Project 2: Clustering on 20NewsGroup

ECE 219
Wenyang Zhu 904947071
Jui Chang 804506544

Introduction:

Clustering algorithms are a method of unsupervised learning, which is for finding groups of data set with similar feature in a proper space. In this project, we use K-means clustering.

K-mean clustering: Given a set of point $\{x_1, x_2, \ldots, x_N\}$ in multidimensional space. It tries to find K clusters, with each point belongs to one cluster, and to minimize the sum of the squares of the distances between each data point and the center of cluster. μ_k is the center of kth cluster.

$$r_{n,k} = \begin{cases} 1, & x_n \; assigned \; to \; cluster \; k \\ 0, & otherwise \end{cases}, \quad n = 1, \ldots, N \quad k = 1, \ldots K$$

$$\underset{\mu_k, r_{n,k}}{\operatorname{argmin}} J, \qquad J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{n,k} \, ||x_n - \mu_k||^2$$

The approach of K-means algorithm is as follows

Randomly initialize the center positions of clusters.

1. (Re)assign the data point to the nearest cluster center
2. (Re)calculate the center position of the clusters: setting the center of cluster to the mean of data points within the current cluster.
3. Repeat 1 and 2 until J converge

Dataset:

We work with "20 Newsgroups" dataset we already used in project 1.

Problem 1. Building TF-IDF matrix

In this part, we use the same 8 categories as Project1. We clean up the document by removing punctuations, stop words, and set minimum document frequency, min_df = 3, and then transform the cleaned document into TFxIDF vectors to do further analysis, similar to the step we do in Project1.

However, unlike the classification training procedures, it doesn't require labels for clustering. Therefore, we can use the whole data set (both the train data and test data) to do the clustering. In this project report, however, we still just use the train data alone to do the clustering. The reason we do so is that we hope to use the same data as the training data in classification for Project 1. We think that changing the input data from "all" to "train" only reduces a bit of the size of data set, therefore won't have much influence on our clustering results in this project.

Table 1: Two well-separated classes

Class1	comp.graphics	comp.os.mswindows.misc	comp.sys.ibm.pc.hardware	comp.sys.mac.hardware
Class2	rec.autos	rec.motorcycles	rec.sport.baseball	rec.sport.hockey

Result:
With min_df = 3, the size of matrix is (4732, 16502), hence the total number of terms in documents is 16502

Problem 2. Apply K-means and cluster the eight given classes into two clusters
In this part, we apply k-means clustering with k = 2 using the TF-IDF vectors we got in problem1. We compare our predicted label from clustering to the known class labels.
(a) We generate a confusion matrix in figure 1 to compare our clustering with the class labels.
(b) To compare the different clustering results, we computed homogeneity score, the completeness score, the V-measure, the adjusted Rand score and the adjusted mutual info score to show the performance. The homogeneity score determines each cluster contains only members of a single class. The completeness score determines all members of a given class are assigned to the same cluster. The V-measure is the harmonic mean of homogeneity and completeness. The adjusted rand score is a measure of similarity between labels and ground truth. The adjusted mutual info score determines the common intersection between the cluster label and the ground truth label distributions. The 5 measure scores of clustering result is shown on table 2.

Result:
We use a heat map to show the confusion matrix in the below Figure 2, and also use Table 2 to show all the measure scores. We can see that the 5 measure scores for our clustering isn't high enough, so we try to improve the performance through the steps in problem 3-4.

Figure 2. Confusion matrix with k=2

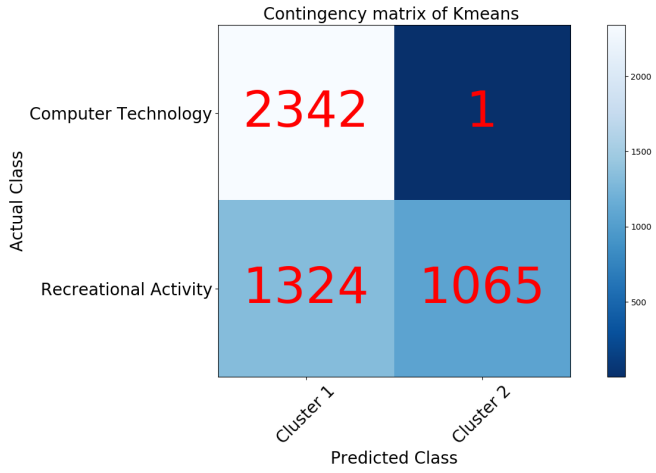


Table 2. The 5 measure scores of clustering result

Metric	Score
homogeneity score	0.266
completeness score	0.346
V-measure	0.301
adjusted rand index	0.193
adjusted mutual info score	0.185

Problem 3. Preprocess the data

Because the high dimensional TFxIDF vectors do not yield a good clustering result, we try to reduce the dimension of data to find a better representation. Here, we use two methods to reduce dimension: Latent Semantic Indexing(LSI) and Non-negative Matrix Factorization(NMF) to fit the clustering algorithms.

First, we implement LSI to reduce dimension. We want to find the effective dimension of data. One method is to see the ratio of the variance of original data is retained after applying LSI to the original data, so we plot percent of variance vs top r principal component, r ranges from 1 to 1000. The other method is that we try with principal component r = {1, 2, 3, 5, 10, 20, 50, 100, 300}, with each r we find confusion matrix and also the 5 measure of scores.

Second, we implement NMF to reduce dimension. Similar step to the above LSI, try r = {1, 2, 3, 5, 10, 20, 50, 100, 300}, we find confusion matrix and also the 5 measure of scores to form a table below

Result:

From below, figure 3.1, we can observe that with top r= 200 we can cover 20% of the variance of original data, and top r = 600 principal components, we can cover 50%, and up to r= 1000, we can cover more than 60%. Compared to the total 4732 terms in

the document, the top 1000(lower than 25% of original data) cover most of the variance of original data, so the principal component can use to represent the data set.
From below, table 3.1 and figure 3.2, we can see with r = 3 in LSI, we have the best 5 measure scores, which means best performance in clustering.
From below, table 3.2 and figure 3.4, we can see with r = 2 in NMF, we have the best 5 measure scores, which means best performance in clustering.

Question: How do you explain the non-monotonic behavior of the measures as r increases?

When the dimension, which is the number of features r, increases, the distances between different documents have less differences among different classes. This is because the fact that the Euclidean distances between different classes become more similar as the dimension increases, so a large r would not fit, and also very small r may not have enough number of clusters to perform well, so the 5 scores of measurement is non-monotonic as r increases. Here, with r = 3 in LSI and with r = 2 in NMF, we have the best performance in clustering.

Figure 3.1 percent of variance vs top r principal component

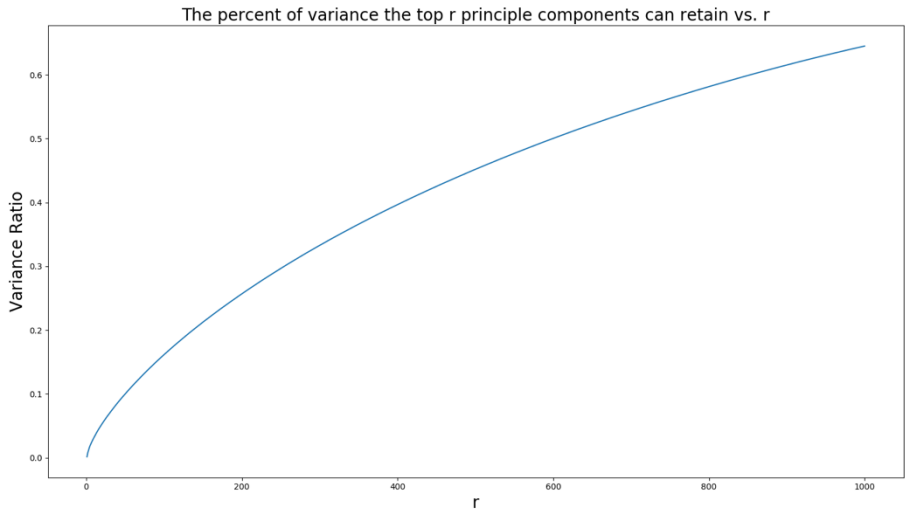


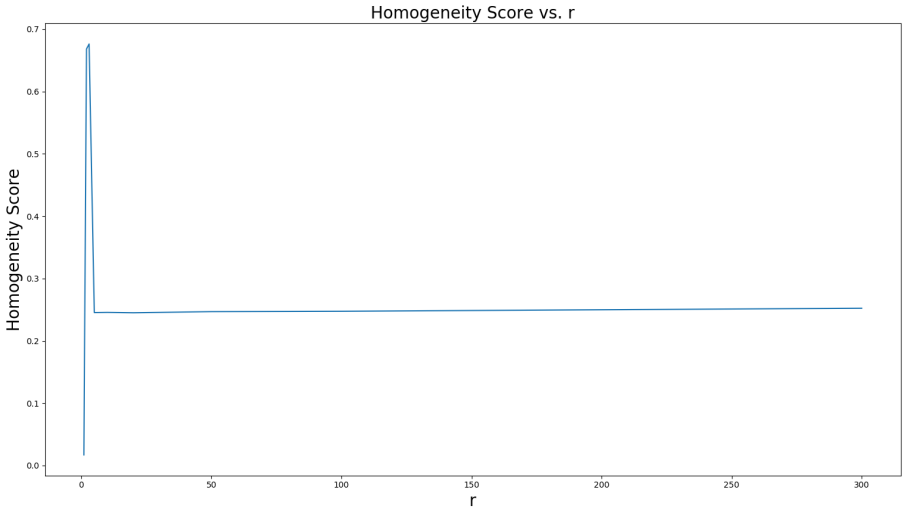
Table 3.1 LSI method, the 5 measure scores with r = {1, 2, 3, 5, 10, 20, 50, 100, 300}

r	Homogeneity score	Completeness score	V-measure score	adjusted rand index	adjusted mutual info score
1	0.017	0.018	0.017	0.023	0.012
2	0.668	0.669	0.668	0.770	0.463

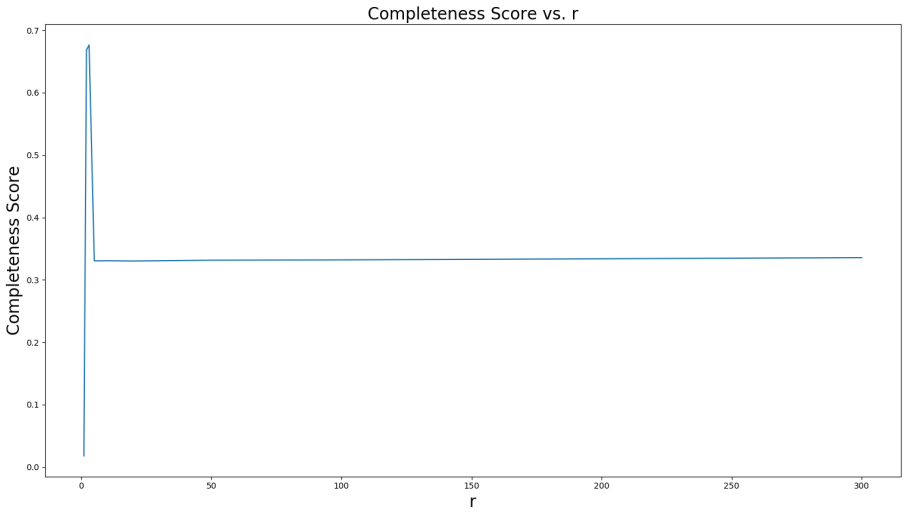
3	**0.676**	**0.676**	**0.676**	**0.777**	**0.469**
5	0.245	0.330	0.282	0.169	0.170
10	0.246	0.331	0.282	0.169	0.170
20	0.245	0.330	0.281	0.168	0.170
50	0.247	0.331	0.283	0.170	0.171
100	0.248	0.332	0.284	0.171	0.172
300	0.252	0.336	0.288	0.177	0.175

Figure 3.2 LSI method, the 5 measure scores vs r

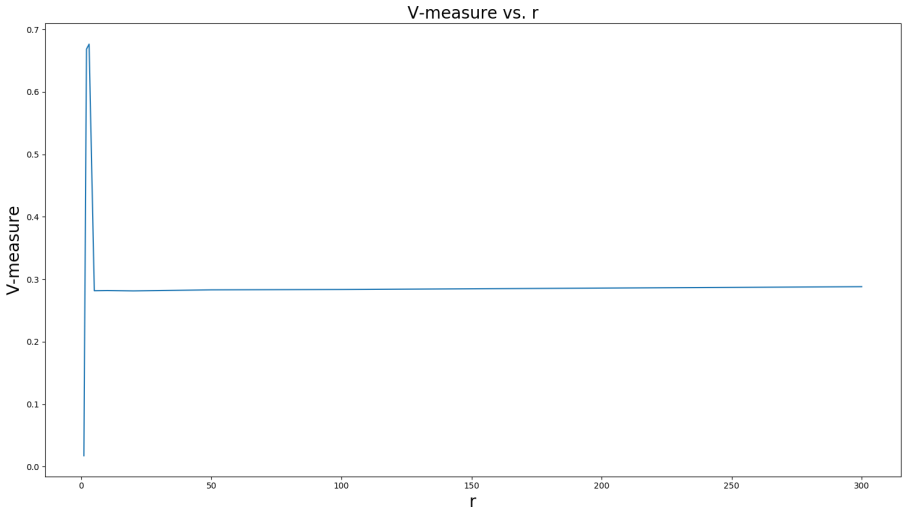
(a) Homogeneity score vs r



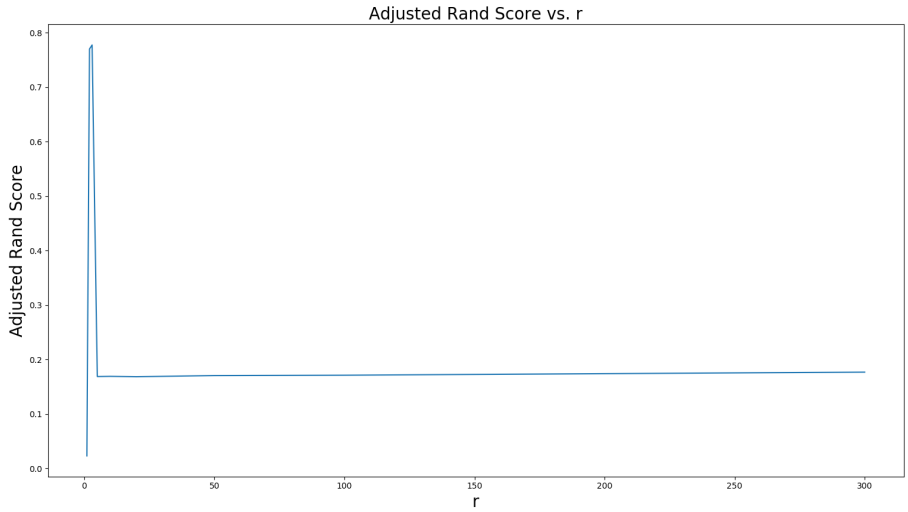
(b) Completeness score vs r



(a) V-measure score vs r



(b) Adjusted rand index vs r



(c) Adjusted mutual info score vs r

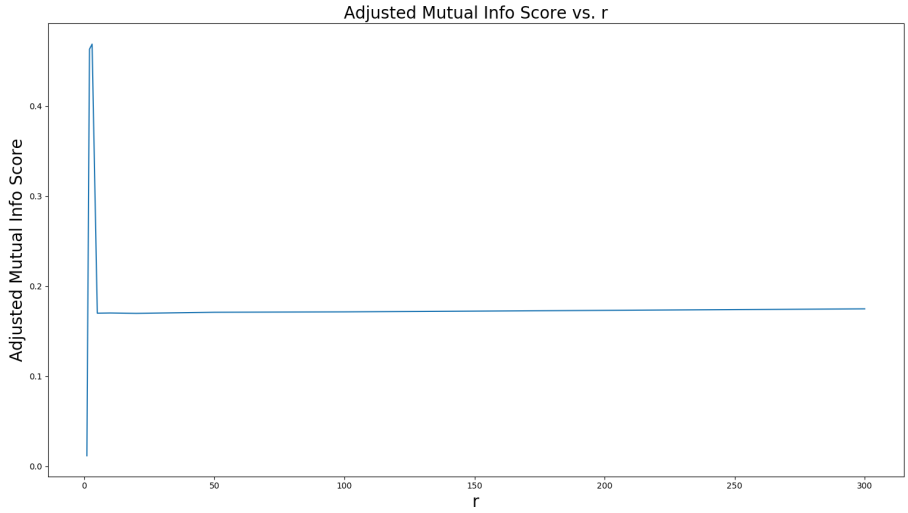
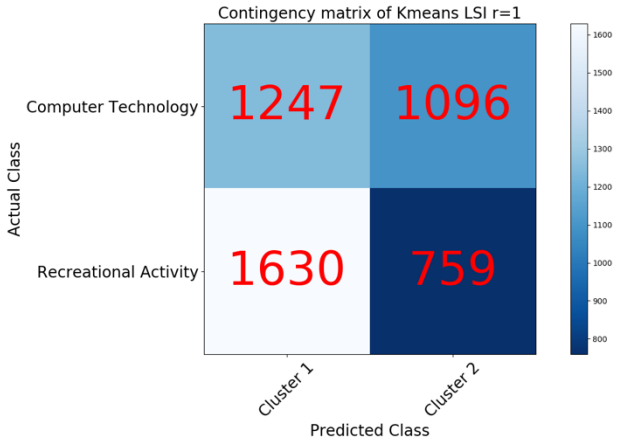
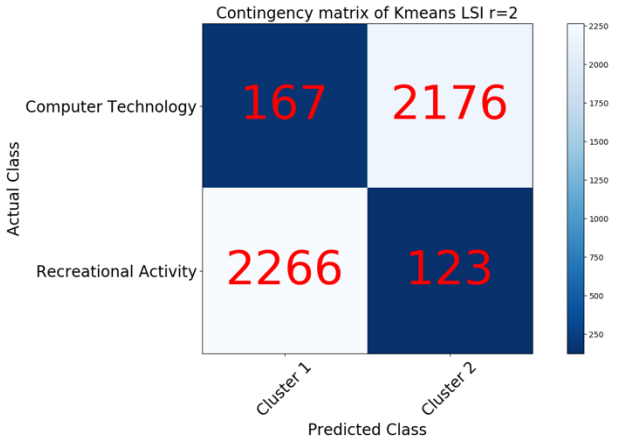


Figure 3.3 LSI method, confusion matrix with r = {1, 2, 3, 5, 10, 20, 50, 100, 300}

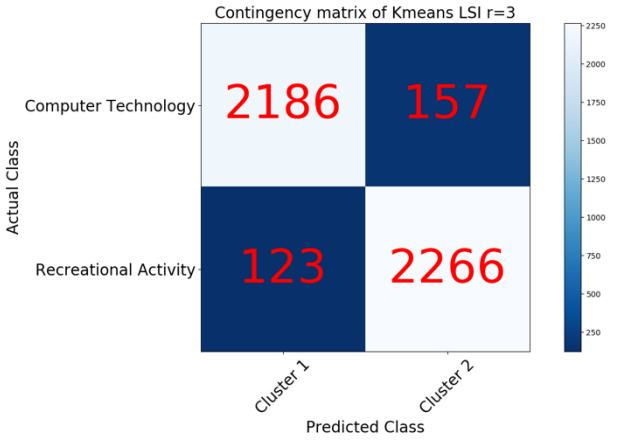
(a) Confusion matrix with r = 1



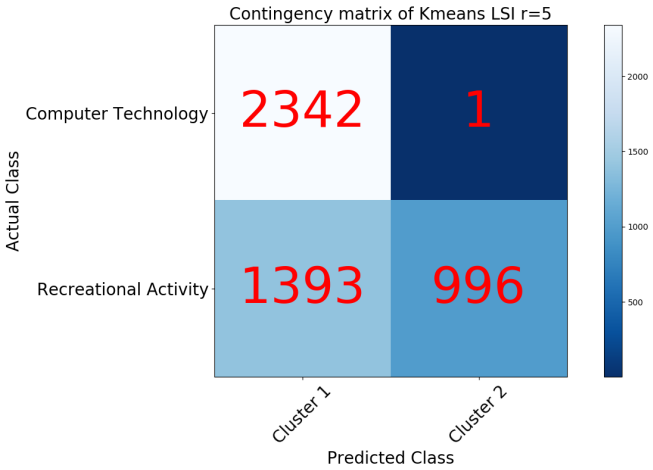
(b) Confusion matrix with r = 2



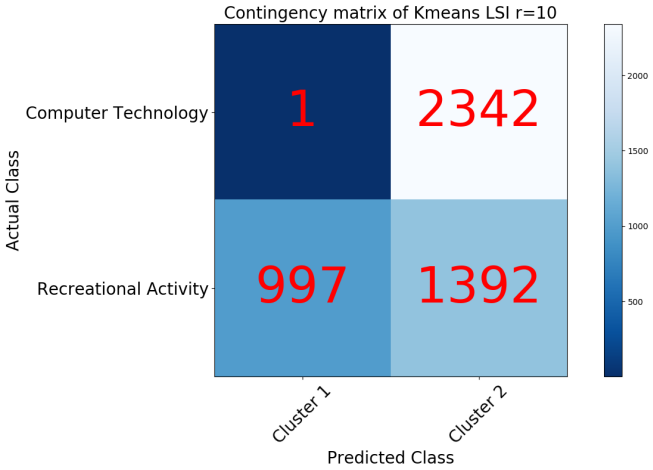
(c) Confusion matrix with r = 3



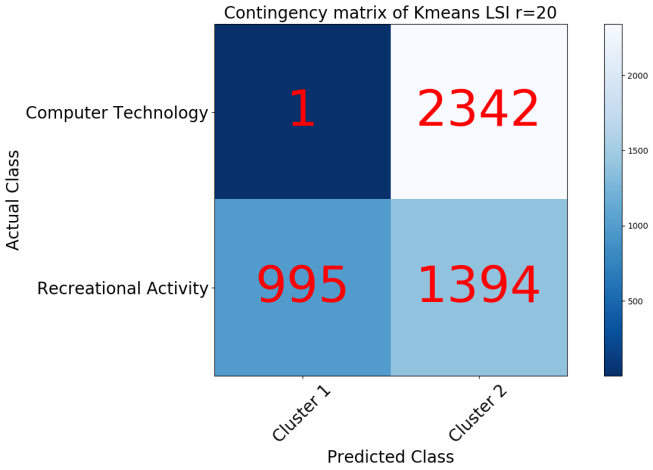
(d) Confusion matrix with r = 5



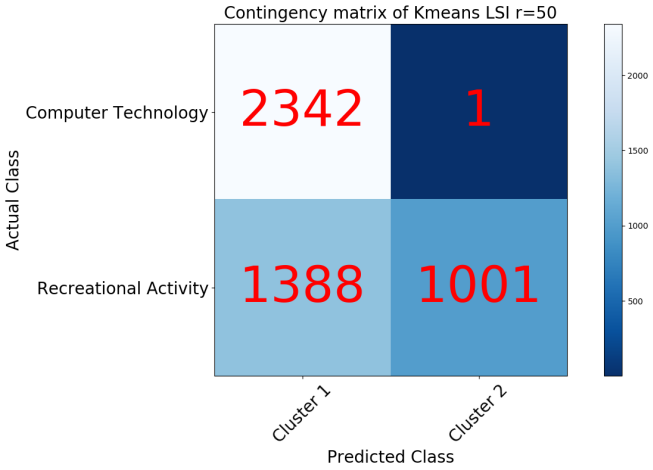
(e) Confusion matrix with r = 10



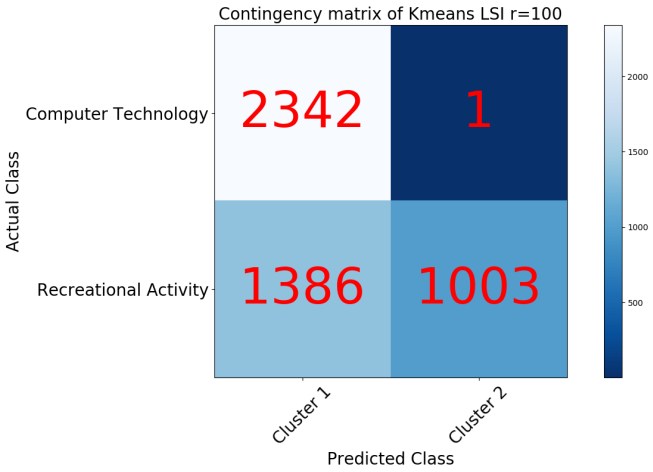
(f) Confusion matrix with r = 20



(g) Confusion matrix with r = 50



(h) Confusion matrix with r = 100



(i) Confusion matrix with r = 300

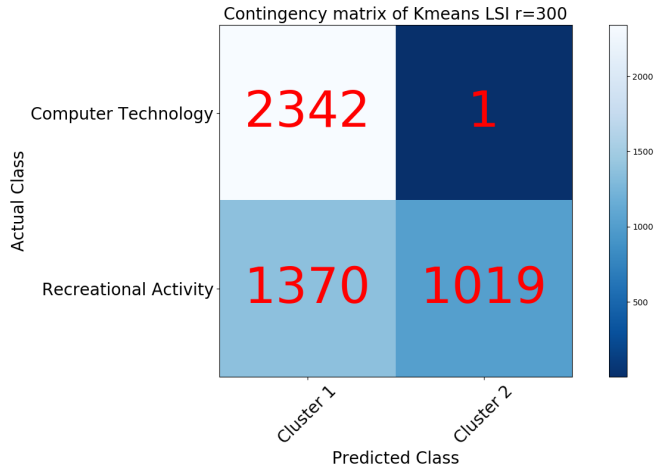
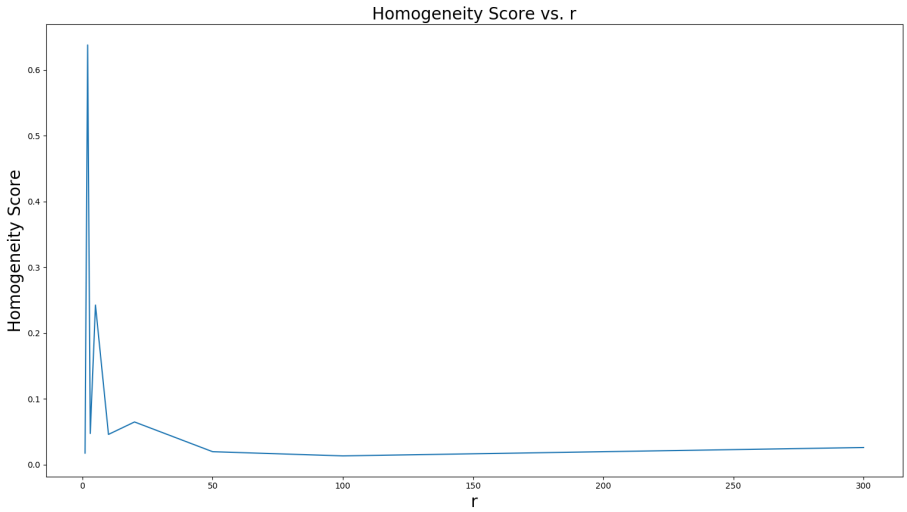


Table 3.2 NMF method, the 5 measure scores with r = {1,2,3, 5, 10, 20, 50, 100, 300}

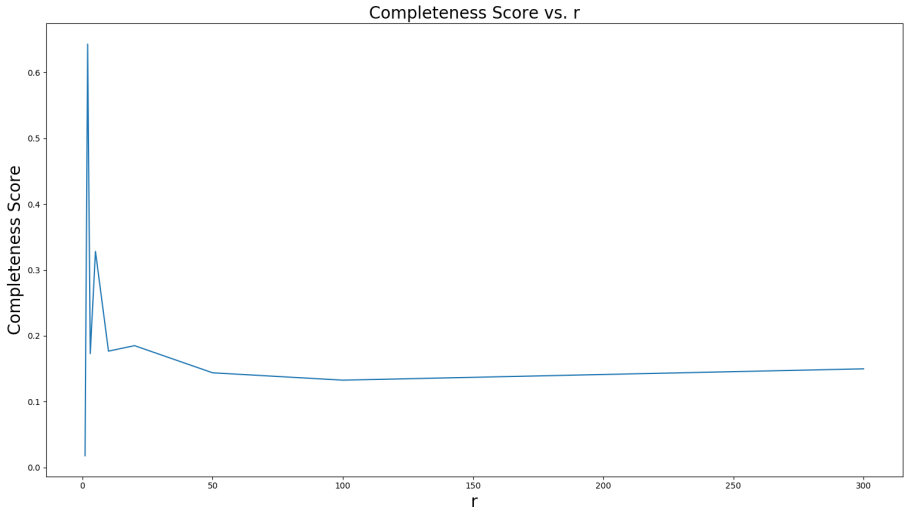
r	Homogeneity score	Completeness score	V-measure score	adjusted rand index	adjusted mutual info score
1	0.017	0.018	0.018	0.023	0.012
2	**0.638**	**0.643**	**0.640**	**0.726**	**0.442**
3	0.047	0.173	0.074	0.010	0.033
5	0.242	0.328	0.279	0.165	0.168
10	0.046	0.177	0.073	0.009	0.032
20	0.065	0.185	0.096	0.019	0.045
50	0.020	0.144	0.034	0.002	0.014
100	0.013	0.133	0.024	0.001	0.009
300	0.026	0.150	0.044	0.002	0.018

Figure 3.4 LSI method, the 5 measure scores vs r

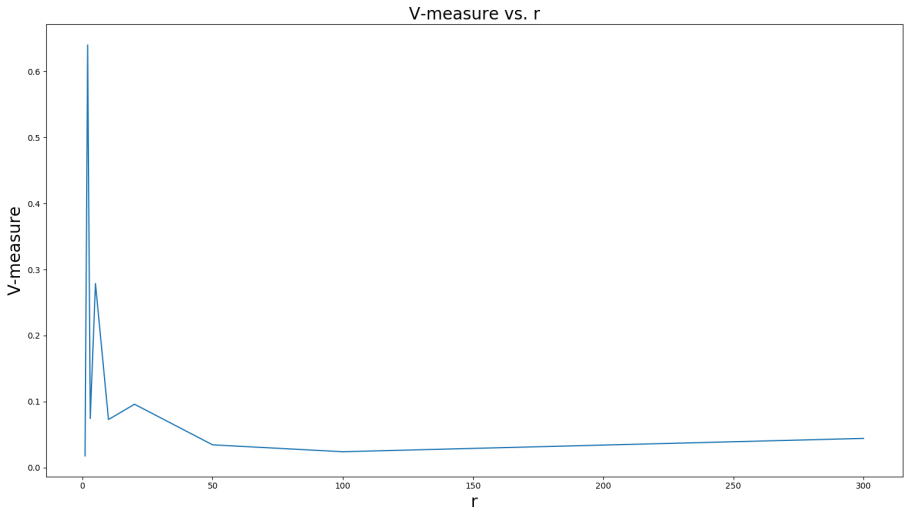
(a) Homogeneity score vs r



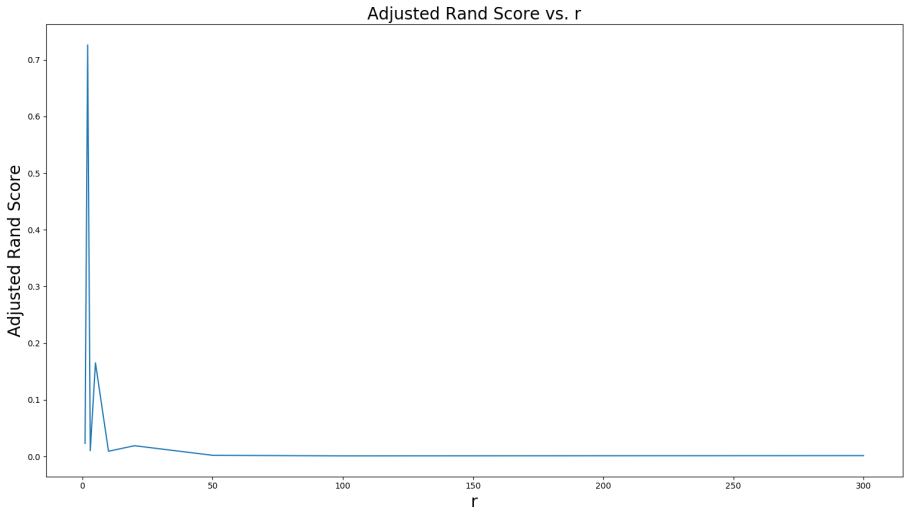
(b) Completeness score vs r



(c) V –measure score vs r



(d) Adjusted rand index vs r



(e) Adjusted mutual info score vs r

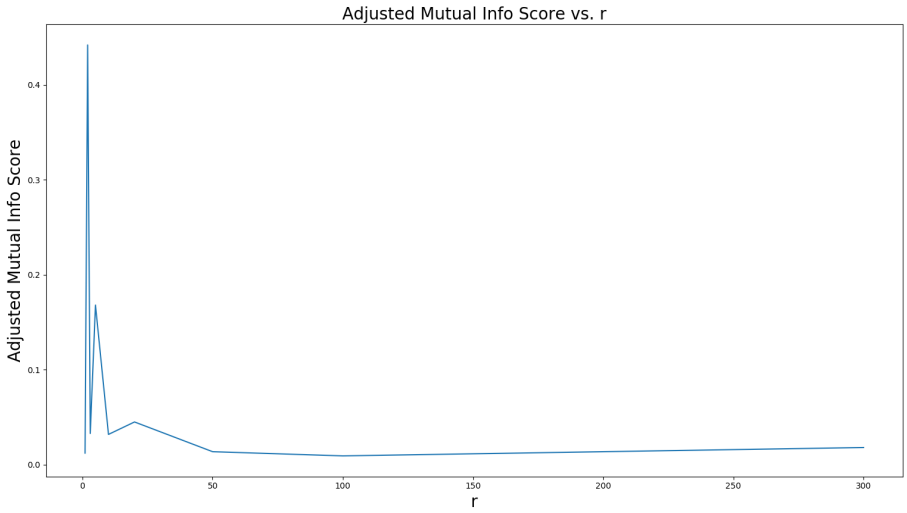
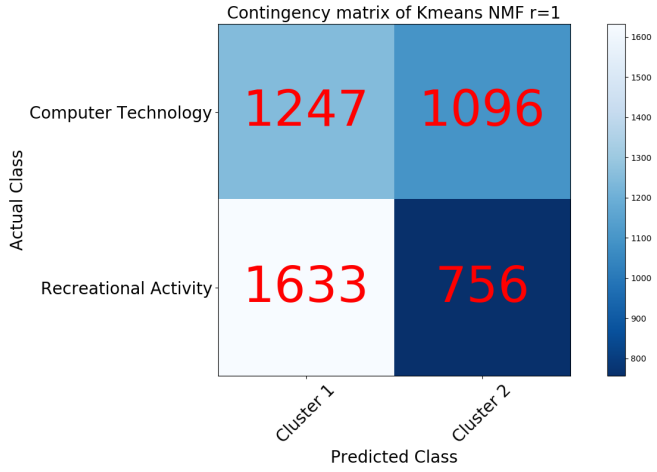
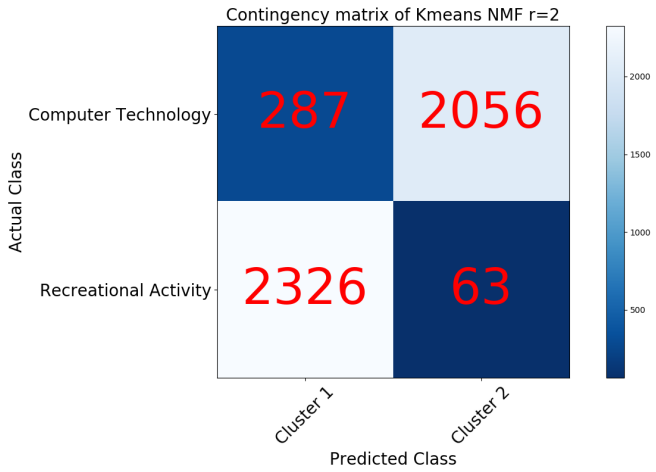


Figure 3.5 NMF method, confusion matrix with r = {1, 2, 3, 5, 10, 20, 50, 100, 300}

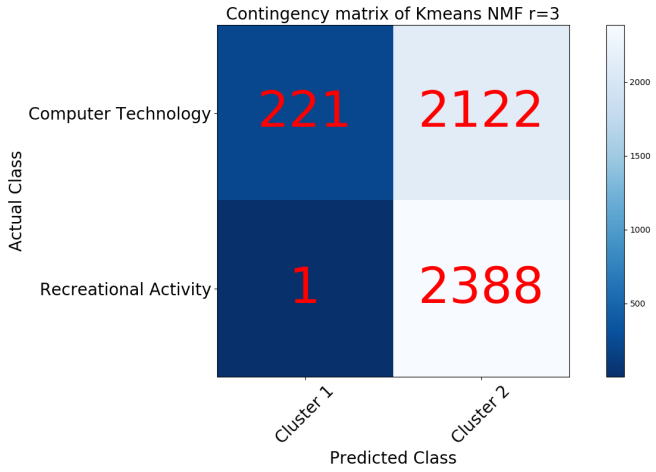
(a) Confusion matrix with r = 1



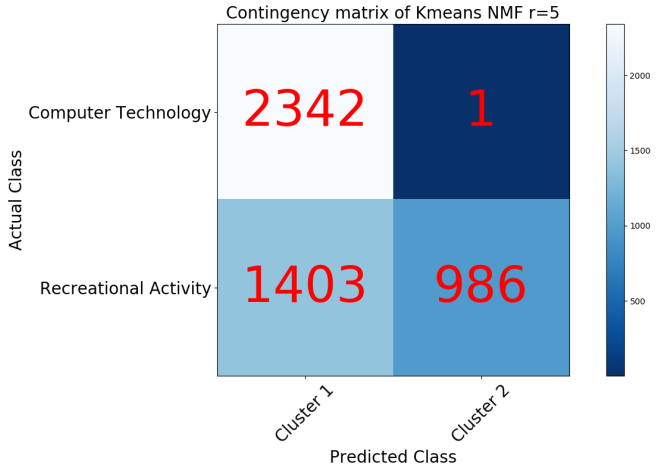
(b) Confusion matrix with r = 2



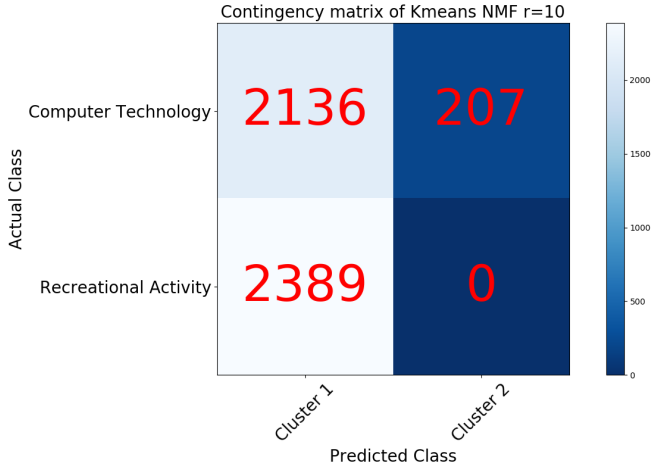
(c) Confusion matrix with r = 3



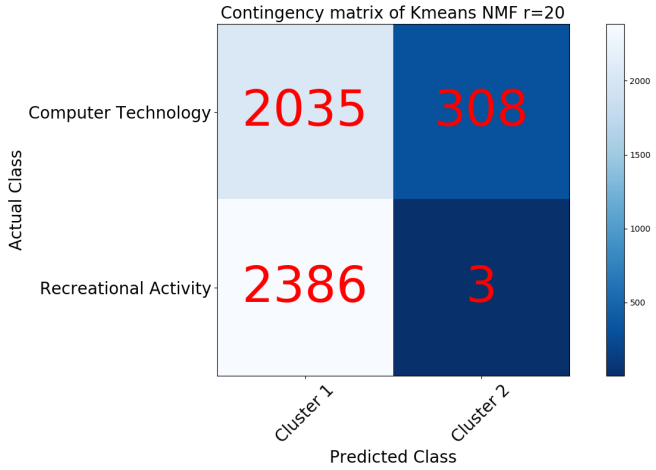
(d) Confusion matrix with r = 5



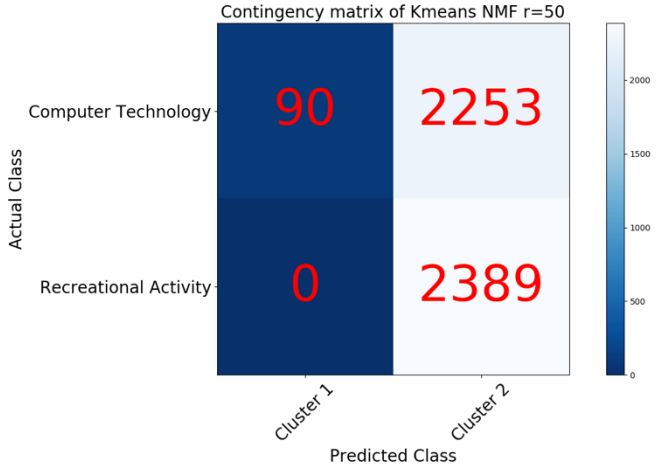
(e) Confusion matrix with r = 10



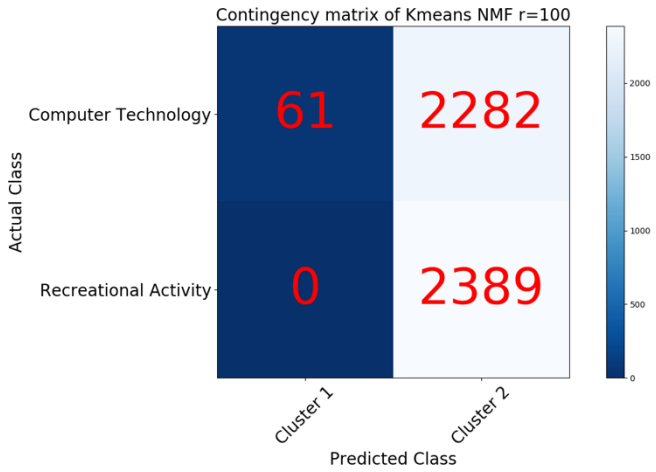
(f) Confusion matrix with r = 20



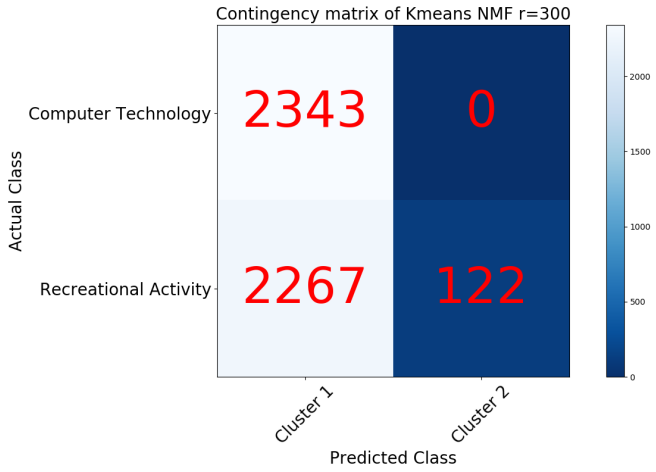
(g) Confusion matrix with r = 50



(h) Confusion matrix with r = 100



(i) Confusion matrix with r = 300



Problem 4. Performance Visualization

(a) To show the performance of our clustering in the previous part, we project our best results to 2 dimensional space.

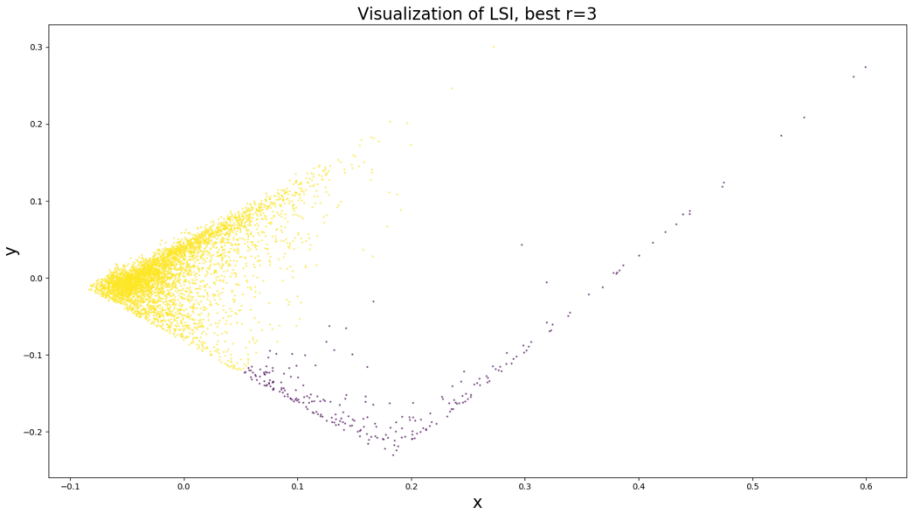
Result:

In below figure 4.2, in both LSI and NMF, we can observe that in 2 dimensional space, there are two clusters: in LSI, one cluster is bottom right, and the other is top left, in NMF, one cluster is on the right and the other is on the left. In both cases, the two clusters are separated by a clear boundary, which means we can classify our data set into this 2 dimension. We can also prove that the result is

satisfiable with the Table 3.1 and Table 3.2 above, since the five measure scores are greatly higher than those without dimension reduction in problem 2.

Figure 4.1 Visualization of the best clustering performance of LSI and NMF

(i) LSI with r = 3



(ii) NMF with r = 2

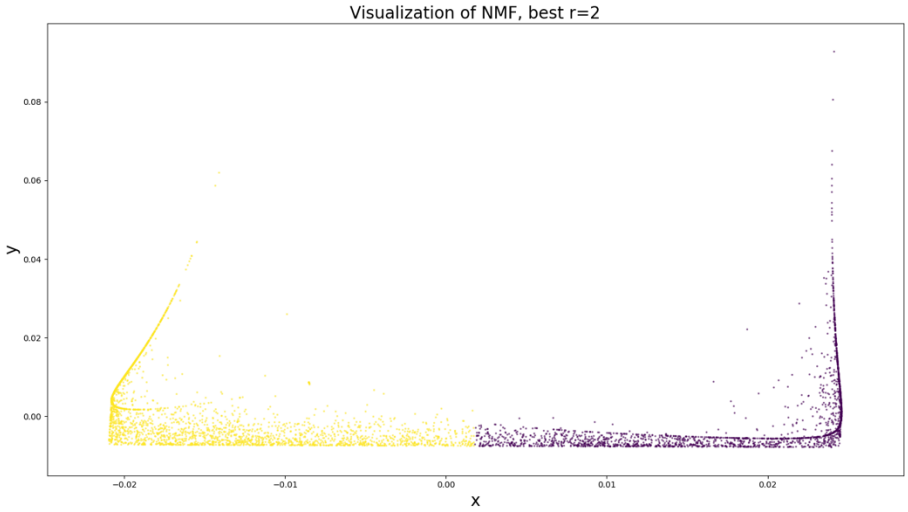
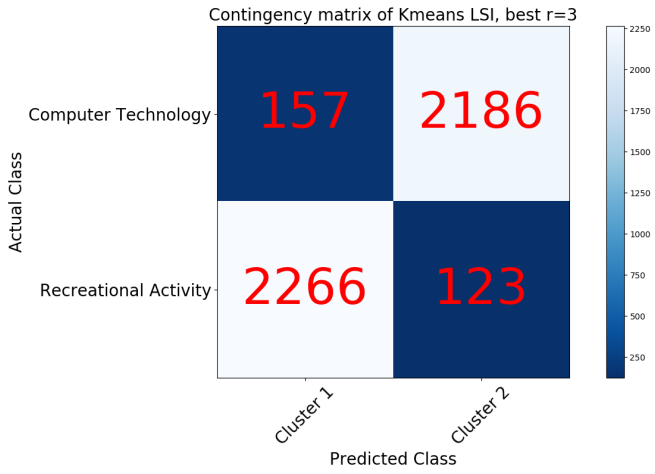
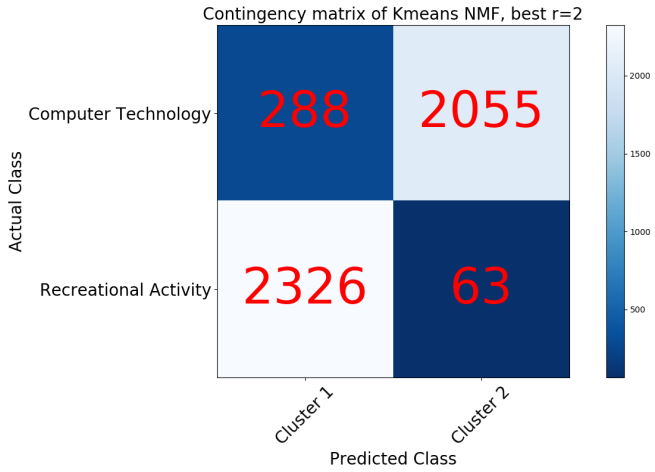


Figure 4.2 Confusion matrix of the best clustering performance of LSI and NMF

(i) Confusion matrix of LSI with r = 3



(ii) Confusion matrix of NMF with r = 2



(b) Here we try three methods to improve our clustering performance. We still use the same r as part (a).

(b.1) We do normalize on the reduced dimension features to let each feature has unit variance. Normalization is used to standardize the range of features. We want to normalized the feature vectors due to the robustness of raw data. The result is summarized and shown below.

(b.2) We apply non-linear transformation to the data vectors on NMF. Here we use logarithm transformation.

(b.3) We combine both b.1 and b.2 transformations on NMF-reduced data(with different order)

Result:

To solve this problem, we apply StandardScalar function to do the normalization at the first step. According to Table 4.1, we can find that with LSI, the results after normalization become worse. According to Table 4.2, we can find with NMF, the results after normalization become better. The improvement with NMF is due to the fact that normalization eliminates the scale among different features. Hence, it leads different features to have the same unit, so that the weights of impacts for features become the same.

Then, we need to perform the logarithm transformation on NMF. We can find that the logarithm operation alone increases the result because this non linear algorithm enlarges the distances between different clusters. In this way, data from different clusters can be more obviously separated.

Finally, for results from combining both normalization and logarithm transformation on NMF, we find that the final outcomes are very similar to that with normalization alone. That is because normalization can have a much larger influence on clustering data set than logarithm operation.

Similarly, we visualize the data with 2 dimensional plot through Fig 4.5.

Question: Can you justify why logarithm transformation may increase the clustering results?

We believe that by applying non-linear transformation(logarithm), the feature would be closer to Gaussian distribution, which would be close to the assumption of k-means clustering dataset. Therefore, we can get a better clustering result.

Table 4.1 LSI with r= 3, the 5 measure scores of different methods on features

method	Homogeneity score	Completeness score	V-measure score	Adjusted rand index	Adjusted mutual info score
Original	0.676	0.676	0.676	0.777	0.469
Normalized	0.286	0.308	0.296	0.324	0.286

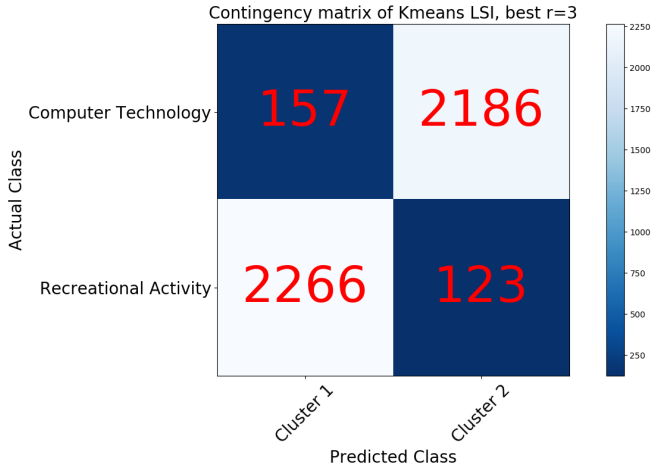
Table 4.2 NMF with r= 2, the 5 measure scores of different methods applied on features

method	Homogeneity score	Completeness score	V-measure score	Adjusted rand index	Adjusted mutual info score
Original	0.638	0.643	0.640	0.726	0.638
Normalized	0.712	0.711	0.711	0.808	0.711

Logarithm	0.648	0.653	0.650	0.738	0.648
Normalized -> log	0.709	0.709	0.709	0.806	0.709
Log -> Normalized	0.711	0.711	0.711	0.807	0.711

Figure 4.3 Confusion matrix of LSI with r= 3 with different method applied on features

(i)LSI with r= 3, Original



(ii) LSI with r= 3, Normalized

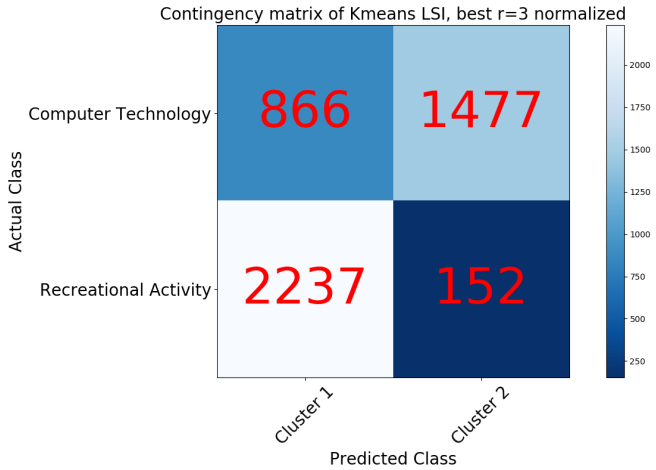
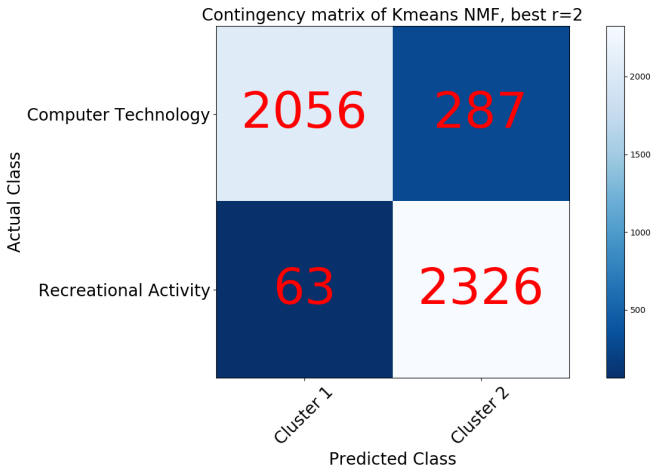
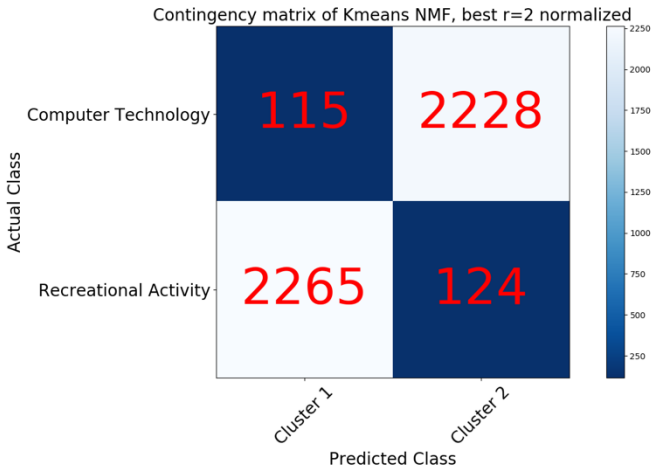


Figure 4.4 Confusion matrix of NMF with r= 2, with different method applied on features

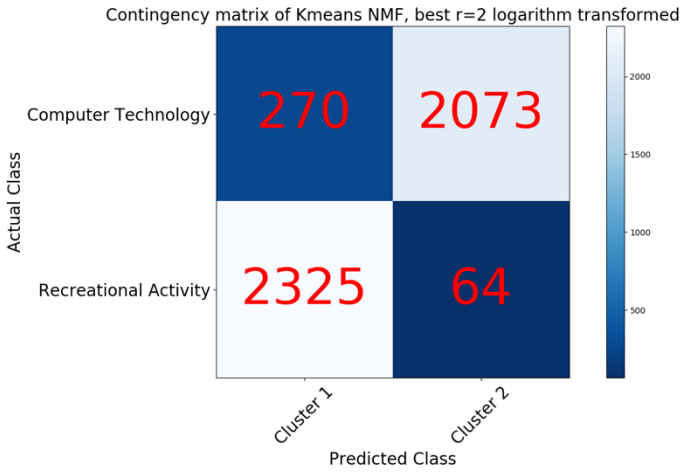
(i)NMF with r= 2, Original



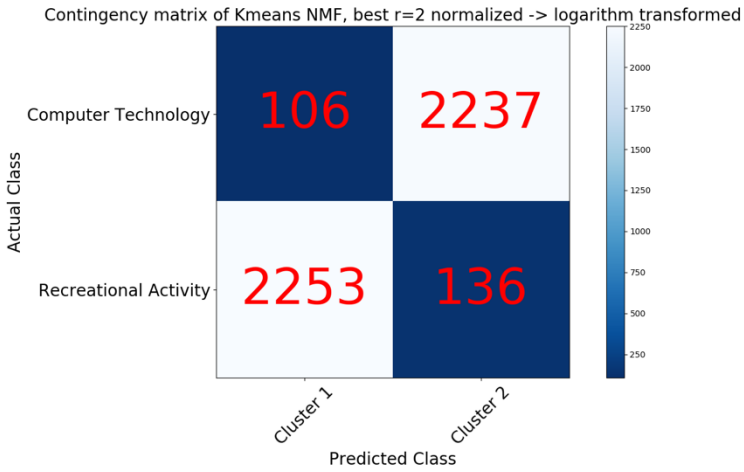
(ii) NMF with r= 2, Normalized



(iii) NMF with r= 2, Logarithm



(iv) NMF with r= 2, Normalized -> logarithm



(v) NMF with r= 2, Logarithm -> Normalized

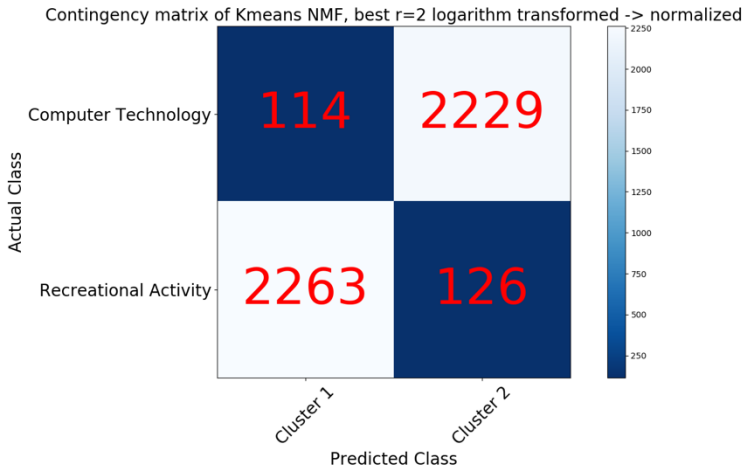
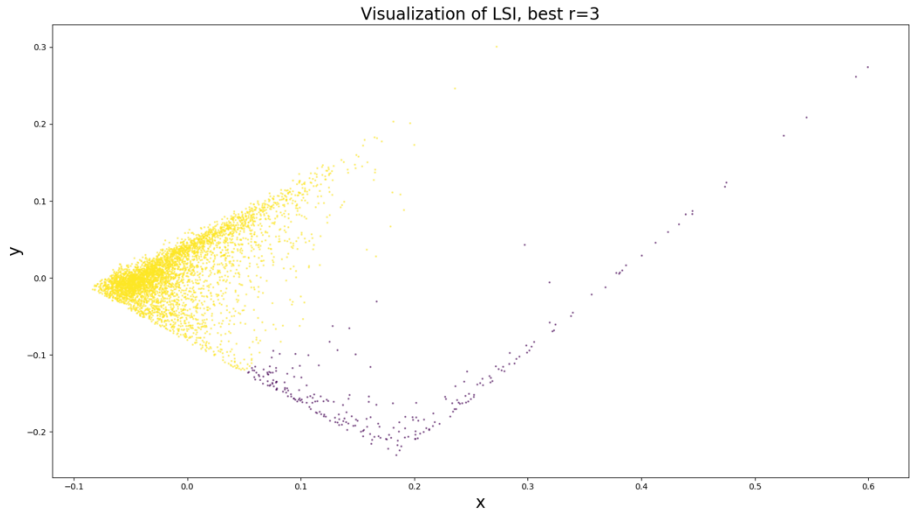


Figure 4.5 Visualization of LSI with r= 3 with different method applied on features

(i)LSI with r= 3, Original



(ii) LSI with r= 3, Normalized

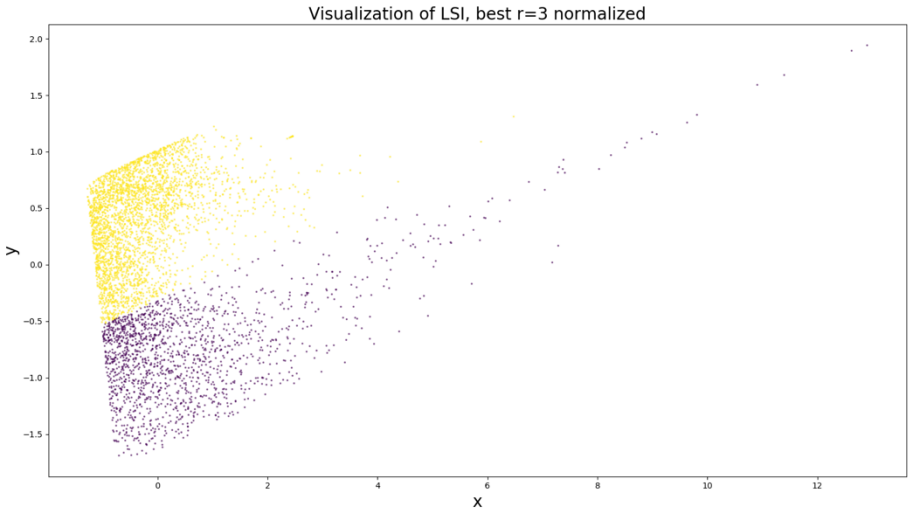
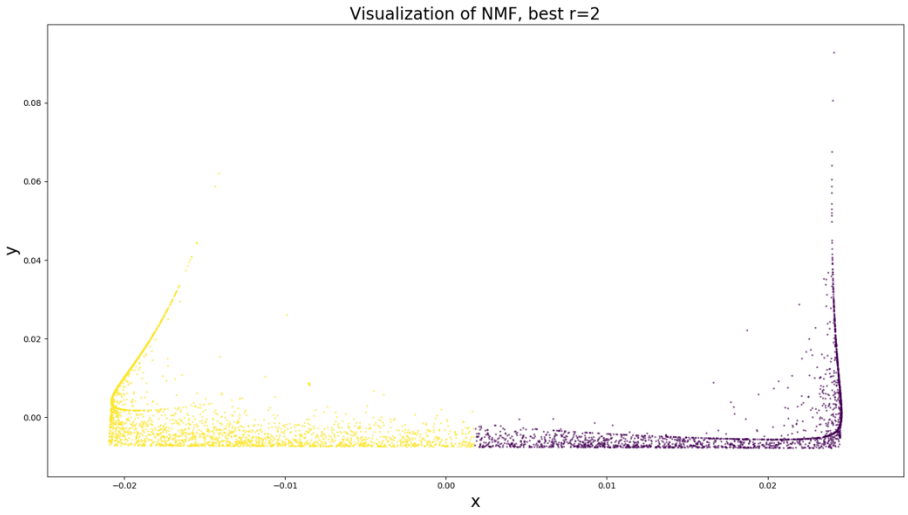
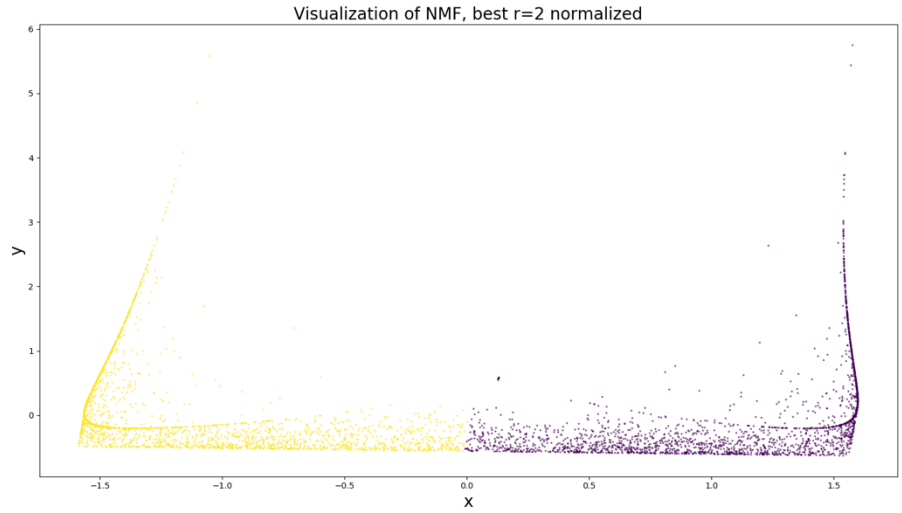


Figure 4.6 Visualization of NMF with r = 2 with different method applied on features

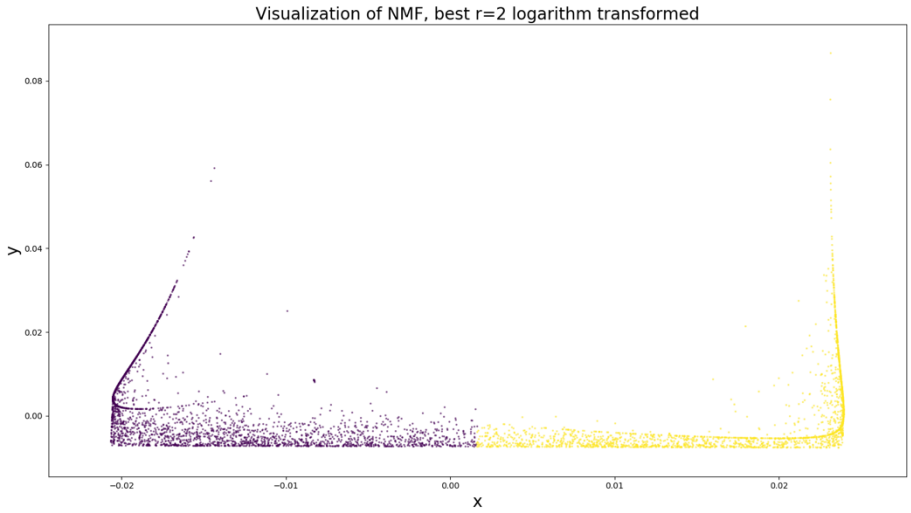
(i)NMF with r = 2, Original



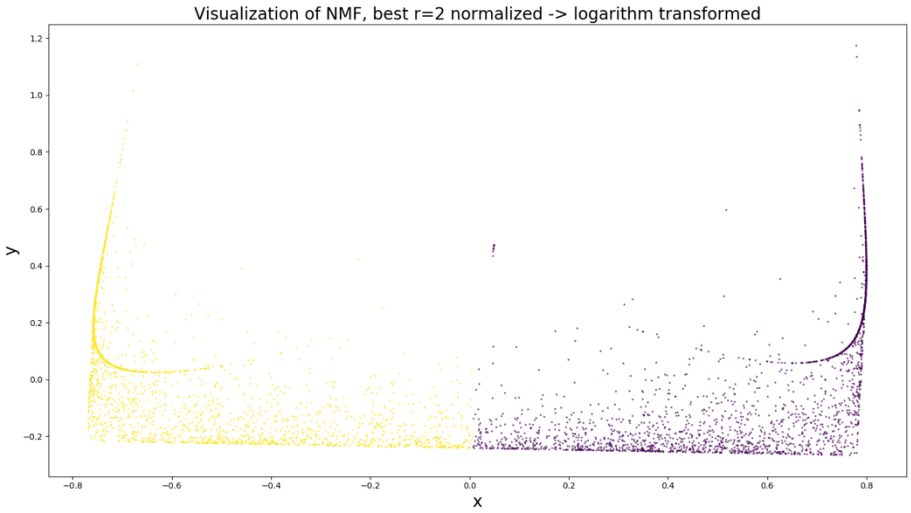
(ii) NMF with r = 2, Normalized



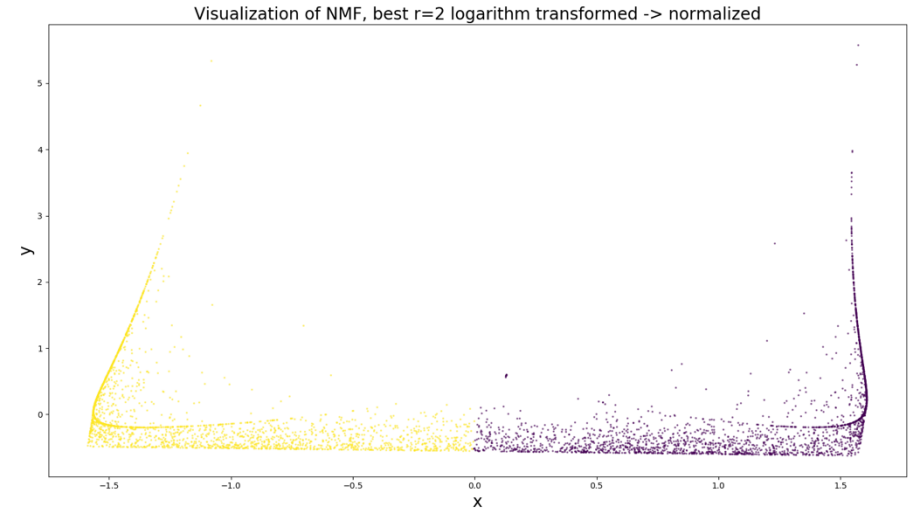
(iii) NMF with r= 2, Logarithm



(iv) NMF with r= 2, Normalized -> logarithm



(v) NMF with r= 2, Logarithm -> Normalized



Problem 5. Expand Dataset into 20 categories

In this part, we perform k-means on 20 original categories rather than 8 categories so we repeat the previous four parts. We have to find the optimal dimension k which gives us the highest purity. Similar to the previous part, we use homogeneity score, completeness score, V-measure, adjusted rand index and adjusted mutual info score to evaluate the clustering performance.

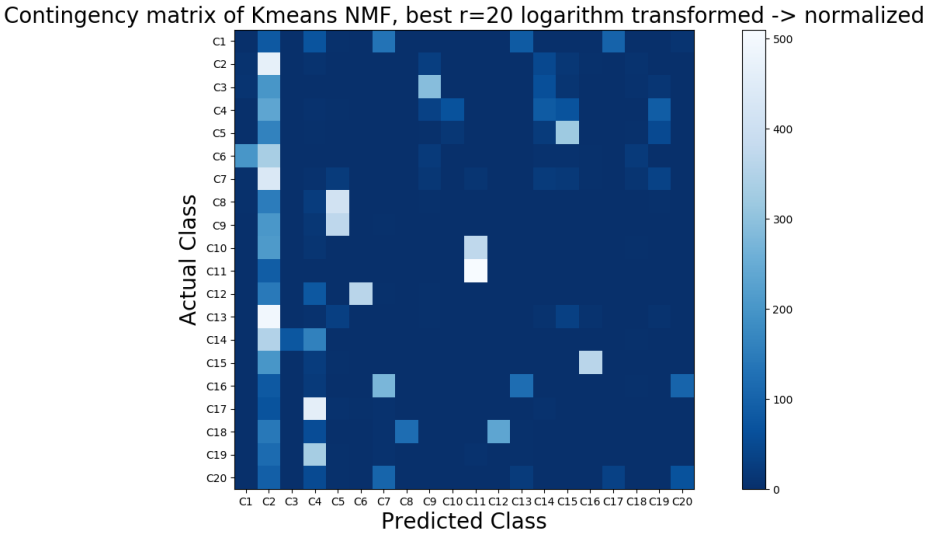
With this approach, we found that the homogeneity score grows as k grows until k>>20, but in this situation, most cluster contains the labels which do not exist in ground truth label set. It does not make sense, so we selected k = 20 which intuitively make sense, and also the performance is better than small k's because small k means a large cluster, for which we do not know which subclass form a superclass to match our cluster.

Result:
the best performance we got is k = 20 and apply logarithm transformed then normalized, and the 5 measure scores and confusion matrix is shown in table 5.1 and figure 5.1

Table 5.1 The 5 measure scores NMF with k = 20, apply logarithm transformed then normalized

Metric	Score
homogeneity score	0.423
completeness score	0.551
V-measure	0.478
adjusted rand index	0.143
adjusted mutual info score	0.408

Figure 5.1 Confusion matrix of NMF with k = 20, with logarithm transformed then normalized



the detail confusion matrix is as follows

```
[[  0  80   0  70   2   0 133   0   0   0   0   1  84   0   0   0 100   1   0   9]
 [  7 468   0   6   1   0   0   0  29   0   0   0   1  49  14   2   0   7   0   0]
 [  8 203   0   0   1   0   0   0 290   1   0   0   0  60  11   0   0   4  13   0]
 [  1 236   0   4   3   1   1   0  34  67   0   0   0  85  68   0   0   1  89   0]
 [  1 162   0   2   0   0   0   0   2  15   0   0   0  23 320   0   0   2  51   0]
 [202 336   0   1   0   1   0   0  20   0   0   0   0   5   4   3   0  21   0   0]
 [  3 435   0   5  23   1   0   0  15   1  10   0   1  23  18   2   0  11  37   0]
 [  3 150   0  24 412   0   0   0   2   0   0   0   0   0   0   1   0   0   2   0]
 [  1 205   0  15 370   0   2   0   1   0   1   0   1   0   1   0   0   0   1   0]
 [  0 211   0  11   0   0   1   0   0   0 372   0   0   0   0   0   0   2   0   0]
 [  0  88   0   1   0   0   0   0   0   0 510   0   0   0   1   0   0   0   0   0]
 [  1 145   0  78   0 362   3   0   3   0   0   0   0   0   1   1   0   0   1   0]
 [  1 496   0   4  31   0   1   0   2   1   0   0   0   7  33   7   0   1   7   0]
 [  0 352  76 160   1   0   1   0   1   0   0   0   0   0   1   0   0   2   0   0]
 [  0 203   0  25   2   0   1   0   0   0   0   0   0   0   0 361   0   1   0   0]
```

[0 81 0 20 0 0 273 0 0 0 0 0 120 0 0 0 0 2 0 103]
[0 69 0 462 4 2 4 0 0 0 0 0 0 5 0 0 0 0 0 0]
[0 143 0 56 1 0 4 121 0 0 0 236 2 0 0 0 0 1 0 0]
[1 116 0 331 3 1 4 0 0 0 5 0 2 0 0 0 1 1 0 0]
[0 93 0 52 2 0 105 0 0 0 0 0 23 0 0 0 35 0 0 67]]

Figure 5.2 Visualization of NMF with k = 20 with logarithm transformed then normalized