# 結合 K-means 及階層式分群法之二階段分群演算法

# A New Two-Phase Clustering Algorithm Based on K-means and Hierarchical Clustering with Single-Linkage Agglomerative Method

陳同孝[1] 陳雨霖[2] 劉明山[2] 許文綬[2] 林志強[1] 邱永興[1,*]

[1]國立臺中技術學院

資訊科技與應用研究所

台中市 404 北區三民路三段 129 號

tschen@ntit.edu.tw

[2]國立臺中技術學院附設進修學院

資訊管理系

台中市 404 北區三民路三段 129 號

rainchen.tw@yahoo.com.tw

Tung-Shou Chen[1] Yu-Lin Chen[2] Ming-Shan Liou[2] When-Shou Hsu[2]
Chih-Chiang Lin[1] Yung-Hsing Chiu[1,*]

1 Graduate School of Computer Science and Information Technology, National Taichung Institute of Technology, Taichung City 404, Taiwan

tschen@ntit.edu.tw *mis.joe@gmail.com

2 Department of Information Management, National Taichung Institute of Technology, Taichung City 404, Taiwan

rainchen.tw@yahoo.com.tw

*Received 30 April 2006; Revised 3 July 2006 ; Accepted 5 July 2006*

**摘 要**

本文提出一個二階段分群演算法：階層式K-means分群法 (HKC，Hierarchical K-means Clustering)。在分割階段，HKC以K-means將資料集合分割成多個群聚。在此增加群聚的數量是爲了降低雜訊及離群值對K-means的影響。在合併階段則採用單一連結聚合演算法來彌補K-means無法探索任意形狀群聚的缺點，並且還能提供樹狀的分群結果。由於K-means將所有要處理的資料減化成數個群聚，所以HKC可以快速的產生樹狀的分群結果。實驗結果顯示，HKC的準確率相當的良好，並且能更有效率地產生樹狀分群結果。

**關鍵詞：**分群演算法、階層式分群法、K-means、單一連結聚合演算法

* 通訊作者

ABSTRACT

We propose a new clustering algorithm: hierarchical K-means clustering algorithm (HKC), in this paper. HKC consists of two phases. In the first phase, HKC employs K-means clustering algorithm to split the original data into some groups. The purpose of the first phase is to handle the outliers and noises. In the second phase, HKC employs single-linkage agglomerative algorithm, which can discover the arbitrarily shaped clusters and produce a clustering tree, to merge the groups. Since the processed data are simplified to some groups by K-means, the clustering tree could be obtained quickly. In this paper, the accuracy of HKC is evaluated and compared with those of K-means and hierarchical clustering. The experimental results indicated that the accuracy of HKC is better than K-means and hierarchical clustering. Hence HKC could assist the researchers to quickly and accurately analyze data.

**Keywords:** Clustering algorithm, Hierarchical clustering, K-means, Single-linkage agglomerative algorithm.

# 一、前言

群聚技術(Clustering Technology)[1]可以將資料依據分群的目標分成許多群聚；而被凝聚在同一群的資料會有某些特性是相近的；也就是說，被分成不同群聚的資料就會有某些特性會明顯不同。工程與科學界常使用這個技術來處理生物資訊(Bioinformatics)[2]、資料挖掘(Data Mining)[3]、人工智慧(Artificial Intelligence)[4]等等問題。本篇文章是希望將群聚技術做深入的探討，並將實務上常用的方法加以改良，期盼能提供使用者在群聚分析上的另一種選擇。

先前被提出的群聚技術大致上可以分二大類：階層式分群演算法(Hierarchical Clustering Algorithms)與非階層式分群演算法(Non-hierarchical Clustering Algorithms)[5]。前者還可以再細分爲聚合法(Agglomerative Algorithm)與分裂法(Divisive algorithm)二種[6]。聚合法是先將每一筆資料視爲一個群聚，然後每次將特性最相近的二個群聚合而爲一，直到群聚數目達到事先所設定的數目爲止。而分裂法是先把整個資料集合看成一個群聚，然後逐次分裂，每次都會在其中一個群聚裡，切割相似度最低的連結，成爲二個較小的群聚，直到群聚數目達到事先所設定的數目爲止。以此類方法所產生的分群結果可以是一個樹狀圖的形態，其較鄰近的節點就是較相似的資料，且群聚合併與分裂的過程也能表現出來。而常見的此類方法還有 BIRCH [7]、CURE [8]、ROCK [9]、CHAMELEON [10]等。

一般應用階層式分群演算法時，較常使用聚合法，而其群聚距離的評估方式有四種[11]，第一種是重心連結聚合演算法（Centroid-linkage Agglomerative Algorithm），如公式(1)所示，計算式中的 $C$ 代表群聚，$m_i$ 代表群聚 $C_i$ 的重心，其評估方式爲二個群聚重心的間距。第二種是平均連結聚合演算法（Average-linkage Agglomerative Algorithm），如公式(2)所示，計算式中的 $x$ 與 $x'$ 分別代表二個群聚內的資料點，其評估方式爲二個群聚間，資料點與資料點的距離總和之平均。第三種是完整連結聚合演算法（Complete-linkage Agglomerative Algorithm），如公式(3)所示，其評估方式爲二個群聚間，最遠的兩個資料點之距離。第四種是單一連結聚合演算法（Single-linkage Agglomerative Algorithm），如公式(4)所示，其評估方式爲不同群聚中，最接近的兩個資料點之距離。

$$d_{\text{mean}}(C_i,C_j)=\|m_i-m_j\| \tag{1}$$

$$d_{\text{avg}}(C_i,C_j)=\frac{1}{|C_i||C_j|}\sum_{x\in C_i}\sum_{x'\in C_j}\|x-x'\| \tag{2}$$

$$d_{\max}(C_i,C_j)=\max_{x\in C_i,x'\in C_j}\|x-x'\| \tag{3}$$

$$d_{\min}(C_i,C_j)=\min_{x\in C_i,x'\in C_j}\|x-x'\| \quad (4)$$

以上四種評估方式中，只有單一連結聚合演算法可以探索任意形狀的群聚[12]，但是也因爲此特性而造成它容易受雜訊及離群值所影響[13]。

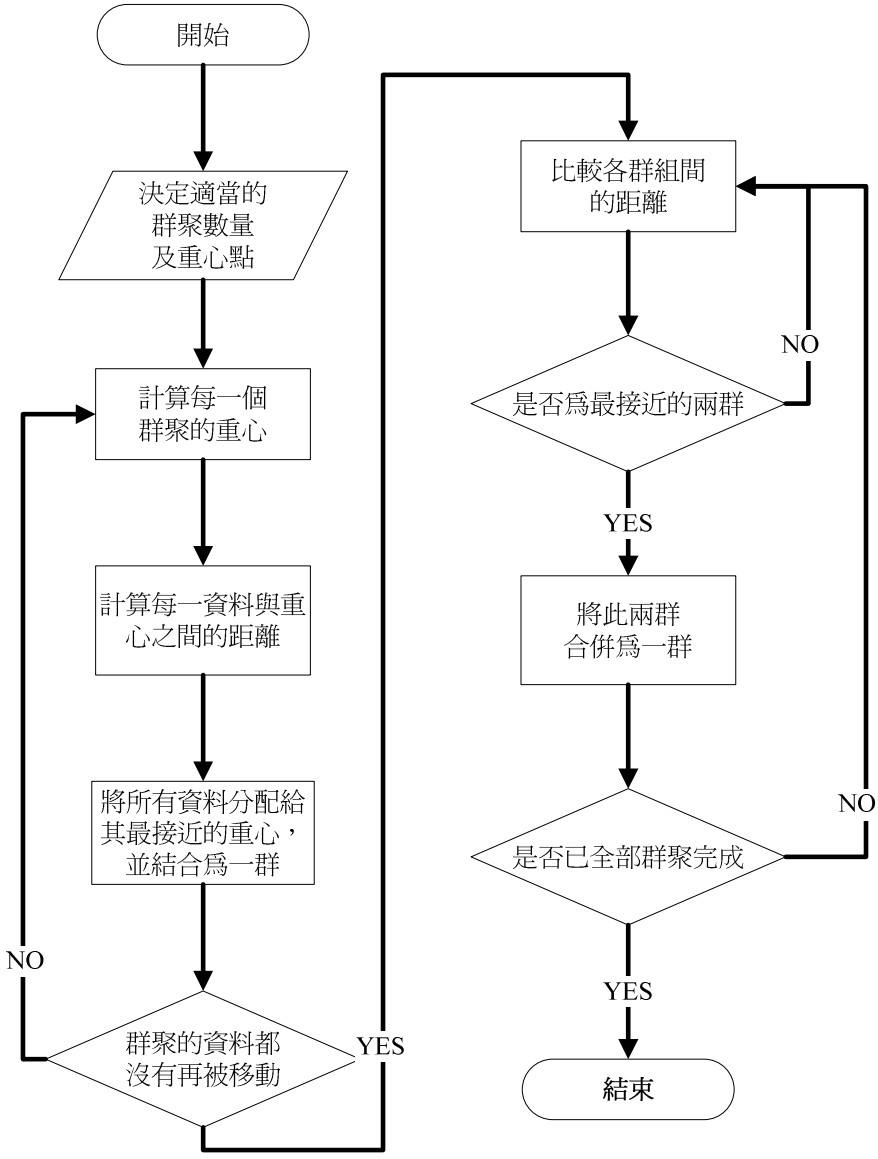
在所有的非階層式分群演算法中，K-means 是最典型的方法[14]。在使用此方法之前，必須先決定分群結果的群聚數量，也就是定義 *K* 的值。當 K-means 初始化時，會先任意選擇 *K* 個資料點做爲群聚的中心點。接著依據所有資料點與每一個群聚中心點的距離，將所有的資料點分配到各自最接近的群聚。然後再從新產生的每一個群聚中，找出新的群聚中心點，並依照新產生的群聚中心點來重新分配所有的資料點。此步驟會反覆執行到每一個群聚中心都不再改變爲止。K-means 以群聚的中心點來代表所有資料點，所以能減少大量的計算，但是隨機選擇的初始中心點不恰當時，會造成分群效率不佳，降低分群可靠度[6]。而且該方法以群聚的重心作爲群聚的代表點（Representative Objects），所以群聚結果很容易被雜訊（Noises）或是離群值（Outliers）所影響，而且無法辨識出非凸邊形的群聚[6]。而後續提出的非階層式群聚演算法中較重要的方法有 PAM[15]、CLARA[15]以及 CLARANS [16]等。

先前曾有學者[17]利用階層式分群演算法的初期分群結果來決定 K-means 的初始中心點，用以解決分群結果不穩定的問題；另外有一份研究[18]是以 K-means 將資料疊代分裂至每一群只有一個資料點爲止，並且快速地以此過程架構出一個樹狀結構。然而，這些方法都無法改善 K-means 無法探索任意形狀群聚的缺點，前者是因爲其後續的分群步驟都是以 K-means 來聚合資料所造成；後者的原因是因爲它只是用 K-means 來將一群資料一分爲二。而本研究所探討的是要能辨識任意形狀的群聚的方法，因此這二份文獻所提出的方法不適合與本研究一同比較。

本篇文章將提出一個新的分群演算法，用以協助使用者快速且準確地分析資料。我們將它命名爲階層式 K-means 分群演算法 (HKC, Hierarchical K-means Clustering Algorithm)。HKC 將結合 K-means 與階層式分群演算法的優點並互補對方缺點。HKC 是一個二階段的分群演算法。在分割階段，HKC 將先採用 K-means 分群演算法快速地將整個資料集合分割成多個群聚。在這個步驟裡，爲了避免雜訊與離群值的干擾，HKC 將會增加群聚的數量。在群聚的數量被增加以後，群聚所包含的資料數量將會減少，而原本被我們視爲雜訊和離群值的資料就只能影響與其較相近的資料，也就是說雜訊和離群的干擾被減輕了。在合併階段，HKC 計劃採用單一連結聚合演算法（Single-linkage Agglomerative Algorithm）來彌補 K-means 分群演算法無法探索任意形狀之群聚的缺點，並且能提供樹狀的分群結果方便使用者分析工作的進行。由於 K-means 分群演算法將所有要處理的資料減化成數個群聚，所以其後再執行的單一連結聚合演算法就可以較快速的產生樹狀的分群結果。本篇文章的第二節，將介紹我們所提出之分群演算法 HKC。第三節爲 HKC 與其他分群演算法比較之實驗與討論。第四節爲本文之結論。

## 二、階層式 K-means 分群演算法 (HKC, Hierarchical K-means Clustering)

HKC 包含二個階段：分割階段與合併階段，其演算法的流程如圖一所示。以下將分別說明各階段內的詳細步驟。




圖一： 階層式K-means分群演算法之流程圖

### 1. 分割階段

在分割階段，HKC 以 K-means 分群演算法將整個資料集合 $X=\{x_1, x_2,\ldots, x_n\}$分割成 $K$ 個群聚 $G=\{g_1, g_2,\ldots, g_K\}$；$K$ 值最大時，每個資料點 $x_i$ 都代表一個群聚 $g_i$；$K$ 值最小時，

整個資料集合 $X$ 就是一個群聚 $g_i$。如圖一的左半邊所示，當 HKC 初始化時，會先任意選擇 $K$ 個資料點 $x_i$ 做爲群聚的重心點 $M=(m_1, m_2,\ldots, m_K)$, $m_i \in g_i$。接著依據所有資料點 $x_i$ 與每一個群聚重心點 $m_i$ 的距離，將所有的資料點 $x_i$ 分配給各自最接近的群聚重心點 $m_i$。然後再從新產生的每一個群聚 $g_i$'裡新的群聚中心點 $m_i$'，並依照新產生的群聚中心點 $m_i$'來重新分配所有的資料點 $x_i$。此步驟會反覆執行到所有的群聚 $G$ 都不再變動爲止。在這個階段裡，HKC 爲了減輕雜訊與離群值的干擾以及初始中心點不恰當所造成的問題，因此增加了群聚的數量 $K$。在群聚的數量被增加以後，群聚所包含的資料數量將會減少，而原本被我們視爲雜訊和離群值的資料就只能與其較相近的資料結合成一個群聚，因此就不會影響到其它的群聚。

如何決定一個適當的 $K$ 是一個值得探討的問題，但是 HKC 的分群結果對 $K$ 的數值大小並不敏感。$K$ 的數值應當隨著 $X$ 的大小而被改變，若給予過大或過小的 $K$ 都會使 HKC 的效率降低。

**2. 合併階段**

在合併階段，HKC 採用單一連結聚合演算法來合併在分割階段所產生的 $G$ 個群聚，如圖一的右半邊所示。在合併過程中，HKC 會逐步將最接近的二個群聚 $g_i, g_j \forall i, j \in \{1,2,...,K\}$ 合而爲一，直到群聚數目達到事先所設定的數目或 1 爲止。而這些群聚之間是以彼此最接近的資料點 $x_i, x_j \forall i \in \{1,2,...,n\}, x_i \in g_i, x_j \in g_j$ 來衡量彼此之間的距離 $d_{\mathrm{g}}$，如公式(5)所示。

$$d_{\mathrm{g}} = \min_{i \neq j} \| x_i - x_j \| \tag{5}$$

由於單一連結聚合演算法能夠探索任意形狀的群聚，所以能彌補 K-means 分群演算法無法探索任意形狀群聚的缺點。在上一個階段中，HKC 已經利用 K-means 分群演算法將所有要處理的資料集合 $X$ 合併成群聚集 $G$，因此提高了單一連結聚合演算法的計算效率，使得 HKC 能更快速地產生樹狀的分群結果，進而方便使用者分析工作的進行。

## 三、實驗結果

本文將以 HKC 與 K-means 及階層式分群演算法的做實驗比較。實驗結果將以分群準確率來表示，如公式(6)所示，其中的 $a_i$ 是代表每一個群聚中，分群結果與原始類別相符合的資料數量。

表一： 分群準確率比較表

| Algorithm / Data Set | HKC | K-means | Hierarchical Clustering | | | |
|---|---|---|---|---|---|---|
| | | | Centroid | Average | Complete | Single |
| A | 100% | 74.6% | 100% | 100% | 61.1% | 80.3% |
| B | 100% | 43.9% | 50.7% | 52.0% | 61.3% | 100% |
| C | 100% | 77.3% | 85.4% | 88.1% | 82.4% | 100% |
| D | 100% | 51.8% | 60.0% | 49.8% | 65.9% | 100% |
| IRIS | 98.6% | 84.1% | 90.7% | 74.7% | 84.0% | 68.0% |

$$\text{分群準確率 (\%)} = \frac{\sum_{i=1}^{m} a_i}{n} = \frac{\text{被正確分群的資料之數量}}{\text{所有資料的數量}} \tag{6}$$

本篇文章的實驗將以五項不同分佈型態的資料集(Data Set)來進行測試，這些資料集的前四項是二維資料，是取自於[5]，在該文中的用途也是用於評估分群準確度。第五項是著名的 IRIS，它是一個 4 維資料集，取自 UCI Machine Learning Database[19]。爲了簡化操作與說明，在前四項實驗中，HKC 的 K 統一被設定爲 30。由於第五項資料集的資料量較少(150 筆)，所以 K 被設定爲 10。

在進行 K-means 的實驗時，由於該方法的初始值會直接影響到最終的分群結果，因此它的實驗結果將以三十次測試數據的算術平均數做爲代表。在進行階層式演算法的測試時，由於該方法有四種群聚距離的評估方式，因此我們將四種評估方式在正確群聚數量下的實驗數據列爲實驗結果。以上所有測試的實驗數據皆記錄於表一。由表一中，我們可以發現 HKC 的分群準確率是最好的。K-means 的分群結果並不好，尤其是以 B 資料集測試時。如圖三～圖五所示，如果測試資料中的群聚間距較寬，則單一連結聚合演算法的會有不錯的準確率。在 IRIS 資料集的實驗中，所有分群方法的準確率都達不到 100%，這是因爲 IRIS 資料集中有二個群聚的部份資料重疊(如圖六所示)在一起所造成。但是在實驗結果中可以看出，HKC 的分群準確率達到 98.6%，高於其它比較對象。

在圖 2(a)中顯示的是 A 資料集，其中有三個群聚，總包含 203 筆資料。上方二個群聚內的資料都很緊密的聚合在一起，而在下方群聚內的資料則是分佈的很鬆散。上方二個群聚邊緣的距離相當的小，而與下方群聚邊緣的距離較大。上方二群聚的大小比下方群聚相對小很多。在 A 資料集的實驗中，K-means 因爲初始值不穩定，因此造成分群準確率不高，如圖 2(b)所示。而完整連結聚合演算法與單一連結聚合演算法在未完成下方群聚的合併之前，就先將上方二個較相近的群聚合併，所以造成分群準確率不佳，如圖 2(c)與 2(d)所示。而 HKC 與其餘的比較對象都能完全無誤地區別出三個群聚。

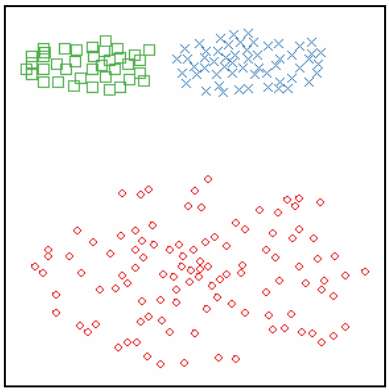


圖 2(a): A資料集

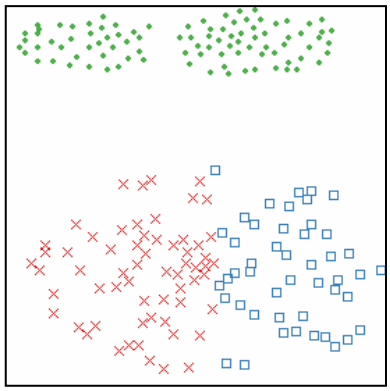


圖 2(b): K-means的結果之一

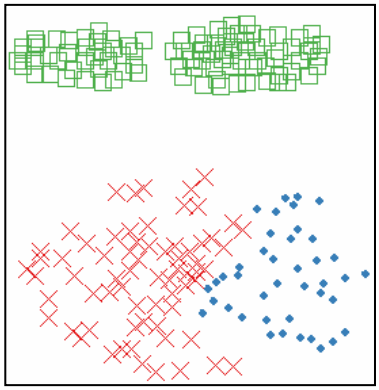


圖 2(c): Complete-linkage的結果

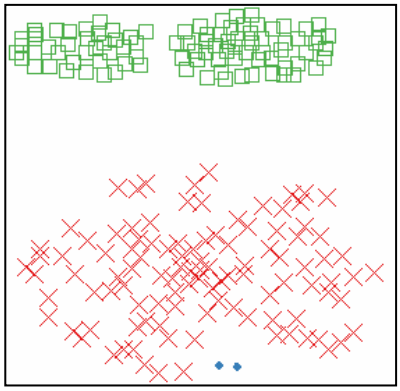


圖 2(d): Single-linkage的結果

圖二：A資料集實驗結果

在圖 3(a)中顯示的是 B 資料集，其中有一個較大的帶狀外環群聚與一個較小的圓形核心群聚，總共包含 150 筆資料。中央群聚內的資料都很緊密的聚合在一起，而外環群聚內的資料則是均勻的分佈。在 B 資料集的實驗中，K-means 因爲無法處理非凸邊形的群聚，所以造成分群準確率不良，如圖 3(b)所示。階層式演算法中，只有 HKC 和單一連結聚合演算法能一樣，將二個群聚完全無誤地區別出來。而其餘三種方法因爲無法有效處理這種分佈型態的資料，所以分群準確率都不高，如圖 3(c)、3(d)與 3(e)所示。

經由以上實驗，我們可以發現 HKC 能有效處理任意形狀的群聚，以及分群結果不會受到隨機決定初始重心點的影響， 而且在實驗的過程中，HKC 產生樹狀分群結果的速度要比階層式分群法來得快。由於以下二個實驗和 B 資料集的實驗結果非常相似，所以我們只對這二個資料集做簡要的描述。在圖四中顯示的是 C 資料集，總共包含 704 筆資料。在圖五中顯示的是 D 資料集，總共含有 1501 筆資料。

在圖六中顯示的是 Iris 資料集的二維投影圖，其中有三個群聚，各包含 50 筆資料。在三個群聚內的資料都很緊密的聚合在一起。左方的群聚與右方二個群聚有明顯的間距，而右方二個群聚有部份資料重疊在一起，我們可以將這些重疊的資料視爲雜訊或離群值。如表 1 所示，在 IRIS 資料集的實驗中， HKC 的分群準確率達到 98.6%，高於其它比較對象。由這個實驗結果，我們可以看出 HKC 能有效地降低雜訊及離群值對分群結果的影響。

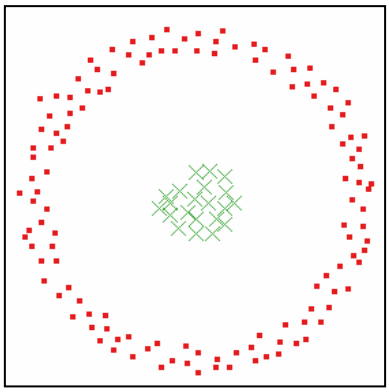


圖3(a): B資料集

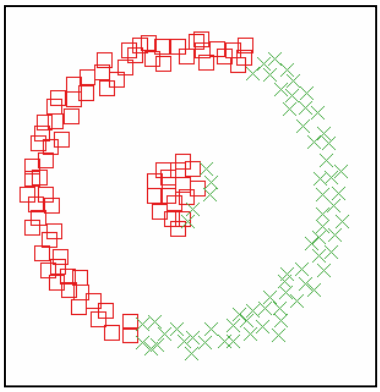


圖3(b): K-means的結果之一

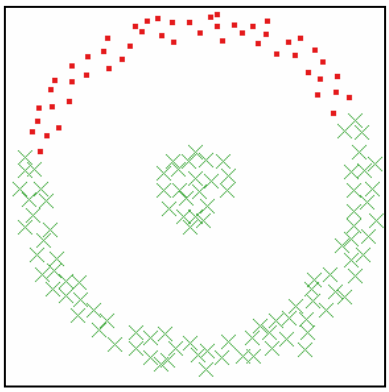


圖3(c): Average-linkage的結果

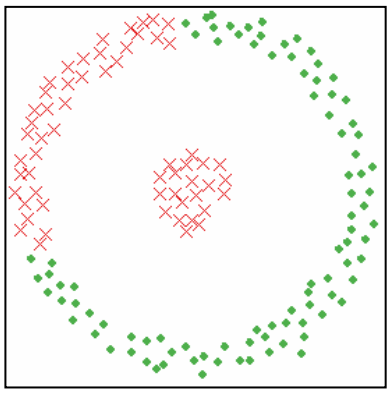


圖3(d): Complete-linkage的結果

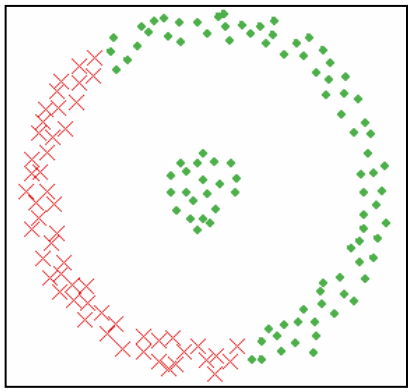
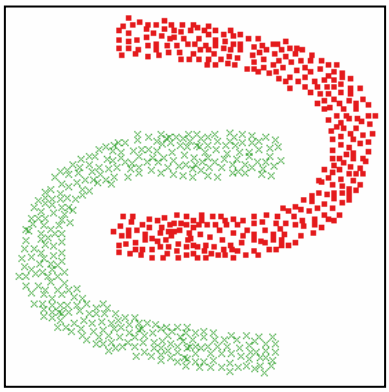
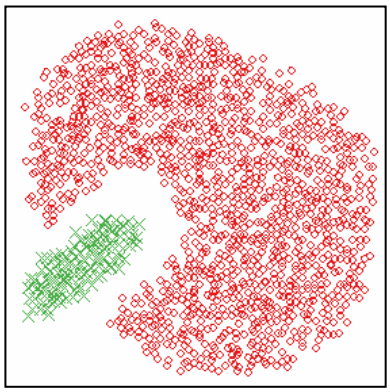


圖3(e): Centroid-linkage的結果

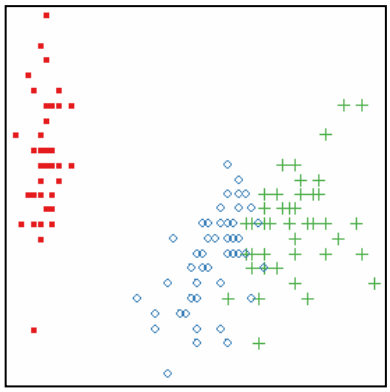
圖三：B資料集實驗結果



圖四：C資料集



圖五：D資料集



圖六：IRIS資料集

## 四、結論

本篇文章提出了一個新的分群演算法：階層式 K-means 分群法 (HKC , Hierarchical K-means Clustering)。HKC 結合了 K-means 與階層式分群演算法的優點並互補對方缺點。本篇文章的實驗結果顯示 HKC 可以探索任意形狀的群聚、更有效率地產生樹狀的分群結果、有效降低雜訊與離群值的干擾及初始中心點不恰當所造成的影響，且 HKC 的分群準確率比 K-means 及階層式分群演算法來得好，將可以協助使用者更快得到既準確又方便的資料分析結果。

## 參考文獻


[1] L. Kaufman, and P.J Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley and Sons, 1990.

[2] T.S. Chen, Y.T. Chen, C.C. Lin, and R.C. Chen, "A Combined K-Means and Hierarchical Clustering Method For Improving the Clustering Efficiency of Microarrary," *In Proceedings of 2005 International Symposium on Intelligent Signal Processing and Communications Systems*, 2005, pp. 405-408.

[3] M.H. Dunham, *Data Ming: Introductory and Advanced Topics*, Prentice Hall, 2003.

[4] R.J. Roiger, and M.W. Geatz, *Data Mining: A Tutorial-Based Primer*, Addison Wesley, 2003.

[5] G. Gautam, and B.B Chaudhuri, "A Novel Genetic Algorithm for Automatic Clustering," *Pattern Recognition Letters*, Vol. 25, 2004, pp. 173–187.

[6] T.S. Chen, C.C. Lin, Y.H. Chiu and R.C. Chen "Combined Density- and Constraint-based Algorithm for Clustering," *In Proceedings of 2006 International Conference on Intelligent Systems and Knowledge Engineering*, 2006.

[7] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An Efficient Data Clustering Method for Very Large Databases," *Proceedings of 1996 ACM-SIGMOD International Conference on Management of Data*, 1996, pp. 103-114.

[8] S. Guha, R. Rastogi, and K. Shim, "CURE: An Efficient Clustering Algorithm for Large Databases," *In Proceedings of 1998 ACM-SIGMOD International Conference on Management of Data*, 1998, pp. 73-84.

[9] S. Guha, R. Rastogi, and K. Shim, "ROCK: A Robust Clustering Algorithm for Categorical Attribute," *In Proceedings of 1999 International Conference on Data Engineering*, 1999, pp. 512-521.

[10] G Karypis, E.H. Han, and V. Kumar, "CHAMELEON: Hierarchical Clustering Using Dynamic Modeling," *IEEE Computer*, Vol. 32, No. 8, 1999, pp. 68-75.

[11] J. Han, and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2000.

[12] A. Dragut, and C.M. Nichitiu, "A Monotonic On-Line Linear Algorithm for Hierarchical Agglomerative Classification," *Information Technology and Management*, Vol. 5, No. 1-2, 2004, pp.111–141.

[13] U. Maulik, and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices" *Pattern Analysis and Machine Intelligence, IEEE Transactions*, Vol. 24, Issue 12, 2002, pp. 1650-1654.

[14] J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," *In Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1, 1967, pp. 281-297.

[15] L. Kaufman, and P.J. Rousseeuw, *Finding Groups in Data: an Introduction to Cluster Analysis*, John Wiley & Sons, 1990.

[16] R. Ng and J. Han, "Efficient and Effective Clustering Method for Spatial Data Mining," *In Proceedings of International Conference on Very Large Databases*, 1994, pp. 144-155.

[17] B. Chen, P.C. Tai, R. Harrison, Yi. Pan, "Novel Hybrid Hierarchical-K-means Clustering Method (H-K-means) for Microarray Analysis" *In Proceedings of 2005 Computational Systems Bioinformatics Conference, Workshops and Poster Abstracts. IEEE*, 2005, pp. 105-108.

[18] A. Boecker, S. Derksen, E. Schmidt, A. Teckentrup, G. Schneider, "A Hierarchical Clustering Approach for Large Compound Libraries" *J. Chem. Inf. Mod.*, Vol. 45, No. 4, 2005, pp. 807-815.

[19] D.J. Newman, S. Hettich, C.L. Blake, and C.J. Merz, UCI Repository of Machine Learning Databases, http://www.ics.uci.edu/~mlearn/MLRepository.html, 1998.