EE219 Large Scale Data Mining

Winter 2018

Project 2 Clustering

Yucong Wang Shizhong Hao

305036163 605035020

1 Introduction

In this project, we work with “20 Newsgroups” dataset that we already explored in Project 1. It is a collection of approximately 20,000 documents, partitioned (nearly) evenly across 20 different newsgroups, each corresponding to a different topic. Each topic can be viewed as a “class”. In order to define the clustering task, we pretend as if the class labels are not available and aim to find groupings of the documents, where documents in each group are more similar to each other than to those in other groups. These groups, or clusters, capture the dependencies among the documents that are known through class labels. We then use class labels as the ground truth to evaluate the performance of the clustering task.

The goal includes:

-To find proper representations of the data, so that the clustering is efficient and gives out reasonable results.

-To perform K-means clustering on the dataset, and evaluate the performance of the clustering.

-To try different preprocess methods which may increase the performance of the clustering.

We are supposed to apply the clustering method on the dataset. And we tried the K-means clustering method with different dimension reduction, optimized the dimension reduction separately and compared the performance with each other. We started with 2-cluster case and then moved on to the multiple cluster case and gave a detailed analysis on each of the outcome results.

2 Question 1

For the purposes of this project we will be working with 8 of the classes as we have done before.

First we loaded the all data for the following 8 subclasses and divided to two major classes ‘Computer Technology’ and ‘Recreational activity’, and then we performed the similar pre-processing procedure like project 1, we replaced the \r \n \t as space for excluding them, removed punctuations and stop words, didn’t stemming the words as we were told, and computed the TF-IDF representation of the overall dataset using a count vectorizer with setting min df=3, and a TF-IDF transformer to fit and transform.

The results of our TF-IDF matrix was size 7882 X 27399 which contains the TF-IDF scores of 27399 terms over 7882 posts.

3 Question 2

In this part, we tried the K-means(2-cluster) clustering algorithm on the raw TF-IDF matrix. We inspected the contingency matrix to get a sense of our clustering result. And evaluated the clustering performance by inspecting these measures of purity for a given partition of the data points with respect to the ground truth, the homogeneity, completeness, V-measure, adjusted rand and adjusted mutual scores of the labels predicted by the clustering algorithm.

Homogeneity is a measure of how “pure” the clusters are. If each cluster contains only data points from a single class, the homogeneity is satisfied.

On the other hand, a clustering result satisfies completeness if all data points of a class are assigned to the same cluster. Both of these scores span between 0 and 1; where 1 stands for perfect clustering.

The V-measure is then defined to be the harmonic average of homogeneity score and completeness score.

The adjusted Rand Index is similar to accuracy measure, which computes similarity between the clustering labels and ground truth labels. This method counts all pairs of points that both fall either in the same cluster and the same class or in different clusters and different classes.

Finally, the adjusted mutual information score measures the mutual information between the cluster label distribution and the ground truth label distributions.

And here are our results.

The dataset shape was (7882, 27399)

Homogeneity: 0.25451173065620636

```
Completeness: 0.3355089128910059
V-measure: 0.28945073364586266
Adjusted Rand Score: 0.18184259008613624
Adjusted Mutual Info Score: 0.25444347300225234

Contigency matrix:
==============
[[3899       4]
  [2256 1723]]
==============
```

class labels	0 (Com)	1 (Rec)
0	3899	4
1	2256	1723

The confusion matrix and the scores were reported. We can see that our clustering didn't do very well as for label 1, since almost equal number of data points were grouped into two different classes i.e. 2256 documents were identified as one class and 1723 as other. Although for the other label, the result was good as 3899 documents were grouped in one cluster and only 4 in the other cluster.

As we know, a perfect clustering would result into a diagonal confusion matrix which denotes least confusion but the matrix obtained here is not diagonal and there is no permutation of rows which result in a diagonal matrix. We can also conclude from the 5 low scores we got that the clustering result obtained in this part did not perform well.

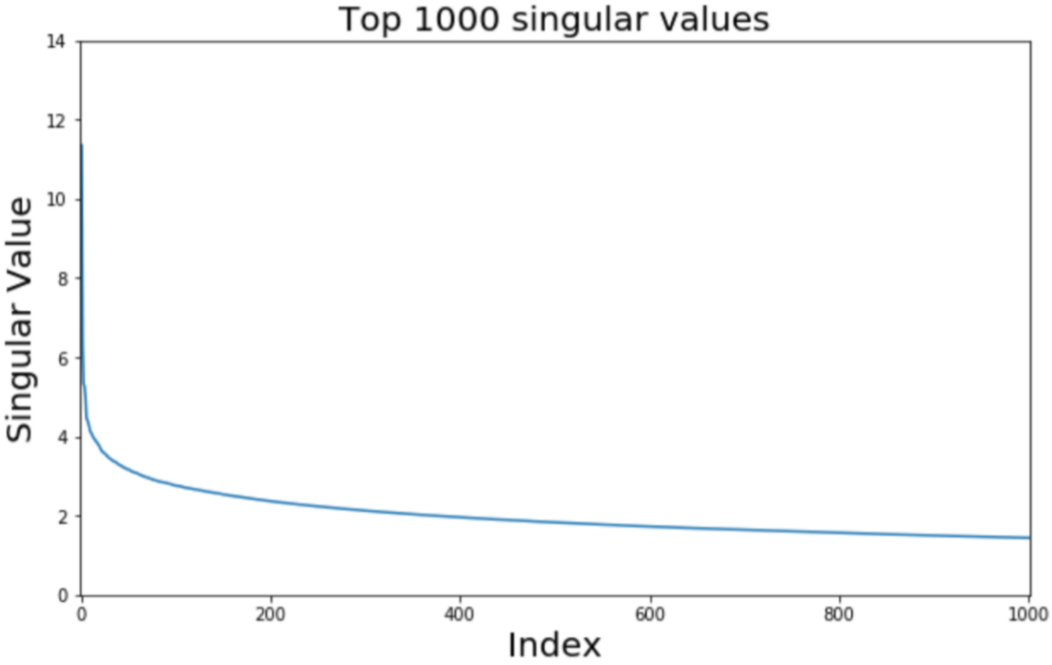
4 Question 3

The precision of the clustering labels in the last question is not as satisfactory as we expect it to be. This is possibly caused by high dimension TF-IDF matrix. Because we use the word count information to build the TF-IDF matrix, with high dimension TF-IDF, too many unimportant words would result in very serious overfitting, and data points is possibly to be distracted far away from where they should be.

To overcome this shortcoming and improve the performance of our algorithm, we will try to reduce the dimension of the TF-IDF matrix. By reducing the dimension, some unrelated information could be eliminated after the process (for example, by doing projection process). Only the most significant dimensions remain and ideally, we can improve the precision of the cluster results. In this project, we will be using Latent Semantic Indexing (LSI) and Non-Negative Matrix Factorization (NMF).

By applying LSI, this technique takes into consideration the fact that words used in the same context tend to have similar (or close) meanings. LSI can help us discover the inner relation between different words. It is based on the singular value decomposition of matrix. Let D be the TF-IDF matrix, then the SVD of the matrix could be expressed as , where U and V are both unitary matrices and Σ is the corresponding singular value matrix. By doing this, we are able to get all the singular values of the TF-IDF matrix and get a reasonable guess on the dimensionality to feed in the K-means algorithm.

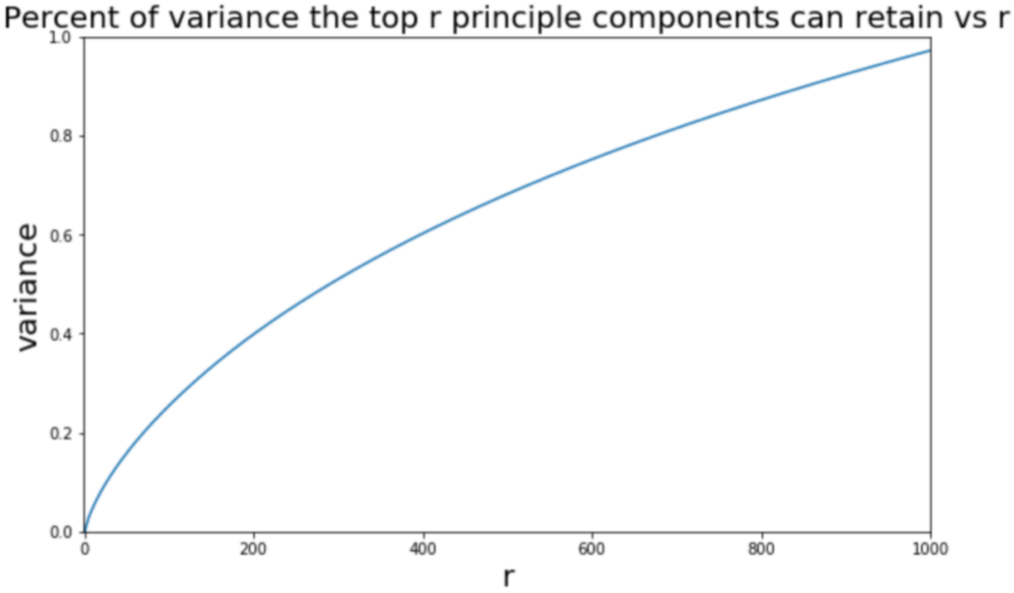
We calculated the top 1000 significant singular values and plotted them as below.



In the plot above, singular values have sorted in descending order. As we can see, the decreasing rate (i.e. the slope of the curve) of the singular values is very high in the beginning and it slows down very fast. The whole curve basically looks like the plot of inverse proportional function.

Actually we want to find the effective dimension of the data through inspection of the top singular values of the TF-IDF matrix and see how many of them are significant in reconstructing the matrix with the truncated SVD representation. A guideline is to see what ratio of the variance of the original data is retained after the dimensionality reduction.

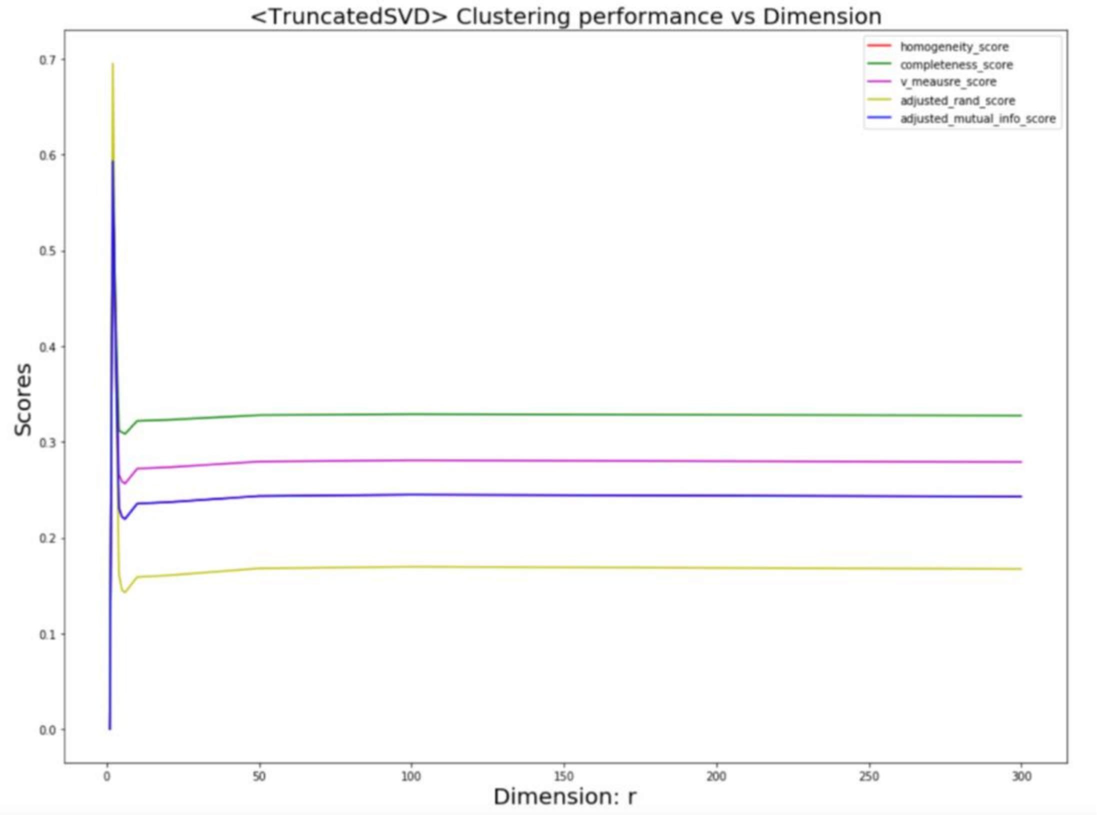
So we plotted of the percent of variance the top r principle components can retain v.s. r, for r = 1 to 1000, and the retained variance is the sum of the squares of top r singular values. Our result is shown below.



We can see that as the number of components increases, the percent of variance the top r principle components can retain also increase, although top 1000 principal components can't explain to 100% of the variance.

After that, we use the both LSI and NMF to reduce the dimensions and try to find the best dimension parameters for each method with the best results.

First, we show the results for LSI. We tried r value from set [1,2,3,4,5,6,10,20,50,100,300], and inspecting the 5 scores. Our results are shown below.



As we know, evaluating the performance of a clustering result is actually not an easy and simple task, which is exactly the reason we plot at all 5 measures, instead of just one. From the plot above, we can easily find that the 5 curves are quite similar and all of them reach their peaks at 2-dimension(r=2). And we reported the scores and contingency matrix below to verify our conclusion.

```
Number of components: 1
Dimensions of TF-IDF vector after LSI: (7882, 1)
Homogeneity: 0.000
Completeness: 0.000
V-measure: 0.000
Adjusted Rand-Index: 0.000
Adjusted Mutual-Index: 0.000

Contigency matrix:
==============
[[2194 1709]
 [2313 1666]]
```

```
===============
=====================================================
Number of components: 2
Dimensions of TF-IDF vector after LSI: (7882, 2)
Homogeneity: 0.577
Completeness: 0.579
V-measure: 0.578
Adjusted Rand-Index: 0.672
Adjusted Mutual-Index: 0.577

Contigency matrix:
===============
[[ 175 3728]
 [3443  536]]
===============
=====================================================
Number of components: 3
Dimensions of TF-IDF vector after LSI: (7882, 3)
Homogeneity: 0.417
Completeness: 0.452
V-measure: 0.434
Adjusted Rand-Index: 0.419
Adjusted Mutual-Index: 0.417

Contigency matrix:
===============
[[   38 3865]
 [2627 1352]]
===============
=====================================================
Number of components: 4
Dimensions of TF-IDF vector after LSI: (7882, 4)
Homogeneity: 0.231
```

```
Completeness: 0.312
V-measure: 0.266
Adjusted Rand-Index: 0.162
Adjusted Mutual-Index: 0.231

Contigency matrix:
===============
[[3890   13]
 [2341 1638]]
===============
=====================================================
Number of components: 5
Dimensions of TF-IDF vector after LSI: (7882, 5)
Homogeneity: 0.222
Completeness: 0.310
V-measure: 0.259
Adjusted Rand-Index: 0.145
Adjusted Mutual-Index: 0.222

Contigency matrix:
===============
[[   5 3898]
 [1546 2433]]
===============
=====================================================
Number of components: 6
Dimensions of TF-IDF vector after LSI: (7882, 6)
Homogeneity: 0.220
Completeness: 0.308
V-measure: 0.257
Adjusted Rand-Index: 0.143
Adjusted Mutual-Index: 0.220
```

```
Contigency matrix:
==============
[[3898    5]
 [2446 1533]]
==============
=================================================
Number of components: 10
Dimensions of TF-IDF vector after LSI: (7882, 10)
Homogeneity: 0.236
Completeness: 0.322
V-measure: 0.272
Adjusted Rand-Index: 0.159
Adjusted Mutual-Index: 0.236

Contigency matrix:
==============
[[   3 3900]
 [1613 2366]]
==============
=================================================
Number of components: 20
Dimensions of TF-IDF vector after LSI: (7882, 20)
Homogeneity: 0.237
Completeness: 0.323
V-measure: 0.274
Adjusted Rand-Index: 0.161
Adjusted Mutual-Index: 0.237

Contigency matrix:
==============
[[   3 3900]
 [1621 2358]]
==============
====================================================
Number of components: 50
Dimensions of TF-IDF vector after LSI: (7882, 50)
Homogeneity: 0.243
Completeness: 0.328
V-measure: 0.279
Adjusted Rand-Index: 0.168
Adjusted Mutual-Index: 0.243

Contigency matrix:
===============
[[   3 3900]
 [1656 2323]]
===============
====================================================
Number of components: 100
Dimensions of TF-IDF vector after LSI: (7882, 100)
Homogeneity: 0.245
Completeness: 0.329
V-measure: 0.281
Adjusted Rand-Index: 0.170
Adjusted Mutual-Index: 0.245

Contigency matrix:
===============
[[3900    3]
 [2314 1665]]
===============
====================================================
Number of components: 300
Dimensions of TF-IDF vector after LSI: (7882, 300)
Homogeneity: 0.247
Completeness: 0.330
```

```
V-measure: 0.282
Adjusted Rand-Index: 0.173
Adjusted Mutual-Index: 0.247

Contigency matrix:
==============
[[    4 3899]
 [1680 2299]]
==============
================================================
```

After comparison, we could draw the conclusion that the optimal value of the number of dimensions was found to be 2 when we use LSI to reduce dimensions, since it showed 5 best purity metric scores.

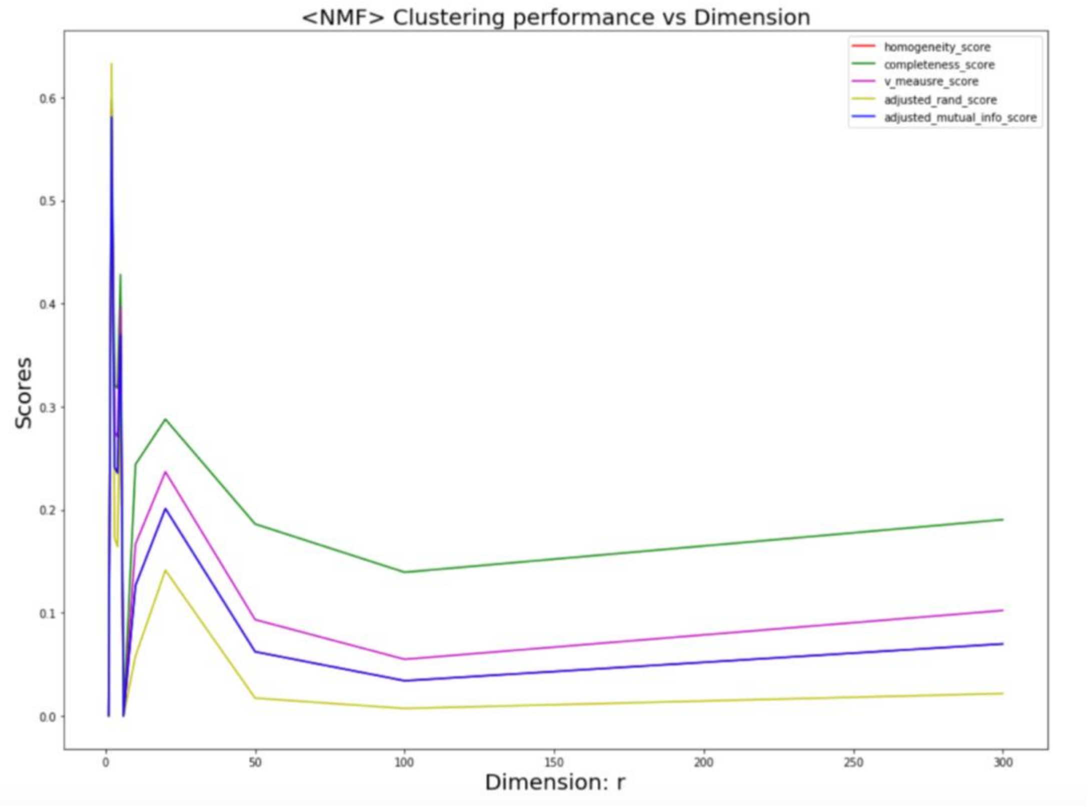
And the best performance on 2-dimension are listed as a table.

Best value for LSI is 2-dimension

Homogeneity:	Completeness:	V-measure:	Adjusted Rand-Index:	Adjusted Mutual-Index:	Contigency matrix:
0.593	0.594	0.594	0.695	0.593	[[212 3691] [3536 443]]

Also we found that the result is slightly different from different runs. In fact, by setting n_init for the KMeans object, the KMeans algorithm will be automatically run for that many times, and the best result in terms of the cost function value will be returned. Because KMeans won't give us a global minimum, this is the way that we try to get a not-too-bad local minimum. However, this best local minimum that we have is still not necessarily the global minimum, and is sensitive to the initialization (which is randomized and the randomization is controlled by the random_state). That is the reason that we don't get the same result each time.

Then, we show the results for NMF. Similarly, we tried r value from set [1,2,3,4,5,6,10,20,50,100,300], and inspecting the 5 scores. Our results are shown below.



From the plot above, we can easily find that the 5 curves are quite similar and all of them reach their peaks at 2-dimension(r=2). And we reported the scores and contingency matrix below to verify our conclusion.

```
Number of components: 1
Dimensions of TF-IDF vector after NMF: (7882, 1)
Homogeneity: 0.000
Completeness: 0.000
V-measure: 0.000
Adjusted Rand-Index: 0.000
Adjusted Mutual-Index: 0.000
Contigency matrix:
==============
[[1714 2189]
 [1671 2308]]
==============
=================================================
Number of components: 2
```

```
Dimensions of TF-IDF vector after NMF: (7882, 2)
Homogeneity: 0.580
Completeness: 0.597
V-measure: 0.589
Adjusted Rand-Index: 0.632
Adjusted Mutual-Index: 0.580
Contigency matrix:
==============
[[ 776 3127]
 [3946   33]]
==============
==================================================
Number of components: 3
Dimensions of TF-IDF vector after NMF: (7882, 3)
Homogeneity: 0.241
Completeness: 0.321
V-measure: 0.276
Adjusted Rand-Index: 0.172
Adjusted Mutual-Index: 0.241
Contigency matrix:
==============
[[   11 3892]
 [1686 2293]]
==============
==================================================
Number of components: 4
Dimensions of TF-IDF vector after NMF: (7882, 4)
Homogeneity: 0.236
Completeness: 0.318
V-measure: 0.271
Adjusted Rand-Index: 0.165
Adjusted Mutual-Index: 0.236
```

```
Contigency matrix:
===============
[[   9 3894]
 [1647 2332]]
===============
=====================================================
Number of components: 5
Dimensions of TF-IDF vector after NMF: (7882, 5)
Homogeneity: 0.365
Completeness: 0.425
V-measure: 0.393
Adjusted Rand-Index: 0.329
Adjusted Mutual-Index: 0.365
Contigency matrix:
===============
[[2229 1674]
 [   5 3974]]
===============
=====================================================
Number of components: 6
Dimensions of TF-IDF vector after NMF: (7882, 6)
Homogeneity: 0.000
Completeness: 0.000
V-measure: 0.000
Adjusted Rand-Index: -0.000
Adjusted Mutual-Index: -0.000
Contigency matrix:
===============
[[3807   96]
 [3883   96]]
===============
=====================================================
Number of components: 10
```

```
Dimensions of TF-IDF vector after NMF: (7882, 10)
Homogeneity: 0.125
Completeness: 0.243
V-measure: 0.166
Adjusted Rand-Index: 0.057
Adjusted Mutual-Index: 0.125
Contigency matrix:
===============
[[2995    908]
 [3977       2]]
===============
=========================================================
Number of components: 20
Dimensions of TF-IDF vector after NMF: (7882, 20)
Homogeneity: 0.093
Completeness: 0.216
V-measure: 0.130
Adjusted Rand-Index: 0.034
Adjusted Mutual-Index: 0.092
Contigency matrix:
===============
[[ 689 3214]
 [      2 3977]]
===============
=========================================================
Number of components: 50
Dimensions of TF-IDF vector after NMF: (7882, 50)
Homogeneity: 0.062
Completeness: 0.186
V-measure: 0.093
Adjusted Rand-Index: 0.017
Adjusted Mutual-Index: 0.062
Contigency matrix:
```

```
==============
[[ 485 3418]
 [    3 3976]]
==============
==================================================
Number of components: 100
Dimensions of TF-IDF vector after NMF: (7882, 100)
Homogeneity: 0.034
Completeness: 0.139
V-measure: 0.055
Adjusted Rand-Index: 0.007
Adjusted Mutual-Index: 0.034
Contigency matrix:
==============
[[ 311 3592]
 [   10 3969]]
==============
==================================================
Number of components: 300
Dimensions of TF-IDF vector after NMF: (7882, 300)
Homogeneity: 0.070
Completeness: 0.190
V-measure: 0.102
Adjusted Rand-Index: 0.022
Adjusted Mutual-Index: 0.070
Contigency matrix:
==============
[[ 550 3353]
 [    5 3974]]
==============
```

After comparison, we could draw the conclusion that the optimal value of the number of dimensions was found is also 2 when we use NMF to reduce dimensions, since it showed 5 best purity metric scores.

And the best performance on 2-dimension are listed as a table.

Best value for NMF is 2-dimension

Homogeneity:	Completeness:	V-measure:	Adjusted Rand-Index:	Adjusted Mutual-Index:	Contigency matrix:
0.586	0.602	0.594	0.641	0.586	[[3153 750] [36 3943]]

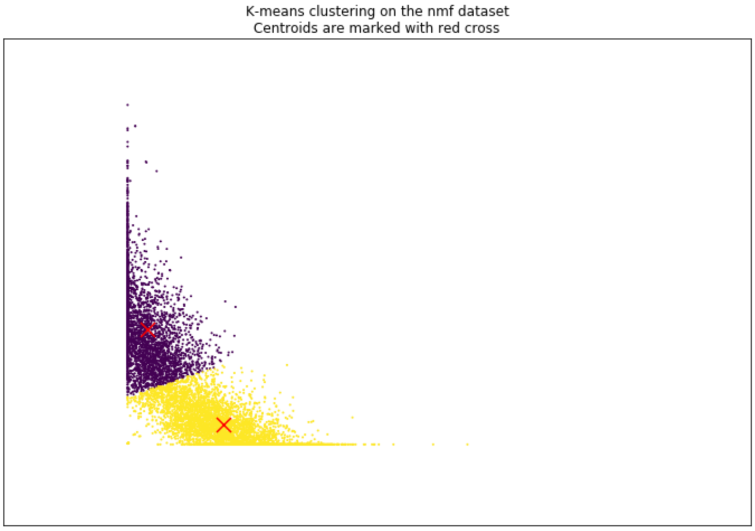
Both the methods LSI and NMF, to be mentioned, all measure results don't monotonically increase with the r value. Because there would a best value on r=2 so that the curves first go up then go down, finally seem to retain. Comparing with the results of SVD, the performance of NMF is a bit worse. In order to get better precision, we are going to apply normalization and non-linear transformations to the reduced TF-IDF matrix, which are exactly what we are supposed to do in the following tasks.

5 Question 4

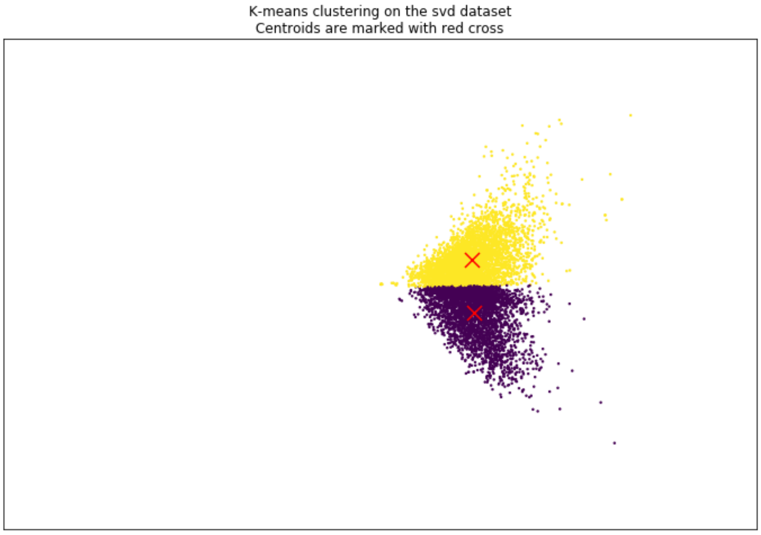
Part a- visualization with original dataset

In question 4(a), we performed visualization toward the cluster data we got from both NMF and SVD dimension reduction. The best value for both SVD and NMF is 2-dimension.

Here is the figure for NMF with dimension n=2. Its metric scores are shown in the last question.



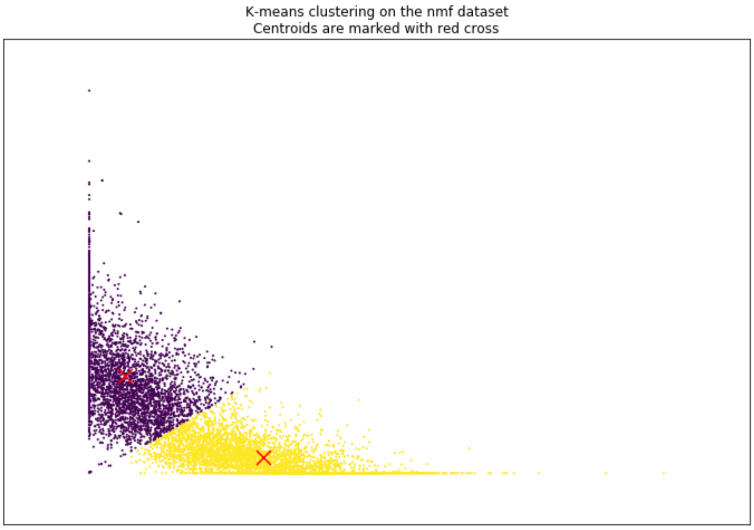
Here is the figure for SVD with dimension n=2. Its metric scores are shown in the last question.



Part b- visualization with normalizing

In this part, we first performed normalizing features toward the dimension reduced datasets. We are using sklearn.preprocessing.scale package in order to achieve unit variance among dataset. Then we performed visualization again and also showed the five-score for the clustering.

Here is the figure for NMF with normalizing.



Homogeneity: 0.686

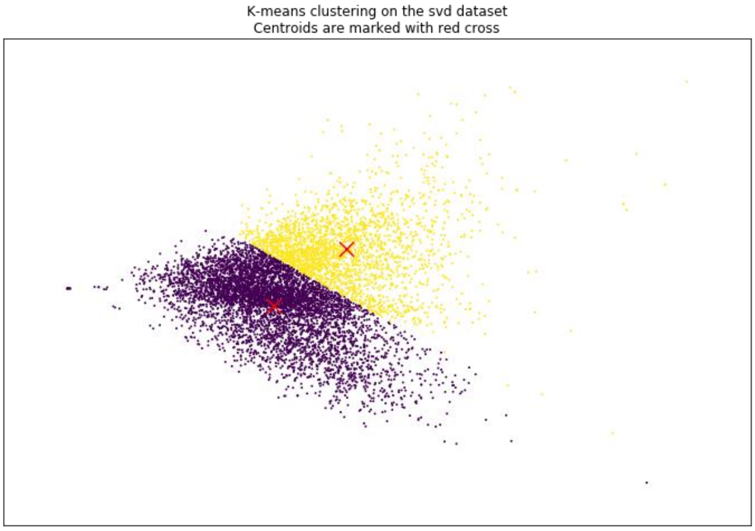
Completeness: 0.689

V-measure: 0.687

Adjusted Rand-Index: 0.777

Adjusted Mutual-Index: 0.686

Below is the figure for SVD with normalizing.



Homogeneity: 0.222

Completeness: 0.252

V-measure: 0.236

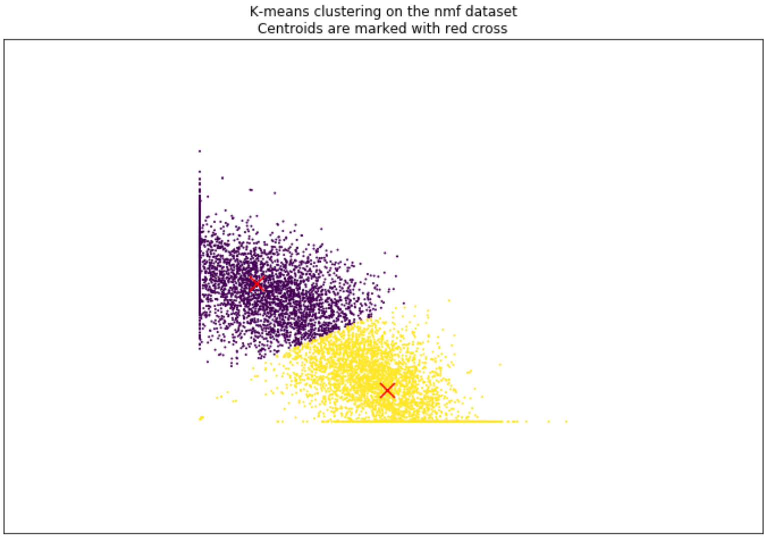
Adjusted Rand-Index: 0.238

Adjusted Mutual-Index: 0.222

Part c- visualization with non-linear transformation

Then we performed non-linear transformation to the NMF dataset. Here we use logarithm transformation towards the NMF dataset since it is all positive. For SVD we are not able to log the dataset due to the negative feature. In order to get relevant result to our dataset, we add 0.01 to our dataset in order to avoid the case log(0) to the nmf dataset.

Below is the figure for NMF with logarithm transformation.



Homogeneity: 0.704

Completeness: 0.705

V-measure: 0.705

Adjusted Rand-Index: 0.799

Adjusted Mutual-Index: 0.704

Analysis:

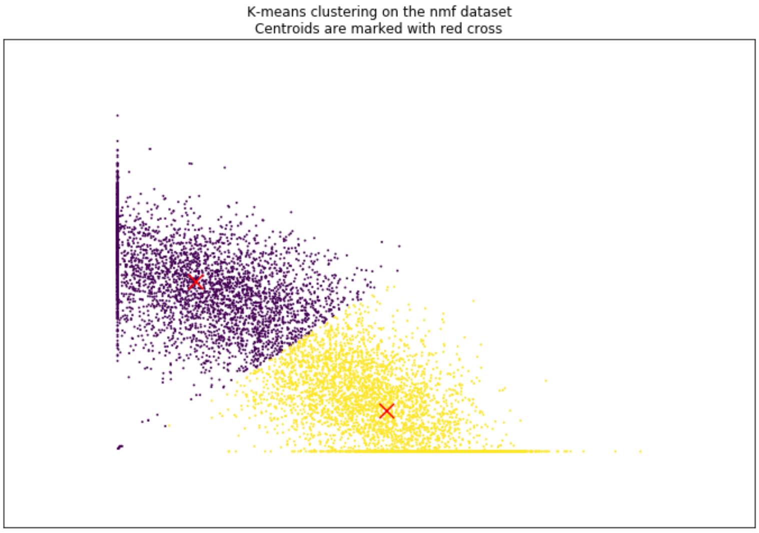
In this case, logarithm transformation increases the clustering result, taking the log usually reduce the skewness and make the distribution closer to normal. With a smaller skewness the data will be better captured by the kmeans. Thus the clustering result are thus better in shape and all of the five score improved with significant amount.

Part d- visualization with combination of 2 methods

Next we tried to perform both normalization and logarithm transformation methods toward the NMF datasets.

We start with logarithm transformation first, followed by scaling of data to unit variance.

Below is the figure for NMF with logarithm transformation first and normalization second.



Homogeneity: 0.702

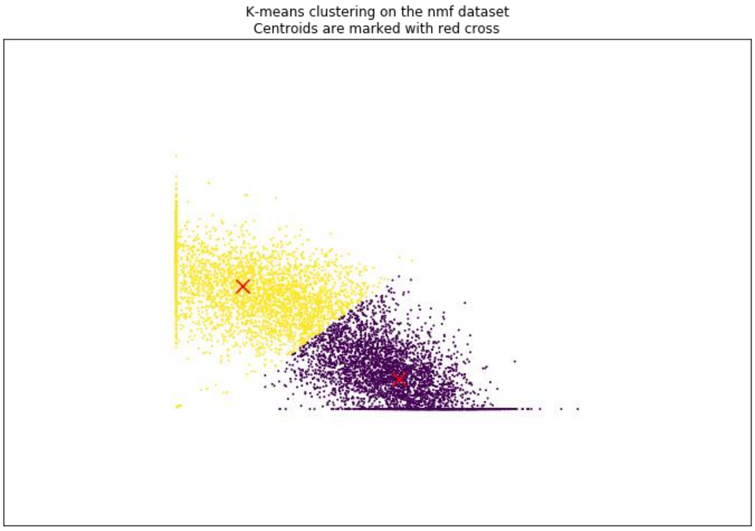
Completeness: 0.704

V-measure: 0.703

Adjusted Rand-Index: 0.797

Adjusted Mutual-Index: 0.702

Then we try normalize data with unit variance then followed by logarithm transformation. We found that there is a lot of negative number in the dataset. Which will become NaN after log transformation. Therefore, instead of adding 0.01 before log, we had to add 2 in order to avoid negative number.



Homogeneity: 0.665

Completeness: 0.669

V-measure: 0.667

```
Adjusted  Rand-Index:  0.753
Adjusted  Mutual-Index:  0.665
```

From the test ,we can see that log transformation and log-norm method will give us better clustering result. Since logged data will become more likely closed to normally distributed with a relative unit variance.

6 Question 5

For this question, we expand the categories to 20 classes, and perform the similar procedures like we did in the past several questions.
We examine how purely we can retrieve all the 20 original sub-class labels with clustering. Therefore, we include all the documents and the corresponding terms in the data matrix and find proper representation through reducing the dimension of TF-IDF matrix representation.

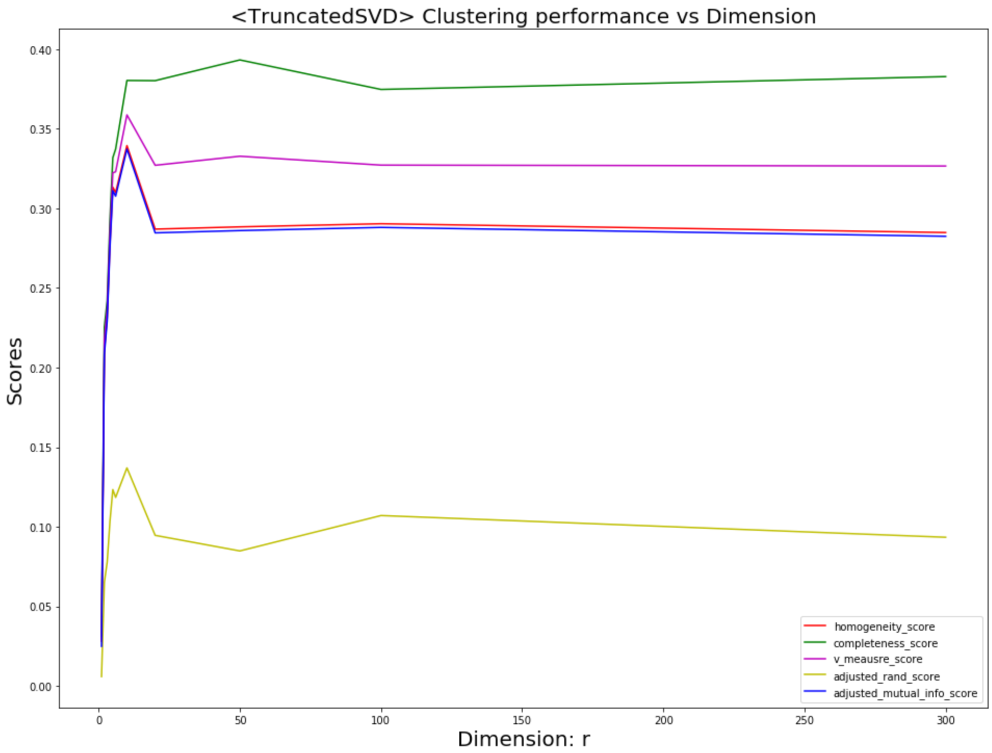
1)We first retrieve all the 20 original sub-class documents and generate TF-IDF as before. And then we cluster them without any reduction in dimensionality. Below table shows results and corresponding purity measures.

```
The dataset shape was (18846, 51729)
Homogeneity:  0.284162983471
Completeness:  0.350029655297
V-measure:  0.313675893009
Adjusted  Rand  Score:  0.102136000485
Adjusted  Mutual  Info  Score:  0.281833502495
```

2)The precision of the clustering labels in the last question is not as satisfactory as we expect it to be. This is possibly caused by high dimension TF-IDF matrix. Because we use the word count information to build the TF-IDF matrix, with high dimension TF-IDF, too many unimportant words would result in very serious overfitting, and data points is possibly to be distracted far away from where they should be.

To overcome this shortcoming and improve the performance of our algorithm, we will try to reduce the dimension of the TF-IDF matrix. By reducing the dimension, some unrelated information could be eliminated after the process (for example, by doing projection process). Only the most significant dimensions remain and ideally, we can improve the precision of the cluster results. In this project, we will be using Latent Semantic Indexing (LSI) and Non-Negative Matrix Factorization (NMF).

First, we show the results for LSI. We tried r value from set [1,2,3,4,5,6,10,20,50,100,300] as before, and inspecting the 5 scores. Our results are shown below.



As we know, evaluating the performance of a clustering result is actually not an easy and simple task, which is exactly the reason we plot at all 5 measures, instead of just one. From the plot above, we treat homogeneity score and completeness score as the most significant, and we can easily find that the 5 curves are quite similar and all of them reach their peaks at 10-dimension(r=10). And we reported the scores and contingency matrix below to verify our conclusion.

Number of components: 1

```
Dimensions of TF-IDF vector after LSI: (18846, 1)
Homogeneity: 0.028
Completeness: 0.031
V-measure: 0.029
Adjusted Rand-Index: 0.006
Adjusted Mutual-Index: 0.025
Number of components: 2
Dimensions of TF-IDF vector after LSI: (18846, 2)
Homogeneity: 0.211
Completeness: 0.225
V-measure: 0.218
Adjusted Rand-Index: 0.066
Adjusted Mutual-Index: 0.208
Number of components: 3
Dimensions of TF-IDF vector after LSI: (18846, 3)
Homogeneity: 0.235
Completeness: 0.244
V-measure: 0.239
Adjusted Rand-Index: 0.081
Adjusted Mutual-Index: 0.232
Number of components: 4
Dimensions of TF-IDF vector after LSI: (18846, 4)
Homogeneity: 0.279
Completeness: 0.294
V-measure: 0.287
Adjusted Rand-Index: 0.105
Adjusted Mutual-Index: 0.277
Number of components: 5
Dimensions of TF-IDF vector after LSI: (18846, 5)
Homogeneity: 0.309
Completeness: 0.327
V-measure: 0.318
Adjusted Rand-Index: 0.123
Adjusted Mutual-Index: 0.307
Number of components: 6
Dimensions of TF-IDF vector after LSI: (18846, 6)
Homogeneity: 0.310
Completeness: 0.332
V-measure: 0.320
Adjusted Rand-Index: 0.119
Adjusted Mutual-Index: 0.307
Number of components: 8
Dimensions of TF-IDF vector after LSI: (18846, 8)
Homogeneity: 0.336
Completeness: 0.371
V-measure: 0.353
Adjusted Rand-Index: 0.138
Adjusted Mutual-Index: 0.334
Number of components: 10
Dimensions of TF-IDF vector after LSI: (18846, 10)
Homogeneity: 0.333
Completeness: 0.368
V-measure: 0.350
Adjusted Rand-Index: 0.148
Adjusted Mutual-Index: 0.331
Number of components: 20
Dimensions of TF-IDF vector after LSI: (18846, 20)
Homogeneity: 0.284
Completeness: 0.358
V-measure: 0.317
Adjusted Rand-Index: 0.102
Adjusted Mutual-Index: 0.282
Number of components: 50
Dimensions of TF-IDF vector after LSI: (18846, 50)
Homogeneity: 0.271
Completeness: 0.360
```

V-measure: 0.309
Adjusted Rand-Index: 0.084
Adjusted Mutual-Index: 0.268
Number of components: 100
Dimensions of TF-IDF vector after LSI: (18846, 100)
Homogeneity: 0.295
Completeness: 0.371
V-measure: 0.329
Adjusted Rand-Index: 0.117
Adjusted Mutual-Index: 0.293
Number of components: 300
Dimensions of TF-IDF vector after LSI: (18846, 300)
Homogeneity: 0.309
Completeness: 0.482
V-measure: 0.377
Adjusted Rand-Index: 0.081
Adjusted Mutual-Index: 0.307

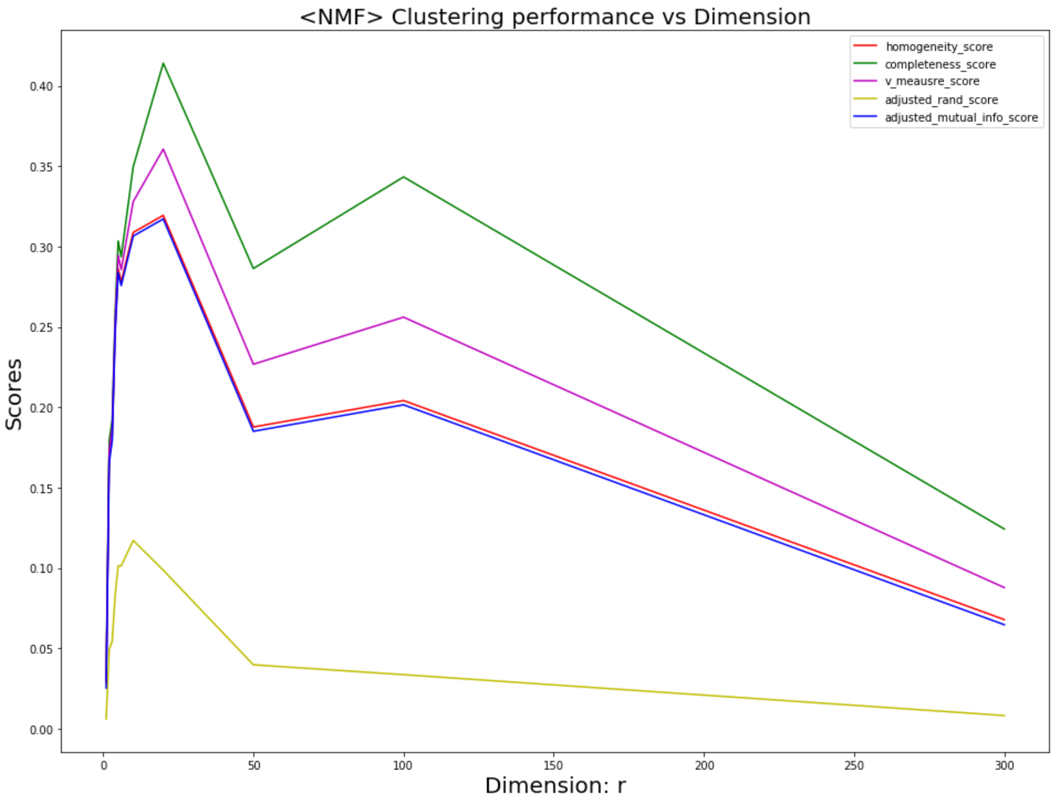
After comparison, we could draw the conclusion that the optimal value of the number of dimensions was found to be 10 when we use LSI to reduce dimensions, since it showed 5 best purity metric scores.

And the best performance on 2-dimension are listed as a table.

Best value for LSI is 10-dimension

Homogeneity:	Completeness:	V-measure:	Adjusted Rand-Index:	Adjusted Mutual-Index:
0.339	0.378	0.358	0.139	0.337

Then, we show the results for NMF. Similarly, we tried r value from set [1,2,3,4,5,6,10,20,50,100,300], and inspecting the 5 scores. Our results are shown below.



From the plot above, we treat homogeneity score and completeness score as the most significant, and we can easily find that the 5 curves are quite similar and all of them reach their peaks approximately at 20-dimention. And we reported the scores from r-set [10,12,15,18,20,21,22,23,24,25,30] to find the best value and contingency matrix below to verify our conclusion.

```
(18846, 51729)
Number of components: 10
Dimensions of TF-IDF vector after NMF: (18846, 10)
Homogeneity: 0.309
Completeness: 0.351
V-measure: 0.329
Adjusted Rand-Index: 0.120
Adjusted Mutual-Index: 0.307
(18846, 51729)
Number of components: 12
Dimensions of TF-IDF vector after NMF: (18846, 12)
Homogeneity: 0.309
```

```
Completeness: 0.362
V-measure: 0.333
Adjusted Rand-Index: 0.118
Adjusted Mutual-Index: 0.307
(18846, 51729)
Number of components: 15
Dimensions of TF-IDF vector after NMF: (18846, 15)
Homogeneity: 0.282
Completeness: 0.331
V-measure: 0.304
Adjusted Rand-Index: 0.099
Adjusted Mutual-Index: 0.279
(18846, 51729)
Number of components: 18
Dimensions of TF-IDF vector after NMF: (18846, 18)
Homogeneity: 0.279
Completeness: 0.336
V-measure: 0.305
Adjusted Rand-Index: 0.095
Adjusted Mutual-Index: 0.277
(18846, 51729)
Number of components: 20
Dimensions of TF-IDF vector after NMF: (18846, 20)
Homogeneity: 0.320
Completeness: 0.415
V-measure: 0.362
Adjusted Rand-Index: 0.099
Adjusted Mutual-Index: 0.318
(18846, 51729)
Number of components: 21
Dimensions of TF-IDF vector after NMF: (18846, 21)
Homogeneity: 0.286
Completeness: 0.391
```

```
V-measure: 0.330
Adjusted Rand-Index: 0.081
Adjusted Mutual-Index: 0.283
(18846, 51729)
Number of components: 22
Dimensions of TF-IDF vector after NMF: (18846, 22)
Homogeneity: 0.303
Completeness: 0.411
V-measure: 0.349
Adjusted Rand-Index: 0.076
Adjusted Mutual-Index: 0.300
(18846, 51729)
Number of components: 23
Dimensions of TF-IDF vector after NMF: (18846, 23)
Homogeneity: 0.312
Completeness: 0.440
V-measure: 0.365
Adjusted Rand-Index: 0.076
Adjusted Mutual-Index: 0.310
(18846, 51729)
Number of components: 24
Dimensions of TF-IDF vector after NMF: (18846, 24)
Homogeneity: 0.268
Completeness: 0.403
V-measure: 0.322
Adjusted Rand-Index: 0.061
Adjusted Mutual-Index: 0.265
(18846, 51729)
Number of components: 25
Dimensions of TF-IDF vector after NMF: (18846, 25)
Homogeneity: 0.297
Completeness: 0.429
V-measure: 0.351
```

```
Adjusted Rand-Index: 0.072
Adjusted Mutual-Index: 0.295
(18846, 51729)
Number of components: 30
Dimensions of TF-IDF vector after NMF: (18846, 30)
Homogeneity: 0.238
Completeness: 0.403
V-measure: 0.299
Adjusted Rand-Index: 0.050
Adjusted Mutual-Index: 0.235
```

After comparison, we could draw the conclusion that the optimal value of the number of dimensions was found to be 23 when we use NMF to reduce dimensions, since it showed almost 5 best purity metric scores, and we treat homogeneity and completeness as the most important two.
And the best performance on 23-dimension are listed as a table.

Best value for NMF is 23-dimension

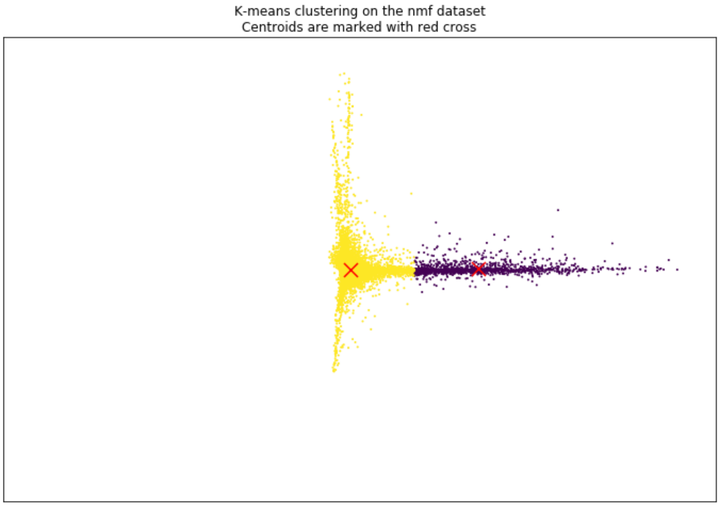
Homogeneity:	Completeness:	V-measure:	Adjusted Rand-Index:	Adjusted Mutual-Index:
0.312	0.440	0.366	0.076	0.310

Comparing with the results of SVD, the performance of NMF is a little bit worse. In order to get better precision, we are going to apply normalization and non-linear transformations on NMF.

3) visualization

We implement the similar method as question 4 to the all_dataset. First we start with visualization of the dataset. However, this time since the best dimension we got is 10 and 23 respectively for SVD and NMF, we use a PCA reduction to reduce the dimension of the dataset to 2 in order to present it into a 2-D format which is good for inspection. Here is the scatter plot we got from each method.

Here is the figure for NMF with 2-dimension inspection.



Homogeneity: 0.060

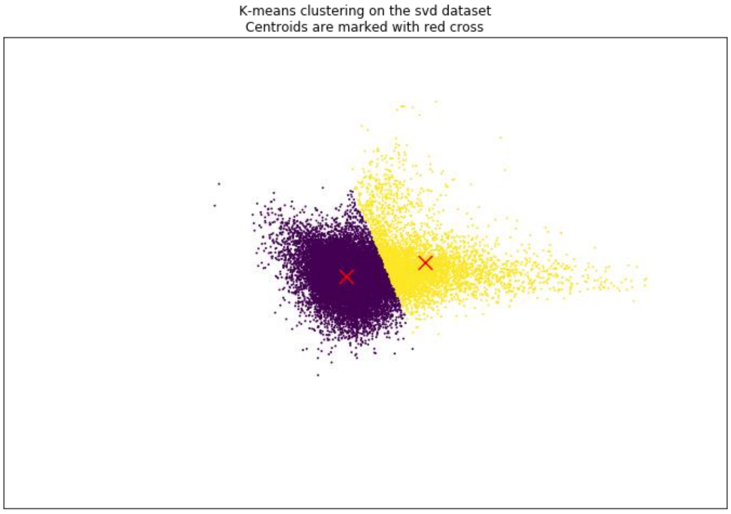
Completeness: 0.711

V-measure: 0.110

Adjusted Rand-Index: 0.010

Adjusted Mutual-Index: 0.060

Below is the figure for LSI with 2-dimension inspection.



Homogeneity: 0.085

Completeness: 0.460

V-measure: 0.143

Adjusted Rand-Index: 0.030

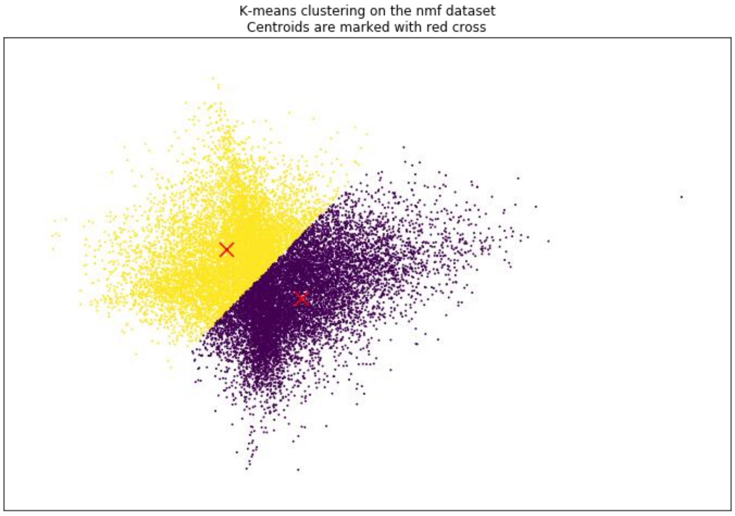
Adjusted Mutual-Index: 0.085

From both method we can see that the clustering result is not very ideal. Therefore, we decide to take a few more attempts to improve the results. We use the similar method as in question 4 to try to scale data to unit variance and perform logarithm transformation.

4) visualization with normalization

We start with normalization to unit variance.

Below is the figure for NMF with normalization.



Homogeneity: 0.066

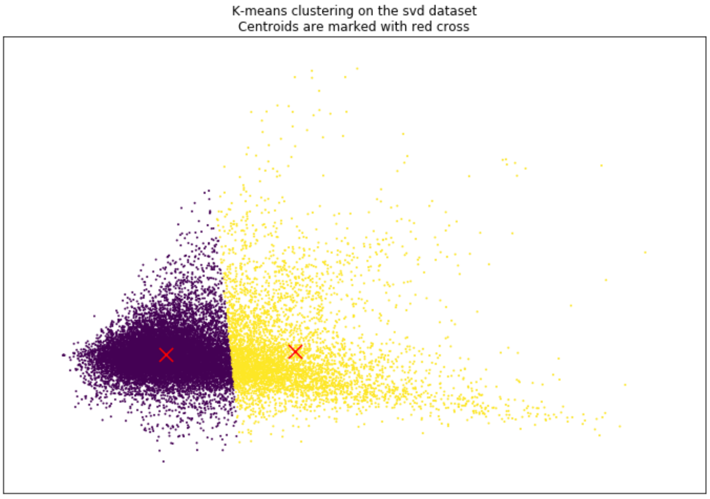
Completeness: 0.284

V-measure: 0.107

Adjusted Rand-Index: 0.035

Adjusted Mutual-Index: 0.065

Below is the figure for SVD with normalization.



Homogeneity: 0.049

Completeness: 0.256

V-measure: 0.082

Adjusted Rand-Index: 0.020

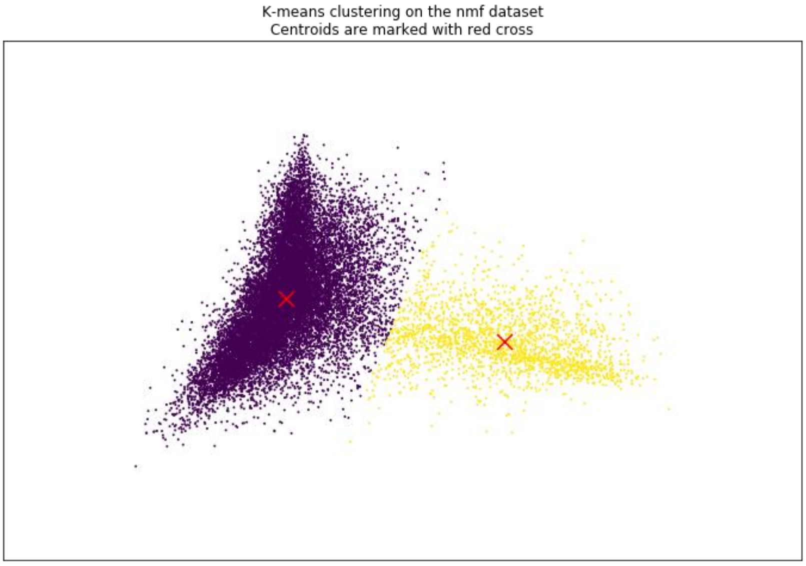
Adjusted Mutual-Index: 0.049

From this method we can observe that the shape of the clustering result are getting better. However, the five score decreases significantly after normalization thus make the normalizing features a bad way to improve the clustering result rather than improving it to a better 2-D representation.

5) visualization with nonlinear transformation

Then we started to try nonlinear transformation and we use logarithm transformation. We can only use log transformation to NMF dataset with the addition of a small amount of number which is 0.01 to avoid log0 case.

Here is the figure for NMF with log transformation.



Homogeneity: 0.088

Completeness: 0.772

V-measure: 0.157

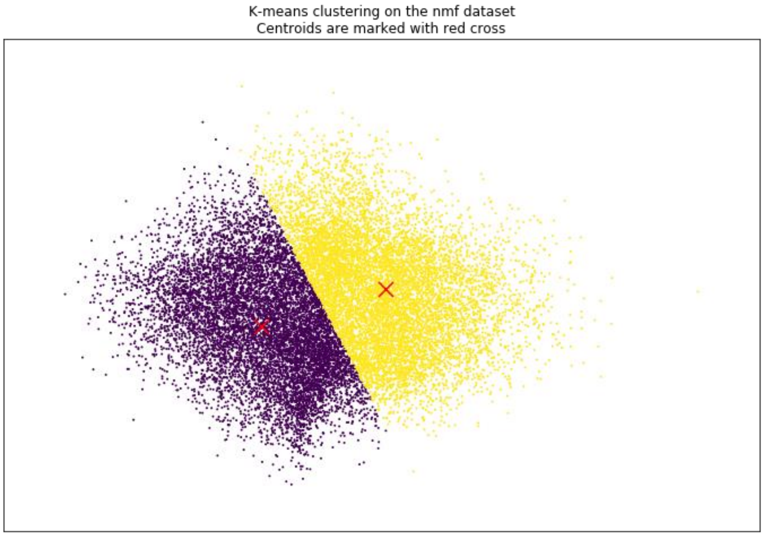
Adjusted Rand-Index: 0.020

Adjusted Mutual-Index: 0.087

From the result of five score, we can see that the V-measure score and adjusted rand and mutual index both dropped but is better than normalization. However, the dataset is seemingly not as good as the normalization feature.

6) visualization with combination of 2 methods

After that, we implemented both method on NMF dataset with different orders. Below is the figure for NMF with log transformation first then normalization.



Homogeneity: 0.110

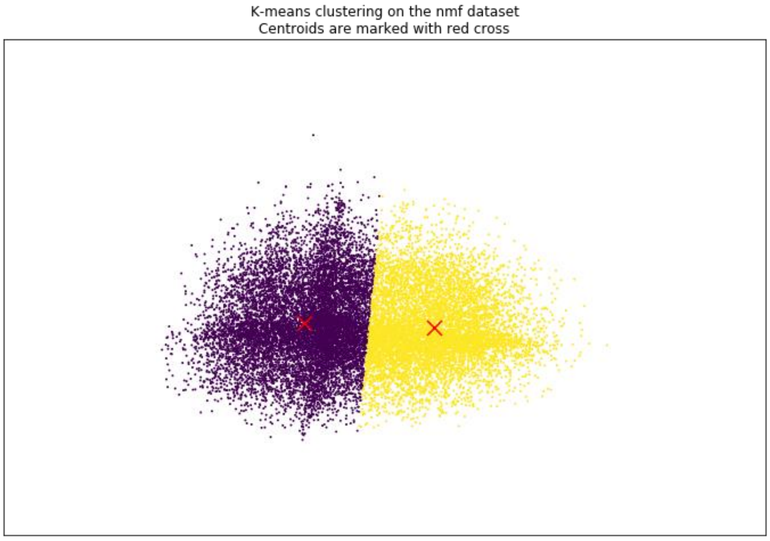
Completeness: 0.473

V-measure: 0.178

Adjusted Rand-Index: 0.054

Adjusted Mutual-Index: 0.110

Below is the figure for NMF with normalization first then log transformation.



Homogeneity: 0.006

Completeness: 0.025

V-measure: 0.009

Adjusted Rand-Index: 0.003

Adjusted Mutual-Index: 0.005

From the result we can see that performing logarithm transformation first followed by the normalizing feature is better than the other way around. However, in this multi-category dataset case, the original result is a little better. Though the dimension reduction in order to reduce data may causing negative effect to our clustering results.