National Taiwan University

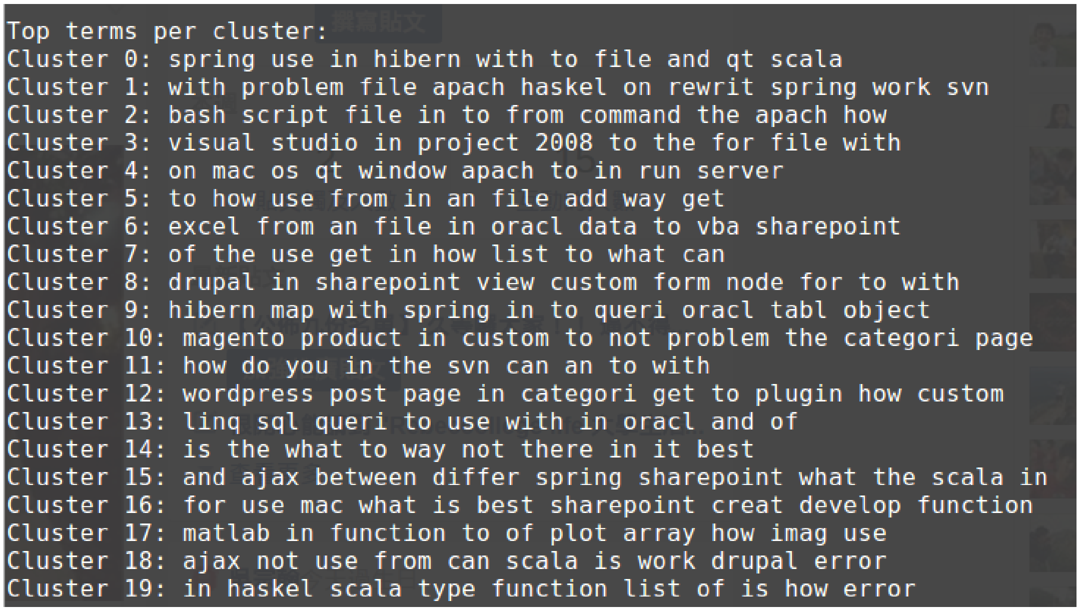
Machine Learning: HW4

EE3 b03901016
陳昊

December 8, 2016

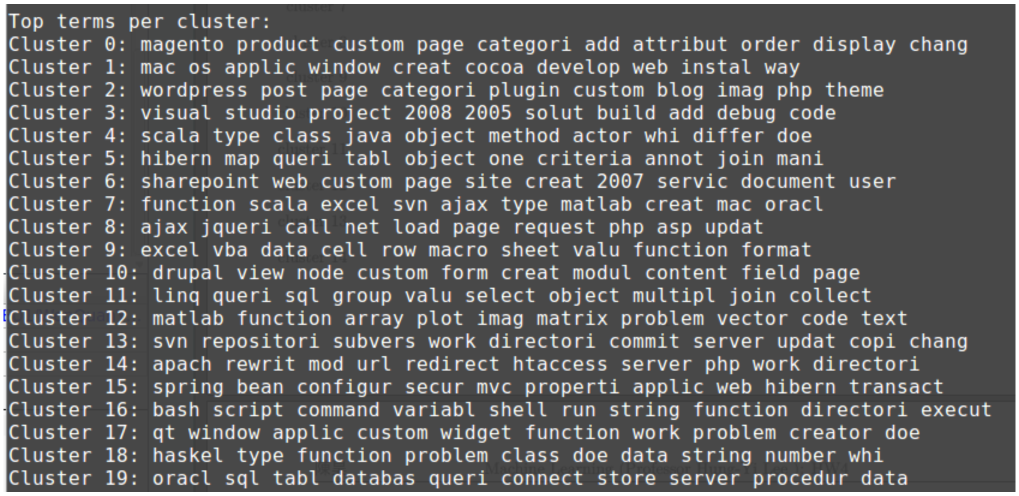
1 Analyze the most common words in the clusters.

After loading the sentences from the document, the first thing I do is preprocessing. The first step is lower all the words and remove all the punctuations; Second, stem the words in each sentences using nltk snowball stemmer, then remove stop words in nltk's english stop words. Furthermore, I add some stop words manually to filter some not useful words after stemming like ”use, get, file, line...”. Some most common words in the clusters without removing the stop words are:



```
Top terms per cluster:
Cluster 0: spring use in hibern with to file and qt scala
Cluster 1: with problem file apach haskel on rewrit spring work svn
Cluster 2: bash script file in to from command the apach how
Cluster 3: visual studio in project 2008 to the for file with
Cluster 4: on mac os qt window apach to in run server
Cluster 5: to how use from in an file add way get
Cluster 6: excel from an file in oracl data to vba sharepoint
Cluster 7: of the use get in how list to what can
Cluster 8: drupal in sharepoint view custom form node for to with
Cluster 9: hibern map with spring in to queri oracl tabl object
Cluster 10: magento product in custom to not problem the categori page
Cluster 11: how do you in the svn can an to with
Cluster 12: wordpress post page in categori get to plugin how custom
Cluster 13: linq sql queri to use with in oracl and of
Cluster 14: is the what to way not there in it best
Cluster 15: and ajax between differ spring sharepoint what the scala in
Cluster 16: for use mac what is best sharepoint creat develop function
Cluster 17: matlab in function to of plot array how imag use
Cluster 18: ajax not use from can scala is work drupal error
Cluster 19: in haskel scala type function list of is how error
```

If remove the stop words in the clusters, the top terms becomes:

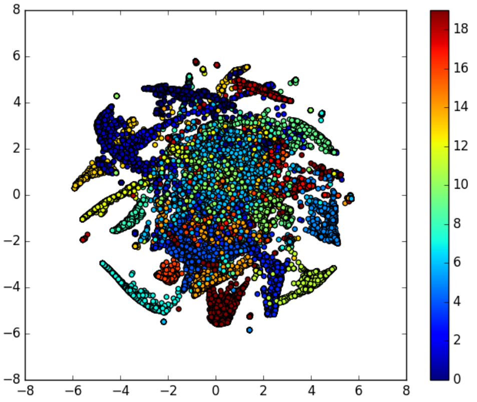


```
Top terms per cluster:
Cluster 0: magento product custom page categori add attribut order display chang
Cluster 1: mac os applic window creat cocoa develop web instal way
Cluster 2: wordpress post page categori plugin custom blog imag php theme
Cluster 3: visual studio project 2008 2005 solut build add debug code
Cluster 4: scala type class java object method actor whi differ doe
Cluster 5: hibern map queri tabl object one criteria annot join mani
Cluster 6: sharepoint web custom page site creat 2007 servic document user
Cluster 7: function scala excel svn ajax type matlab creat mac oracl
Cluster 8: ajax jqueri call net load page request php asp updat
Cluster 9: excel vba data cell row macro sheet valu function format
Cluster 10: drupal view node custom form creat modul content field page
Cluster 11: linq queri sql group valu select object multipl join collect
Cluster 12: matlab function array plot imag matrix problem vector code text
Cluster 13: svn repositori subvers work directori commit server updat copi chang
Cluster 14: apach rewrit mod url redirect htaccess server php work directori
Cluster 15: spring bean configur secur mvc properti applic web hibern transact
Cluster 16: bash script command variabl shell run string function directori execut
Cluster 17: qt window applic custom widget function work problem creator doe
Cluster 18: haskel type function problem class doe data string number whi
Cluster 19: oracl sql tabl databas queri connect store server procedur data
```

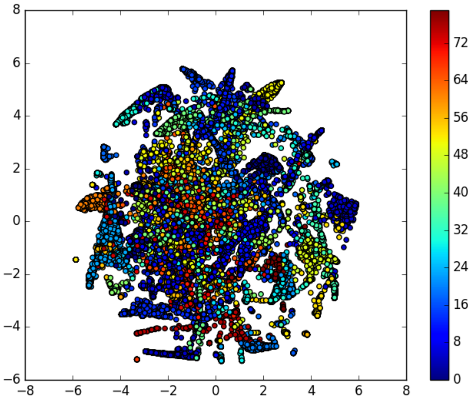
(Note: These words are already stemmed.)

2 Visualize the data by projecting onto 2-D space.

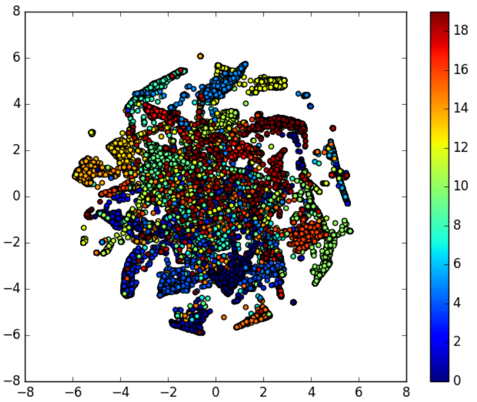
If directly use the result classified by K-means and project it to 2-D space using t-SNE, the plot looks like (a), (b), they have different number of cluster. While (c) is the plot of using true labels with t-SNE.



(a)



(b)



(c)

Since (a) is the result of using 20 clusters by k-Means, (b) used 80 clusters, it's easy to find that (b) looks more complicated. And for all plots, the middle part are too complicated that it's hard to classify each cluster.

3 Compare different feature extraction methods.

There are some feature extraction methods I've tried to implement, like

1. TF-IDF + LSA
2. TF-IDF + PCA
3. BoW + LSA
4. BoW + PCA

The performance of using TF-IDF + LSA to extract feature is better than others, while TF-IDF + PCA has the second high performance. I thought the reason may be that using TF-IDF has great improvement than using BoW since the words which occur too frequently would have lower weight, while BoW just directly count how many times the words occur.

Second, we compare the different about PCA and LSA. PCA and LSA are both analyses which use SVD. PCA is a general class of analysis and could in principle be applied to enumerated text corpora in a variety of ways. In contrast LSA is a very clearly specified means of analyzing and reducing text. Both of them are leveraging the idea that meaning can be extracted from context. In LSA the context is provided in the numbers through a term-document matrix. In the PCA your proposed context is provided in the numbers through providing a term covariance matrix. In my implementation, I found that LSA works a little better than PCA.

4 Try different cluster numbers and compare them.

First, I try 20 clusters to do K-means clustering, and I got a score of about 0.4, and then I try to rise the number of clusters to 30, then the score arise to 0.68. I got the best score by using 85 clusters, but I found that the score would be worse when the number of clusters excess a threshold value, because the score I got using 100 clusters is worse than using 85.

Cluster number	20	30	60	85	100	150
Public score	0.43838	0.68670	0.80284	0.84531	0.83395	0.75833
Private score	0.43898	0.68519	0.80114	0.84552	0.83329	0.75620

Beside using K-means cluster, there's a more efficient way to determine if two sentences are in the same tags. I got a better result directly using cosine similarity to evaluate it. If the value is greater than 0.9 after calculating cosine similarity, I consider they are in the same tags. After using this method, I got my best score of 0.91656.