Machine Learning HW4 Report

電機四 B02901073 楊靖平

1. Analyze the most common words in the clusters. Use TF-IDF to remove irrelevant words such as “the”. (1%)

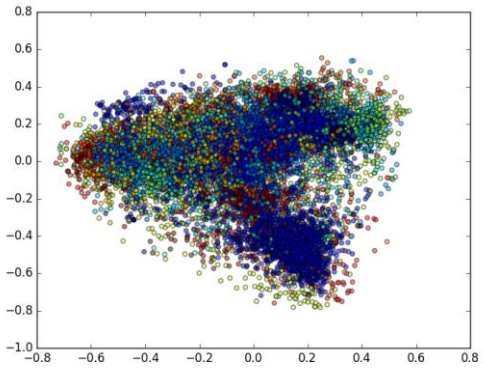
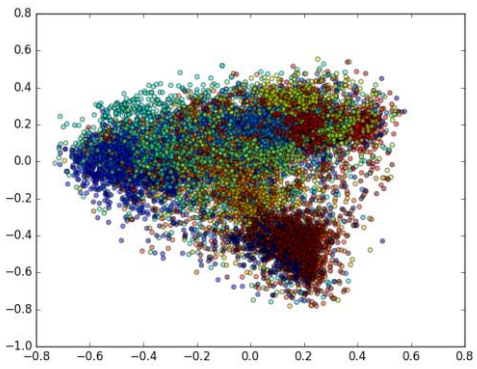
Ans.

我使用 sklearn 中的 TF-IDF 作為 feature extraction，以下是 20 個 cluster 中最常出現的字

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| view | script | window | vba | java | jquery | list | bean | rewrite | 2008 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| subver | cocoa | function | Os_x | product | function | map | post | sql | queri |

2. Visualize the data by projecting onto 2-D space. Plot the results and color the data points using your cluster predictions. Comment on your plot. Now plot the results and color the data points using the true labels. Comment on this plot. (1%)

Ans.

| **Mypredict** | **Using label data** |
|---|---|
|  |  |

My predict 是根據我在 kaggle 上最好的參數所取下來的圖(about 0.85)，其中我將所有資料分為 85 個 clusters，可以看到其中有幾群是比較大群的，可能因為取參數的關係所以顯的有些雜亂，其中又有一群(藍色)是特別大群的

Label data 使用相同降維方法畫出來的圖，可以明顯看到每個 cluster 之間的界線，我覺得我自己降維的時候調整的不太好，不然結果應該會更接近 Label data 的圖

3. Compare different feature extraction methods. (2%)

Ans.
這次作業一開始我都是使用 SVD 來降維，之後分群的時候主要使用兩種方法
Method 1. KMeans
隨機取想要的 cluster 數的點，再將每個點附近的點歸類到自己的 cluster，歸類玩會重新取每個 cluster 的中心點，重複下去直到收斂
Method 2. Consine Similarity
直接比較兩個點和原點連成的 vector 的夾角，取相似度大於某個值，就識為同一群

一開始任意調參數下，結果是 KMeans 的 performance 較好，猜測是因為 KMeans 分群會比直接比較兩 vector 夾角來的好，因為分群自動就會包含 vector 夾角相近的點了

4. Try different cluster numbers and compare them. You can compare the scores and also visualize the data. (1%)

Ans.
這裡我主要比較 Keans 的 cluster 數，如下表：

| 50 維 50 群 | 50 維 35 群 | 50 維 20 群 | 20 維 100 群 | 20 維 85 群 | 20 維 70 群 | 25 維 85 群 |
|---|---|---|---|---|---|---|
| 0.62488 | 0.57686 | 0.38515 | 0.84794 | 0.85005 | 0.84208 | 0.79933 |

變數有：降維數、cluster 數
觀察結果：降得維數越多，效果越好。cluster 雖然題目是給 20 群，但是觀察結果顯示，分越多群結果越好，猜測是因為分太少群容易有一大 cluster 吃掉所有的點，讓每筆 data 看起來都會出自於同一個 cluster。