# [Machine Learning HW4]　R04921006 謝仲凱

## Unsupervised Clustering & Dimensionality Reduction

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

BoW

```
top 20 words

in -> 5941
using -> 1168
the -> 2833
excel -> 888
how -> 4017
is -> 1645
on -> 1566
from -> 1420
wordpress -> 882
for -> 1829
hibernate -> 934
of -> 2084
with -> 2033
to -> 6567
and -> 1854
can -> 1060
an -> 1151
what -> 894
do -> 1294
magento -> 880
```

BoW + TF-IDF

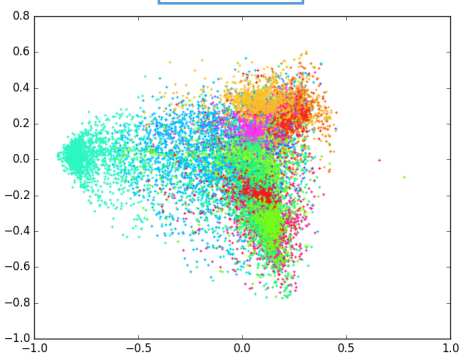
```
top 20 words

using -> 279.741587301
magento -> 254.186147411
wordpress -> 255.18272803
linq -> 262.049602333
hibernate -> 265.734059892
matlab -> 243.765496409
ajax -> 226.639359675
scala -> 229.961382355
drupal -> 239.782049193
sharepoint -> 214.820578355
haskell -> 223.452223611
excel -> 238.908117026
oracle -> 215.989065705
file -> 215.179098408
spring -> 233.830797378
bash -> 203.603414133
studio -> 188.597462383
svn -> 191.561347513
qt -> 185.678785233
visual -> 197.237251422
```

可以看到如果沒有使用TF-IDF的BoW 會出現 in, the is,for 等對於分辨是不是相同群的topic沒啥意義的字，因此在做cluster上會造成結果不是很理想。

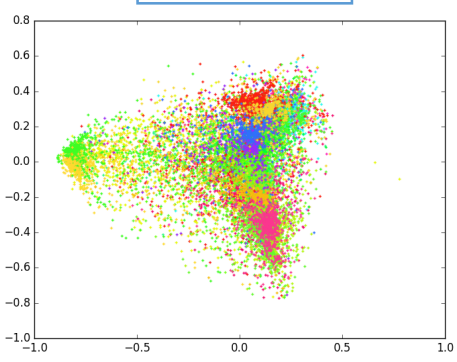
再看看有加入TF-IDF的字中就沒有這些無意義的字，還可以看到qt,svn,visual,studio等等對於分辨topic群組很有意義的字。

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%)

Predict




True Label



Predict:

這張看得出來他其實每群（一種顏色代表一群）都滿相近的，看圖可以看得出來他們是同一群。其實到現在也還滿覺得不可思議的 topic 最後轉成cluster到現在畫出來可以看出每群的分佈其實是有差異，同群是有相似性的。

True Label:

大致上同群還是有一定的相似性，但是感覺好像有點noise沒有分布得很好，我個人覺得是因為我做出來正確率大概8成 所以有些跟True Label不像造成noise是很正常的。

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

在以下條件相同的情況下 比較feature extraction的差異

Stop_words=English, lowercase=True, N_cluster=20 normalized = True

[BoW](10876 features) :0.16165

[BoW] + SVD(20 features) : 0.55845

[BoW+tf-idf] (10876 features):0.24306

[BoW + tf-idf] + SVD(20 features):0.62960

這邊有兩個心得：

第一個 從10876的feature 使用SVD降成20個feature:

可以明顯看出來效果變得非常好一個是從 0.16165->0.55845 另一個是從0.24306->0.62960，我認為是它降維的時候把他投影到比較好切的的dim上，因此正確率才能顯著的提升。

第二個 有無TF-IDF:

我們可以看到第一個是從0.16165->0.24306 ，另一個是從0.55845->0.6296都有明顯的提升，我認為tf-idf可以解決一個字如果在文件中出現越多次代表這個字越重要，但是他如果在語料庫中出現的頻率非常多反而不重要ex:the。

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

[BoW + tf-idf] + SVD(20 features)

n_classes = 20 : 0.62960

n_classes = 40: 0.81719

n_classes = 60: 0.85446

這一題讓我想了很久，因為我們的topic明明只有20個分類，為什麼在K-means 中 classes 設成40 , 60 performance卻直直上升呢？我覺得是因為如果分得越多群20->40 每個cluster 其實越集中，那相信此topic是此cluster 的信心度就會越提升，performance就會跟著提升，但是自己有試過設到80 performance就開始掉了，因此他還是有一個fine-tune 可以到最好的值，我的case是 60類