以 Python 進行 K-means、階層式分群與 DBSCAN 進行文件分群預測及分群結果品質

B10423004 蔡宗穎，國立雲林科技大學

B10423013 曾鈺雯，國立雲林科技大學

B10423044 王彥淇，國立雲林科技大學

摘要

本團隊使用 K-means、階層式分群與 DBSCAN，針對文件進行分群，並比較分群所花費的時間，使用 SSE(Sum of Squared Error)、Purity 比較分群結果品質，及劃出階層式分群的階層數，最後評估結果為花費時間排序：DBSCAN > K-means > Hierarchical，Purity 排序：Hierarchical > K-means > DBSCAN。

演算法：K-means、階層式分群(hierarchical clustering)、BSCAN(Density-based spatial clustering of applications with noise)

一、緒論

1.1 動機

本實驗欲研究新聞文章分群，此數據集(mini_newsgroups)包含了來自 20 個新聞組的 20000 條訊息。我們不確定每種新聞文章的結構，希望可以透過分群法來區分出這 20 種新聞文章。

1.2 目的

本實驗目標希望根據 K-means、階層式分群、DBSCAN 三種分群法，來比較三種分群法的時間效率及分群結果品質，藉此了解文章與文章之間的關聯性，達成自動分群新聞文章。

二、方法

2.1 程式架構

1. 將檔案解壓縮，並轉碼，方便讀取資料集
2. 對文字出現頻率進行預先處理
3. 紀錄開始時間
4. 使用 K-means 套件建立模型
5. 用現在時間減去開始時間
6. 驗證輪廓係數、SSE、purity
7. 使用 DBSCAN 套件建立模型
8. 用現在時間減去開始時間
9. 驗證 purity
10. 使用階層式分群套件建立模型
11. 用現在時間減去開始時間
12. 驗證 purity

2.2 執行程式的方式

透過 Spyder 執行 main.py 檔，使用 python3.6 版本，注意 matplotlib(畫樹狀圖套件)須為 2.2.3 版，其他需安裝的套件為 sklearn(演算法套件)、nltk(語言分析套件)、numpy(科學計算套件)。

三、實驗

3.1 資料集

使用 mini_newsgroups 資料集是用於文本分群、文本探勘和信息檢索研究的國際標準數據集之一，此資料集被分為 6 個類別，分別為 ：

- 表 1 資料集

名稱	筆數	型態
alt.atheism	100	字串
comp.graphics	100	字串
misc.forsale	100	字串
rec.autos	100	字串
sci.crypt	100	字串
talk.politics.guns	100	字串

comp.* – 關於電腦的相關話題

sci.* – 關於科學方面的討論

rec.* – 關於娛樂活動的討論（遊戲、愛好等等）

talk.* – 關於社會及宗教熱點話題的討論

misc.* - 其他無法歸入現有層級的討論

alt.* – 無法歸於其他類別或不願歸入其他類別的話題

3.2 前置處理

利用亂數來打亂原本資料集的順序，藉此來達到每次執行所抓取的資料集不一樣。

TFIDF ： 用來衡量文字字詞出現的頻率，並降低停用詞的權重

$$w_t = tf_t \times \boldsymbol{idf_t} = tf_t \times \log \frac{\boldsymbol{N}}{\boldsymbol{df_t}}$$

- 表 2 輸入資料

名稱	型態	次數
test	int32	45731
zoo	int32	50608
ask	int32	13047
seem	int32	42228
respect	int32	40574

3.3 實驗設計

3.3.1 K-means ： 是訊號處理中的一種向量量化方法，現在則更多地作為一種聚類分析方法流行於資料探勘領域。功用為把個點劃分到 k 個聚類中，使得每個點都屬於離他最近的均值對應的聚類，以之作為聚類的標準。

- 表 3 K-mean 參數定義

名稱	定義
n_clusters	簇的個數，即你想聚成幾類
init	初始簇中心的獲取方法
n_init	獲取初始簇中心的更迭次數，為了彌補初始質心的影響，算法默認會初始 10 次質心，實現算法，然後返回最好的結果
max_iter	最大迭代次數
tol	容忍度，即 kmeans 運行準則收斂的條件
precompute_distances	是否需要提前計算距離，這個參數會在空間和時間之間做權衡
verbose	冗長模式
random_state	隨機生成簇中心的狀態條件
copy_x	對是否修改數據的一個標記，如果 True，即復制了就不會修改數據。bool 在 scikit-learn 很多接口中都會有這個參數的，就是是否對輸入數據繼續 copy 操作，以便不修改用戶的輸入數據
n_jobs	並行設置
algorithm	kmeans 的實現算法，有：'auto', 'full', 'elkan', 其中'full'表示用 EM 方式實現

3.3.2 階層式分群(hierarchical clustering)： 透過一種階層架構的方式，將資料層層反覆地進行分裂或聚合，以產生最後的樹狀結構，常見的方式有兩種：

- 如果採用聚合的方式，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
- 如果採用分裂的方式，則由樹狀結構的頂端開始，將群聚逐次分裂。是一種構建分類器的簡單方法。

- 表 4 階層式分群參數定義

名稱	定義
linkage	如何衡量群與群之間的距離
n_clusters	分成幾個群
ward(single)	兩個群中最近的點
complete	兩個群中最遠的點
average	兩個群的重心
optimal_ordering	如果為 True，則將重新排序鏈接矩陣，使得連續葉之間的距離最小。當數據可視化時，這會產生更直觀的樹結構。默認為 False，因為此算法可能很慢，尤其是在大型數據集上
metric	在 y 是觀察向量的集合的情況下使用的距離度量

3.3.3 DBSCAN(Density-Based Spatial Clustering of Applications with Noise)： 是一個比較有代表性的基於密度的聚類算法。與劃分和層次聚類方法不同，它將簇定義為密度相連的點的最大集合，能夠把具有足夠高密度的區域劃分為簇，並可在噪聲的空間數據庫中發現任意形狀的聚類。

- 表 5 階層式分群參數定義

名稱	定義
eps	兩個樣本之間的最大距離，以便將它們視為在同一鄰域中。
min_samples	對於要被視為核心點的點，鄰域中的樣本數（或總權重）。
metric	計算要素數組中實例之間距離時使用的度量標準
metric_params	度量函數的其他關鍵字參數
algorithm	近鄰算法求解方式，有四種 brute 為蠻力實現、kd_tree 為 KD 樹實現、ball_tree 為球樹實現、auto 為上面三種算法中做權衡，選擇一個擬合最好的最優算法。
leaf_size	使用 ball_tree 或 kd_tree 時,停止建子樹的葉子節點數量的值
p	只用於閔可夫斯基距離和帶權重柴可夫斯基距離中 p 值的選擇，p=1 為曼哈頓距離， p=2 為歐式距離。如果使用默認的歐式距離不需要管這個參數
n_jobs	CPU 並行數，若值為 -1，則用所有的 CPU 進行運算
core_sample_indices_	核心點的索引，因為 labels_不能區分核心點還是邊界點，所以需要用這個索引確定核心點
components_	訓練樣本的核心點
labels_	每個點所屬集羣的標籤，-1 代表噪聲點

3.3.4 SSE (Sum of Squared Error) ： 是觀察每個值與其組平均值之間的平方差異的總和。它可以用作群集內變異的度量。如果群集中的所有情況都相同，則 SSE 將等於 0。

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist(m_i, x)^2$$

3.3.5 Purity(純質) ： 同質性越高，purity 越大。

purity Using the terminology derived for entropy, the purity of cluster j, is given by $purity_j = \max p_{ij}$ and the overall purity of a clustering by $purity = \sum_{i=1}^{K} \frac{m_i}{m} purity_j$.

3.4 實驗結果

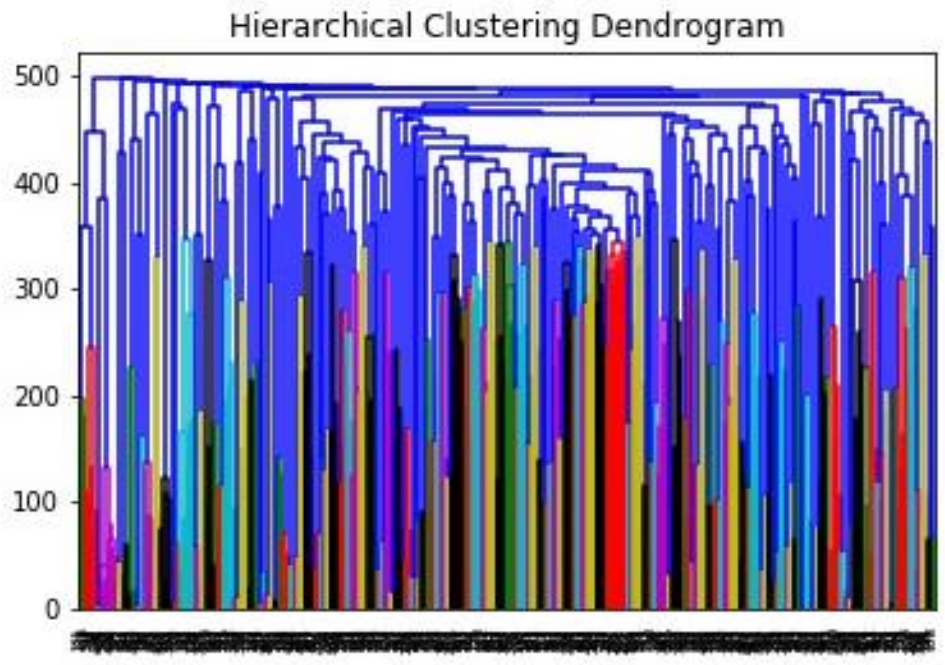
3.4.1 K-means

- 表 6 使用 K-means，計算花費時間、SSE、purity 與側影係數

```
K-means
花費時間: 405.65873098373413 秒
Silhouette Coefficient: 0.012510772718172717
SSE: 1848.969212646806
purity: 0.22650000000000006
```

3.4.2 階層式分群(hierarchical clustering)

- 表 7 使用階層式分群，計算花費時間與 purity




```
hierarchical
花費時間: 12.49722146987915 秒
purity: 0.4390000000000001
```

3.4.3 DBSCAN(Density-Based Spatial Clustering of Applications with Noise)

- 表 8 使用 DBSCAN，計算花費時間與 purity，半徑為 0.911，鄰域中的樣本數為 3，可以把資料分成 20 群

```
DBSCAN
花費時間: 652.0506982803345 秒
purity: 0.016500000000000004
```

```
dbscan = DBSCAN(eps = 0.911, min_samples = 3).fit(trans_dataset)
```

四、結論

1. 花費時間排序：DBSCAN > K-means > Hierarchical
2. Purity 排序：Hierarchical > K-means > DBSCAN
3. K-means：花費時間約為：406 秒、purity 約為：0.2265
4. Hierarchical ： 花費時間約為：13 秒、purity 約為：0.439
5. DBSCAN ： 花費時間約為：652 秒、purity 約為：0.0165