Machine Learning Homework#4

--Unsupervised Clustering & Dimensionality Reduction

b02901120 羅志軒

Analyze the most common words in the clusters

My cluster:

wordpress, oracl, use, apach, excel, matlab, file, magento, hibern, drupal, linq, scala, spring, sharepoint, visual, haskel, ajax, studio, bash, qt

True tages:

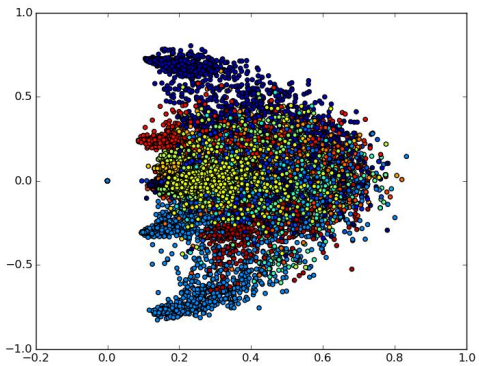
wordpress, oracle, svn, apache, excel, matlab, visual-studio, cocoa, osx, bash, spring, hibernate, scala, sharepoint, ajax, qt, drupal, linq, haskell, magento

討論：

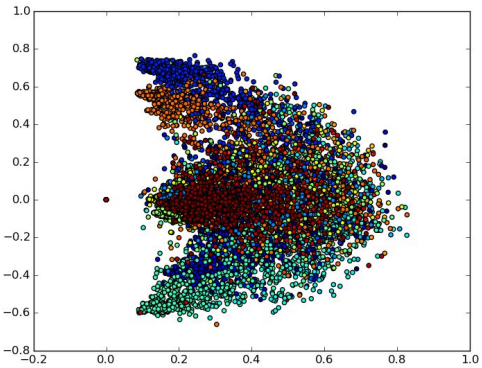
因為Tf-idf基本上還是利用特定字出現的頻率來判斷字的重要性，所以像use, file...類似於stopwords但並沒有被列入的字還是會被保留下來，若我們提高對字出現頻率過多的懲罰，或許可以解決此類問題

Visualize the data by projecting onto 2-D space

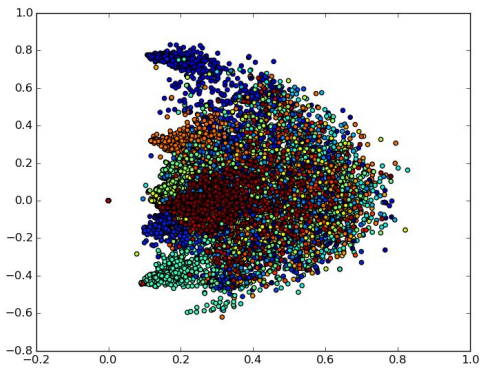
My cluster (20 clusters)



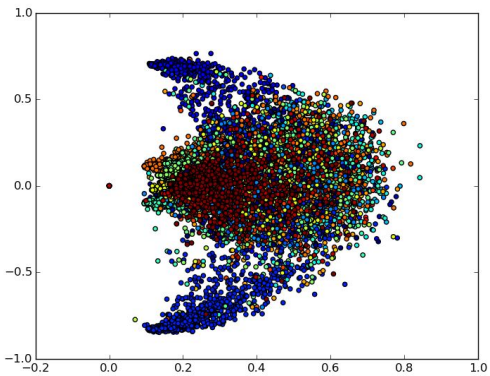
True cluster



My cluster (60 clusters)



My cluster(100 clusters)



討論：

這裡我取TruncatedSVD後20維component中的第1維和第20維當作投影後的2D平面數值，可以發現基本上分成20個cluster的結果已經相當不錯，增加cluster數可以一定程度上將資料分的更開，但相對於正確資料界線較為明確的分群還有一些差距。

Compare different methods

Nomalization of Tf-df：

normalizaion可以有效幫助kmeans於相同的基準下對Tf-idf後的字分群，正確率從0.30027上升至0.46831

Data preprocessing：

在本次作業中我使用三種preprocess方式，**1. 消除標點符號及將字全部轉換為小寫**

2. 刪除stopwords：利用nltk的'stopwords'字庫去除掉常見但不重要的字

3. Stem：英文中包含許多詞性及分詞形式，基本上表達的意義是相近的，為了避免Tf-idf重複將這些意義相似的字列為最重要的幾個字，所以可以使用stem的方式簡化字詞，使重點明確被表達出來，正確率大約可以上升0.02～0.03左右。

Tf-idf vs Bag-of-Word：

cluster number	Tf-idf	Bag-of-Word
20	0.4695	0.5155
60	0.7651	0.7498
100	0.7958	0.8047
200	0.6795	0.7714

LSA vs PCA：

cluster number	LSA	PCA
100	0.7958	0.7498
200	0.6795	0.7817
500	0.6970	0.7585

討論：

在所有方法中以調整cluster數對正確率的影響最大，原因和使用F-measure計算正確率有關，若cluster數上升，正確分辨資料在不同群的機率上升，即FP上升，正確率隨之上升，但若cluster數上升太多，正確分辨資料在同群的機率下降，即FN上升，正確率隨之下降