Spectral Clustering on Star War

- 一、建置 Laplacian Matrix (L): 根據資料之 Adjacency matrix,可以計算每列(或每行)的總和得到
 - diagonal matrix (D),利用 L=D-A 之關係式即可得到 L (69*69)。
- 二、尋找 L 的特徵向量與特徵值: 使用 Numpy 套件的 eig (L) 功能即可對 L 取出特徵向量及特徵值(需注意 eig (L) 會取出 L 被 normalized 後的特徵向量)。
- 三、依照前三小的特徵值取出其對應的特徵向量,並以之為 69 個角色的分群 representation vectors (X_hat): 在此處是採有取 eigenvalue = 0 的 constant eigenvector, 故所獲得的 X hat 是 69*3 的矩陣。

四、撰寫 K-means 邏輯:

- (1) 先對 69 個角色指定一個隨機的分群結果 $(0 \cdot 1 \cdot 2)$, 並且隨機指定 3 個中心點位置存入 reps (3*1 vector)
- (2) 設定一個目標值(J): 其意義為「依據每一個點的指定分群計算該點的 representation vector 到該分群中心點的距離」,此處以 2-norm 作為計算 距離的指標,使用 Numpy 的 LA.norm() 計算之。將 J 的初始值設為無限大,目的是要在多次迭代之後找到一個穩定存在的最小 J 值,亦即:我們的目標是盡可能最小化 J。
- (3) 迭代過程中,盡可能將點分入使J值更小的分群,得到新的分組和中心點。
- (4) 當 J 值達到穩定(設定 tolerance 值 = 10⁻⁶) 即:當 J 和前次 Jprevious 的差值 < tolerance,則停止迭代,並回傳結果:69 個角色的最終分群 (ASS)、3 個中心點 (reps)。
- - (1) 共 6 次迭代結果 (Jprevious):

```
Iteration = 0 : Jclust = 0.026468466849872585
Iteration = 1 : Jclust = 0.016152278494252867
Iteration = 2 : Jclust = 0.013941637750953022
Iteration = 3 : Jclust = 0.013114590346018902
Iteration = 4 : Jclust = 0.01293563206740359
Iteration = 5 : Jclust = 0.01293563206740359
```

(2) 分群結果:

可以發現點被分為兩個比較主要的群體(1,2);被分入0的則較少,僅4個 nodes。

六、結果呈現與觀察:

為了確認本次的分群結果合理性,我們可以將所有 representation vectors (X_{hat}) 標出,並且檢視 K_{means} 給予的中心點是否確實分布在分群當中。結果如下圖:

- (1) 所有 nodes 依據其分組的分別標為藍、綠、黃三色
- (2) 紅點則為最終的3個中心點)

