HW4 Report

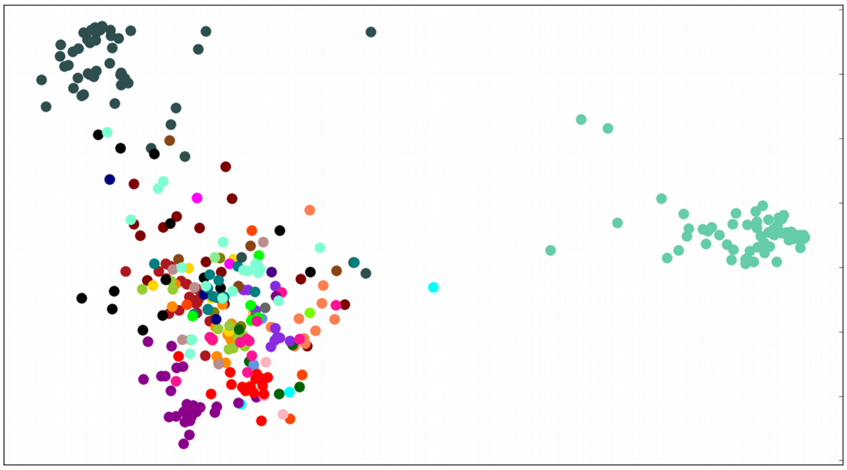
b03901070 羅啟心

1.
Analyze the most common words in the clusters. Use TF-IDF to remove irrelevant words such as “the”. (1%)
The top 10 most common words in each clusters are :
(after removing stop words)
Cluster 0:
spring mvc security bean using framework use configuration web application
Cluster 1:
magento product custom products page add category order checkout module
Cluster 2:
scala qt apache svn spring oracle linq ajax haskell hibernate
Cluster 3:
qt mac os x file sharepoint spring ajax application oracle
Cluster 4:
svn apache bash files excel visual repository sharepoint script studio
Cluster 5:
apache ajax svn wordpress magento spring hibernate matlab haskell oracle
Cluster 6:
wordpress page post posts plugin category custom blog theme add
Cluster 7:
haskell type function list 's error does problem types data
Cluster 8:
bash script command files shell variable string line output function
Cluster 9:
drupal node custom module views content form page view menu
Cluster 10:
excel vba data cell macro sheet files range formula function
Cluster 11:
oracle sql query table database use 's stored server 10g
Cluster 12:
scala java type 's does use list function class actors
Cluster 13:
linq sql query multiple group use list xml join data
Cluster 14:
hibernate mapping query criteria table using object problem join use
Cluster 15:
ajax jquery page asp.net request php using javascript problem post
Cluster 16:
matlab function matrix array plot image using 's data text
Cluster 17:
apache rewrite mod_rewrite server redirect error use url php files
Cluster 18:
sharepoint list web custom site page services document create workflow
Cluster 19:
mac os x application development osx cocoa 's terminal windows
Cluster 20:
qt application window windows widget does custom creator use c++
Cluster 21:
file svn subversion repository cocoa apache using best way directory

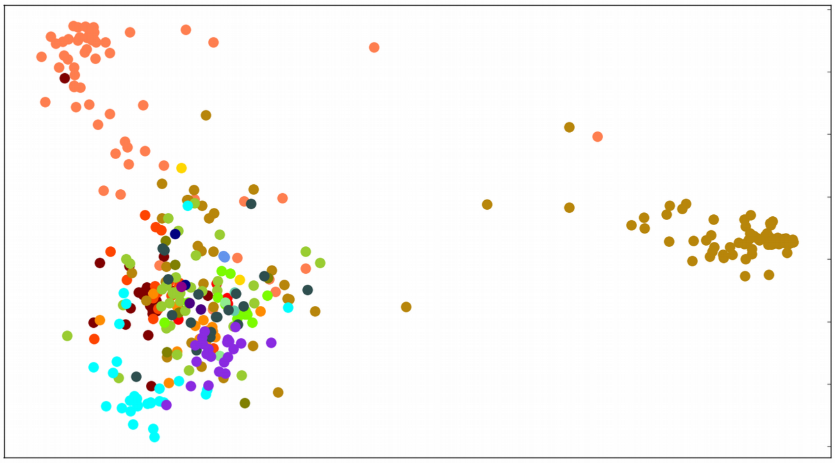
Cluster 22:
visual studio qt svn scala matlab mac excel ajax apache
Cluster 23:
using cocoa list subversion files image mod_rewrite svn scala apache
Cluster 24:
svn files repository subversion copy server directory update revision working
Cluster 25:
visual studio project files projects solution 's code does build
Cluster 26:
drupal sharepoint magento wordpress custom qt page ajax excel list
Cluster 27:
haskell scala matlab bash excel oracle linq function type hibernate
Cluster 28:
file subversion cocoa text line best way xml data reading
Cluster 29:
hibernate spring scala using use ajax query mapping linq does
Cluster 30:
linq using sql query list multiple group xml join cocoa
Cluster 31:
file oracle sharepoint scala spring qt mac excel drupal svn
Cluster 32:
bash file script matlab subversion using command mac line cocoa
Cluster 33:
linq oracle hibernate sql excel query using drupal sharepoint scala
Cluster 34:
file using mac subversion qt os cocoa x scala application
Cluster 35:
excel file matlab oracle linq data qt haskell ajax vba

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%)

cluster prediction by cluster.py



true label



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

method	procedure	F-measure score
method1	1.remove stopwords, stemming, and tokenizing 2.tfidf : fit_transform(titles) 3.k means	0.27558
method2	1.remove stopwords, stemming, and tokenizing 2.tfidf : fit(titles+docs) , transform(titles) 3.LSA 4.k means	0.78782
method3	1.remove stopwords, stemming, and tokenizing 2.tfidf : fit(titles+docs) , transform(titles) 3.PCA 4.k means	0.70987
method4	1.remove stopwords, stemming, and tokenizing 2.bag of words : fit(titles+docs),transform(titles) 3.PCA 4.k means	0.53619
method5	1.remove stopwords, stemming, and tokenizing 2.bag of words : fit(titles+docs) , transform(titles) 3.LSA 4.k means	0.65517

In general, LSA is performs better than PCA and tfidf performs better than bag of words.

4.

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

number of clusters in cluster.py	F-measure score
16	0.45789
21	0.59269
26	0.72786
31	0.76308
36	0.77954