群聚分析（Cluster Analysis）

> Reference：
>
> 1. Cluster Analysis - Dr. Tun-Wen Pai

群聚分析

主要分成兩種不同的群聚分析：

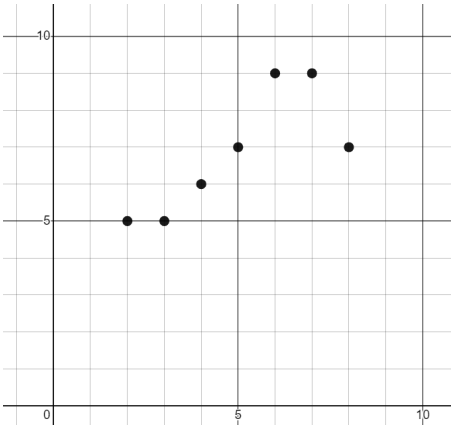
1. 分層式群聚：分成由上至下（top-down）或由下至上（bottom-up）演算法，層層迭代運算。
2. 分割式群聚：一次把所有的分群結果納入考慮。

分割式群聚

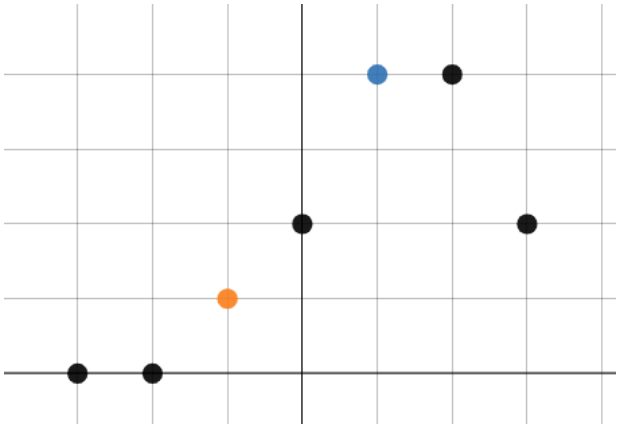
這邊舉一個最經典的例子：K-Mean。

簡單的步驟舉例

舉個例子，我們有個資料集。

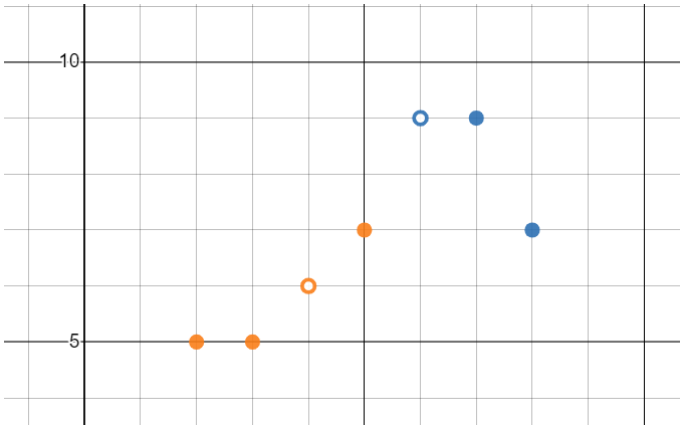


既然要分群，我們先***雖然說是隨機，但是為了做黑魔法所以不隨機的***挑選兩個群心。

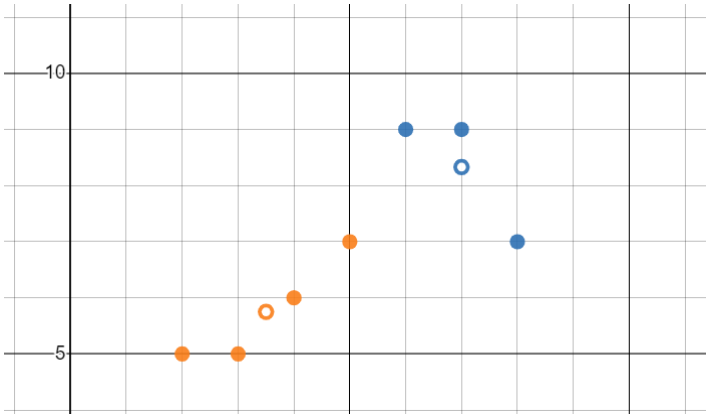


接下來我們開始迭代每個點，來知道距離哪個群心比較近，就屬於那個群心。

根據上圖來說，我們用眼睛可以看出這個分類結果，為了方便辨識群心在哪，我將群心用空心圓代替。



此時我們可以更新群心，利用每個向度 x, y 的平均。

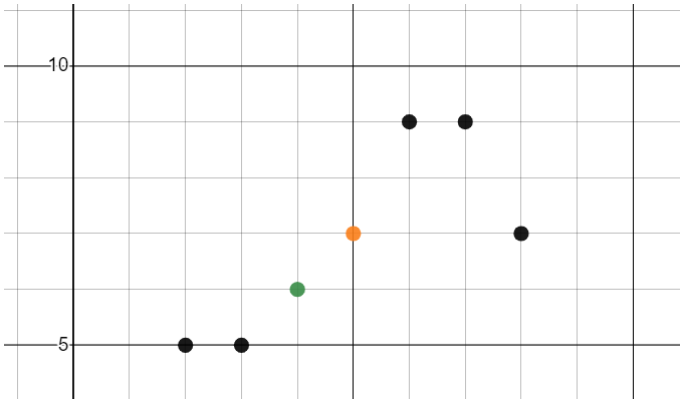


經過多次迭代之後就可以找到一個穩定的分群。

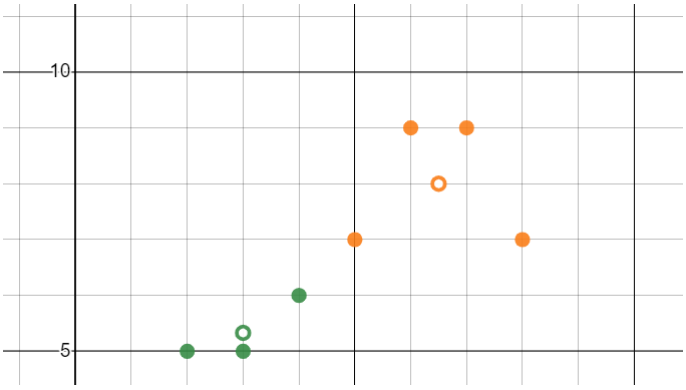
問題探討

顯然，我們用***雖然說是隨機，但是為了做黑魔法所以不隨機的***方式挑選群心是一個很大的問題。

因為隨機挑選群心可能會挑出一個完全不同的結果，舉個例子：



群心這樣挑選會得到這樣的結果。



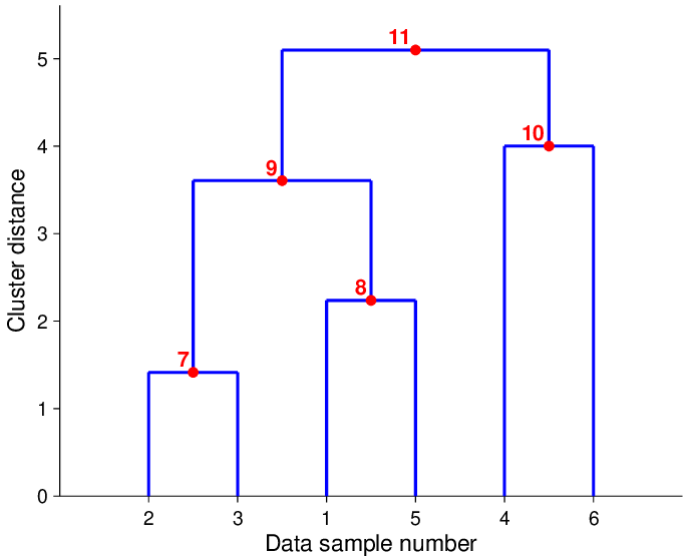
跟前面的結果幾乎完全不一樣，所以挑選適當的群心是非常重要的。

分層式群聚

簡介

分層式群聚可以將群聚的步驟畫成一樹枝狀圖。

> Image source：Hierarchical Clustering for Large Data Sets



在這份筆記會介紹四種不同的分層式群聚。

令 A 為某個群集的點，B 為某個群集的點，以下是群集表格。

分群演算法	群聚概念	演算表達式
Single Linkage	以兩個群聚的最近點	$\min\{d(a,b) : a \in A, b \in B\}$
Complete Linkage	以兩個群聚的最遠點	$\max\{d(a,b) : a \in A, b \in B\}$
UPGMA	以兩群的平均距離	$\frac{1}{\lvert A\rvert + \lvert B\rvert}\sum_{a\in A}\sum_{b\in B} d(a,b)$
WPGMA	以某點至兩群和的平均距離	$d(i \cup j, k) = \frac{d(i,k) + d(j,k)}{2}$

群集步驟

以上的演算法其實都大同小異，只有差別在演算法更新「群集距離表」的方式不同。

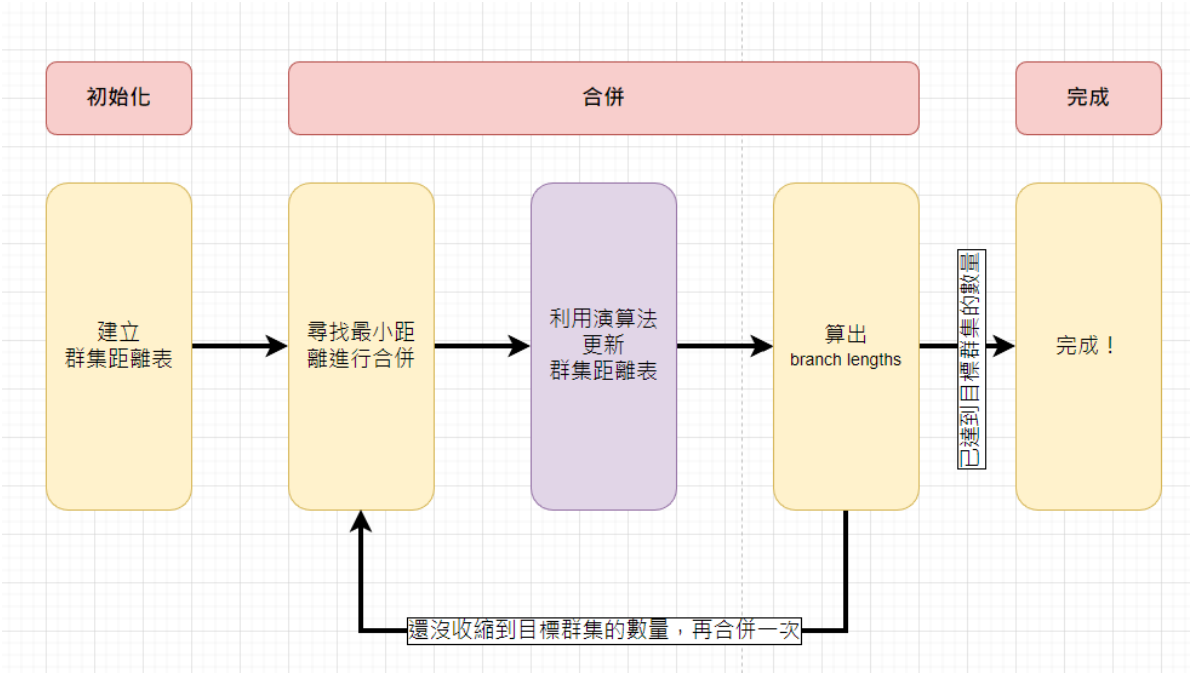
群集距離表即為每個群集到其他群集的距離，與自己的群集距離為 0。

例如下方表格即為群集距離表。

以圖上來說，群集 A 至群集 B 的距離為 25，群集 B 至群集 C 的距離為 25，群集 A 至群集 C 的距離為 15。

	A	B	C
A	0	25	15
B	25	0	25
C	15	25	0

這張流程圖是我自製的，用來解釋群集分析的流程。



利用演算法更新群集距離表

基本上群聚的方法只差在利用演算法來合併群集與更新群聚分析表，也就是更新除了合併群集以外群集距離。

	A	B	C	D
A	0	25	15	18
B	25	0	25	35
C	15	25	0	45
D	18	35	45	0

以上方這個表格為例，要尋找最小距離進行合併，該合併的兩個群集即為 A 與 C。

請注意，A 與 C 非常重要。

例如當你欲合併的結果為 (B,C) 與 D 得到 (B,C,D) 時，你應該要在更新時使用 (B,C) 欄的值與 D 欄的值。

否則你可能會用 B 與 (C,D) 的值，但表格上 (C,D) 的值並不存在，所以記住什麼與什麼合併是非常重要。

	(A, C)	B	D
(A, C)	0	?	?
B	?	0	35
D	?	35	0

合併完成之後，此時我們想要知道群集 (A,C) 與 B 的距離，此時我們根據演算法的定義：$\frac{d(i,k)+d(j,k)}{2}$

$d(i,k)=d(A,B)=25$，$d(j,k)=d(C,B)=25$，可以得到距離為 $d((A,C),B)=\frac{25+25}{2}=25$。

故$d((A,C),B)=25$

同理我們可以算出 $d((A,C),D)=\frac{d(A,D)+d(C,D)}{2}=\frac{18+45}{2}=\frac{63}{2}=31.5$

	(A, C)	B	D
(A, C)	0	25	31.5
B	25	0	35
D	31.5	35	0

且我們可以得到 branch length = $\frac{15}{2}=7.5$，也就是我們合併時使用的最短距離除以 2。

通常來說會越來越大，就能在樹枝狀圖上畫出 branch length（或者你可以說他是 thoushold）。

舉例其他演算法更新群集距離表

此時就能完成表格更新的步驟，其餘的合併方法只差在演算法的不同，照著做就行。

以其他三種不同的演算法來說：

1. Single Linkage：更新其他群集距離的方式為，取合併群集與該群集的**最短距離**。
2. Complete Linkage：更新其他群集距離的方式為，取合併群集與該群集的**最長距離**。
3. UPGMA：更新其他群集距離的方式為，若合併群集為 A 與 B 的合併，且欲合併的群集為 C。
   則距離更新方式為 $d((A,C),B)=\frac{|A|\times d(A,C)+|B|\times d(B,C)}{|A|+|B|}$，其中 $|A|$ 與 $|B|$ 為該群集的點數量。
4. WPGMA：更新其他群集距離的方式為，若合併群集為 A 與 B 的合併，且欲合併的群集為 C。
   則距離更新方式為 $d((A,C),B)=\frac{d(A,C)+d(B,C)}{2}$

優化分群結果

- 什麼樣的分群結果是好的？
  - 我們可以將不同的 K 值進行計算 total variation，並且找出拐點。