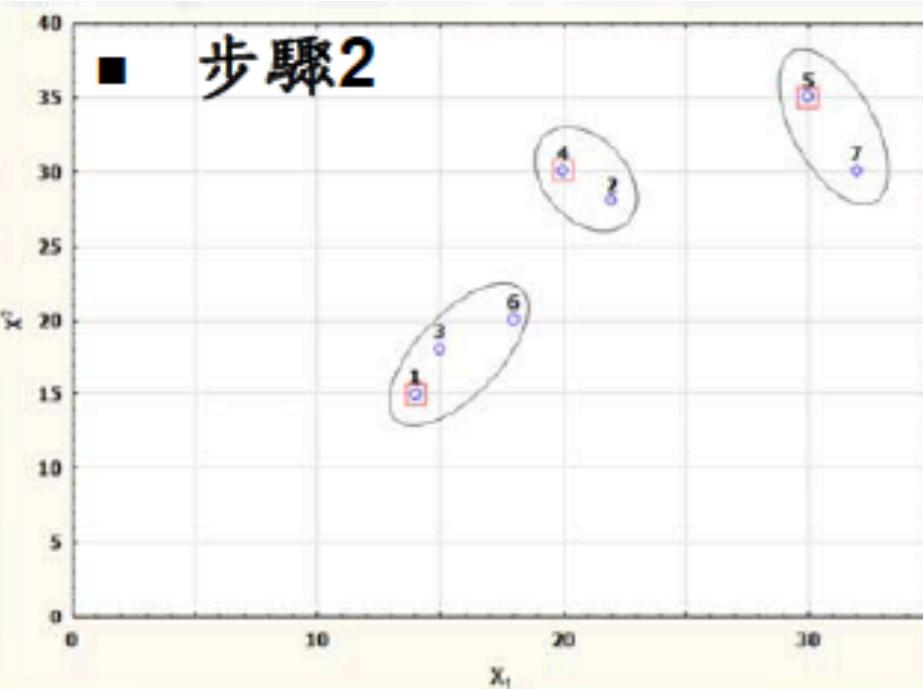


(b) 劃分群集

# Example

選取任一非群集中心的資料點，取代任一群集中心，  
若總距離改變量  $< 0$ ，則取代原有的群集中心

ex. 以資料點 5 取代群集 C 的中心資料點 6



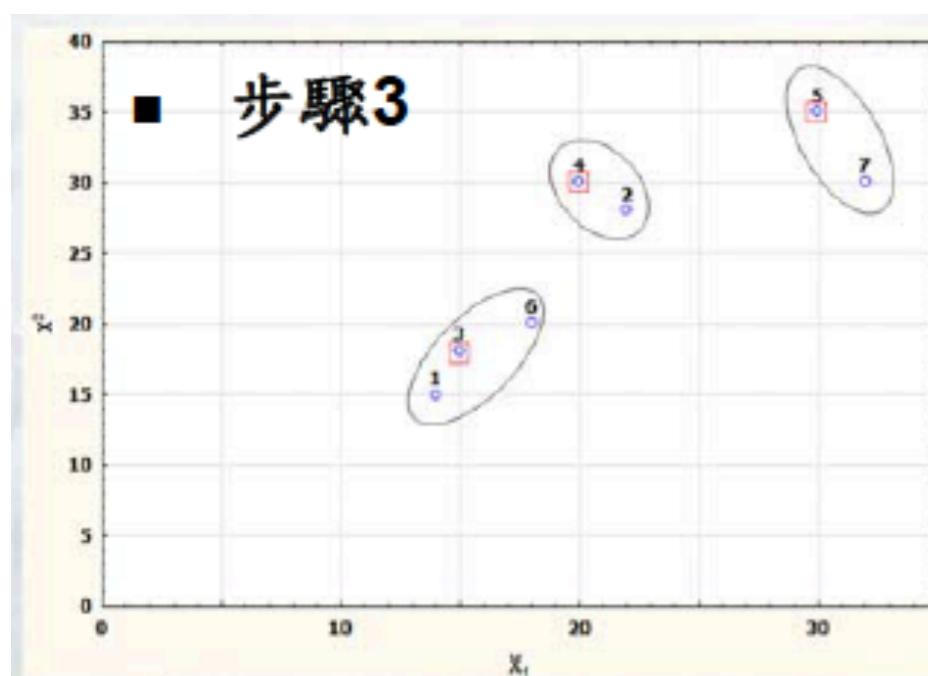
| # | 與群集A (1)<br>的相異度 | 與群集B (4)<br>的相異度 | 與群集C (5)<br>的相異度 | 最小相<br>異度 | 分配的<br>群集 |
|---|------------------|------------------|------------------|-----------|-----------|
| 1 | 0                | 261              | 656              | 0         | A         |
| 2 | 233              | 8                | 113              | 8         | B         |
| 3 | 10               | 169              | 514              | 10        | A         |
| 4 | 261              | 0                | 125              | 0         | B         |
| 5 | 656              | 125              | 0                | 0         | C         |
| 6 | 41               | 104              | 369              | 41        | A         |
| 7 | 549              | 144              | 29               | 29        | C         |

$$\text{Sum} = 88$$

總距離改變量  $S = 88 - 287 = -199 < 0$ , 所以將資料點 5 作為群集 C 的中心

# Example

如此不斷重複選擇資料點取代原有中心，直到群集中心不再變動為止



| # | 與群集A (3)<br>的相異度 | 與群集B (4)<br>的相異度 | 與群集C (5)<br>的相異度 | 最小相<br>異度 | 分配的<br>群集 |
|---|------------------|------------------|------------------|-----------|-----------|
| 1 | 10               | 261              | 656              | 10        | A         |
| 2 | 149              | 8                | 113              | 8         | B         |
| 3 | 0                | 169              | 514              | 0         | A         |
| 4 | 169              | 0                | 125              | 0         | B         |
| 5 | 514              | 125              | 0                | 0         | C         |
| 6 | 13               | 104              | 369              | 13        | A         |
| 7 | 433              | 144              | 29               | 29        | C         |

# K-Means 的缺點

- 無法直接處理類別型資料，可改用 **K眾數法 (K-mode)**
- 須事先決定群集數目
- 易受到離群值或雜訊影響，可改用 **K中心點法 (K-medoids)**
- 起始群集中心選擇的不同會造成不同的分群結果

# K-means++ 演算法

1. 初始化一個空的集合M來儲存被選取的質心
2. 從輸入的樣本中，隨機選取第一個質心  $\mu^{(j)}$ ，並加入M
3. 對於每一個不在M裡的樣本  $x^{(i)}$  計算出對M中所有質心的最小距離平方  $d(x^{(i)}, M)^2$
4. 使用加權機率分配 
$$\frac{d(x^{(i)}, M)^2}{\sum_i d(x^{(i)}, M)^2}$$
來隨機選取下一個質心，並加入M
5. 重複步驟3, 4直到選取了k個質心
6. 以古典K-means演算法完成後續工作

在選取初始質心時，盡可能讓他們彼此遠離

# K-Means 的缺點

- 無法直接處理類別型資料，可改用 **K眾數法 (K-mode)**
- 須事先決定群集數目
- 易受到離群值或雜訊影響，可改用 **K中心點法 (K-medoids)**
- 起始群集中心選擇的不同會造成不同的分群結果
- 無法適用於所有的資料群集形態

# Python code

## ■ 利用sklearn生成數據

```
from sklearn.datasets import make_blobs
X, y = make_blobs(n_samples=150,
                    n_features=2,
                    centers=3,
                    cluster_std=0.5,
                    random_state=0)

import matplotlib.pyplot as plt
plt.scatter(X[:, 0], X[:, 1],
            c='white', marker='o', edgecolor='black', s=50)
plt.grid()
plt.show()
```

隨機生成150個點

密度很高的3群

# Python code

## ■ K-means

```
from sklearn.cluster import KMeans  
km = KMeans(n_clusters=3,  
             init='random',  
             n_init=10,  
             max_iter=300,  
             tol=1e-04,  
             random_state=0)  
  
y_km = km.fit_predict(X)
```

以獨立不同、隨機選取的質心，  
執行 K-means 演算法 10 次，並  
以最低 SSE 的模型作為最終模型

指定每次執行的最大迭代次數

控制“聚類內誤差平方和”的“可  
容許誤差”

某些情況下可能不會收斂，處理此問題的方法是選擇較大的 tol 值

在 Scikit-learn 的 k-means 實作中，若有聚類是空的，則演算法會找出  
與該“空聚類”質心最遠的點，接著將該點重新設定為“空聚類的質心”

在使用“歐式距離”來計算時，要記得做標準化

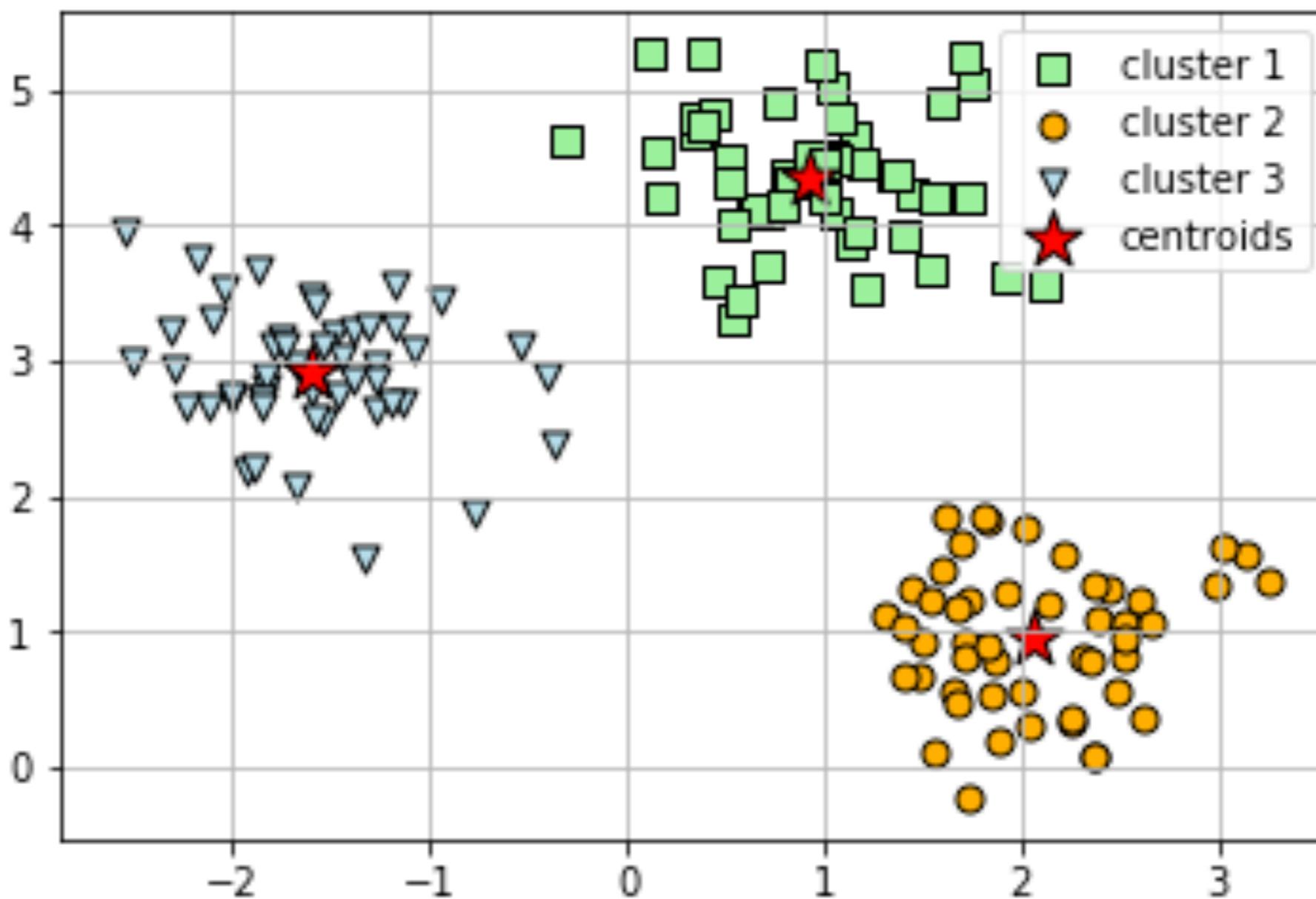
# Python code

## ■ 視覺化K-means所確認的聚類與這些聚類的質心

```
plt.scatter(X[y_km == 0, 0],  
            X[y_km == 0, 1],  
            s=50, c='lightgreen',  
            marker='s', edgecolor='black',  
            label='cluster 1')  
plt.scatter(X[y_km == 1, 0],  
            X[y_km == 1, 1],  
            s=50, c='orange',  
            marker='o', edgecolor='black',  
            label='cluster 2')  
plt.scatter(X[y_km == 2, 0],  
            X[y_km == 2, 1],  
            s=50, c='lightblue',  
            marker='v', edgecolor='black',  
            label='cluster 3')  
plt.scatter(km.cluster_centers_[:, 0],  
            km.cluster_centers_[:, 1],  
            s=250, marker='*',  
            c='red', edgecolor='black',  
            label='centroids')  
plt.legend(scatterpoints=1)  
plt.grid()  
plt.show()
```

質心

# Python code



# Python code

## ■ K-means++

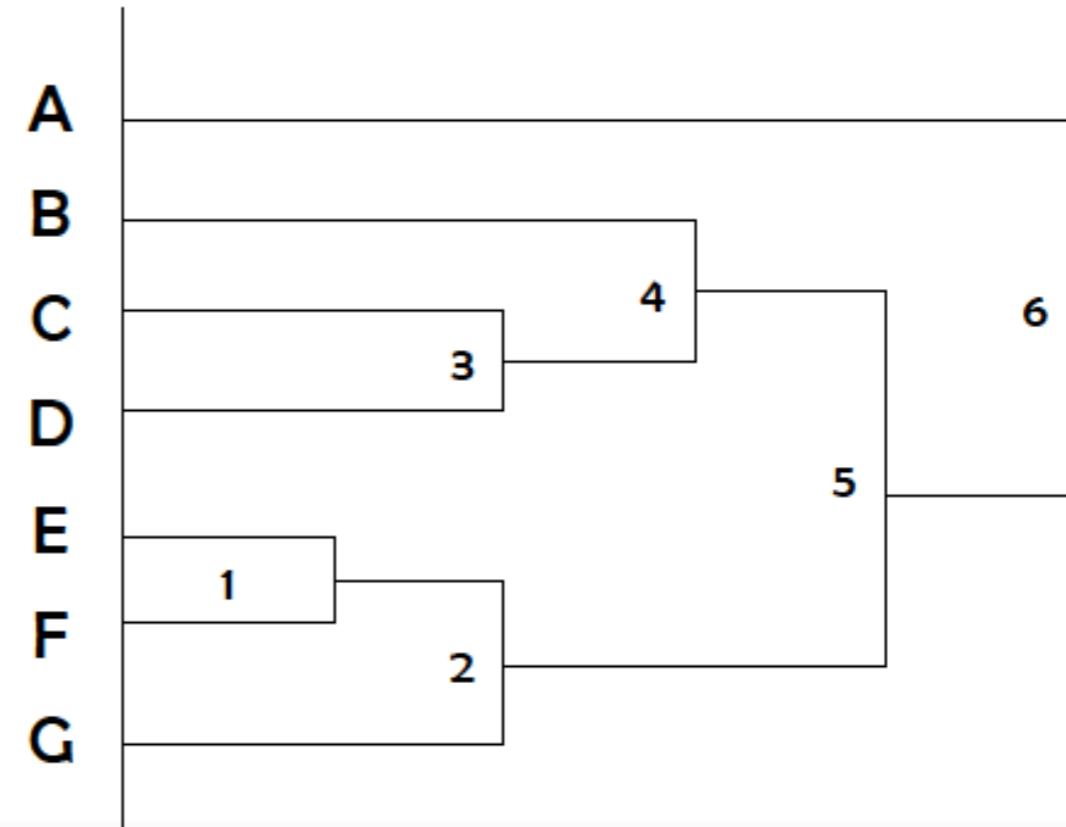
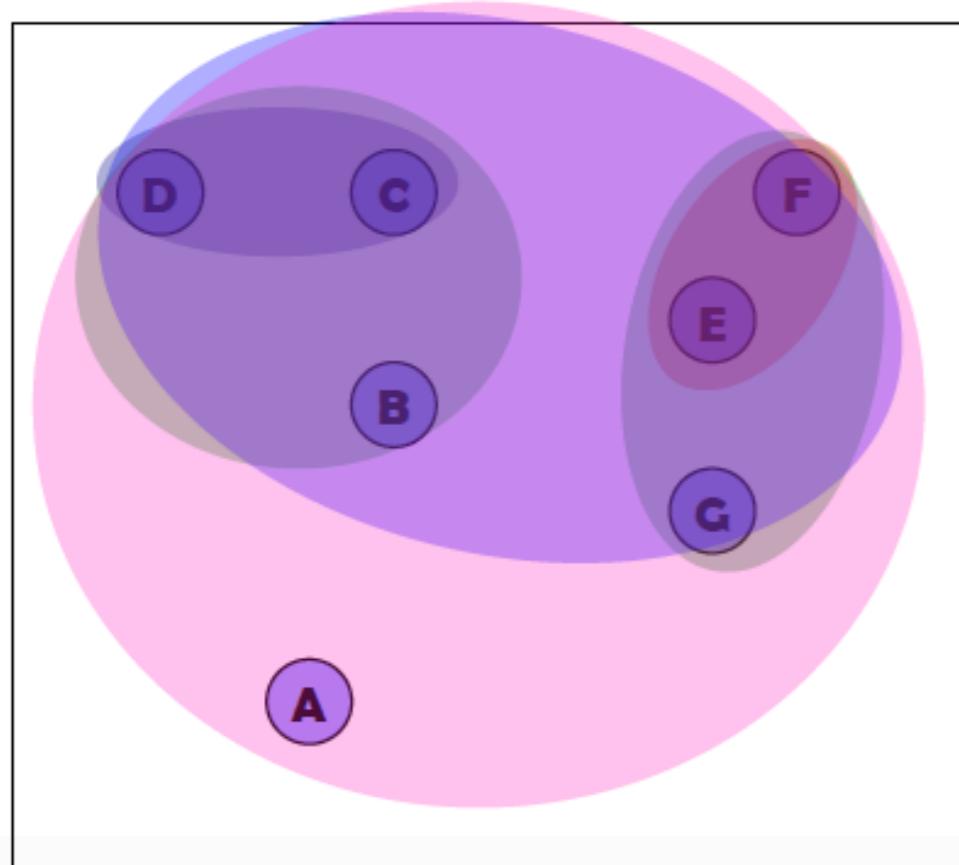
```
from sklearn.cluster import KMeans  
km = KMeans(n_clusters=3,  
             init='k-means++',  
             n_init=10,  
             max_iter=300,  
             tol=1e-04,  
             random_state=0)  
  
y_km = km.fit_predict(X)
```

初始質心選取方式

在實作中，強烈建議使用 k-means++

# 階層式聚類

- 階層聚類技術是第二重要的聚類方法。如同K-means，這些方法和許多聚類演算法比起來相對較久遠，但它們仍然被廣泛使用
- 階層聚類分析法通常可用**樹狀圖(Dendrogram)**表示。可顯示聚類-子聚類的關係，以及聚類被合併/分割順序。
- 使用**距離矩陣**作為聚類條件



# 產生階層聚類的方法

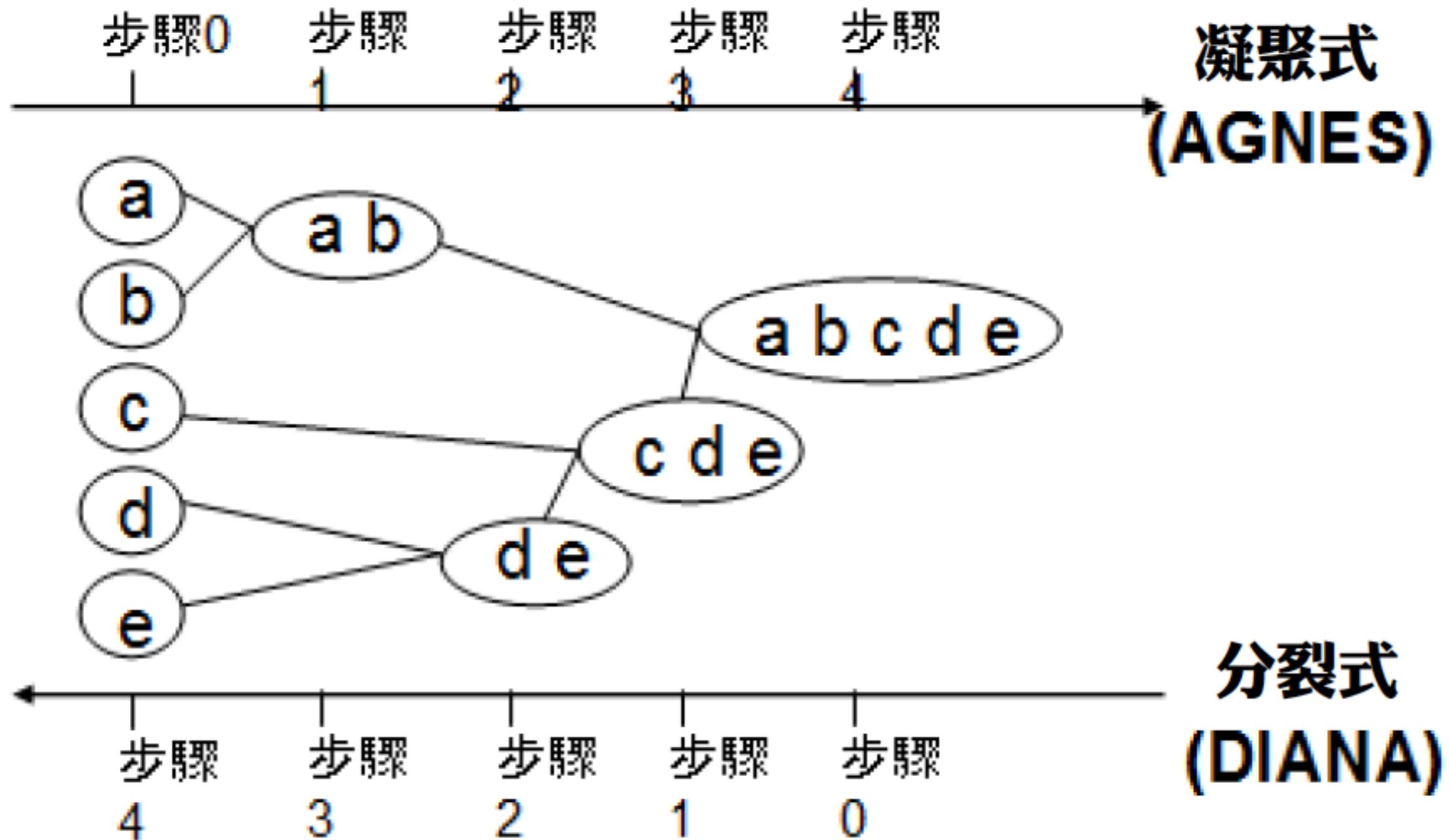
## ■ 凝聚式(Agglomerative)：

- 開始將每個對象視為單獨一組，然後相繼合併鄰近的對象或組別，直到所有的組別合併為一個，或者達到終止條件。
- 這需要定義聚類鄰近值(cluster proximity)的概念。

## ■ 分裂式(Divisive)：

- 開始將所有的對象置於同一個聚類中，在迭代的每一步，一個聚類被分裂為多個更小的聚類，直到最終每個對象在一個單獨的聚類中，或達到一個終止條件
- 在這個情況下，需要決定在每一步驟中哪一個聚類要被分裂，以及如何做分裂。

# 產生階層聚類的方法



# 凝聚式階層聚類

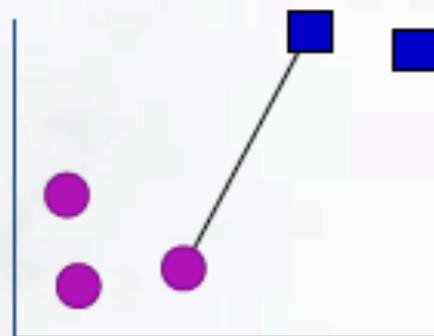
■ 以下將討論四種凝聚式階層聚類法：

- Min**
- Max**
- 群平均**
- 華德氏**

# 衡量群集間的距離

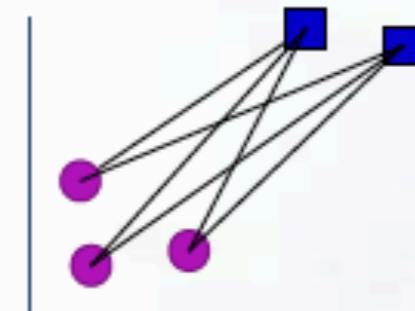
最小距離 (minimum distance)

$$D_{\min}(C_i, C_j) = \min_{a \in C_i, b \in C_j} D_{(a, b)}$$



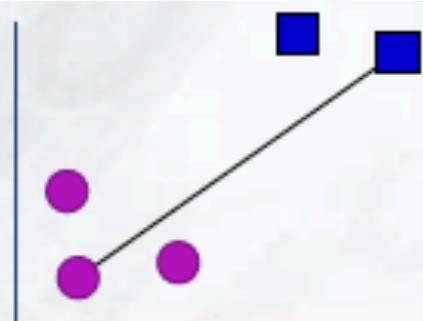
平均距離 (average distance)

$$D_{\text{average}}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{a \in C_i} \sum_{b \in C_j} D_{(a, b)}$$



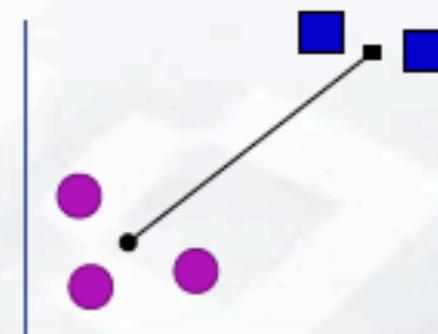
最大距離 (maximum distance)

$$D_{\max}(C_i, C_j) = \max_{a \in C_i, b \in C_j} D_{(a, b)}$$



中心值距離 (centroid distance)

$$D_{\text{centroid}}(C_i, C_j) = D_{(m_i, m_j)}$$

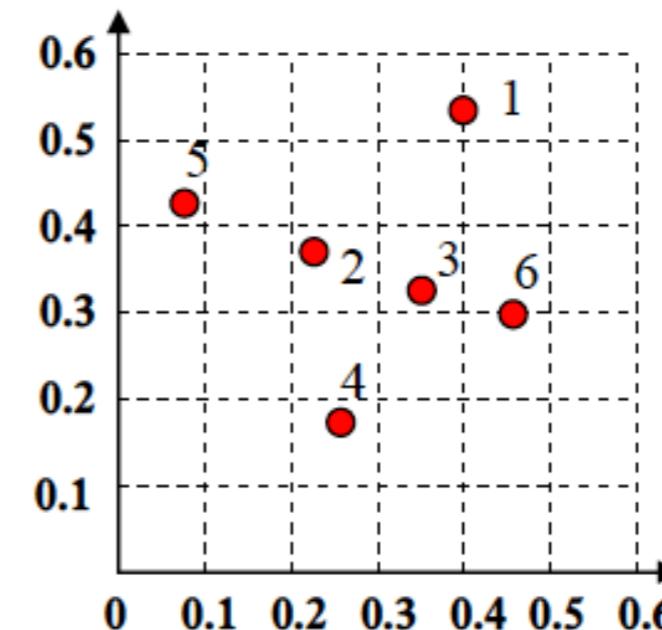


# Example

## ■ 樣本資料

□ 假設有6個二維資料點之樣本資料  $G = \{1, 2, 3, 4, 5, 6\}$  如下。

- 資料點座標
- 資料點之間的歐幾里德距離

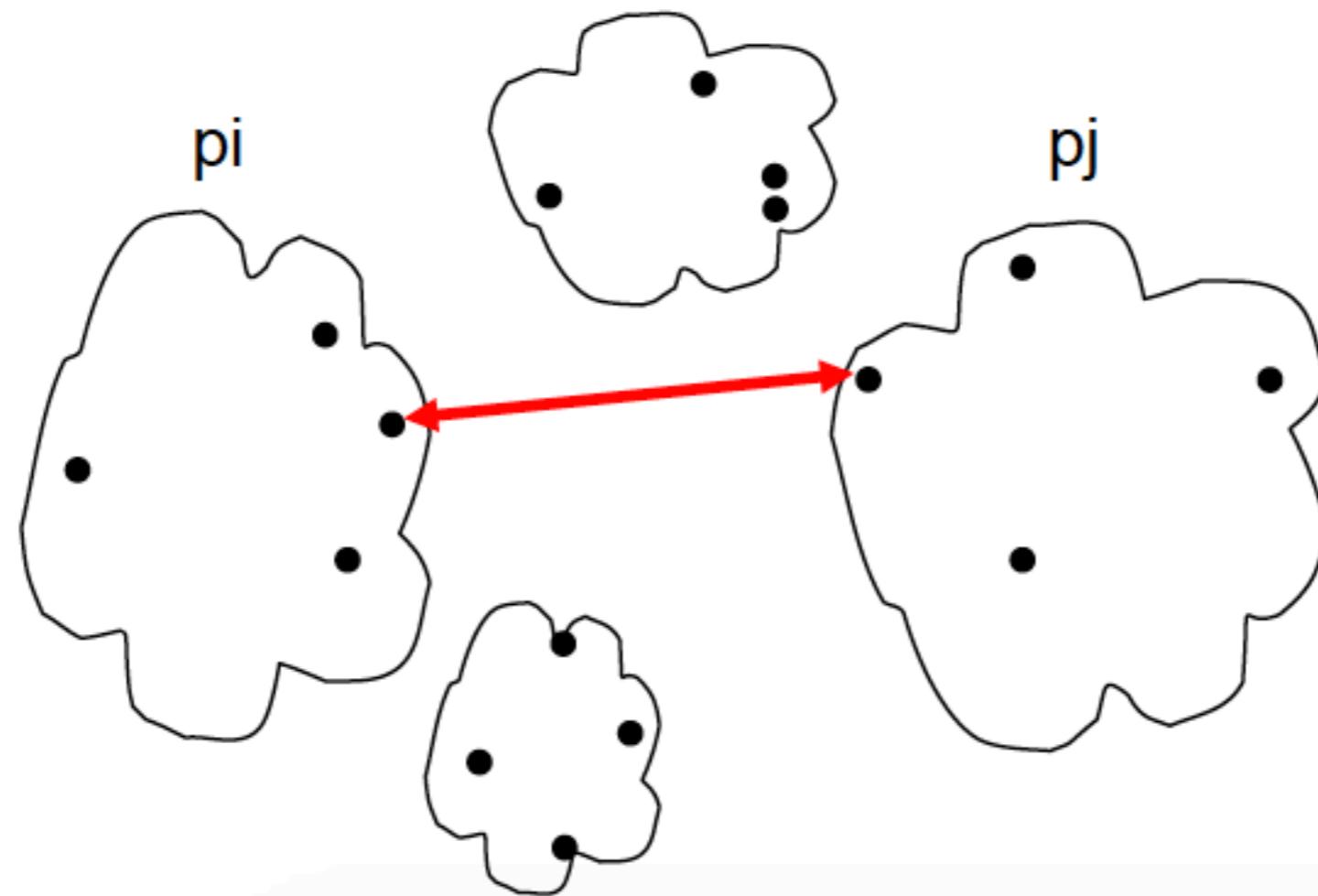


|    | X座標  | Y座標  |
|----|------|------|
| 點1 | 0.4  | 0.53 |
| 點2 | 0.22 | 0.38 |
| 點3 | 0.35 | 0.32 |
| 點4 | 0.26 | 0.19 |
| 點5 | 0.08 | 0.41 |
| 點6 | 0.45 | 0.3  |

|    | 點1   | 點2   | 點3   | 點4   | 點5   | 點6   |
|----|------|------|------|------|------|------|
| 點1 | 0.00 | 0.24 | 0.22 | 0.37 | 0.34 | 0.23 |
| 點2 | 0.24 | 0.00 | 0.15 | 0.20 | 0.14 | 0.25 |
| 點3 | 0.22 | 0.15 | 0.00 | 0.15 | 0.28 | 0.11 |
| 點4 | 0.37 | 0.20 | 0.15 | 0.00 | 0.29 | 0.22 |
| 點5 | 0.34 | 0.14 | 0.28 | 0.29 | 0.00 | 0.39 |
| 點6 | 0.23 | 0.25 | 0.11 | 0.22 | 0.39 | 0.00 |

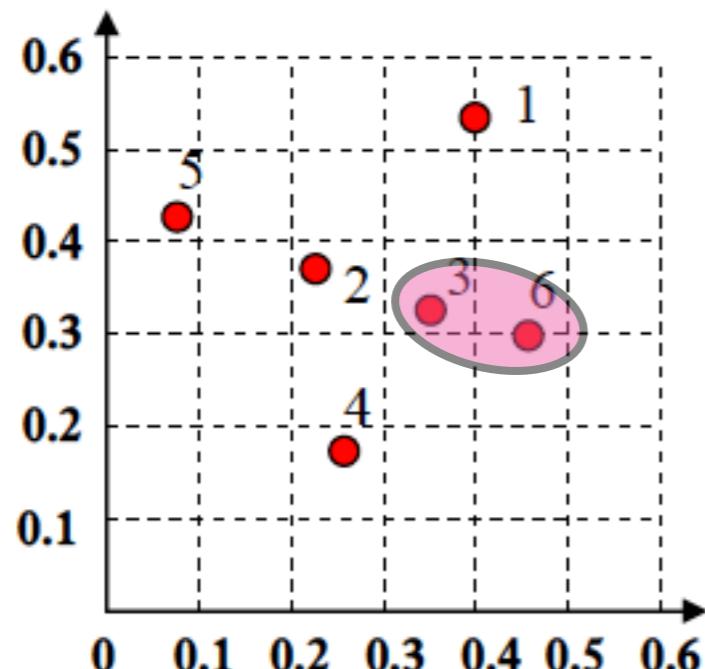
# 單一連結法 (Single linkage method)

- 是將不同聚類中之兩個最近的資料點之間的距離，當成是聚類的距離值



# Example

- 合併過程中，一開始各聚類皆僅單一值，經由計算得知具有最短距離的資料點集合為3和6，故將之合併為一聚類。(即： $G = \{1, 2, (3, 6), 4, 5\}$ )



|    | 點1   | 點2   | 點3   | 點4   | 點5   | 點6   |
|----|------|------|------|------|------|------|
| 點1 | 0.00 | 0.24 | 0.22 | 0.37 | 0.34 | 0.23 |
| 點2 | 0.24 | 0.00 | 0.15 | 0.20 | 0.14 | 0.25 |
| 點3 | 0.22 | 0.15 | 0.00 | 0.15 | 0.28 | 0.11 |
| 點4 | 0.37 | 0.20 | 0.15 | 0.00 | 0.29 | 0.22 |
| 點5 | 0.34 | 0.14 | 0.28 | 0.29 | 0.00 | 0.39 |
| 點6 | 0.23 | 0.25 | 0.11 | 0.22 | 0.39 | 0.00 |

# Example

$$\begin{aligned} \text{dist}(\{3, 6\}, 1) &= \min(\text{dist}(3, 1), \text{dist}(6, 1)) \\ &= \min(0.22, 0.23) \\ &= 0.22 \end{aligned}$$

|    | X座標  | Y座標  |
|----|------|------|
| 點1 | 0.4  | 0.53 |
| 點2 | 0.22 | 0.38 |
| 點3 | 0.35 | 0.32 |
| 點4 | 0.26 | 0.19 |
| 點5 | 0.08 | 0.41 |
| 點6 | 0.45 | 0.3  |

|        | 點1   | 點2   | {3, 6} | 點4   | 點5   |
|--------|------|------|--------|------|------|
| 點1     | 0.00 | 0.24 | 0.22   | 0.37 | 0.34 |
| 點2     | 0.24 | 0.00 |        | 0.20 | 0.14 |
| {3, 6} | 0.22 |      | 0.00   |      |      |
| 點4     | 0.37 | 0.20 |        | 0.00 | 0.29 |
| 點5     | 0.34 | 0.14 |        | 0.29 | 0.00 |

# Example

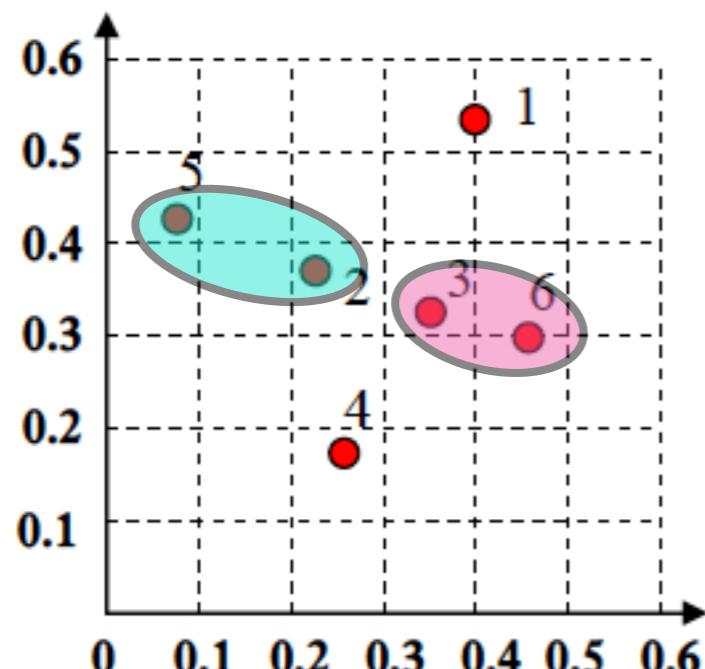
$$\begin{aligned} \text{dist}(\{3, 6\}, 4) &= \min(\text{dist}(3, 4), \text{dist}(6, 4)) \\ &= \min(0.15, 0.22) \\ &= 0.15 \end{aligned}$$

|    | X座標  | Y座標  |
|----|------|------|
| 點1 | 0.4  | 0.53 |
| 點2 | 0.22 | 0.38 |
| 點3 | 0.35 | 0.32 |
| 點4 | 0.26 | 0.19 |
| 點5 | 0.08 | 0.41 |
| 點6 | 0.45 | 0.3  |

|        | 點1   | 點2   | {3, 6} | 點4   | 點5   |
|--------|------|------|--------|------|------|
| 點1     | 0.00 | 0.24 | 0.22   | 0.37 | 0.34 |
| 點2     | 0.24 | 0.00 | 0.15   | 0.20 | 0.14 |
| {3, 6} | 0.22 | 0.15 | 0.00   | 0.15 | 0.28 |
| 點4     | 0.37 | 0.20 | 0.15   | 0.00 | 0.29 |
| 點5     | 0.34 | 0.14 | 0.28   | 0.29 | 0.00 |

# Example

- 接著，聚類 $\{3, 6\}$ 和其它聚類相互做比較，發現資料點2和5是所有可能距離中最短的，故將之合併為一聚類。(即： $G = \{1, (2, 5), (3, 6), 4\}$ )



|            | 點 1  | 點 2  | $\{3, 6\}$ | 點 4  | 點 5  |
|------------|------|------|------------|------|------|
| 點 1        | 0.00 | 0.24 | 0.22       | 0.37 | 0.34 |
| 點 2        | 0.24 | 0.00 | 0.15       | 0.20 | 0.14 |
| $\{3, 6\}$ | 0.22 | 0.15 | 0.00       | 0.15 | 0.28 |
| 點 4        | 0.37 | 0.20 | 0.15       | 0.00 | 0.29 |
| 點 5        | 0.34 | 0.14 | 0.28       | 0.29 | 0.00 |

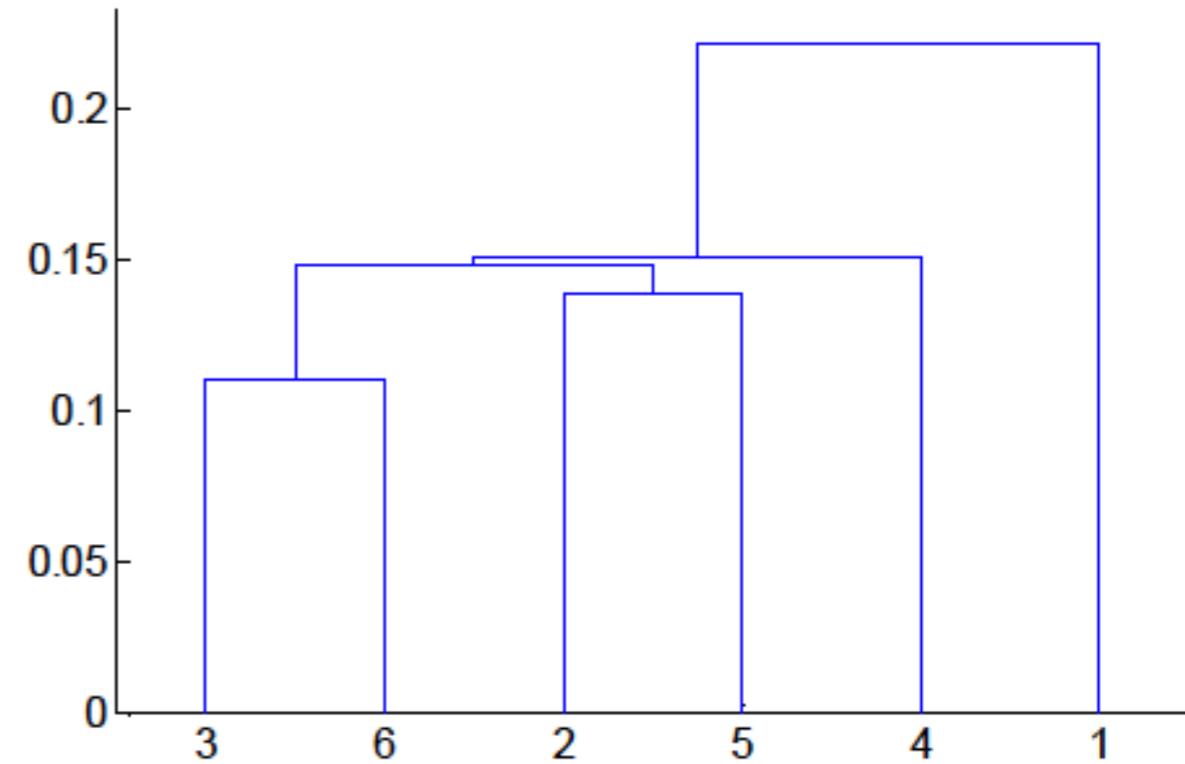
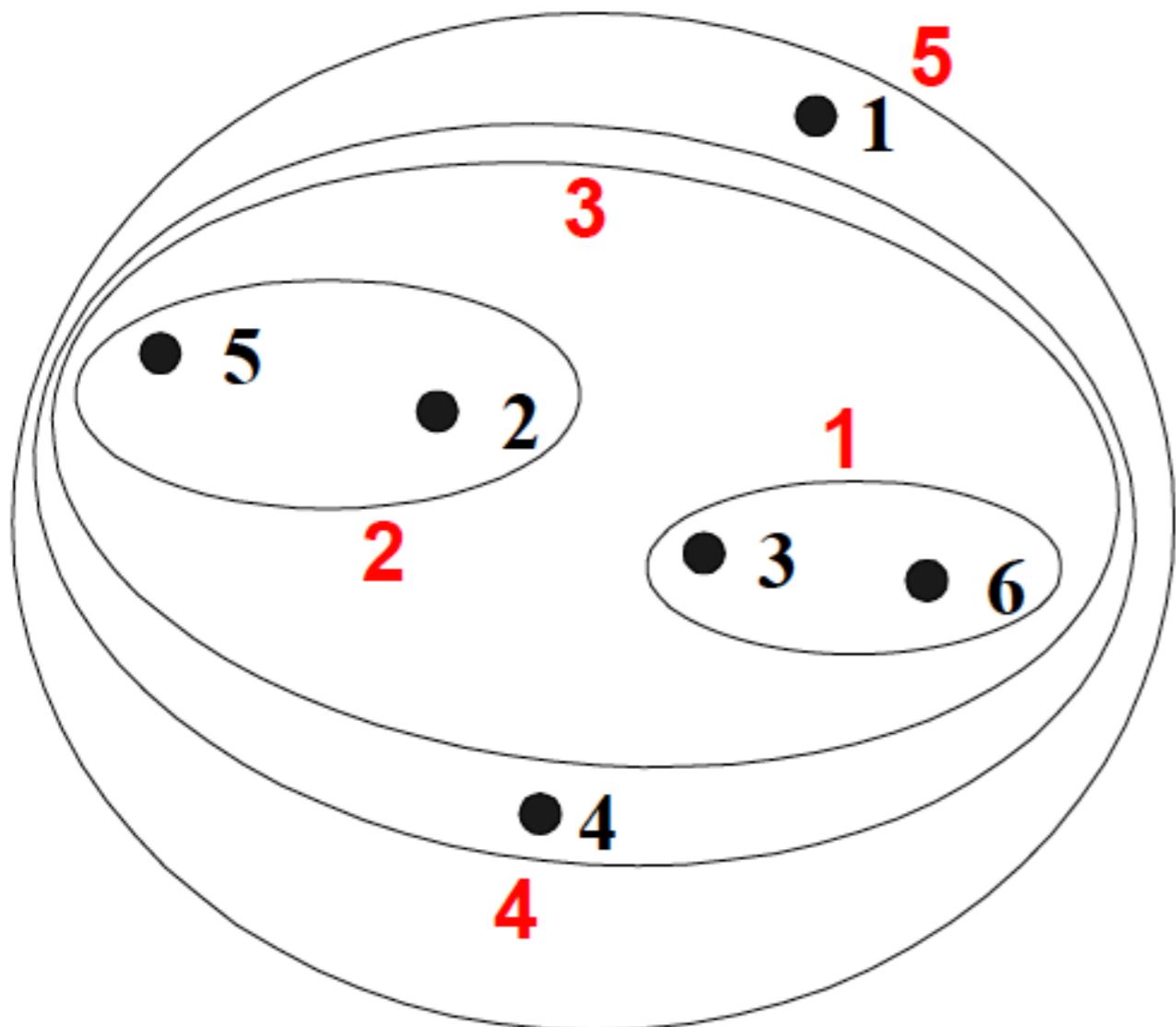
# Example

$$\begin{aligned} \text{dist}(\{3, 6\}, \{2, 5\}) &= \min(\text{dist}(3, 2), \text{dist}(6, 2), \text{dist}(3, 5), \text{dist}(6, 5)) \\ &= \min(0.15, 0.25, 0.28, 0.39) \\ &= 0.15 \end{aligned}$$

|    | X座標  | Y座標  |
|----|------|------|
| 點1 | 0.4  | 0.53 |
| 點2 | 0.22 | 0.38 |
| 點3 | 0.35 | 0.32 |
| 點4 | 0.26 | 0.19 |
| 點5 | 0.08 | 0.41 |
| 點6 | 0.45 | 0.3  |

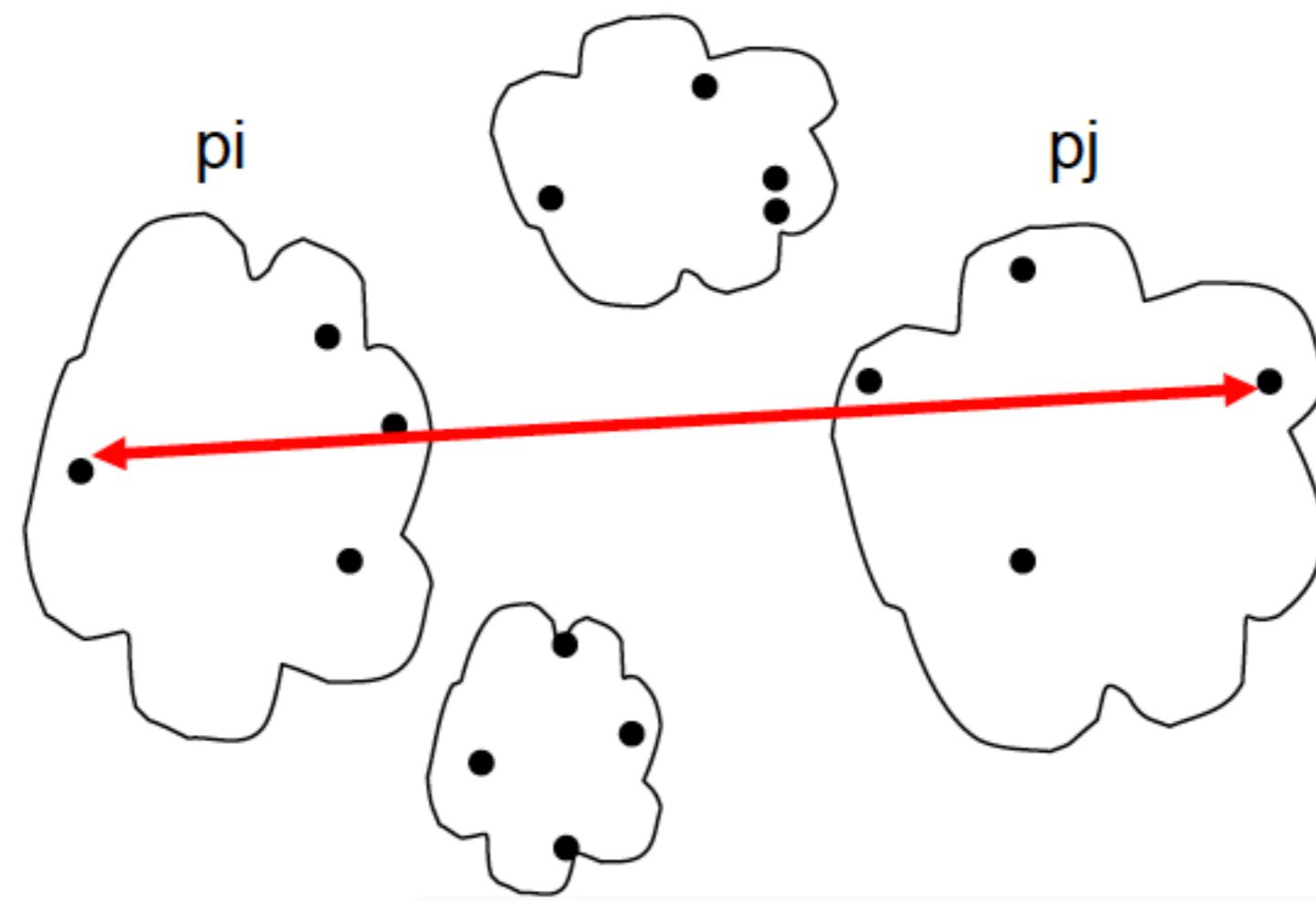
|        | 點 1  | {2, 5} | {3, 6} | 點 4  |
|--------|------|--------|--------|------|
| 點 1    | 0.00 | 0.24   | 0.22   | 0.37 |
| {2, 5} | 0.24 | 0.00   | 0.15   | 0.20 |
| {3, 6} | 0.22 | 0.15   | 0.00   | 0.15 |
| 點 4    | 0.37 | 0.20   | 0.15   | 0.00 |

# Example



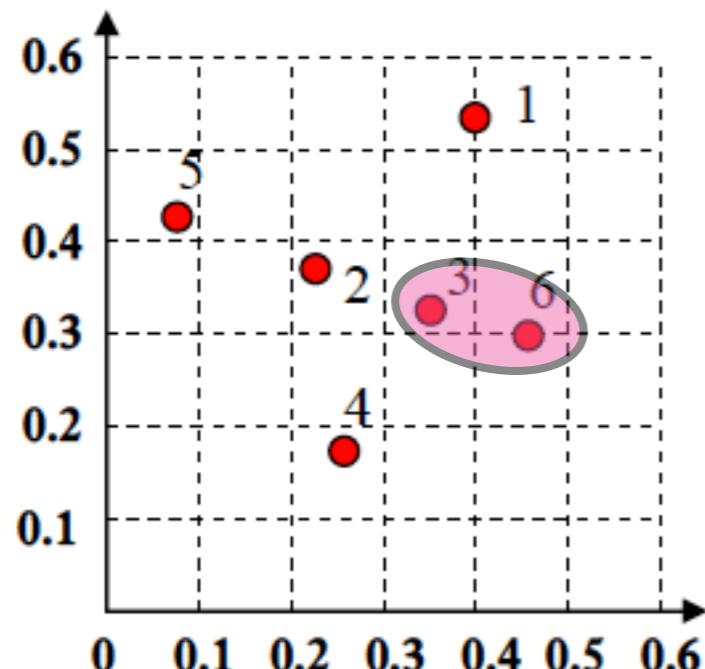
# 完全連結法 (complete linkage method)

- 是將不同聚類中之兩個最遠的資料點之間的距離，當成是聚類的距離值



# Example

- 合併過程中，一開始各聚類皆僅單一值，經由計算得知具有最短距離的資料點集合為3和6，故將之合併為一聚類。(即： $G = \{1, 2, (3, 6), 4, 5\}$ )



|    | 點1   | 點2   | 點3   | 點4   | 點5   | 點6   |
|----|------|------|------|------|------|------|
| 點1 | 0.00 | 0.24 | 0.22 | 0.37 | 0.34 | 0.23 |
| 點2 | 0.24 | 0.00 | 0.15 | 0.20 | 0.14 | 0.25 |
| 點3 | 0.22 | 0.15 | 0.00 | 0.15 | 0.28 | 0.11 |
| 點4 | 0.37 | 0.20 | 0.15 | 0.00 | 0.29 | 0.22 |
| 點5 | 0.34 | 0.14 | 0.28 | 0.29 | 0.00 | 0.39 |
| 點6 | 0.23 | 0.25 | 0.11 | 0.22 | 0.39 | 0.00 |

# Example

$$\begin{aligned} \text{dist}(\{3, 6\}, 1) &= \max(\text{dist}(3, 1), \text{dist}(6, 1)) \\ &= \max(0.22, 0.23) \\ &= 0.23 \end{aligned}$$

|    | X座標  | Y座標  |
|----|------|------|
| 點1 | 0.4  | 0.53 |
| 點2 | 0.22 | 0.38 |
| 點3 | 0.35 | 0.32 |
| 點4 | 0.26 | 0.19 |
| 點5 | 0.08 | 0.41 |
| 點6 | 0.45 | 0.3  |

|        | 點1   | 點2   | {3, 6} | 點4   | 點5   |
|--------|------|------|--------|------|------|
| 點1     | 0.00 | 0.24 | 0.23   | 0.37 | 0.34 |
| 點2     | 0.24 | 0.00 |        | 0.20 | 0.14 |
| {3, 6} | 0.23 |      | 0.00   |      |      |
| 點4     | 0.37 | 0.20 |        | 0.00 | 0.29 |
| 點5     | 0.34 | 0.14 |        | 0.29 | 0.00 |

# Example

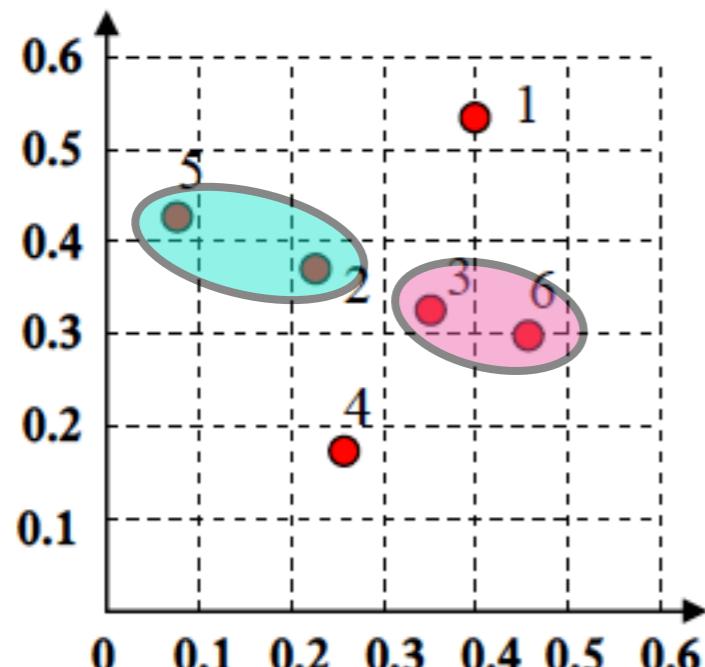
$$\begin{aligned} \text{dist}(\{3, 6\}, 1) &= \max(\text{dist}(3, 1), \text{dist}(6, 1)) \\ &= \max(0.22, 0.23) \\ &= 0.23 \end{aligned}$$

|    | X座標  | Y座標  |
|----|------|------|
| 點1 | 0.4  | 0.53 |
| 點2 | 0.22 | 0.38 |
| 點3 | 0.35 | 0.32 |
| 點4 | 0.26 | 0.19 |
| 點5 | 0.08 | 0.41 |
| 點6 | 0.45 | 0.3  |

|        | 點1   | 點2   | {3, 6} | 點4   | 點5   |
|--------|------|------|--------|------|------|
| 點1     | 0.00 | 0.24 | 0.23   | 0.37 | 0.34 |
| 點2     | 0.24 | 0.00 | 0.25   | 0.20 | 0.14 |
| {3, 6} | 0.23 | 0.25 | 0.00   | 0.22 | 0.39 |
| 點4     | 0.37 | 0.20 | 0.22   | 0.00 | 0.29 |
| 點5     | 0.34 | 0.14 | 0.39   | 0.29 | 0.00 |

# Example

- 接著，聚類 $\{3, 6\}$ 和其它聚類相互做比較，發現資料點2和5是所有可能距離中最短的，故將之合併為一聚類。(即： $G = \{1, (2, 5), (3, 6), 4\}$ )



|        | 點 1  | 點 2  | {3, 6} | 點 4  | 點 5  |
|--------|------|------|--------|------|------|
| 點 1    | 0.00 | 0.24 | 0.23   | 0.37 | 0.34 |
| 點 2    | 0.24 | 0.00 | 0.25   | 0.20 | 0.14 |
| {3, 6} | 0.23 | 0.25 | 0.00   | 0.22 | 0.39 |
| 點 4    | 0.37 | 0.20 | 0.22   | 0.00 | 0.29 |
| 點 5    | 0.34 | 0.14 | 0.39   | 0.29 | 0.00 |

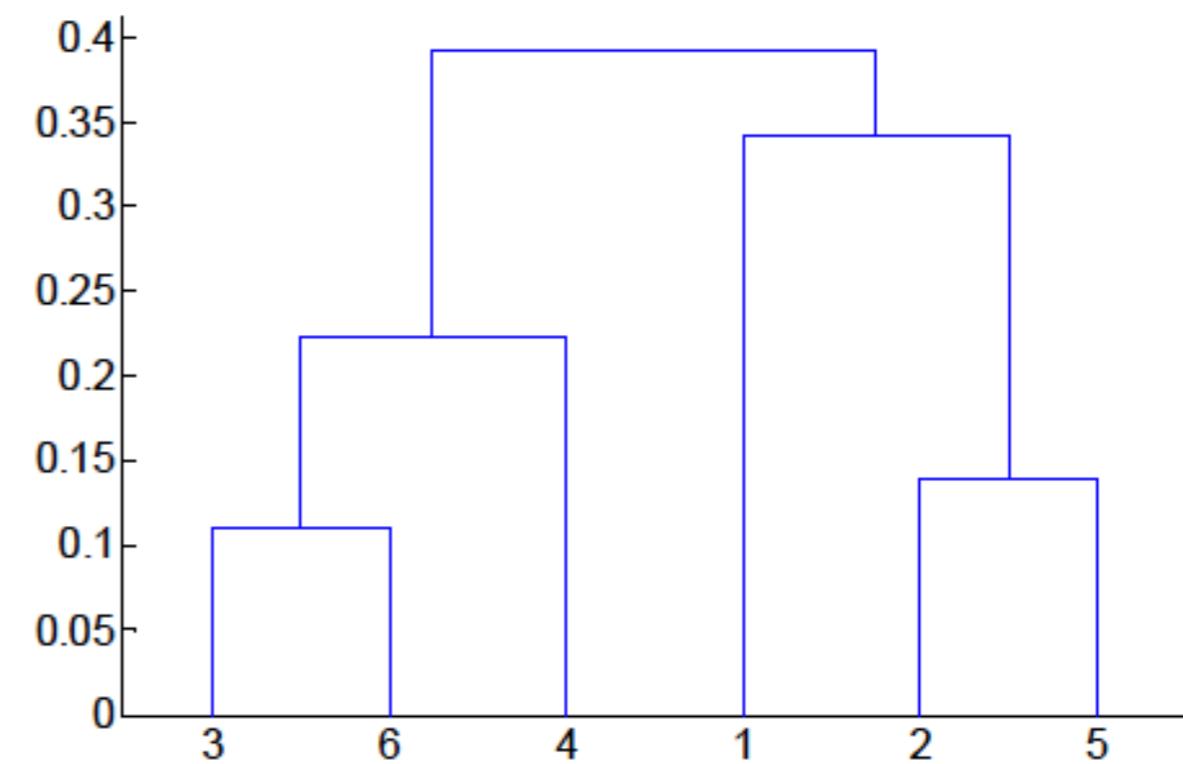
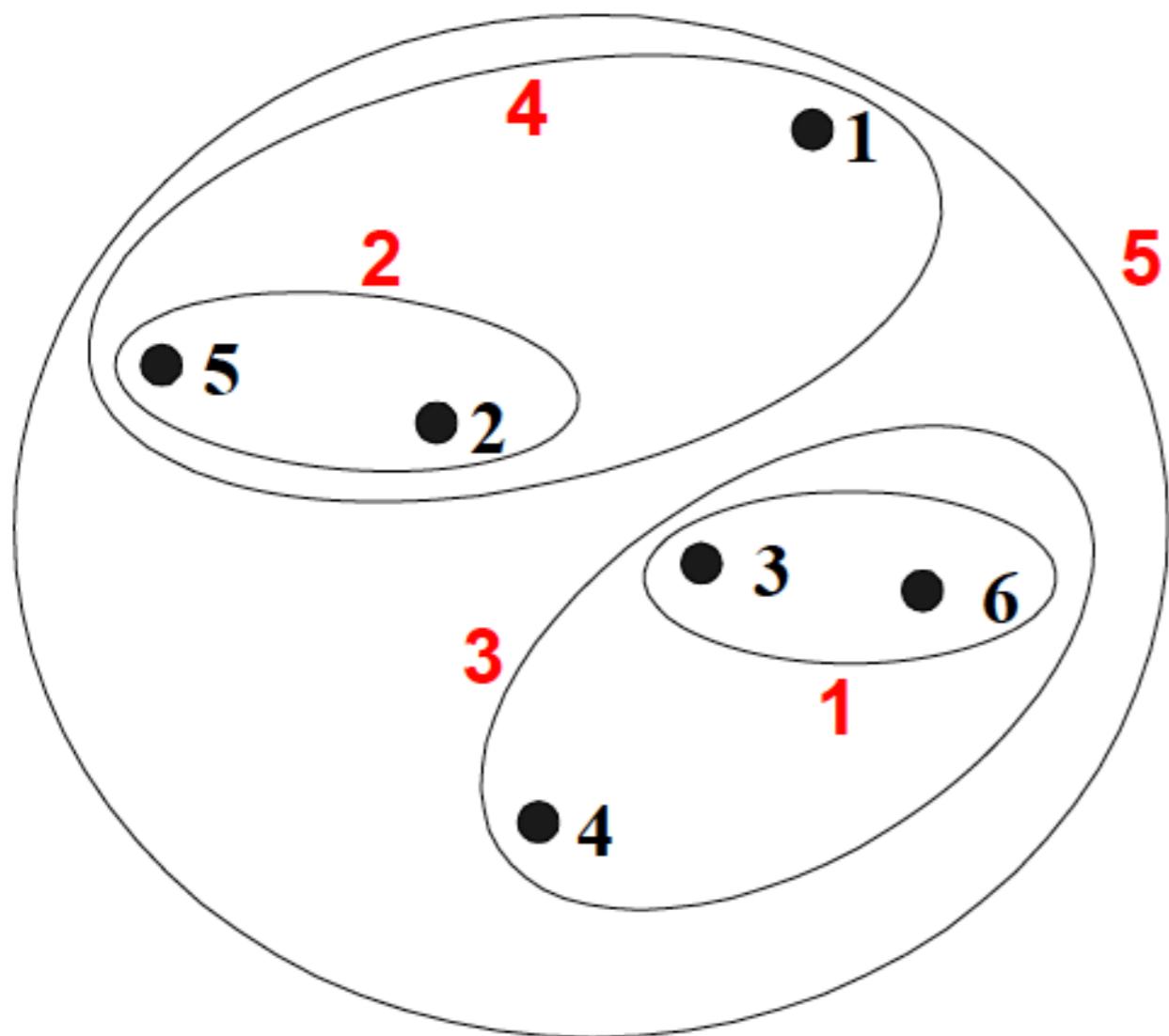
# Example

$$\begin{aligned} \text{dist}(\{3, 6\}, \{2, 5\}) &= \max(\text{dist}(3, 2), \text{dist}(6, 2), \text{dist}(3, 5), \text{dist}(6, 5)) \\ &= \max(0.15, 0.25, 0.28, 0.39) \\ &= 0.39 \end{aligned}$$

|    | X座標  | Y座標  |
|----|------|------|
| 點1 | 0.4  | 0.53 |
| 點2 | 0.22 | 0.38 |
| 點3 | 0.35 | 0.32 |
| 點4 | 0.26 | 0.19 |
| 點5 | 0.08 | 0.41 |
| 點6 | 0.45 | 0.3  |

|        | 點 1  | {2, 5} | {3, 6} | 點 4  |
|--------|------|--------|--------|------|
| 點 1    | 0.00 | 0.34   | 0.22   | 0.37 |
| {2, 5} | 0.34 | 0.00   | 0.39   | 0.29 |
| {3, 6} | 0.22 | 0.39   | 0.00   | 0.15 |
| 點 4    | 0.37 | 0.29   | 0.15   | 0.00 |

# Example

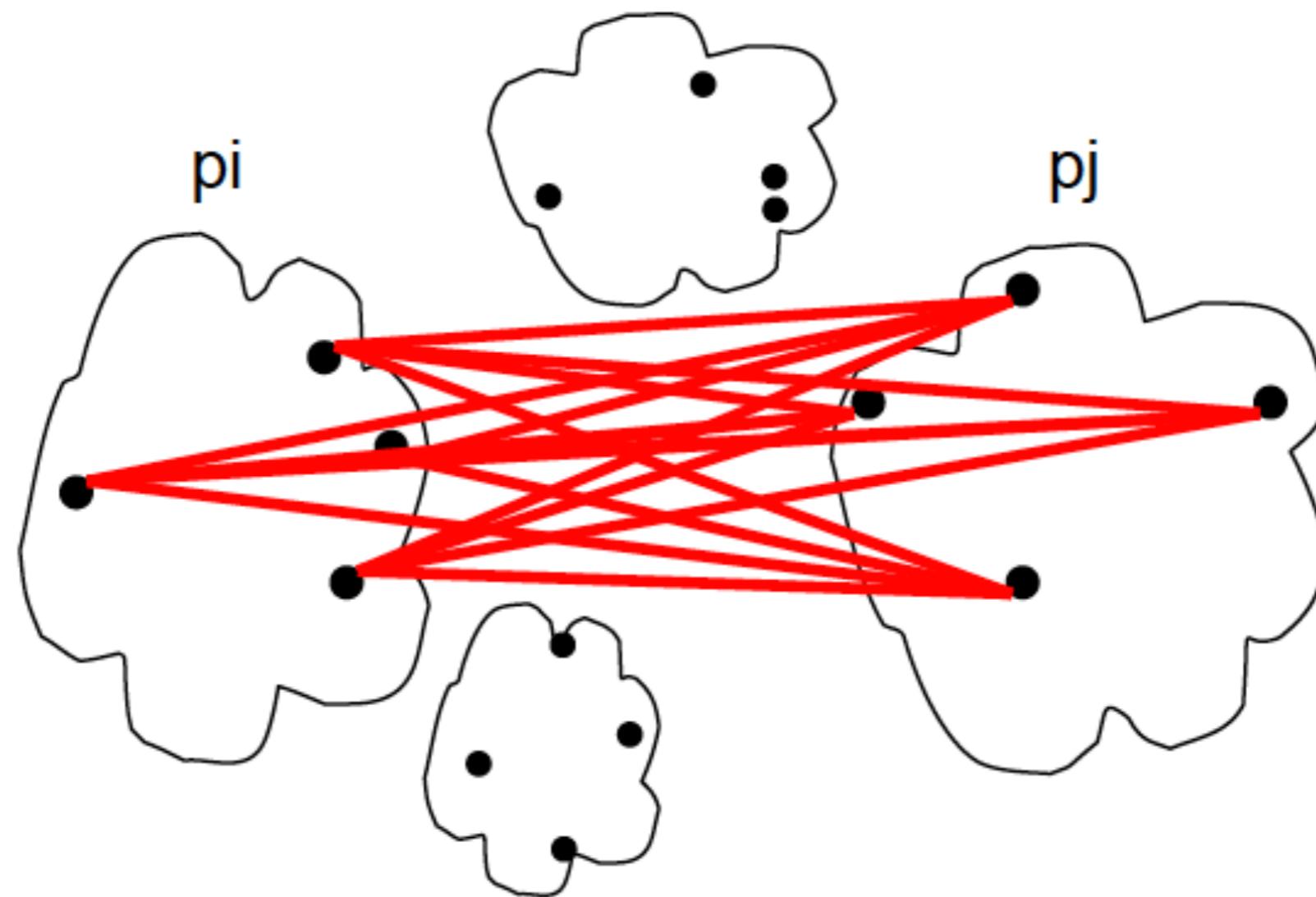


# 比較

- 單一連結法利用區域的鄰近性來進行階層式分群，而完全連結法傾向尋找全域相似性的群集
- 兩者皆考慮在量測群集間距離時極端的狀況，所以對離群值與雜訊資料皆十分敏感
- 使用平均距離是這兩者的折衷方案，可以克服對離群值與雜訊資料敏感的問題
- 使用中心點距離亦可解決對離群值與雜訊資料敏感的問題

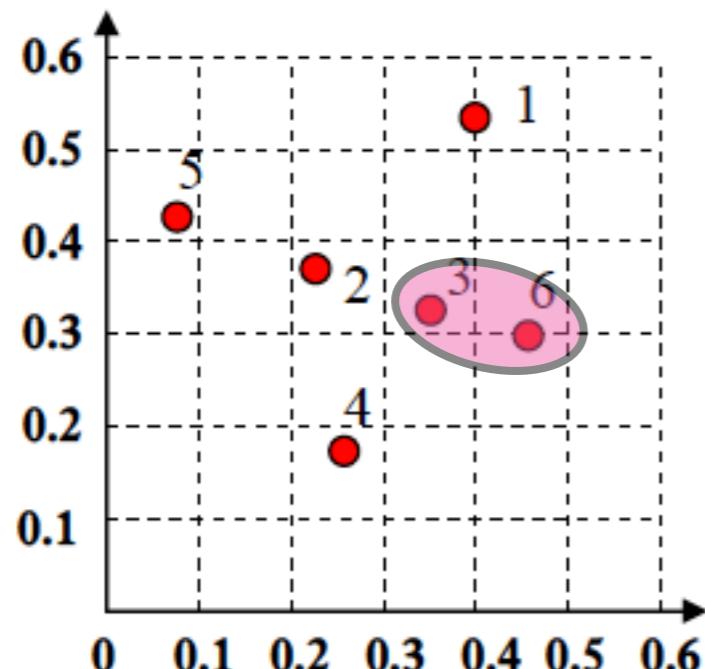
# 平均連結法 (average linkage method)

- 是將**不同聚類中所有成對資料點之平均成對距離**，當成聚類間的距離值。



# Example

- 合併過程中，一開始各聚類皆僅單一值，經由計算得知具有最短距離的資料點集合為3和6，故將之合併為一聚類。(即： $G = \{1, 2, (3, 6), 4, 5\}$ )



|    | 點1   | 點2   | 點3   | 點4   | 點5   | 點6   |
|----|------|------|------|------|------|------|
| 點1 | 0.00 | 0.24 | 0.22 | 0.37 | 0.34 | 0.23 |
| 點2 | 0.24 | 0.00 | 0.15 | 0.20 | 0.14 | 0.25 |
| 點3 | 0.22 | 0.15 | 0.00 | 0.15 | 0.28 | 0.11 |
| 點4 | 0.37 | 0.20 | 0.15 | 0.00 | 0.29 | 0.22 |
| 點5 | 0.34 | 0.14 | 0.28 | 0.29 | 0.00 | 0.39 |
| 點6 | 0.23 | 0.25 | 0.11 | 0.22 | 0.39 | 0.00 |

# Example

$$\begin{aligned} \text{dist}(\{3, 6\}, 1) &= (\text{dist}(3, 1) + \text{dist}(6, 1)) / (2 \times 1) \\ &= (0.22 + 0.23) / 2 \\ &= 0.225 \end{aligned}$$

|    | X座標  | Y座標  |
|----|------|------|
| 點1 | 0.4  | 0.53 |
| 點2 | 0.22 | 0.38 |
| 點3 | 0.35 | 0.32 |
| 點4 | 0.26 | 0.19 |
| 點5 | 0.08 | 0.41 |
| 點6 | 0.45 | 0.3  |

|        | 點1    | 點2   | {3, 6} | 點4   | 點5   |
|--------|-------|------|--------|------|------|
| 點1     | 0.00  | 0.24 | 0.225  | 0.37 | 0.34 |
| 點2     | 0.24  | 0.00 |        | 0.20 | 0.14 |
| {3, 6} | 0.225 |      | 0.00   |      |      |
| 點4     | 0.37  | 0.20 |        | 0.00 | 0.29 |
| 點5     | 0.34  | 0.14 |        | 0.29 | 0.00 |

# Example

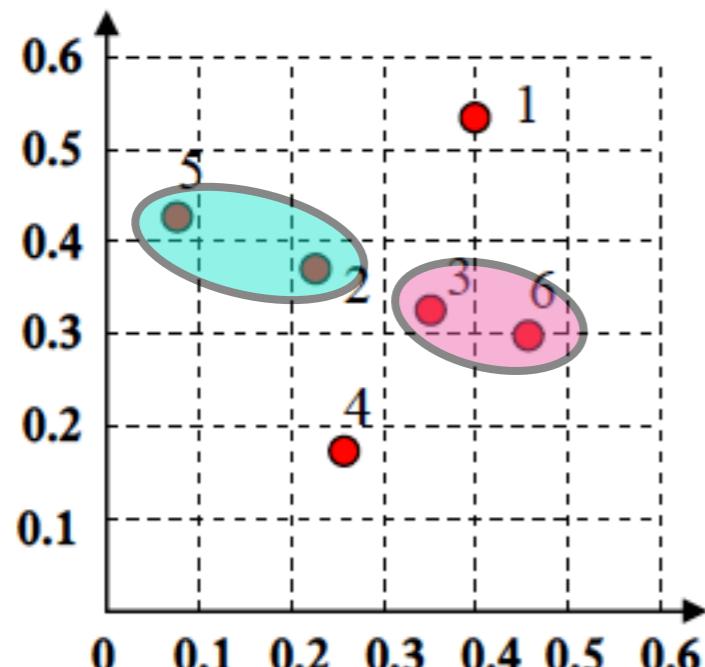
$$\begin{aligned} \text{dist}(\{3, 6\}, 1) &= (\text{dist}(3, 1) + \text{dist}(6, 1)) / (2 \times 1) \\ &= (0.22 + 0.23) / 2 \\ &= 0.225 \end{aligned}$$

|    | X座標  | Y座標  |
|----|------|------|
| 點1 | 0.4  | 0.53 |
| 點2 | 0.22 | 0.38 |
| 點3 | 0.35 | 0.32 |
| 點4 | 0.26 | 0.19 |
| 點5 | 0.08 | 0.41 |
| 點6 | 0.45 | 0.3  |

|        | 點1    | 點2   | {3, 6} | 點4    | 點5    |
|--------|-------|------|--------|-------|-------|
| 點1     | 0.00  | 0.24 | 0.225  | 0.37  | 0.34  |
| 點2     | 0.24  | 0.00 | 0.2    | 0.20  | 0.14  |
| {3, 6} | 0.225 | 0.2  | 0.00   | 0.185 | 0.335 |
| 點4     | 0.37  | 0.20 | 0.185  | 0.00  | 0.29  |
| 點5     | 0.34  | 0.14 | 0.335  | 0.29  | 0.00  |

# Example

- 接著，聚類 $\{3, 6\}$ 和其它聚類相互做比較，發現資料點2和5是所有可能距離中最短的，故將之合併為一聚類。(即： $G = \{1, (2, 5), (3, 6), 4\}$ )



|            | 點 1   | 點 2  | $\{3, 6\}$ | 點 4   | 點 5   |
|------------|-------|------|------------|-------|-------|
| 點 1        | 0.00  | 0.24 | 0.225      | 0.37  | 0.34  |
| 點 2        | 0.24  | 0.00 | 0.2        | 0.20  | 0.14  |
| $\{3, 6\}$ | 0.225 | 0.2  | 0.00       | 0.185 | 0.335 |
| 點 4        | 0.37  | 0.20 | 0.185      | 0.00  | 0.29  |
| 點 5        | 0.34  | 0.14 | 0.335      | 0.29  | 0.00  |

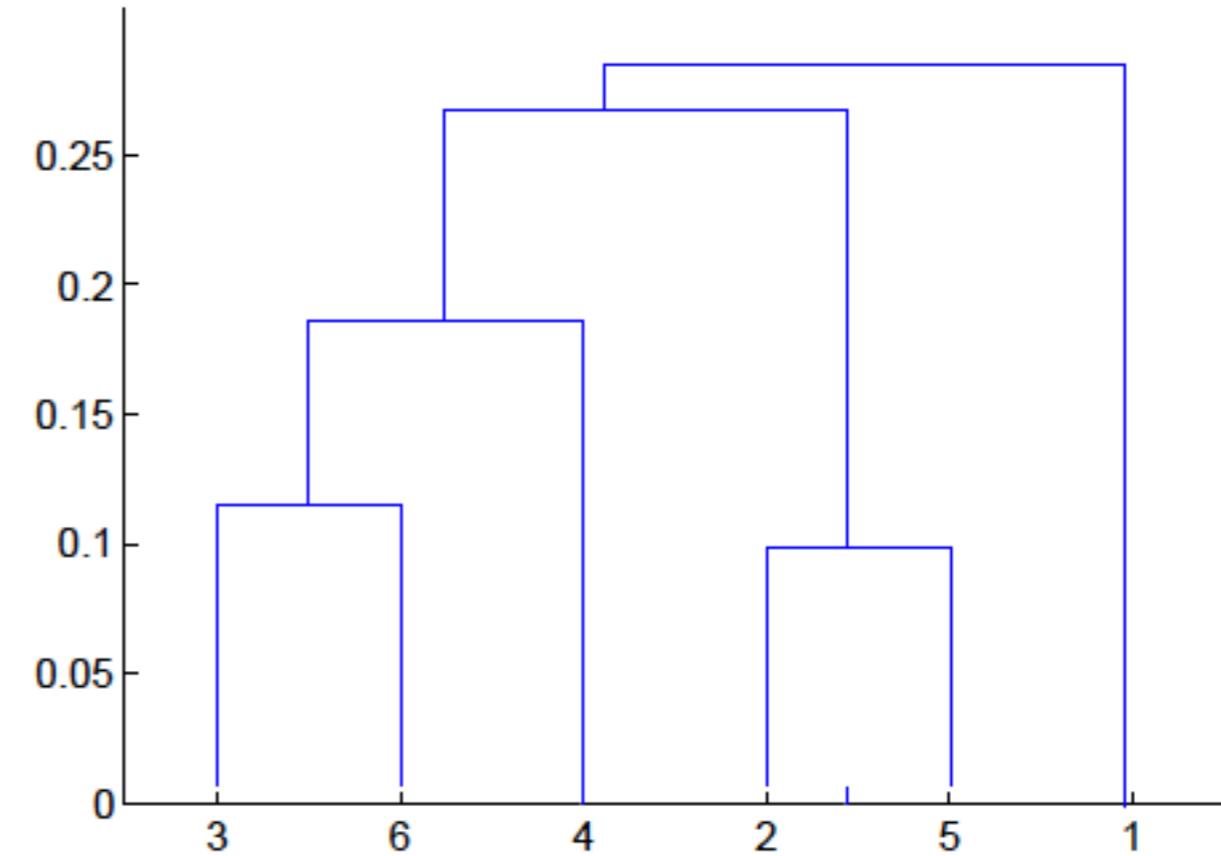
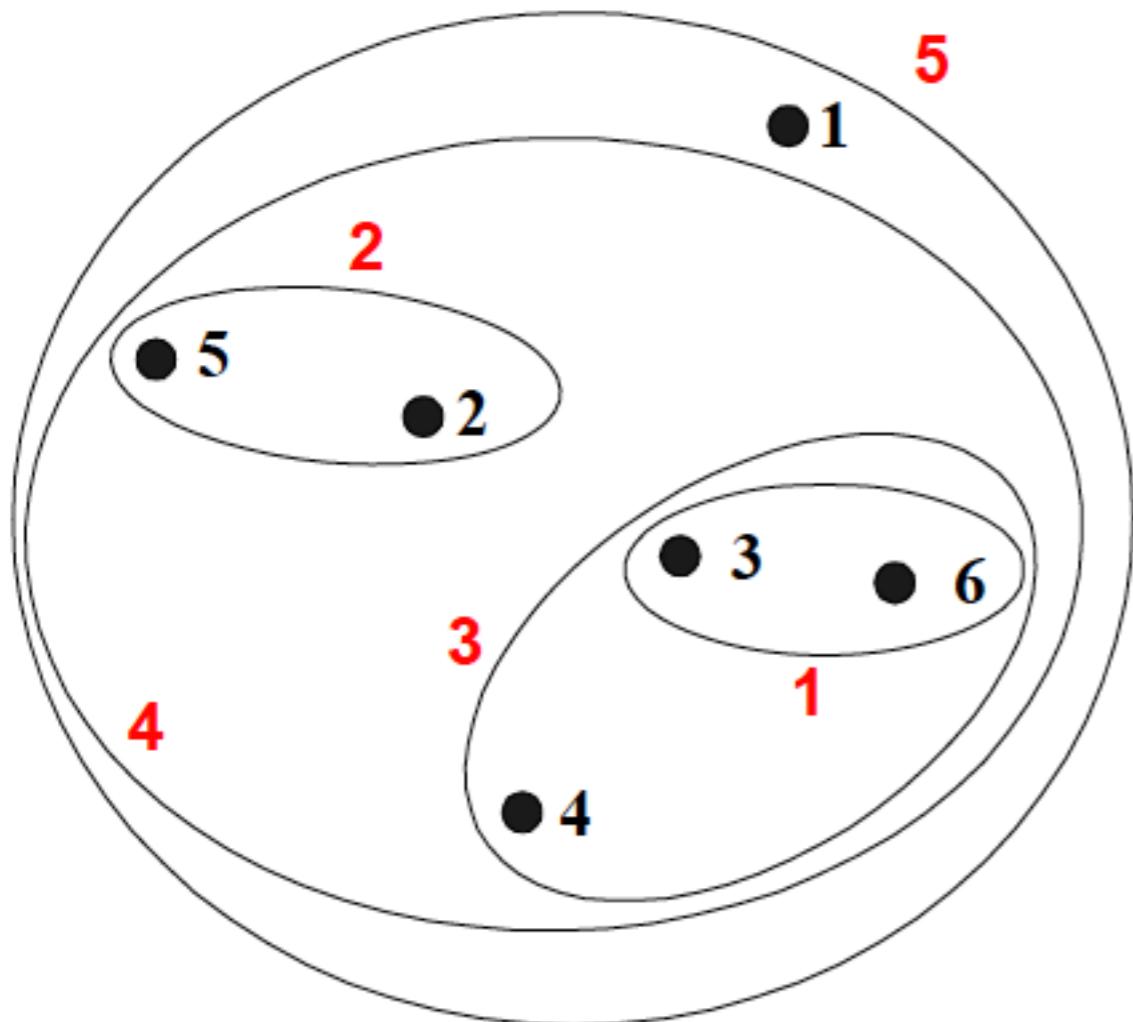
# Example

$$\begin{aligned} \text{dist}(\{3, 6\}, \{2, 5\}) &= (\text{dist}(3, 2) + \text{dist}(6, 2) + \text{dist}(3, 5) + \text{dist}(6, 5)) / (2 \times 2) \\ &= (0.15 + 0.25 + 0.28 + 0.39) / 4 \\ &= 0.2675 \end{aligned}$$

|    | X座標  | Y座標  |
|----|------|------|
| 點1 | 0.4  | 0.53 |
| 點2 | 0.22 | 0.38 |
| 點3 | 0.35 | 0.32 |
| 點4 | 0.26 | 0.19 |
| 點5 | 0.08 | 0.41 |
| 點6 | 0.45 | 0.3  |

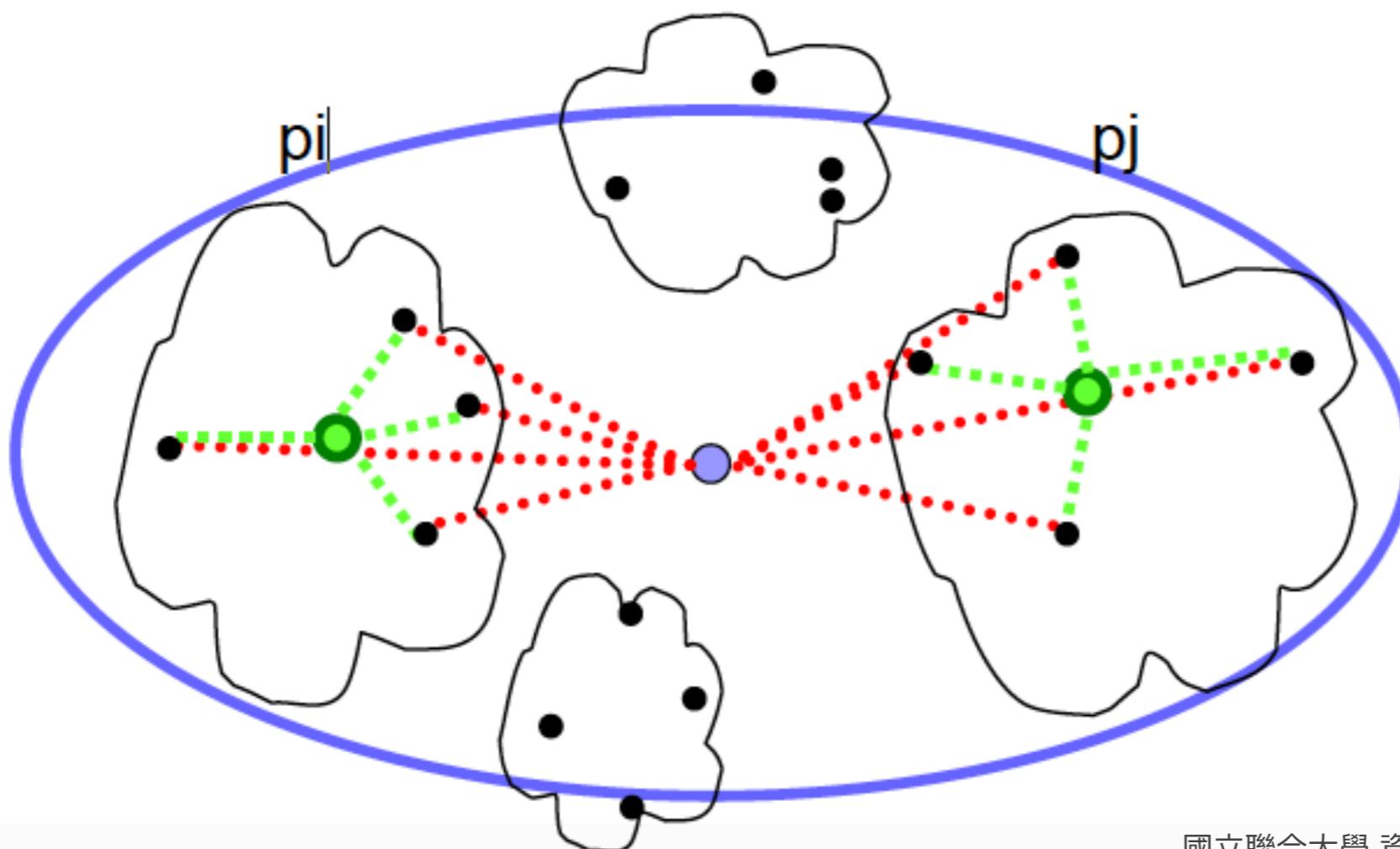
|        | 點 1   | {2, 5} | {3, 6} | 點 4   |
|--------|-------|--------|--------|-------|
| 點 1    | 0.00  | 0.29   | 0.225  | 0.37  |
| {2, 5} | 0.29  | 0.00   | 0.2675 | 0.245 |
| {3, 6} | 0.225 | 0.2675 | 0.00   | 0.185 |
| 點 4    | 0.37  | 0.245  | 0.185  | 0.00  |

# Example



# Ward's linkage

- 是以兩個聚類合併後所增加的SSE，來評估兩個聚類之間的鄰近性。
  - 凡使群內距離平方和最小之兩聚類即予以優先合併，愈早合併之聚類表示其間的相似性愈高。
  - 目的：希望合併後之聚類內的組內變異量達到最小。



# Ward's linkage

- 兩個聚類的距離值定義為：

- 兩群合併後的群中心之距離平方和(即： $SSE_{c_i \cup c_j}$ )，與原本兩群體之SSE(即： $SSE_{c_i}$  與  $SSE_{c_j}$ )的差距。

$$\begin{aligned} dist^2(C_i, C_j) &= \sum_{x \in C_i \cup C_j} \|(x - m_{C_i \cup C_j})\|^2 - \sum_{x \in C_i} \|(x - m_{C_i})\|^2 - \sum_{x \in C_j} \|(x - m_{C_j})\|^2 \\ &= \frac{n_{C_i} n_{C_j}}{n_{C_i \cup C_j}} \|m_{C_i} - m_{C_j}\|^2 \end{aligned}$$

- 其中， $C_i$ 為第  $i$  個聚類； $m_{c_i}$ 、 $m_{c_j}$ 、 $m_{c_i \cup c_j}$ 是兩聚類  $C_i$ 、 $C_j$  及其合併後的資料中心點、 $x$  為任一資料點。
- 假設兩聚類  $C_i$ 、 $C_j$  及其合併後的資料點數目分別為  $n_{c_i}$ 、 $n_{c_j}$ 、 $n_{c_i \cup c_j}$ ，則中心點  $m_{c_i}$ 、 $m_{c_j}$ 、 $m_{c_i \cup c_j}$ 的公式分別為：

$$m_{c_i} = \frac{\sum_{x \in C_i} x}{n_{c_i}}$$

$$m_{c_j} = \frac{\sum_{x \in C_j} x}{n_{c_j}}$$

$$m_{c_i \cup c_j} = \frac{\sum_{x \in C_i \cup C_j} x}{n_{c_i \cup c_j}}$$

# Example

$$dist^2(C_i, C_j) = \frac{n_{C_i}n_{C_j}}{n_{C_i \cup C_j}} \|m_{C_i} - m_{C_j}\|^2$$

$$m_{\{3,6\}} = \left( \frac{0.35 + 0.45}{2}, \frac{0.32 + 0.30}{2} \right) = (0.4, 0.31)$$

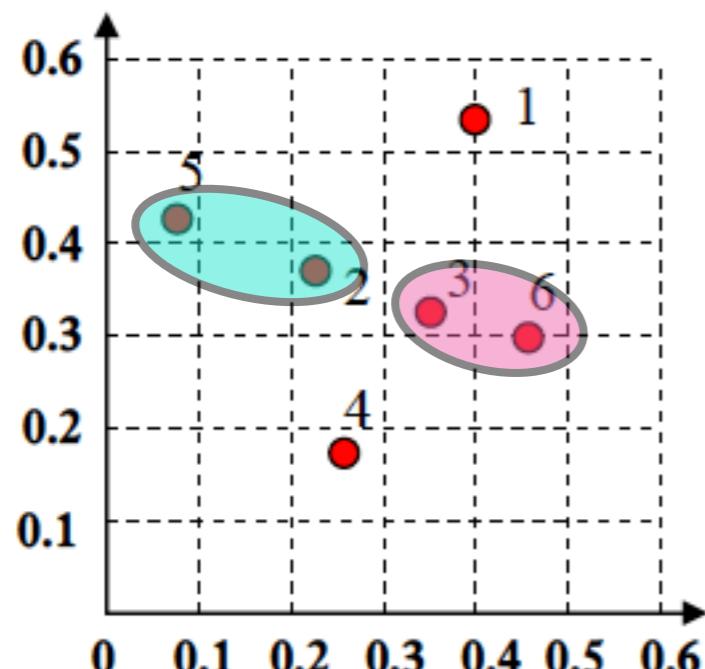
$$\begin{aligned} dist(\{3,6\}, 4) &= \sqrt{\frac{2}{3}[(0.4 - 0.26)^2 + (0.31 - 0.19)^2]} \\ &= 0.1506 \end{aligned}$$

|    | X座標  | Y座標  |
|----|------|------|
| 點1 | 0.4  | 0.53 |
| 點2 | 0.22 | 0.38 |
| 點3 | 0.35 | 0.32 |
| 點4 | 0.26 | 0.19 |
| 點5 | 0.08 | 0.41 |
| 點6 | 0.45 | 0.3  |

|        | 點1   | 點2   | {3, 6} | 點4   | 點5   |
|--------|------|------|--------|------|------|
| 點1     | 0.00 | 0.24 | 0.18   | 0.37 | 0.34 |
| 點2     | 0.24 | 0.00 | 0.16   | 0.20 | 0.14 |
| {3, 6} | 0.18 | 0.16 | 0.00   | 0.15 | 0.27 |
| 點4     | 0.37 | 0.20 | 0.15   | 0.00 | 0.29 |
| 點5     | 0.34 | 0.14 | 0.27   | 0.29 | 0.00 |

# Example

- 接著，聚類 $\{3, 6\}$ 和其它聚類相互做比較，發現資料點2和5是所有可能距離中最短的，故將之合併為一聚類。 (即： $G = \{1, (2, 5), (3, 6), 4\}$ )



|            | 點 1  | 點 2  | $\{3, 6\}$ | 點 4  | 點 5  |
|------------|------|------|------------|------|------|
| 點 1        | 0.00 | 0.24 | 0.18       | 0.37 | 0.34 |
| 點 2        | 0.24 | 0.00 | 0.16       | 0.20 | 0.14 |
| $\{3, 6\}$ | 0.18 | 0.16 | 0.00       | 0.15 | 0.27 |
| 點 4        | 0.37 | 0.20 | 0.15       | 0.00 | 0.29 |
| 點 5        | 0.34 | 0.14 | 0.27       | 0.29 | 0.00 |

# Example

$$dist(C_i, C_j) = \frac{n_{C_i} n_{C_j}}{n_{C_i \cup C_j}} \|m_{C_i} - m_{C_j}\|^2$$

$$m_{\{3,6\}} = \left( \frac{0.35 + 0.45}{2}, \frac{0.32 + 0.30}{2} \right) = (0.4, 0.31)$$

$$m_{\{2,5\}} = \left( \frac{0.22 + 0.08}{2}, \frac{0.38 + 0.41}{2} \right) = (0.15, 0.395)$$

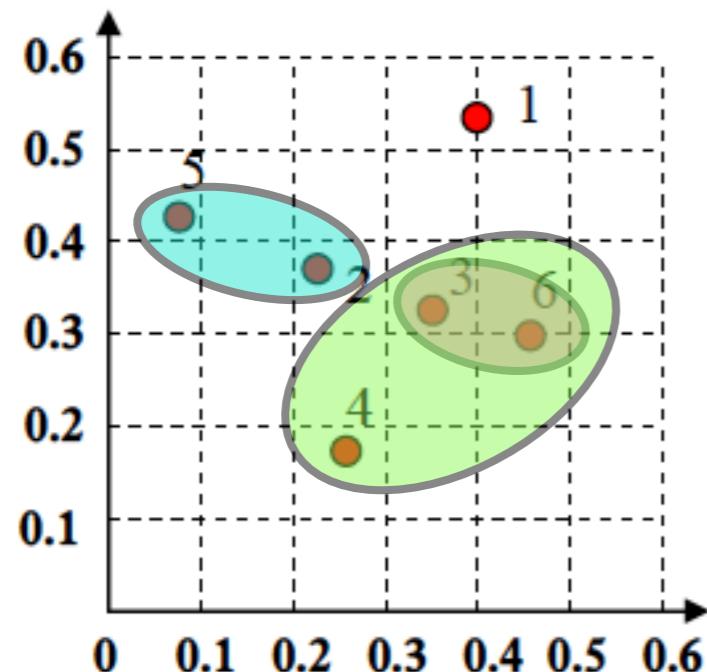
$$dist(\{3,6\}, \{2,5\}) = \sqrt{\frac{4}{4}[(0.4 - 0.15)^2 + (0.31 - 0.395)^2]} \\ = 0.264$$

|    | X座標  | Y座標  |
|----|------|------|
| 點1 | 0.4  | 0.53 |
| 點2 | 0.22 | 0.38 |
| 點3 | 0.35 | 0.32 |
| 點4 | 0.26 | 0.19 |
| 點5 | 0.08 | 0.41 |
| 點6 | 0.45 | 0.3  |

|        | 點 1  | {2, 5} | {3, 6} | 點 4  |
|--------|------|--------|--------|------|
| 點 1    | 0.00 | 0.23   | 0.18   | 0.37 |
| {2, 5} | 0.23 | 0.00   | 0.26   | 0.19 |
| {3, 6} | 0.18 | 0.26   | 0.00   | 0.15 |
| 點 4    | 0.37 | 0.19   | 0.15   | 0.00 |

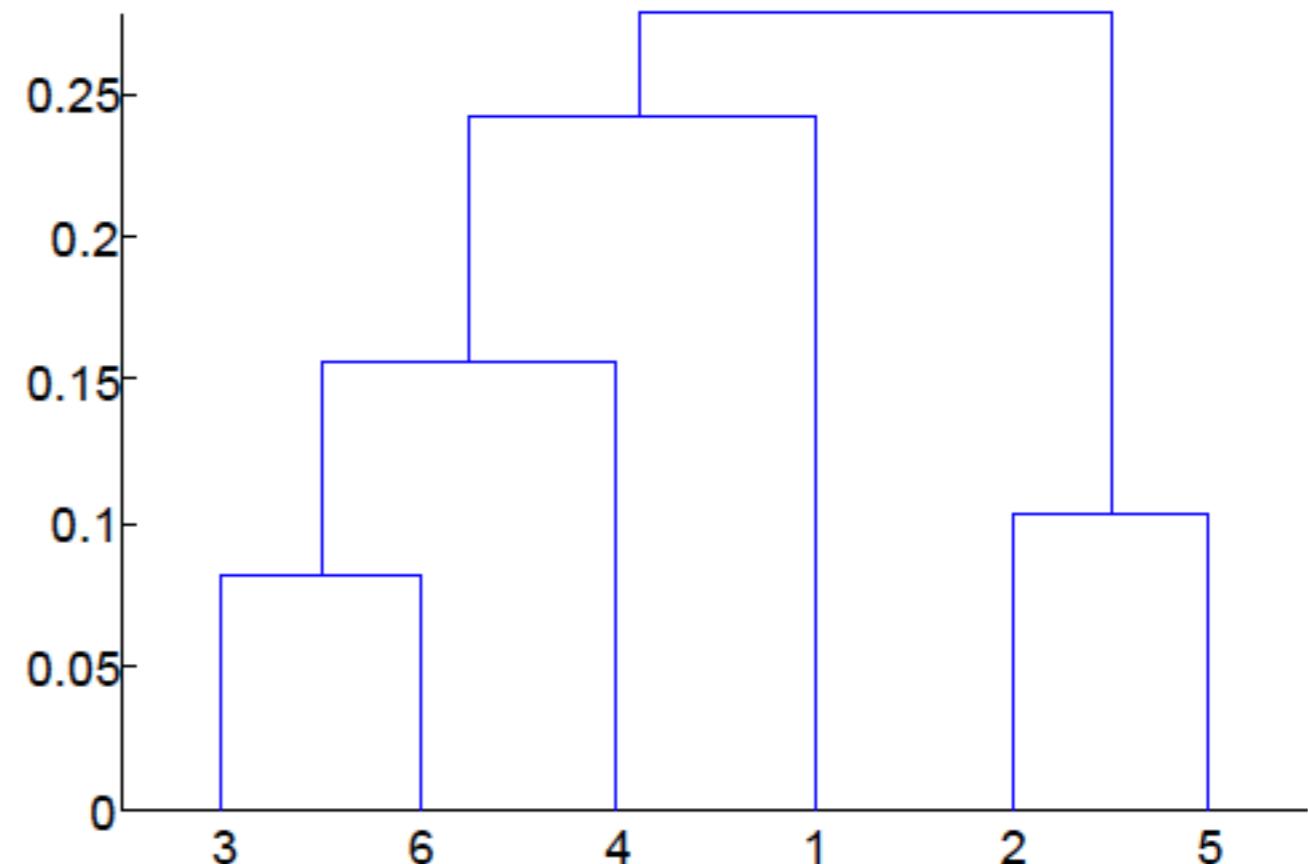
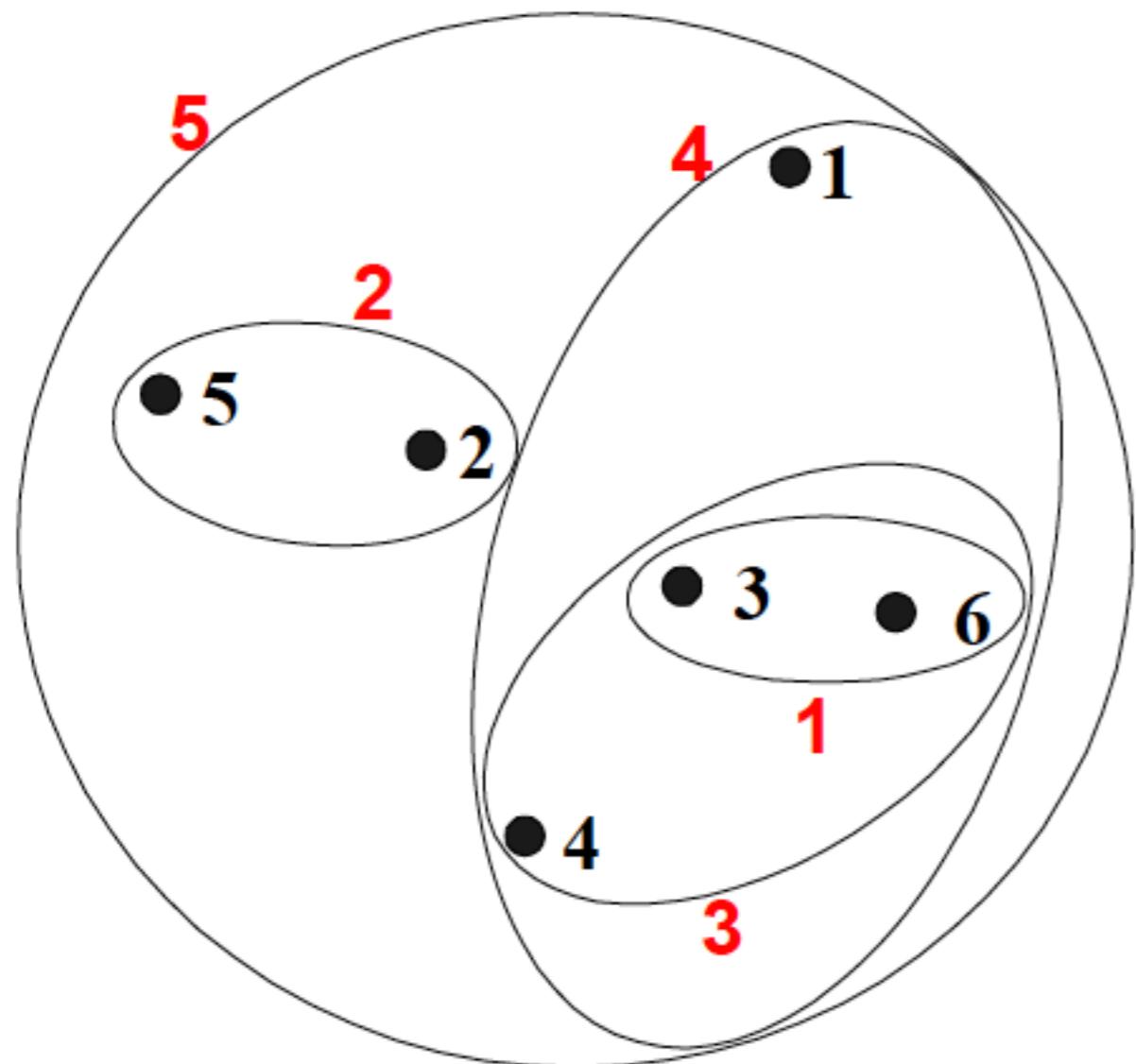
# Example

- 聚類 $\{3, 6\}$ 、 $\{2, 5\}$ 和其它聚類相互做比較，發現聚類 $\{3, 6\}$ 與4是所有可能距離中最短的，故將之合併為一聚類。(即： $G = \{1, (2, 5), ((3, 6), 4)\}$ )



|            | 點 1  | $\{2, 5\}$ | $\{3, 6\}$ | 點 4  |
|------------|------|------------|------------|------|
| 點 1        | 0.00 | 0.23       | 0.18       | 0.37 |
| $\{2, 5\}$ | 0.23 | 0.00       | 0.26       | 0.19 |
| $\{3, 6\}$ | 0.18 | 0.26       | 0.00       | 0.15 |
| 點 4        | 0.37 | 0.19       | 0.15       | 0.00 |

# Ward's linkage



# Python code

## ■ 凝聚式階層聚類

```
from scipy.cluster.hierarchy import linkage
from scipy.cluster.hierarchy import dendrogram

Complete = linkage(X, method = 'complete', metric = 'euclidean')
dendrogram(Complete)
ax = plt.gca()
plt.xlabel("Sample index")
plt.ylabel("Cluster distance")
plt.show()
```

選擇不同的集群距離  
'single', 'average'

獲得目前的座標軸

# Python code

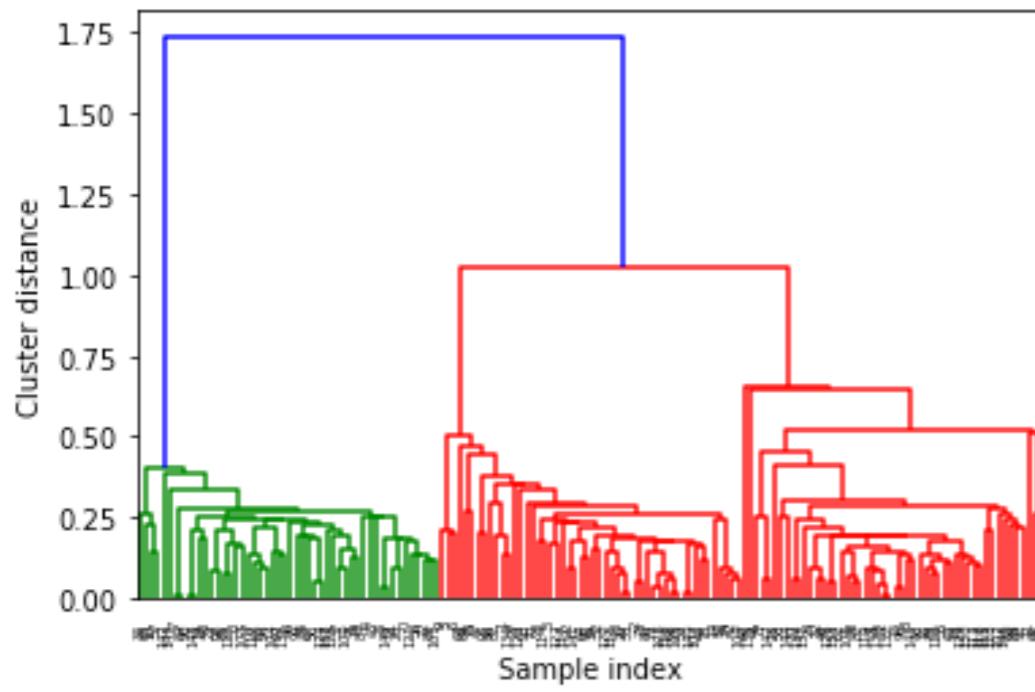
## ■ Ward's Method

```
from scipy.cluster.hierarchy import ward
from scipy.cluster.hierarchy import dendrogram

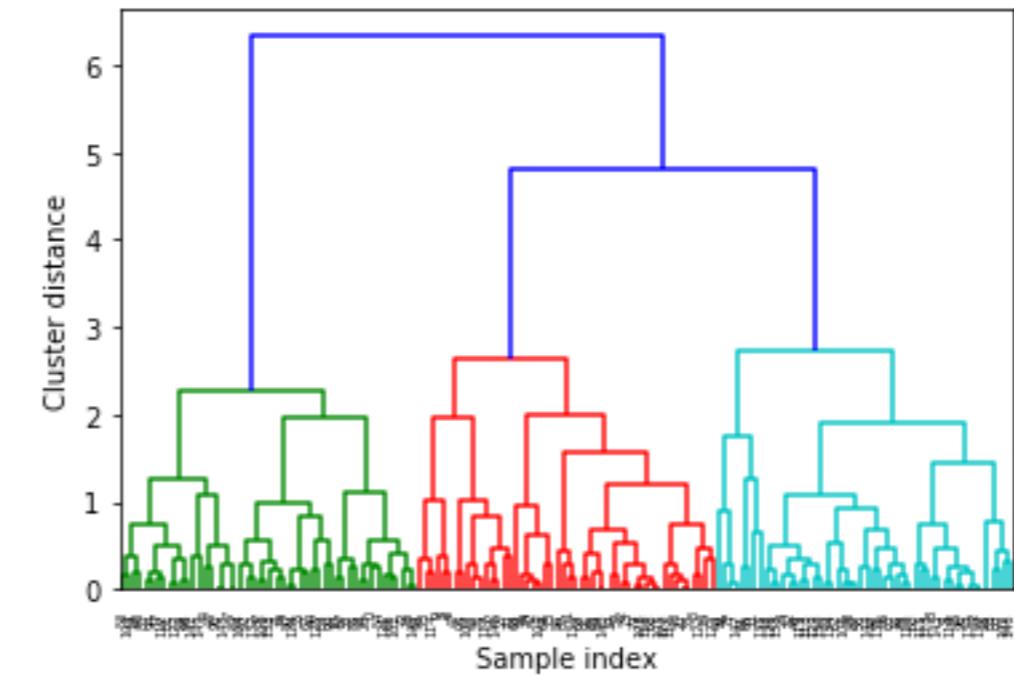
linkage = ward(X)
dendrogram(linkage)
ax = plt.gca()
plt.xlabel("Sample index")
plt.ylabel("Cluster distance")
plt.show()
```

# Python code

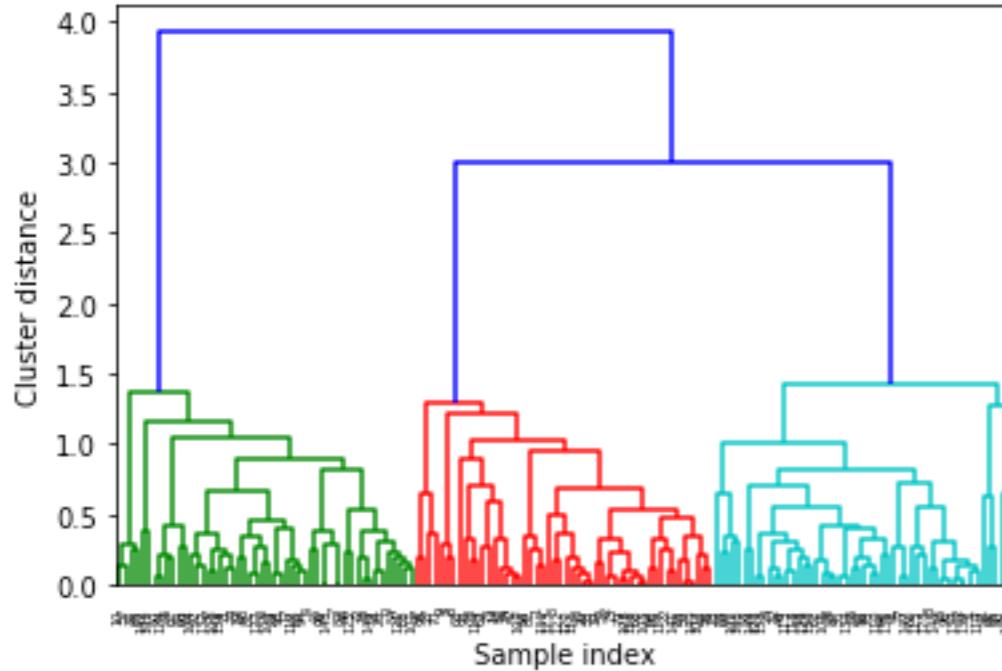
Single linkage



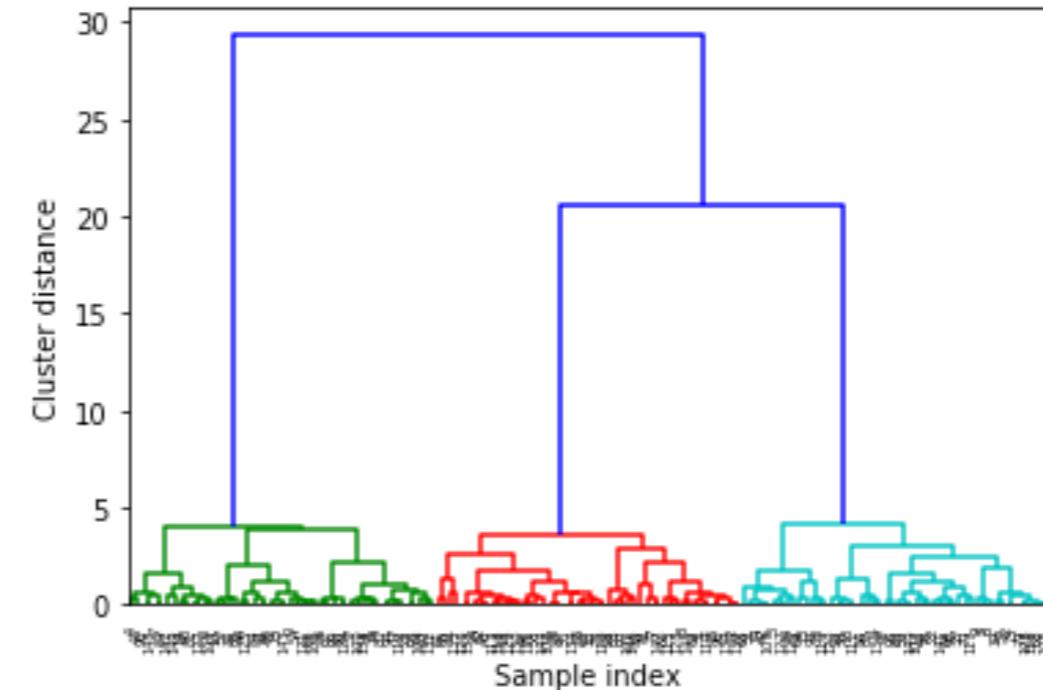
Complete linkage



Average linkage



Ward's linkage



# Python code

## ■ 利用Scikit-learn來完成凝聚分層

```
from sklearn.cluster import AgglomerativeClustering  
  
ac = AgglomerativeClustering(linkage = 'complete',  
                               affinity = 'euclidean',  
                               n_clusters = 3)  
labels = ac.fit_predict(X)
```

將資料分成 3 個聚類

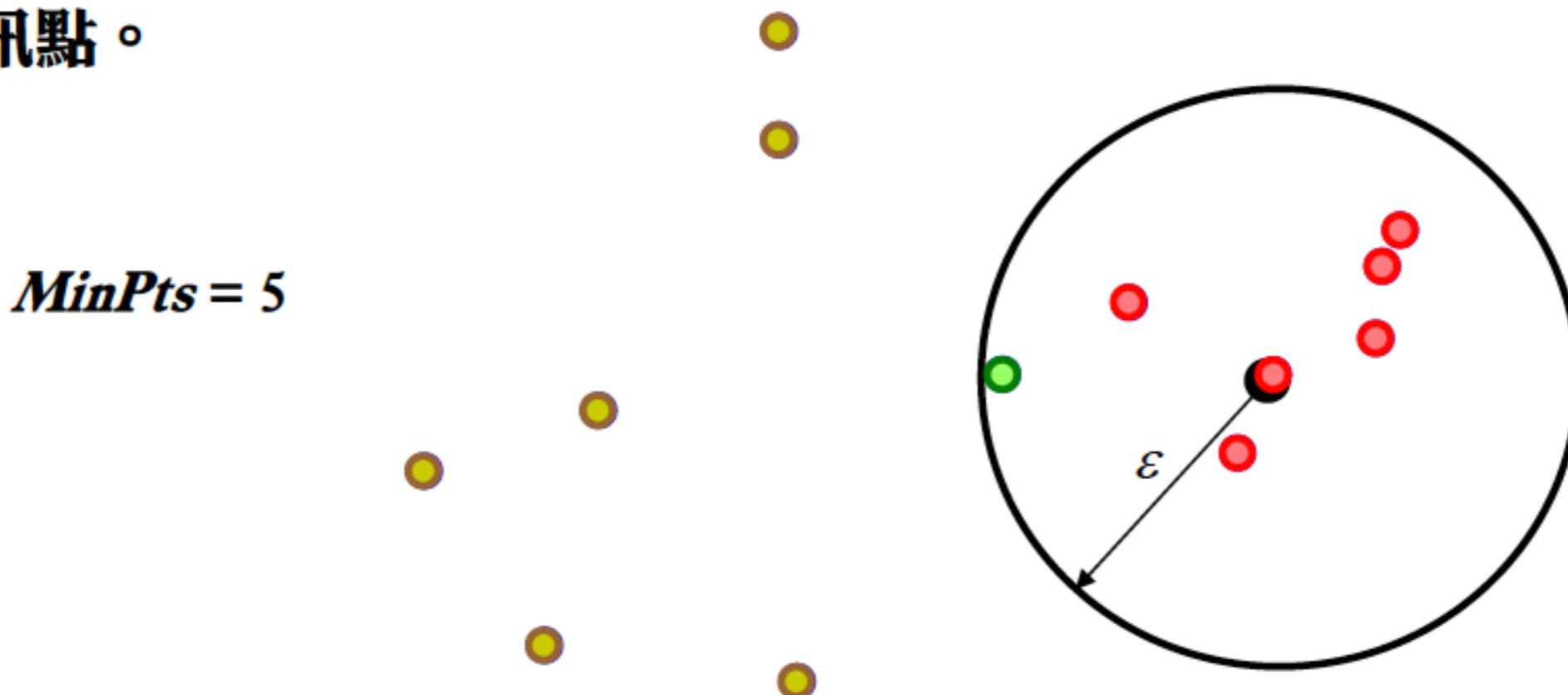
# 密度式聚類

<https://www.youtube.com/watch?v=zAbnJ7kERXk>

- 以密度為基礎之聚類法（density-based clustering）會找出遠離低密度區域之高密度的區域。
- DBSCAN (Density-Based Spatial Clustering of Applications with Noise) 為一個簡單且有效之以密度為基礎聚類演算法，並可用密度的概念剔除不屬於聚類資料的所有雜訊點。
- DBSCAN基本名詞定義如下：
  - $\varepsilon$  (或 *Eps*)：以某資料點為圓心所設的半徑長度。
  - *MinPts*：通常做為密度門檻值之用。以某資料點為圓心、 $\varepsilon$  為半徑所構成之區域內所包含的資料點最小個數。若滿足此一門檻值，表示區域內的資料點密度達到最小要求。

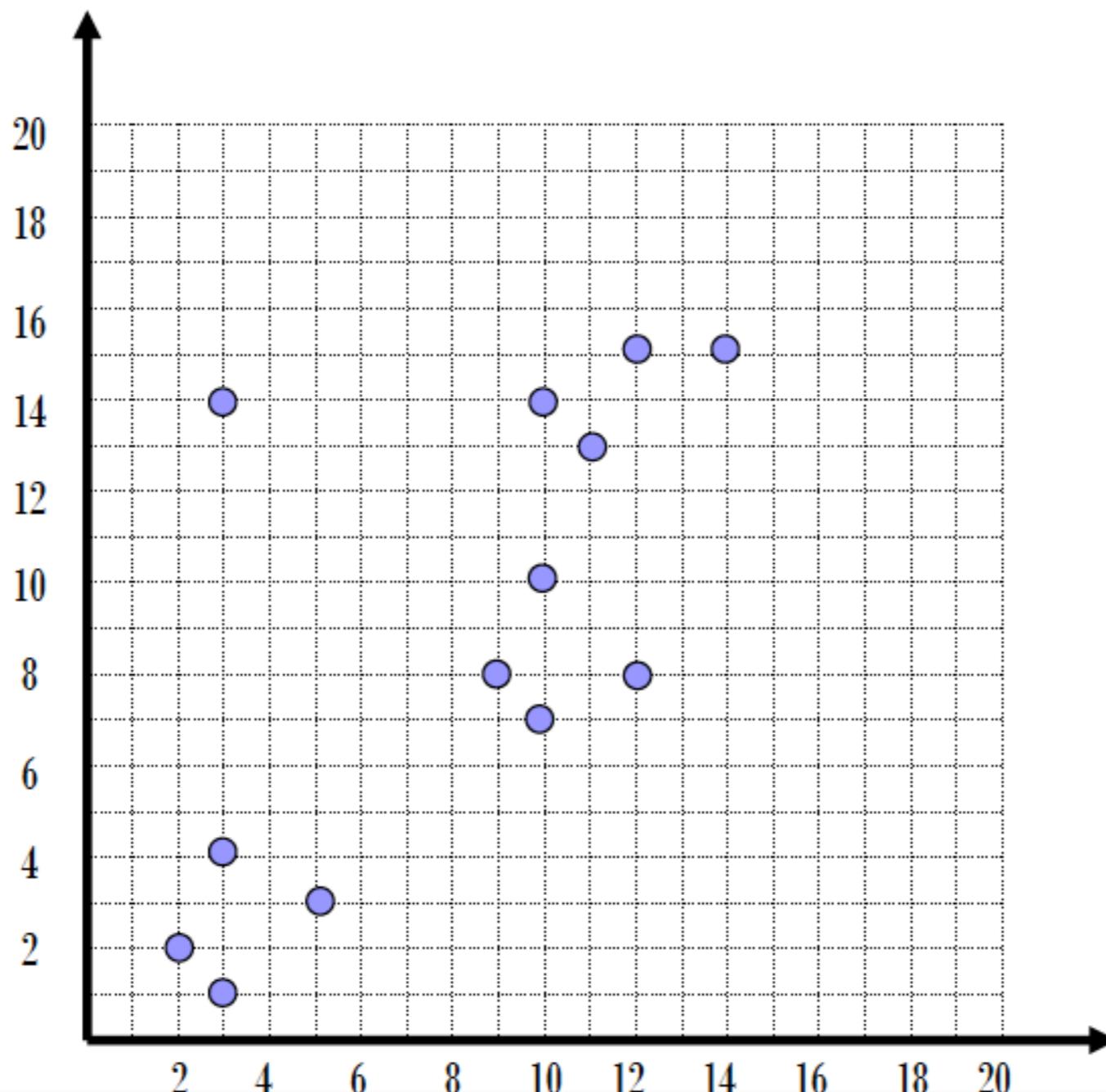
# 密度式聚類

- **Core Points (核心點)**：以某一點為圓心， $\varepsilon$ 為半徑所圍繞出來的範圍能包含超過 $MinPts$ 指定的資料點數目（包含該點），則此一圓心點即為核心點。
- **Border Points (邊緣點)**：若有一點被某個核心點包含，但若以它為圓心卻沒辦法包含超過 $MinPts$ 指定的資料點數目，則該點即為邊緣點
- **Noise Points (雜訊點)**：不屬於核心點，也不屬於邊界點，即為雜訊點。



# Example

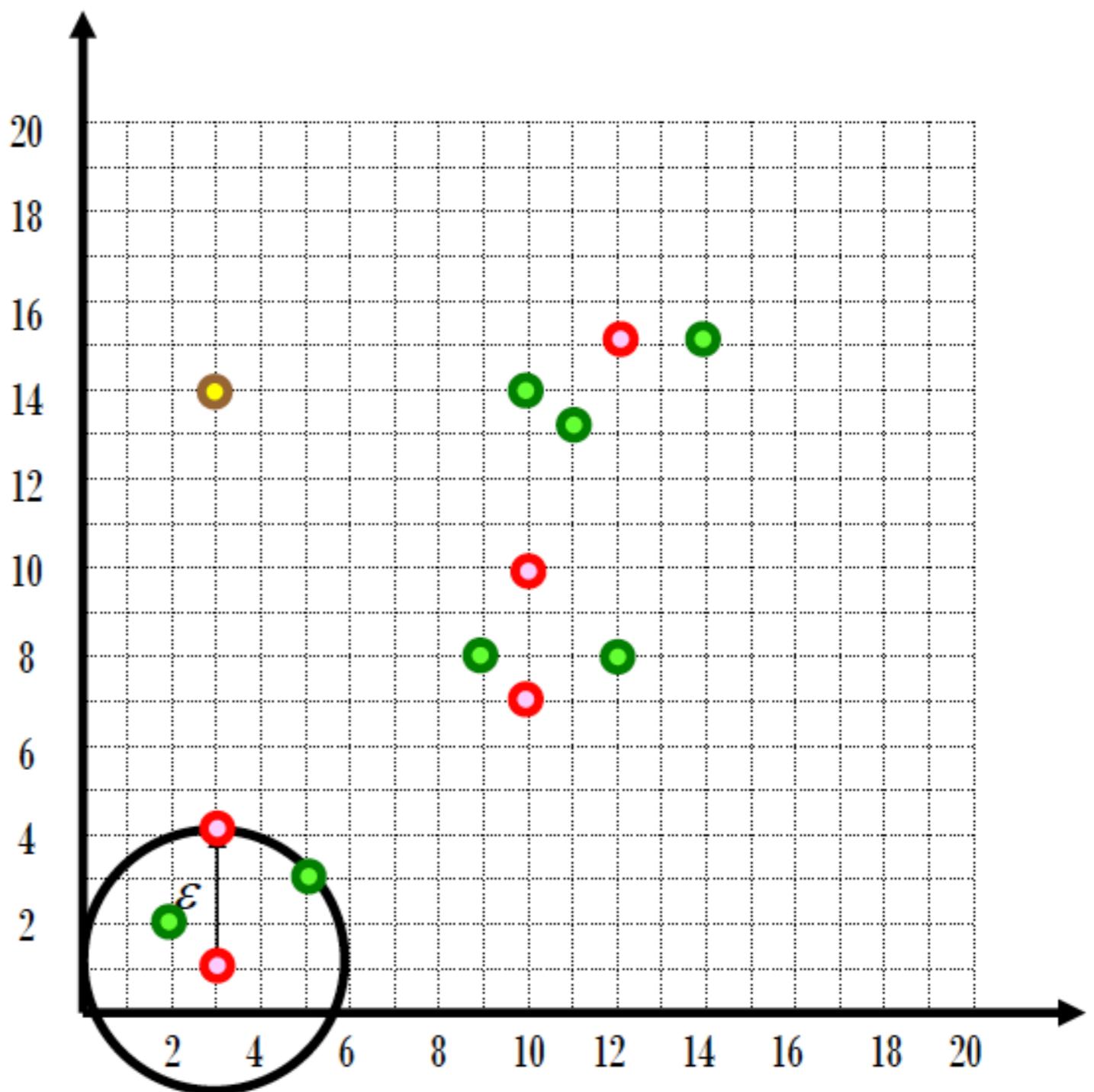
- 假設有13個資料點如下。令 $\varepsilon = 3$ ， $MinPis = 4$ ，請利用DBSCAN演算法加以聚類。



# Example

1. 將所有的點做過一次搜尋，找出核心點、邊界點、雜訊點

- 核心點
- 邊界點
- 雜訊點



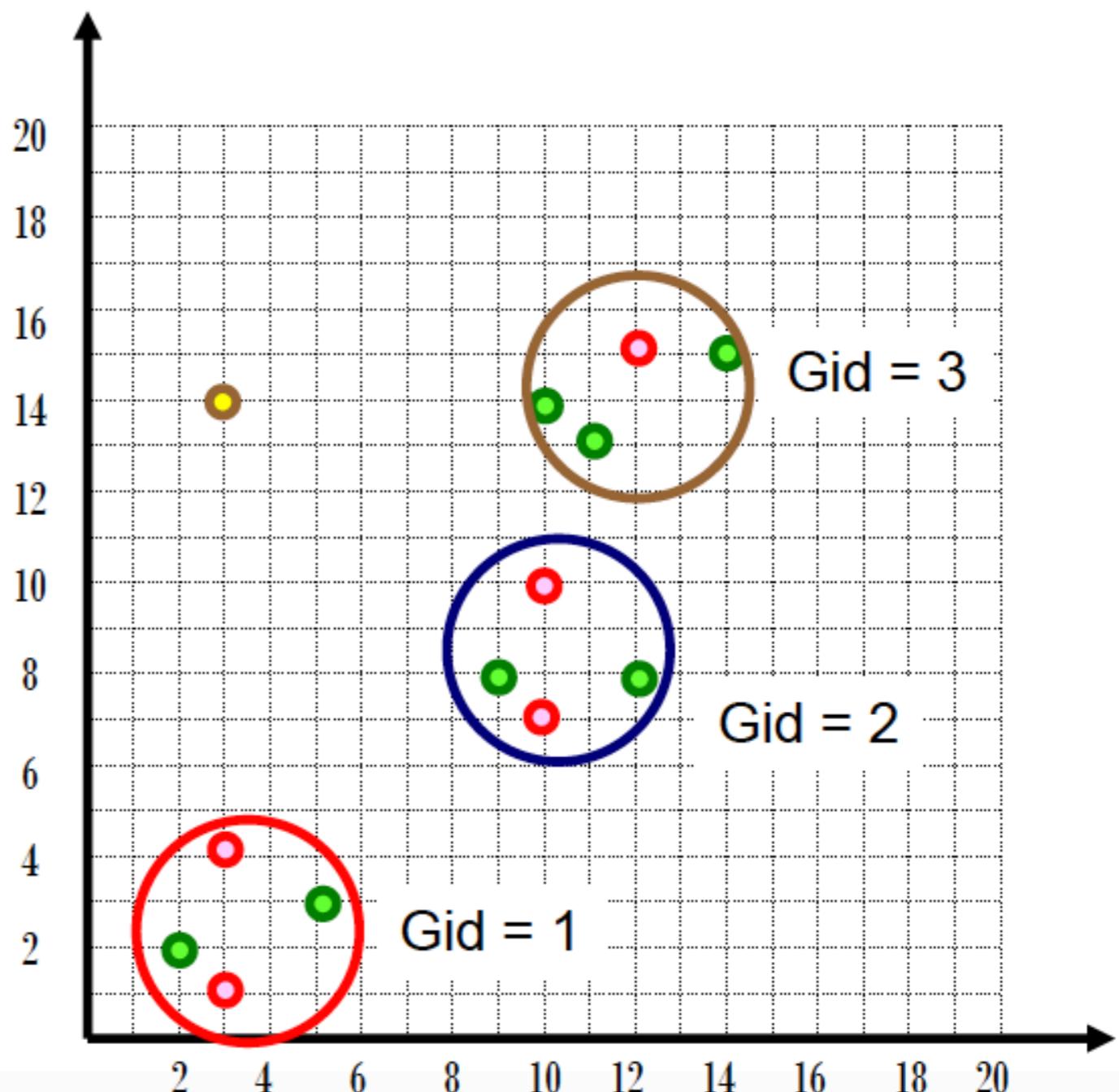
2. 忽略所有雜訊點

# Example

3.  $\text{Gid} = 0;$

4. 執行找尋聚類的for迴圈

- 核心點
- 邊界點
- 雜訊點



# Python code - 聚類方法的比較

## ■ 使用DBSCAN作聚類分析

```
from sklearn.cluster import DBSCAN  
db = DBSCAN(eps=0.2, min_samples=5, metric='euclidean')  
y_db = db.fit_predict(X)
```

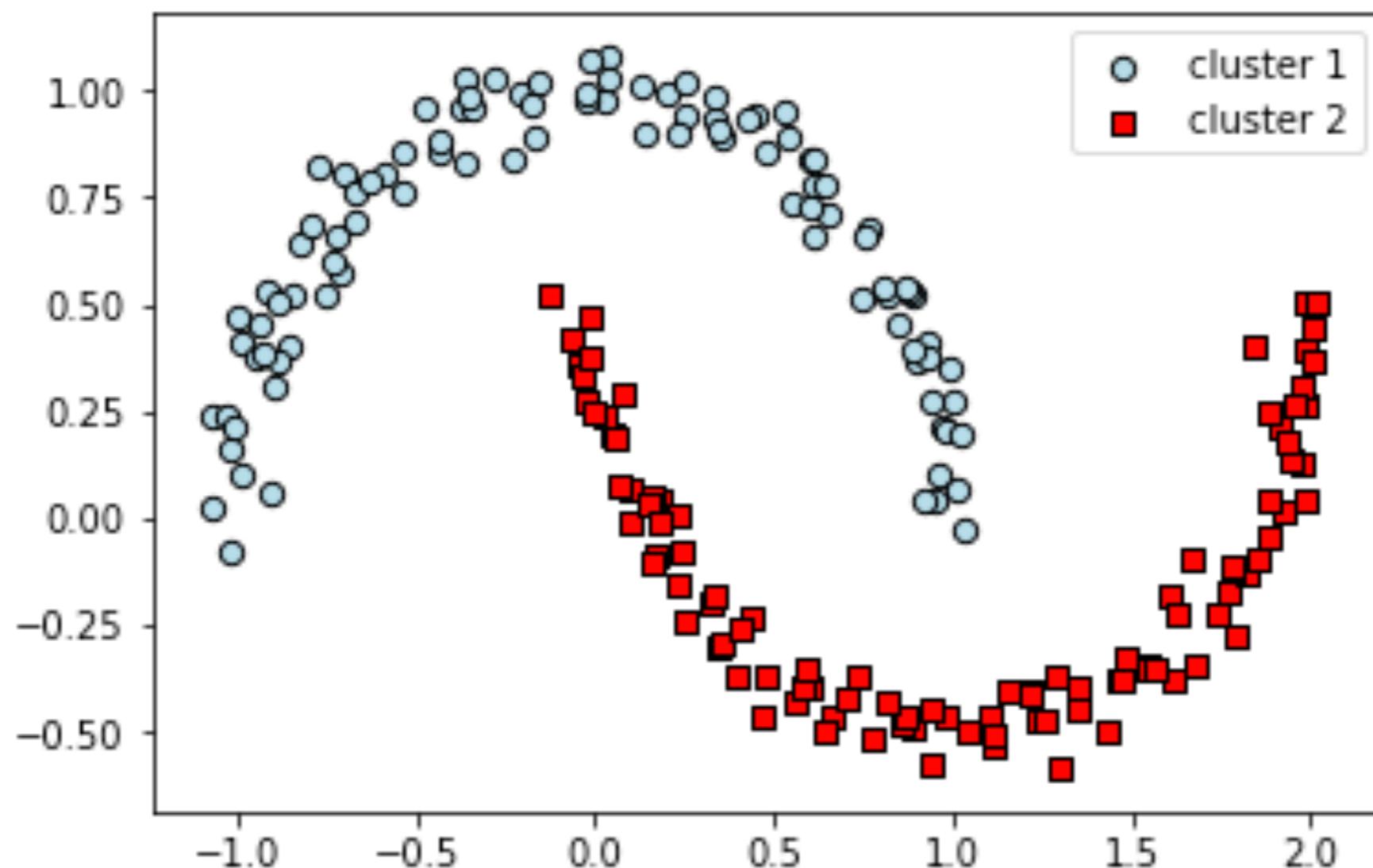
# Python code - 聚類方法的比較

## ■ DBSCAN

```
from sklearn.cluster import DBSCAN
db = DBSCAN(eps=0.2, min_samples=5, metric='euclidean')
y_db = db.fit_predict(X)
plt.scatter(X[y_db == 0, 0], X[y_db == 0, 1],
            c='lightblue', marker='o', s=40,
            edgecolor='black',
            label='cluster 1')
plt.scatter(X[y_db == 1, 0], X[y_db == 1, 1],
            c='red', marker='s', s=40,
            edgecolor='black',
            label='cluster 2')
plt.legend()
plt.show()
```

# Python code - 聚類方法的比較

## ■ DBSCAN



# 聚類分析方法的比較

- 聚類分析的方法非常繁多，但到現在還沒有任何方法被確定為最優異的方法，而每個方法所得的結果有時又略有出入。
- 此外，目前聚類分析的結果應保留多少聚類尚無定論。雖然學者已發展出各種聚類顯著性考驗，但迄今尚乏一個大家公認的理想方法。
- 因此，目前一般研究者採用的應對措施是在做研究時兼採幾種不同的聚類分析，再根據各種結果的意義性和可解釋性，從中挑選一個。