EE 219 Project 2: Clustering

Jinxi Zou 605036454

Cheng Ma 105033453

I. Introduction:

In this project, we are going to find the proper representation of the data so that it can efficiently cluster. We will perform K-means clustering on different dataset and evaluate their performance to observe how K-means algorithm works on clustering. We will also try some preprocessing methods and try to improve clustering performance.

The dataset we used is "20 Newsgrooup", we would use two groups of dataset from it. One is shown in Table 1, the other is shown in Table 2.

Table 1: eight subset group

Class 1	Class 2
comp.graphics	rec.autos
comp.os.ms-windows.misc	rec.motorcycles
comp.sys.ibm.pc.hardware	rec.sport.baseball
comp.sys.mac.hardware	rec.sport.hockey

Table 2: twenty subset group

comp.graphics	sci.crypt
comp.os.ms-windows.misc	sci.electronics
comp.sys.ibm.pc.hardware	sci.med
comp.sys.mac.hardware	sci.space
comp.windows.x	talk.politics.misc
misc.forsale	talk.politics.guns
rec.autos	talk.politics.mideast
rec.motorcycles	talk.religion.misc
rec.sport.baseball	alt.atheism
rec.sport.hockey	soc.religion.christia

II. Problem 1:

The first thing to do with texture dataset is to transform it to a proper representation. Like in Project 1, we used TF-IDF matrix to represent the first 8 subset group using min_df = 3 and exclude Engilish stop words. The dimension of TF-IDF matrix we got from training set of eight subset group is in Table 3.

Table 3: Dimension of TF-IDF matrix

row	column
4732	20295

III. Problem 2:

We applied K-means algorithm to cluster eight subset to two groups and compared clustering labels and corresponding true labels and measured their contingency matrix, homogeneity score, completeness score, V-measure, adjusted Rand score and adjusted mutual info score. The reason why we used five measure score to observe the performance is listed below:

Homogeneity represent how purity, if each cluster is from only one class, homogeneity will be 1.

Completeness represent if all data of a class are assigned to the same cluster, if so, it will be 1.

V-measure is the harmonic average if homogeneity and completeness.

The adjusted Rand Index computes similarity between clustering labels and ground true labels.

The adjusted mutual information score measures the mutual information between cluster distribution and ground true label distribution.

After applying K-means algorithm to cluster the data TF-IDF matrix we got from problem 1, the 8 classes were given two labels. We got the contingency matrix in Table 4. We computed the five measure scores to inspect performance of clustering. The results are listed in Table 5.

Table 4: Contingency matrix

2	2341
992	1397

Table 5: Five measures of clustering

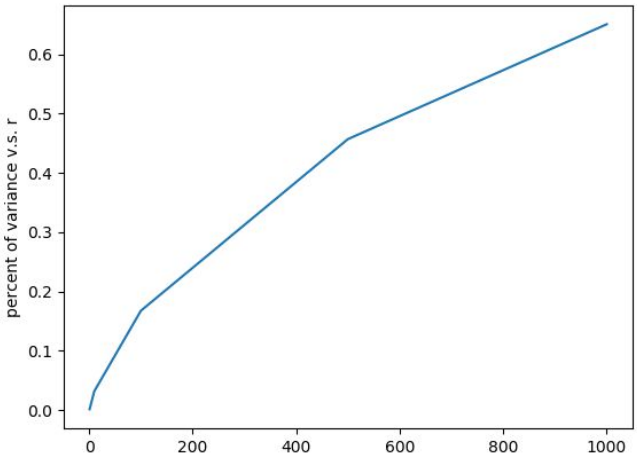
Homogeneity	Completeness	V-measure	The adjusted Rand Index	The adjusted mutual information score
0.242	0.327	0.278	0.167	0.242

From Table 4 and 5, it's obvious that performance of K-means clustering was not very good. This may because the dimension of matrix was too big, so Euclidean distance wouldn't be a good metric.

VI. Problem 3:

In this problem, we used two different dimension reduction methods, Latent Semantic Indexing and Non-negative Matrix Factorization. To see how many singular values are significant, we recorded ratio of the variance of the original data of LSI. The plot of the percent of variance of the top r LSI are shown in Figure 1.

Figure 1: Ratio of variance of the original data by LSI



From Figure 1, we knew that top 1000 principle components have include over a half of total variance, which mean that clustering would also be accurate enough with dimension reduced DF-IDF matrix.

Then we tried TruncatedSVD and NMF to do dimension reduction to r principle components. We set r to 1, 2, 3, 5, 10, 20, 50, 100 and 300 and plotted five measure scores to find the r with the best performance. The result of SVD is shown in Figure 2, The performance result of NMF is shown in Figure 3.

Figure 2: performance of Clustering with SVD

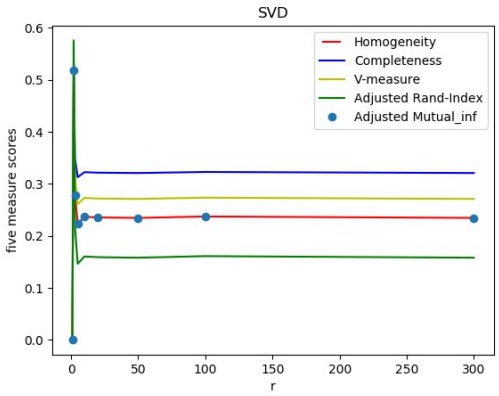
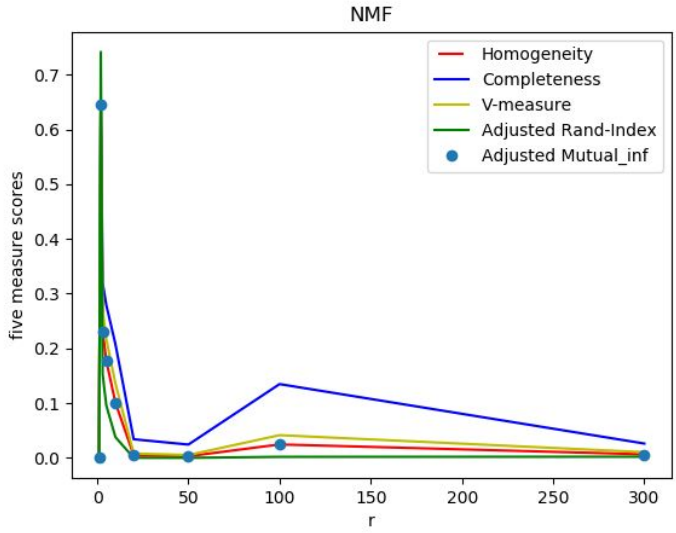


Figure 3: performance of Clustering with NMF



From Figure 2 and Figure 3, it's obvious that lines peak at 2, so r with best performance for both SVD and NMF are 2. The reason why performance decreased when r was increasing is that Euclidean distance would not be a good metric to represent distance when dimensions are high. In addition, as r increase, more variance which means more information are included, however, news group are formed by words, the more principle components are included, the more unimportant words are considered may reduce the accuracy of clustering.

V. Problem 4:

From problem 3, we chose r to be 2. Then we visualize the clustering results of SVD and NMF, which are shown in Figure 4 and Figure 5.

Figure 4: Visualization of Clustering with NMF

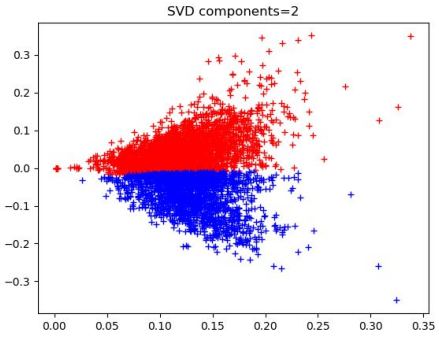
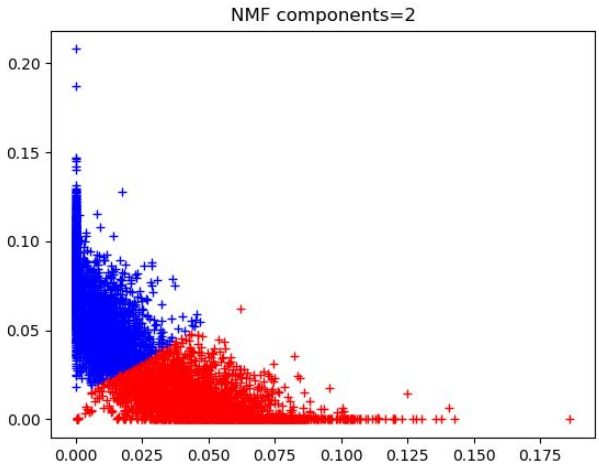


Figure 5: Visualization of Clustering with SVD



Both in Figure 4 and Figure 5, all data were roughly equally assigned to two clusters, which correlated to the ground truth that the number of original data with each label are roughly equal.

Then we tried three different preprocessing methods to improve clustering performance, normalizing features, non-linear transformation and their combination. The result visualization of Normalized SVD clustering is shown in Figure 6, its contingency matrix and five measure scores are shown in Table 6 and 7.

Figure 6: Visualization of Clustering with SVD after normalizing features

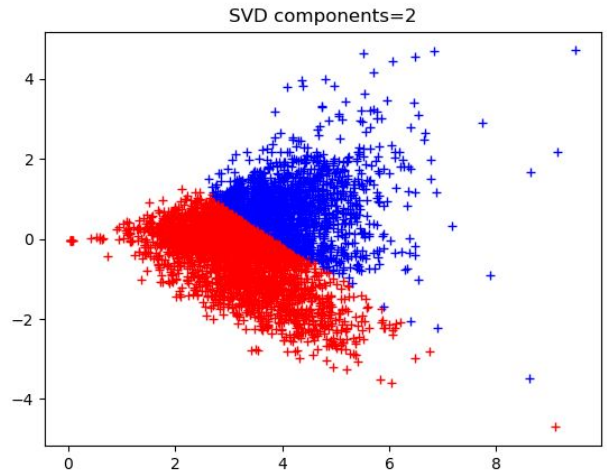


Table 6: Contingency matrix of Clustering with SVD after normalizing features

1033	1310
2187	202

Table 7: Five measures of clustering of Clustering with SVD after normalizing features

Homogeneity	Completeness	V-measure	The adjusted Rand Index	The adjusted mutual information score
0.203	0.224	0.213	0.228	0.203

From Figure 6, table 6 and table 7, we know that normalization wouldn't improve and even damage performance of clustering with reduced dimension matrix by SVD.

The result visualization of Normalized NMF clustering is shown in Figure 7, its contingency matrix and five measure scores are shown in Table 8 and 9.

Figure 7: Visualization of Clustering with NMF after normalizing features

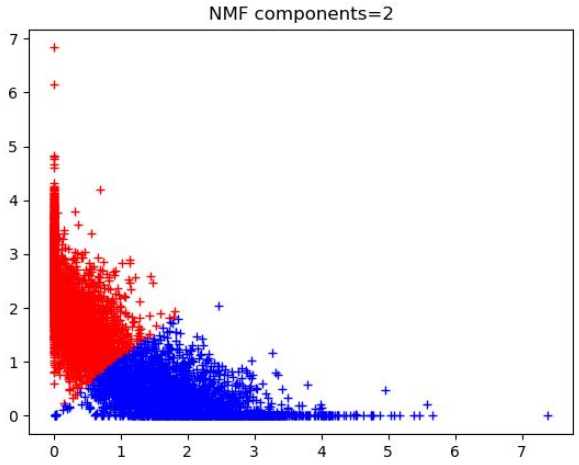


Table 8: Contingency matrix of Clustering with NMF after normalizing features

159	2184
2301	88

Table 9: Five measures of clustering of Clustering with NMF after normalizing features

Homogeneity	Completeness	V-measure	The adjusted Rand Index	The adjusted mutual information score
0.707	0.708	0.707	0.702	0.707

From Figure 7, Table 8 and Table 9, we know that normalize features can improve a little performance of clustering with NMF. This is because all features are normalized so that features with big variance before wouldn't influence the clustering a lot.

After applying non-linear transformation to the data vector with NMF, the result clustering is shown in Figure 8, Table 10 and Table 11. To avoid zero term, we used log one plus term as the non-linear transformation function.

Figure 8: Visualization of Clustering with NMF after non-linear transformation

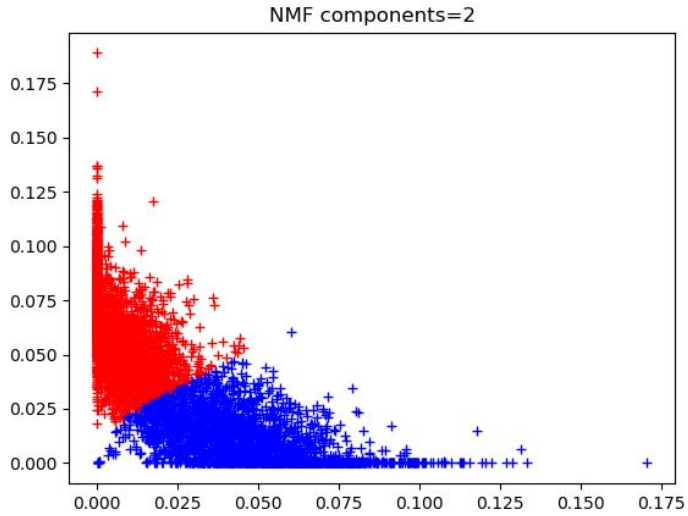


Table 8: Contingency matrix of Clustering with NMF after non-linear transformation

125	2218
2252	137

Table 9: Five measures of clustering of Clustering with NMF after non-linear transformation

Homogeneity	Completeness	V-measure	The adjusted Rand Index	The adjusted mutual information score
0.691	0.691	0.691	0.791	0.691

From Figure 8, Table 8 and Table 9, there isn't an obviously difference between clustering using log transformation and not. This is because previous data is two dimensions and don't have too many outliers, so constrain variance or smooth out unusual term wouldn't improve performance a lot. For some dataset which have terms with big differences and many outliers, log transformation will help to reduce the difference and makes them comparable, so clustering results would be increased.

Then I tried to combine normalizing transformation and log transformation and then also exchange them to look the clustering performance. The results are shown in Figure 9, 10 and Table 10, 11 and Table 12, 13.

Figure 9: Visualization of Clustering with NMF after normalizing transformation then Log transformation

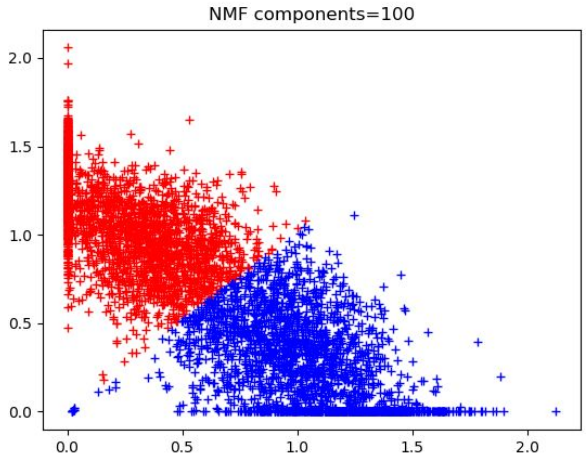


Table 10: Contingency matrix of Clustering with NMF after normalizing transformation then Log transformation

160	2183
2302	87

Table 11: Five measures of clustering of Clustering with NMF after normalizing transformation then Log transformation

Homogeneity	Completeness	V-measure	The adjusted Rand Index	The adjusted mutual information score
0.707	0.708	0.707	0.802	0.707

Figure 9: Visualization of Clustering with NMF after log transformation then normalizing transformation

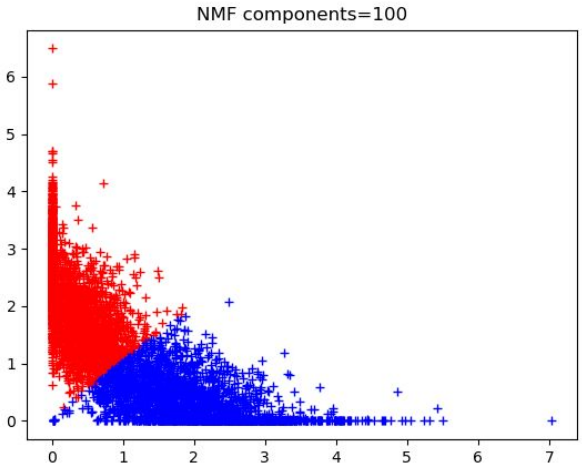


Table 10: Contingency matrix of Clustering with NMF after log transformation then normalizing transformation

159	2184
2300	89

Table 11: Five measures of clustering of Clustering with NMF after log transformation then normalizing transformation

Homogeneity	Completeness	V-measure	The adjusted Rand Index	The adjusted mutual information score
0.706	0.707	0.706	0.801	0.706

The rank of non-linear transformation and log transformation didn't influence clustering results a lot. This may because our data don't have many outliers so both transformations have less influence.

V. Problem 4:

In this problem, we changed the dataset to 20 classes group and repeated the process we have done with 8 classes group. Firstly, we plotted the clustering performance respect to different dimensions to find out the best r choices for SVD and NMF, the results are shown in Figure 10 and Figure 11.

Figure 10: performance of Clustering with SVD

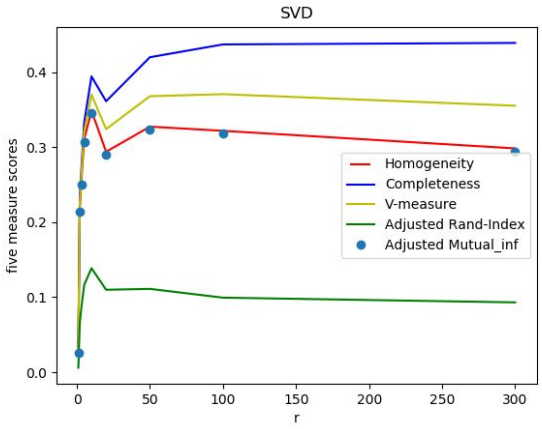
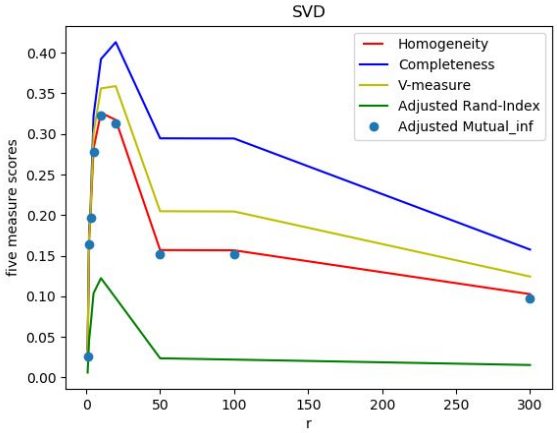


Figure 11: performance of Clustering with NMF



From Figure 10 and Figure 11, we found that the best r for SVD is r =10 and the best r for NMF is r =10. The reason why performance decreased when r was increasing is that Euclidean distance would not be a good metric to represent distance when dimensions are high.

We apply dimension reduction to dataset and got their dimension reduced TF-IDF matrix, then we explored how normalizing features and non-linear transformation would influence them. The results are shown in following figures and tables.

Figure 12: Visualization of Clustering with SVD

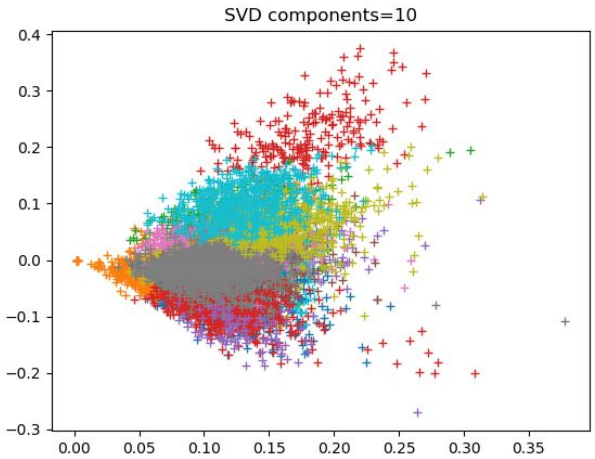


Figure 13: Visualization of Clustering with SVD after normalizing

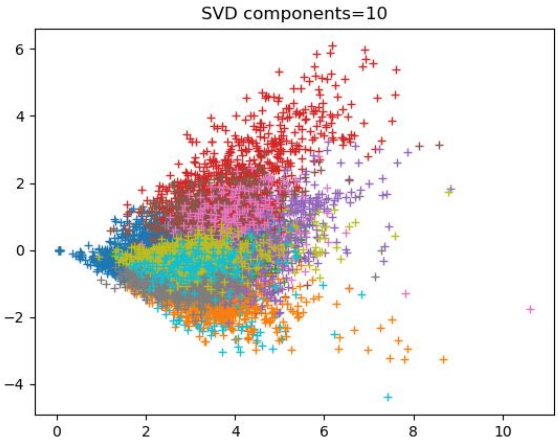


Table 12: Five measures of clustering of Clustering with SVD

Homogeneity	Completeness	V-measure	The adjusted Rand Index	The adjusted mutual information score
0.353	0.396	0.373	0.144	0.350

Table 13: Five measures of clustering of Clustering with SVD after normalizing

Homogeneity	Completeness	V-measure	The adjusted Rand Index	The adjusted mutual information score
0.338	0.381	0.358	0.130	0.334

From Table 12 and Table 13, we can't see obvious difference between normalized and non-normalized SVD clustering. This may because the dimension is too low to see any change after normalizing features and there isn't features with much larger variance.

Figure 14: Visualization of Clustering with NMF

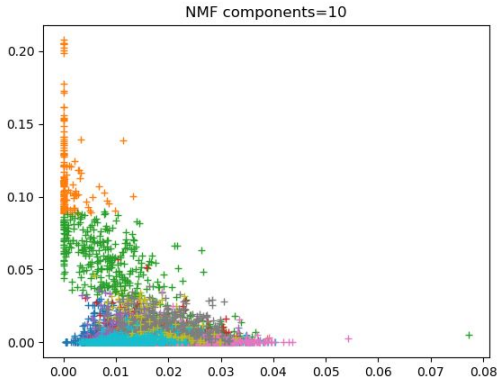


Table 14: Five measures of clustering of Clustering with NMF

Homogeneity	Completeness	V-measure	The adjusted Rand Index	The adjusted mutual information score
0.305	0.348	0.325	0.118	0.302

Figure 15: Visualization of Clustering with NMF after normalizing

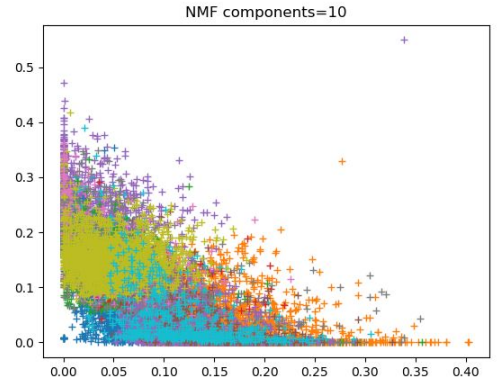


Table 15: Five measures of clustering with NMF after normalizing

Homogeneity	Completeness	V-measure	The adjusted Rand Index	The adjusted mutual information score
0.311	0.350	0.329	0.115	0.307

Figure 16: Visualization of Clustering with NMF after log transformation

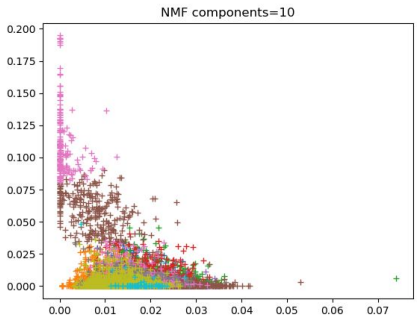


Table 16: Five measures of clustering with NMF after log transformation

Homogeneity	Completeness	V-measure	The adjusted Rand Index	The adjusted mutual information score
0.323	0.360	0.341	0.129	0.320

From Figure 15, 16 and Table 15, 16, we see that normalizing features and log transformation can both improve clustering a little, this is because normalizing can reduce the effect of features with much larger variance and log transformation can constrain the effect of outliers.

Figure 17: Visualization of Clustering with NMF log trans then normalizing

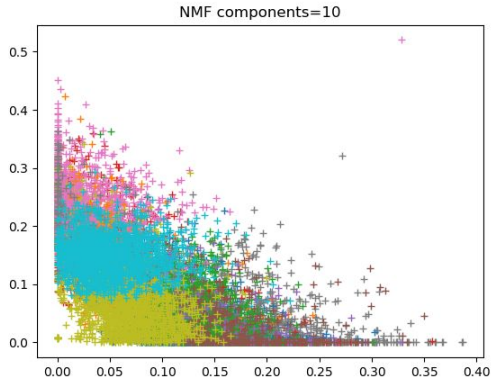


Table 17: Five measures of clustering of Clustering with NMF log trans then normalizing

Homogeneity	Completeness	V-measure	The adjusted Rand Index	The adjusted mutual information score
0.322	0.357	0.338	0.122	0.318

Figure 18: Visualization of Clustering with NMF after normalizing then log trans

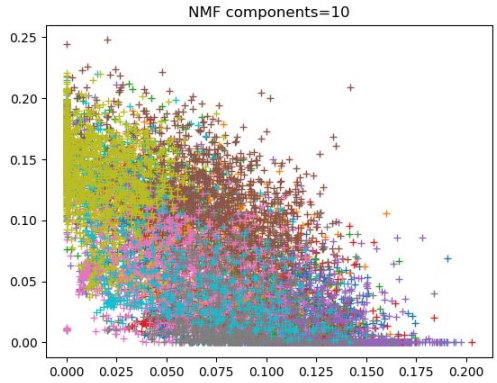


Table 18: Five measures of clustering of Clustering with NMF normalizing then log trans

Homogeneity	Completeness	V-measure	The adjusted Rand Index	The adjusted mutual information score
0.388	0.403	0.396	0.190	0.385

From Figure 17 and 18, we can see that first do normalizing and then log transformation would get a better clustering performance, this may because we can find outliers easier after applying normalizing.