EE219

Large-Scale Data Mining

Project 2

Clustering

Winter 2018

By Zeyu Jin, Xudong Li

February 11, 2018

Introduction

Clustering algorithms are unsupervised methods for finding groups of data points that have similar representations in a proper space. Clustering differs from classification in that no a priori labeling (grouping) of the data points is available. In this project, we work with the datasets provided in project1, using K-means method to make clustering.

K-means clustering method is a popular algorithm. It works mainly by minoring the distance between cluster centers and each data vector belonging to the cluster. In the project, we firstly apply the algorithm in 2-catigory 2-cluster situation and investigate the homogeneity score, the completeness score, the V-measure, the adjusted Rand score and the adjusted mutual info score in different dimension reduction functions. Then we preprocess the data by normalizing and non-linear transformation, investigating the change of various measures of purity. At last, we change the expand the target dataset to 20 categories.

Part (1)

In this part, we work with two well-separated classes, the documents into TF-IDF vectors with min df = 3, exclude the stop-words.

After transforming the documents, we get the TF-IDF array with the shape of (7882, 27768)

Part (2)

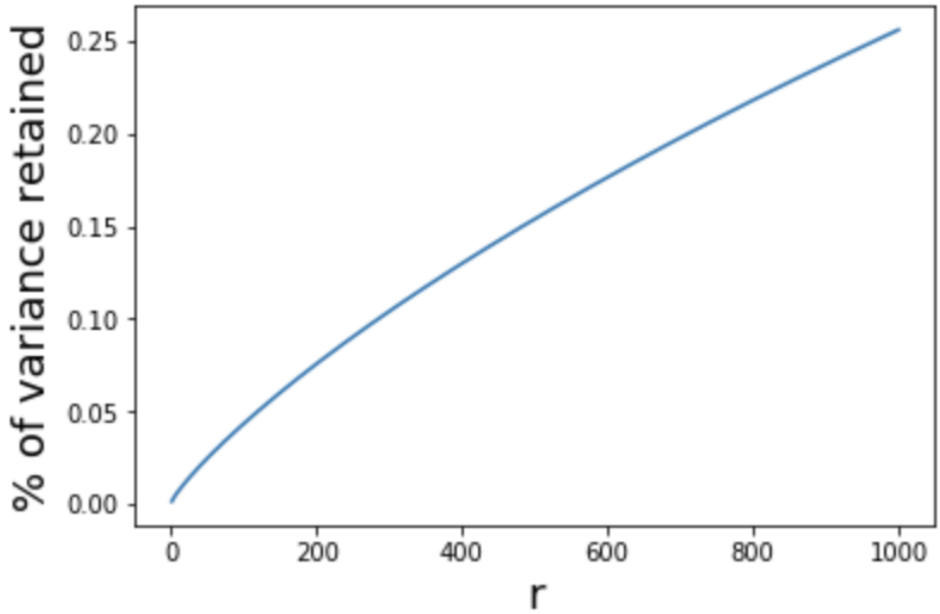
We applying K-means clustering with k = 2 when using the TF-IDF data, and get the measurements:

K-means clustering with k = 2	
contingency matrix	[[3900 3] [2265 1714]]
measures	homogeneity=0.253915 completeness=0.335755 v-measure=0.289156 adj rand index=0.180115 adj mutual info=0.253847

Part (3a)

In this part, we use Latent Semantic Indexing (LSI) and Non-negative Matrix Factorization (NMF) to decrease the dimensions and see what ratio of the variance of the original data is retained after the dimensionality reduction.

i.
The percent of variance the top r principle components can retain v.s. r, for r = 1 to 1000:



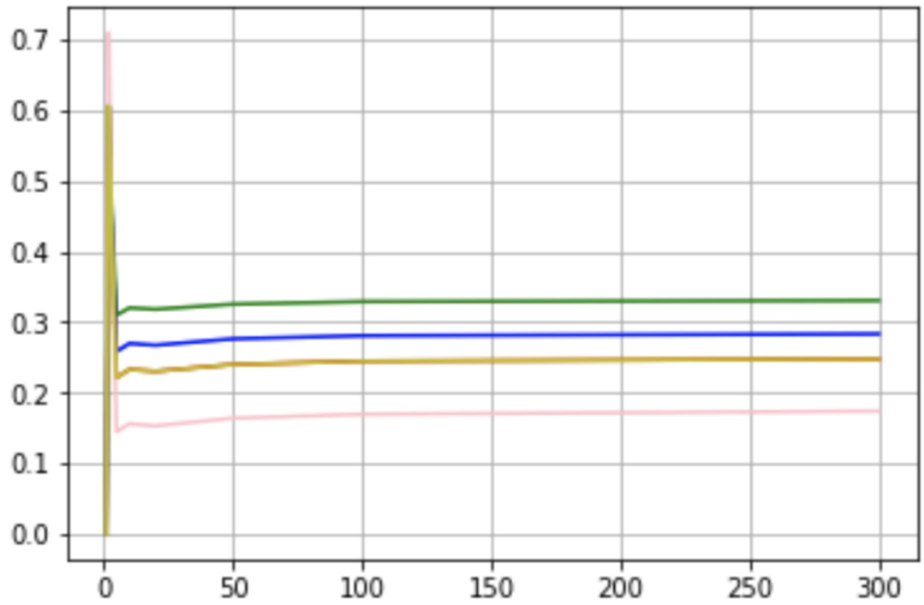
ii.

For LSI, contingency matrics and measures in different r values:

LSI		
r	contingency matrix	measures
1	[[1716 2187] [1672 2307]]	homogeneity=0.000279 completeness=0.000283 v-measure=0.000281 adj rand index=0.000307 adj mutual info=0.000187
2	[[3672 231] [390 3589]]	homogeneity=0.605257 completeness=0.605628 v-measure=0.605442 adj rand index=0.709644 adj mutual info=0.605221
3	[[3857 46] [1249 2730]]	homogeneity=0.437143 completeness=0.466996 v-measure=0.451577 adj rand index=0.450714 adj mutual info=0.437092
5	[[3898 5] [2435 1544]]	homogeneity=0.221520 completeness=0.309832

		v-measure=0.258337 adj rand index=0.144962 adj mutual info=0.221449
10	[[3900 3] [2381 1598]]	homogeneity=0.233028 completeness=0.320017 v-measure=0.269681 adj rand index=0.155989 adj mutual info=0.232958
20	[[3900 3] [2396 1583]]	homogeneity=0.230375 completeness=0.318020 v-measure=0.267194 adj rand index=0.152996 adj mutual info=0.230305
50	[[3 3900] [1637 2342]]	homogeneity=0.239976 completeness=0.325248 v-measure=0.276180 adj rand index=0.163907 adj mutual info=0.239907
100	[[3 3900] [1664 2315]]	homogeneity=0.244830 completeness=0.328905 v-measure=0.280707 adj rand index=0.169503 adj mutual info=0.244761
300	[[4 3899] [1686 2293]]	homogeneity=0.247765 completeness=0.330405 v-measure=0.283179 adj rand index=0.173921 adj mutual info=0.247696

For LSI, plot the measures in different r values:

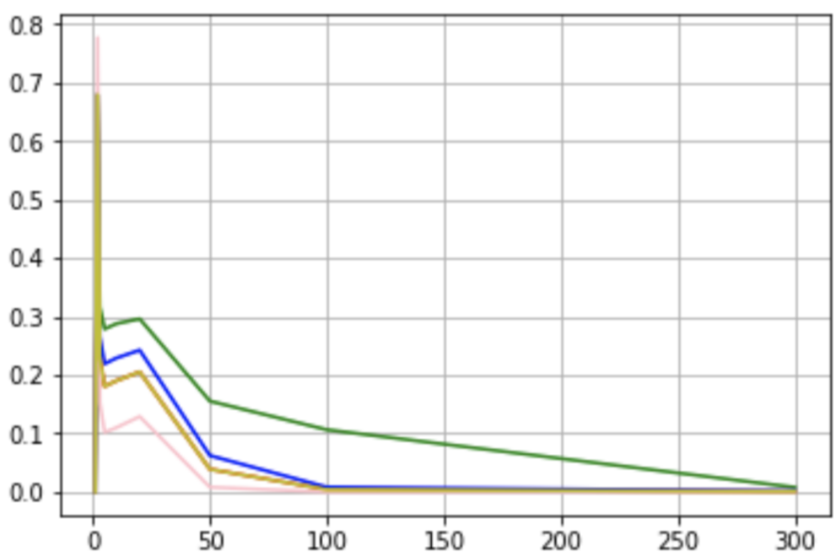


For NMF, contingency matrics and measures in different r values:

NMF		
r	contingency matrix	measures
1	[[2200 1703] [2323 1656]]	homogeneity=0.000299 completeness=0.000304 v-measure=0.000302 adj rand index=0.000339 adj mutual info=0.000208
2	[[309 3594] [3821 158]]	homogeneity=0.679048 completeness=0.680132 v-measure=0.679590 adj rand index=0.777018 adj mutual info=0.679019
3	[[4 3899] [1583 2396]]	homogeneity=0.229343 completeness=0.316484 v-measure=0.265957 adj rand index=0.152797 adj mutual info=0.229272
5	[[3898 5] [2677 1302]]	homogeneity=0.180631 completeness=0.278709 v-measure=0.219199 adj rand index=0.101956 adj mutual info=0.180556

10	[[3 3900] [1348 2631]]	homogeneity=0.190155 completeness=0.287701 v-measure=0.228972 adj rand index=0.109886 adj mutual info=0.190081
20	[[7 3896] [1461 2518]]	homogeneity=0.205373 completeness=0.296097 v-measure=0.242528 adj rand index=0.128997 adj mutual info=0.205300
50	[[325 3578] [5 3974]]	homogeneity=0.038962 completeness=0.155346 v-measure=0.062299 adj rand index=0.008154 adj mutual info=0.038874
100	[[0 3903] [34 3945]]	homogeneity=0.004267 completeness=0.106408 v-measure=0.008206 adj rand index=-0.000093 adj mutual info=0.004175
300	[[3813 90] [3927 52]]	homogeneity=0.001029 completeness=0.007904 v-measure=0.001821 adj rand index=0.000277 adj mutual info=0.000937

For NMF, plot the measures in different r values:



The best r for each algorithm:

The best r	
LSI	2
NMF	2

Q&A1:

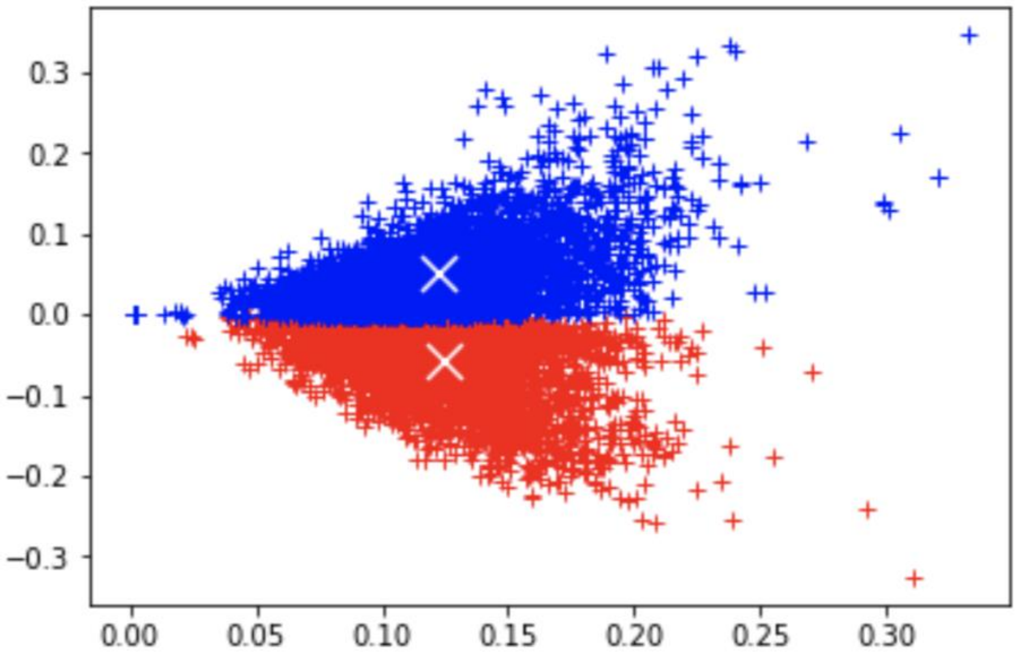
Question: How do you explain the non-monotonic behavior of the measures as r increases?

Answer: As r grows, the variance of data increases, which means more and more features are included. However, when r is bigger, the Euclidean distance performs worse, because the Euclidean distances between high dimensional data points tends to be almost the same, i.e. the points essentially become uniformly distant from each other. The premise of nearest neighbor search is that "closer" points are more relevant than "farther" points, but if all points are essentially uniformly distant from each other, the distinction is meaningless. Thus, the (r=2) gets better results than bigger r’s.

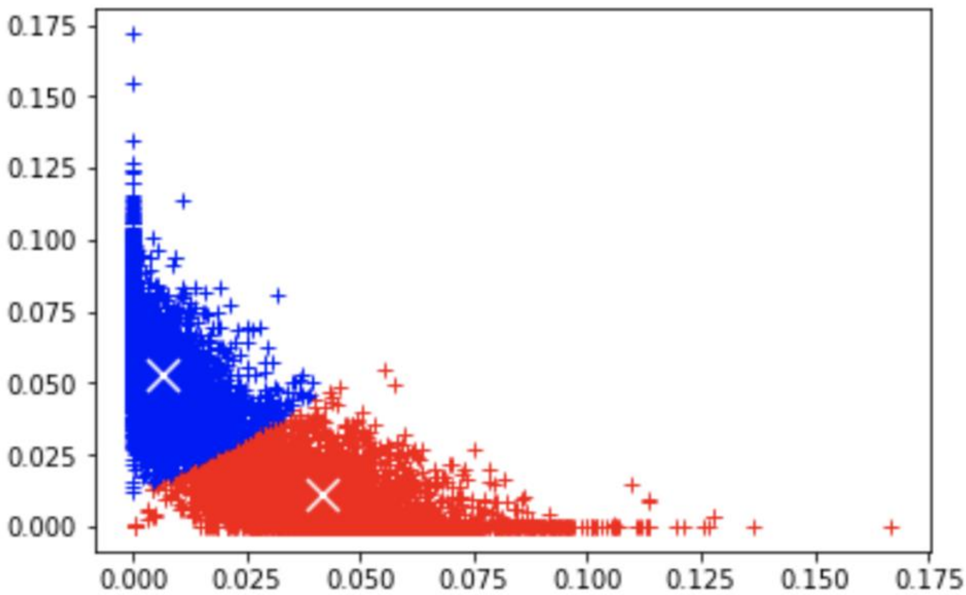
Part (4a)

In this part, we visualize the performance of the case with best clustering results in the previous part. We plot the clustered data vectors in different color points and put them in graph. The cluster center is noted as white “X”.

LSI:

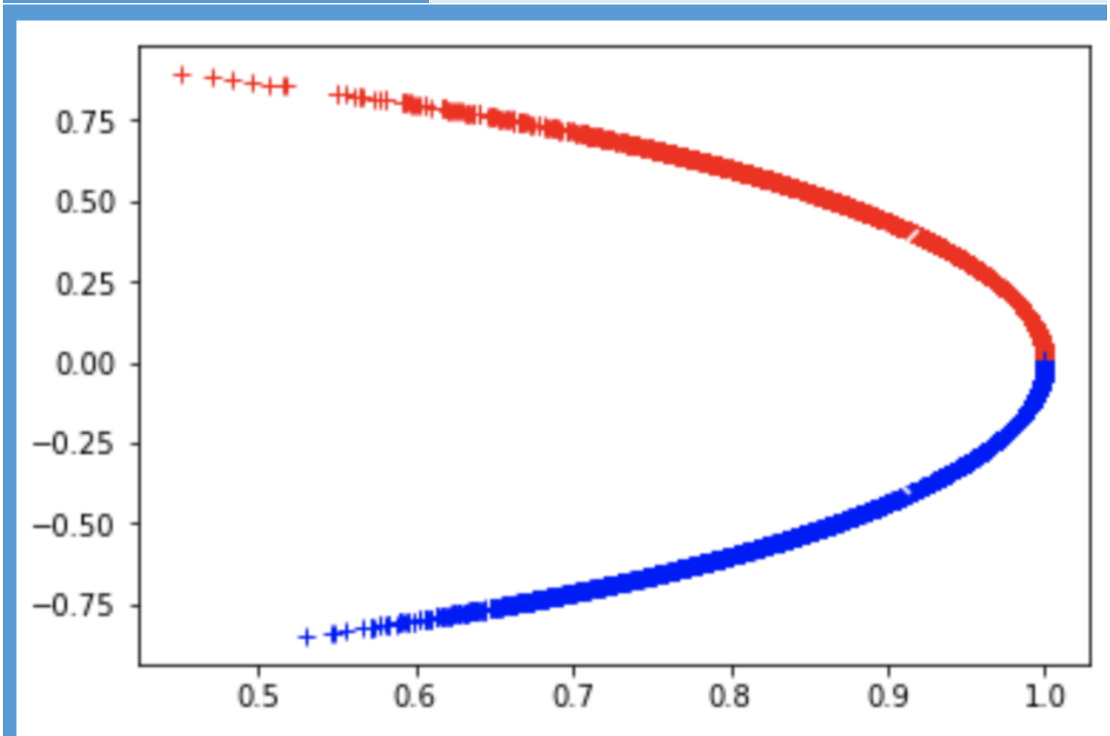


NMF:

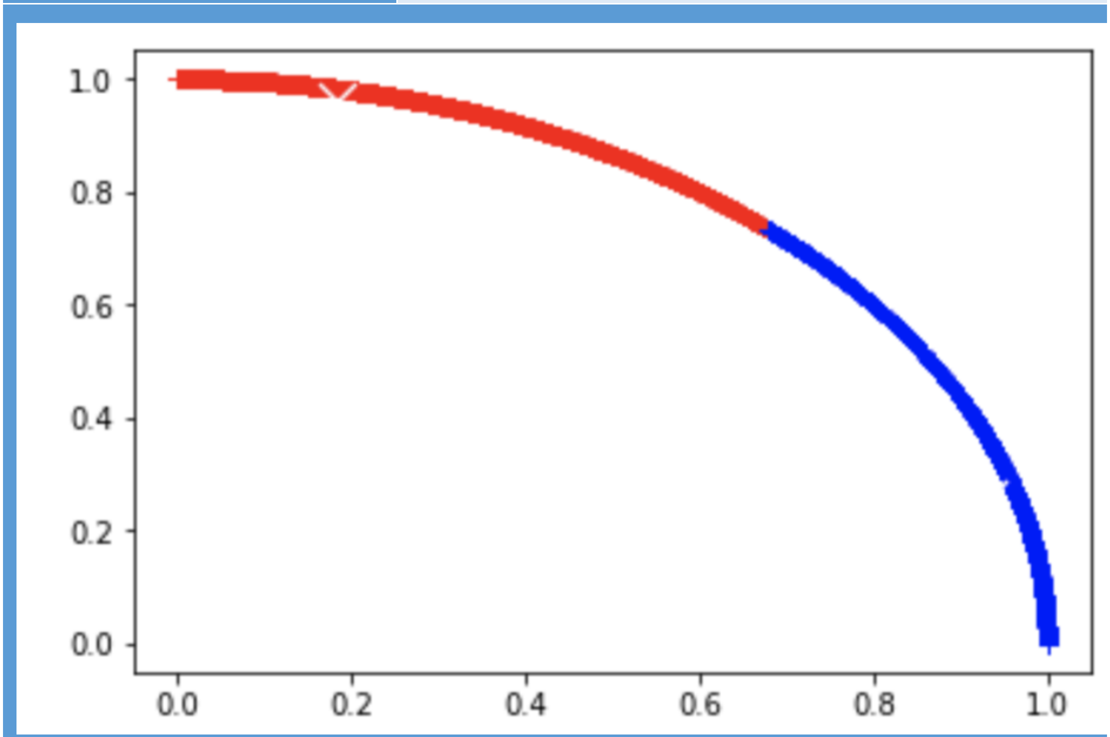


Part (4b)

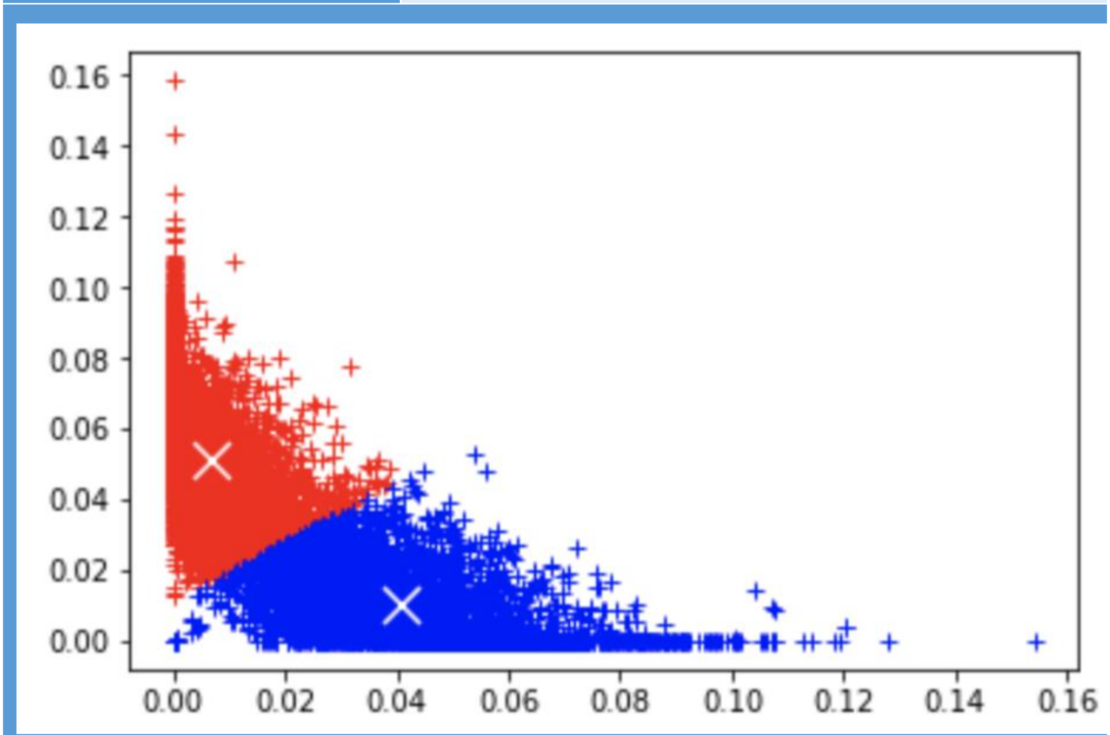
In this part, we try the three methods below to see whether they increase the clustering performance. We plot the clustered data vectors in different color points and put them in graph. The cluster center are noted as white "X"

Normalizing features after SVD	
contingency matrix	[[3813 90] [3927 52]]
measures	homogeneity=0.615275 completeness=0.615279 v-measure=0.615277 adj rand index=0.722098 adj mutual info=0.615239
	

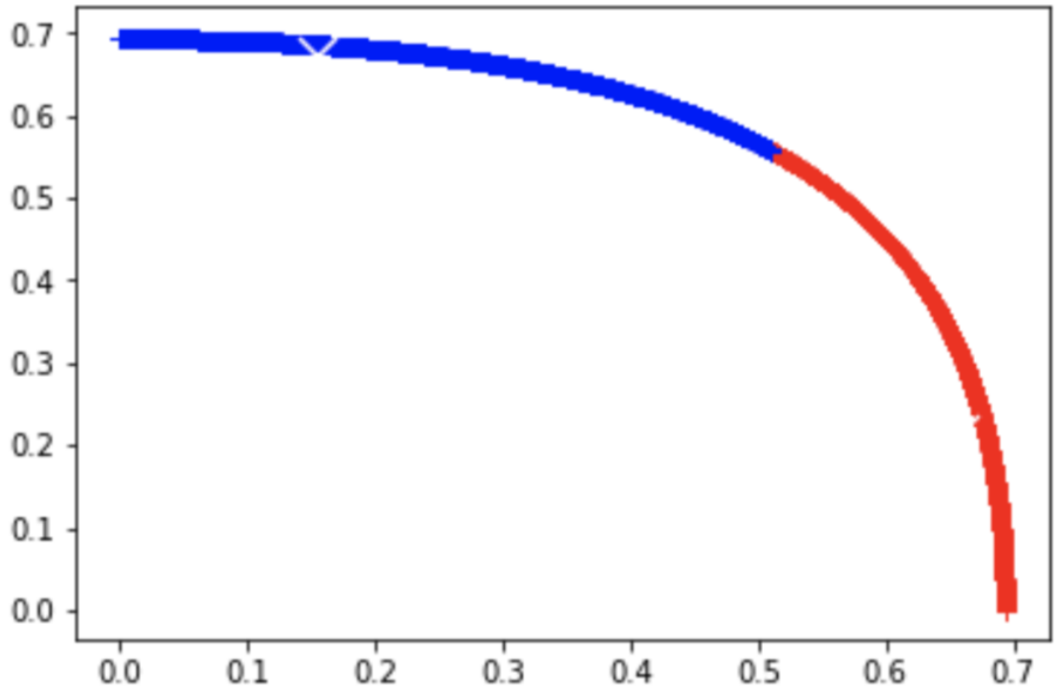
Normalizing features after NMF	
contingency matrix	[[3813 90] [3927 52]]
measures	homogeneity=0.654307 completeness=0.660829 v-measure=0.657552 adj rand index=0.734225 adj mutual info=0.654275



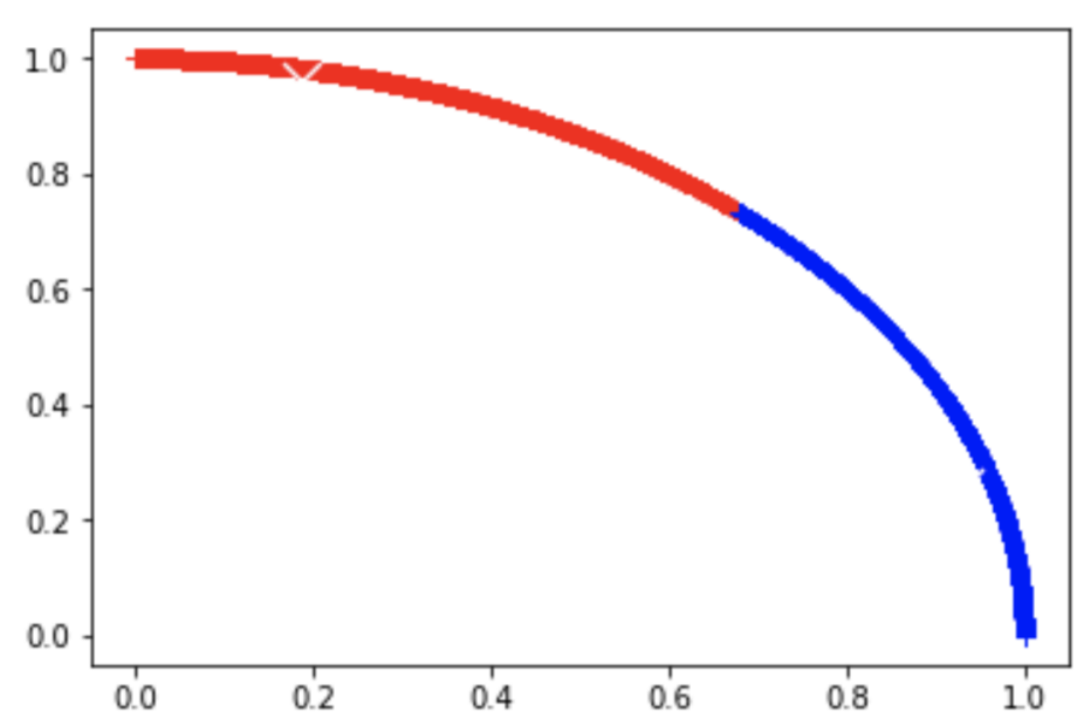
logarithmic transformation after NMF	
contingency matrix	[[3813 90] [3927 52]]
measures	homogeneity=0.676199 completeness=0.677582 v-measure=0.676890 adj rand index=0.773443 adj mutual info=0.676169



norm + log after NMF	
contingency matrix	[[3813 90] [3927 52]]
measures	homogeneity=0.659313 completeness=0.665373 v-measure=0.662329 adj rand index=0.740327 adj mutual info=0.659281



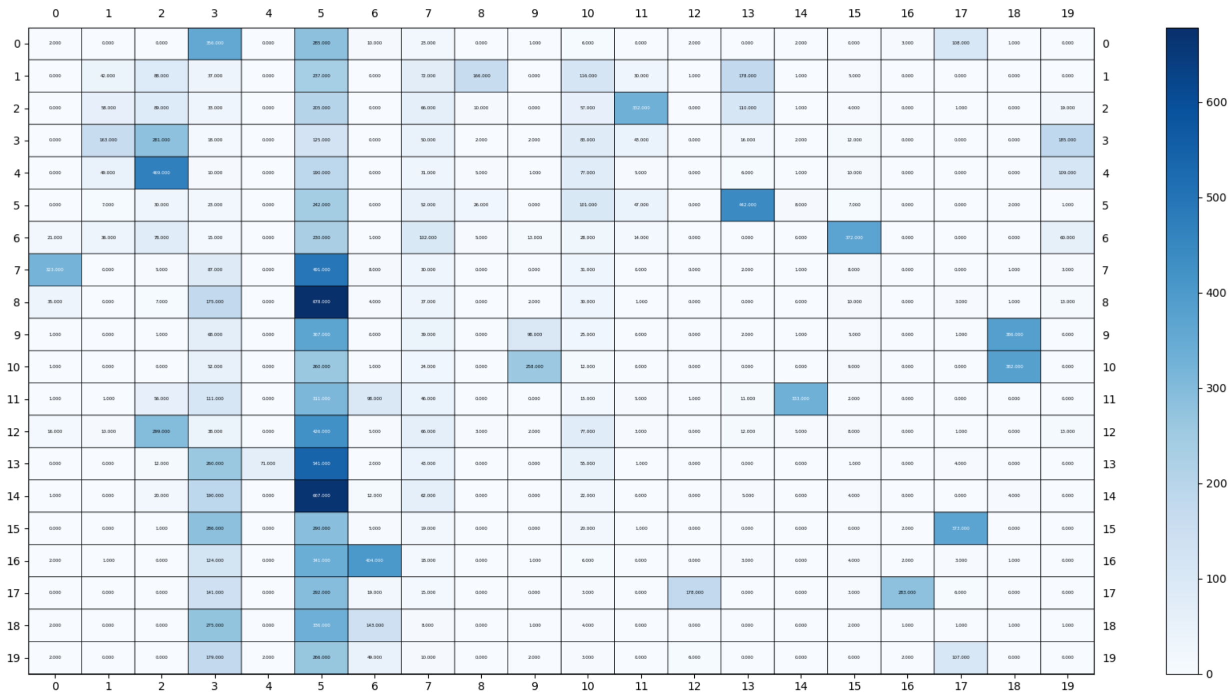
log + norm after NMF	
contingency matrix	[[3813 90] [3927 52]]
measures	homogeneity=0.654307 completeness=0.660829 v-measure=0.657552 adj rand index=0.734225 adj mutual info=0.654275



Q&A2:
Question: Can you justify why logarithm transformation may increase the clustering results?
Answer: From a theoretical perspective, the logarithmic transformation can give us a better result for skewed datasets since taking a log of something will decrease the weight of large data points and therefore make the distribution more uniform. However, in this case, at least from out experiment results, the logarithm transformation does not help in terms of contingency matrix. There is still the same number of correct classification, but the measures seem to increase in contrast of normalization.

Part (5)

We first import the whole 20 newsgroup dataset and using K-means clustering to classify the categories without using any dimension reduction algorithm:

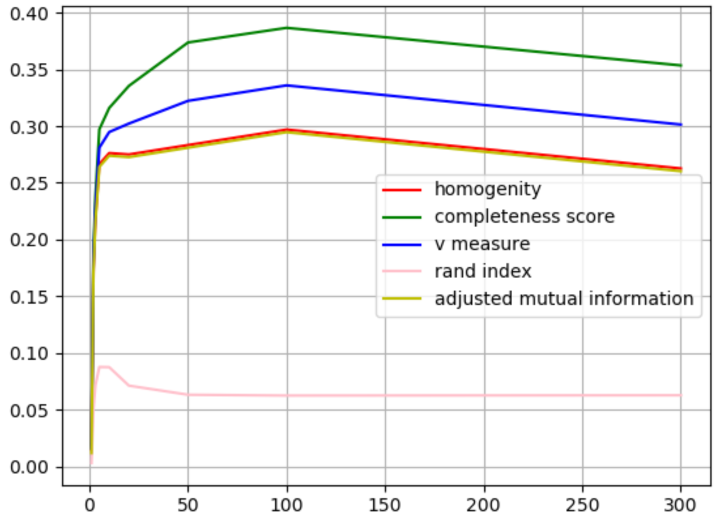


Confusion Matrix for all data without dimension reduction

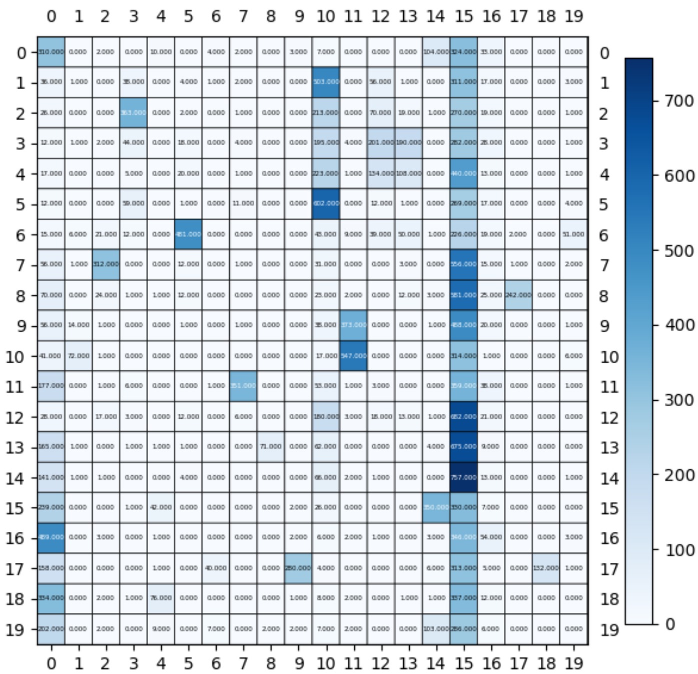
homogeneity	completeness	v-measure	adj rand index	adj mutual info
0.274481	**0.348996**	**0.307286**	**0.065318**	**0.272123**

Measurement Scores

Best r for SLI



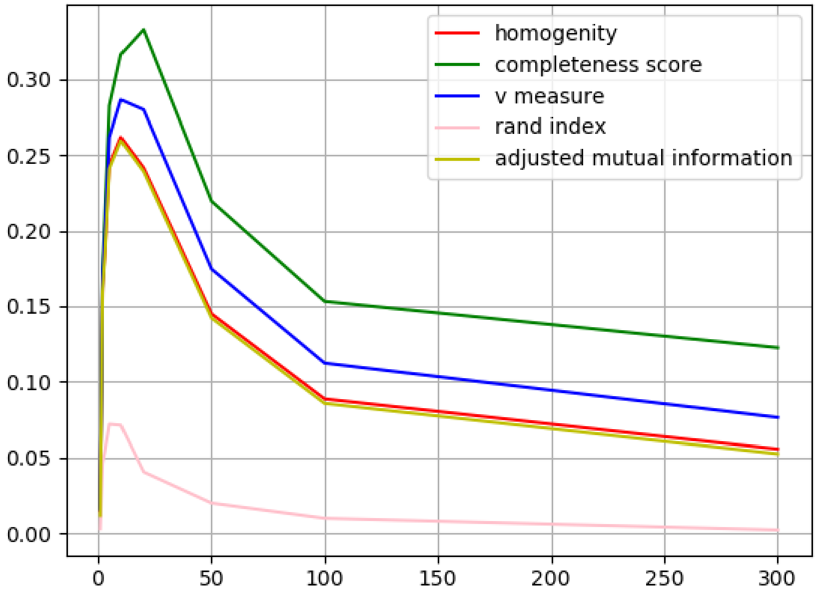
SLI Best Scores



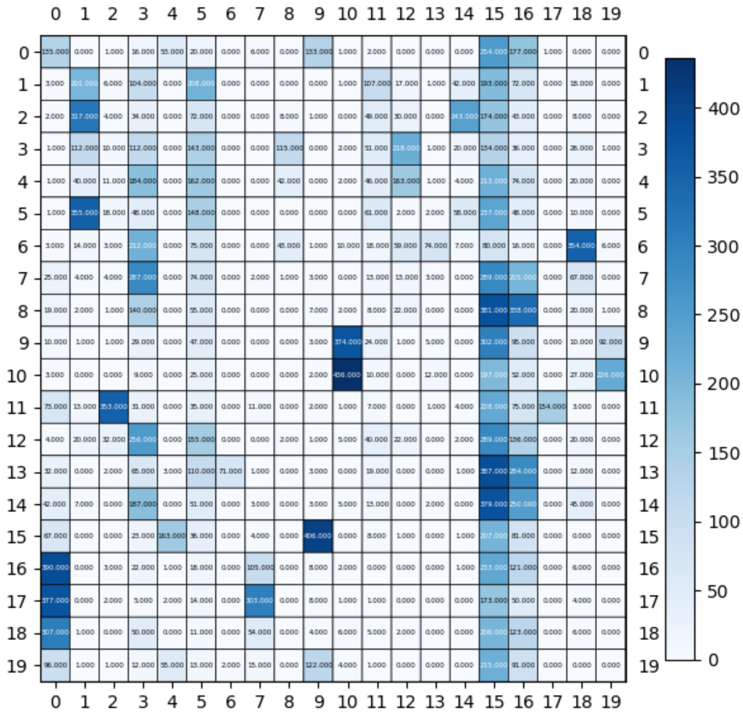
LSI Confusion Matrix

After choosing varies r values, which ranges from 2 to 300, and the plot and the confusion matrix shows the best r as 100.

Best r for NMF



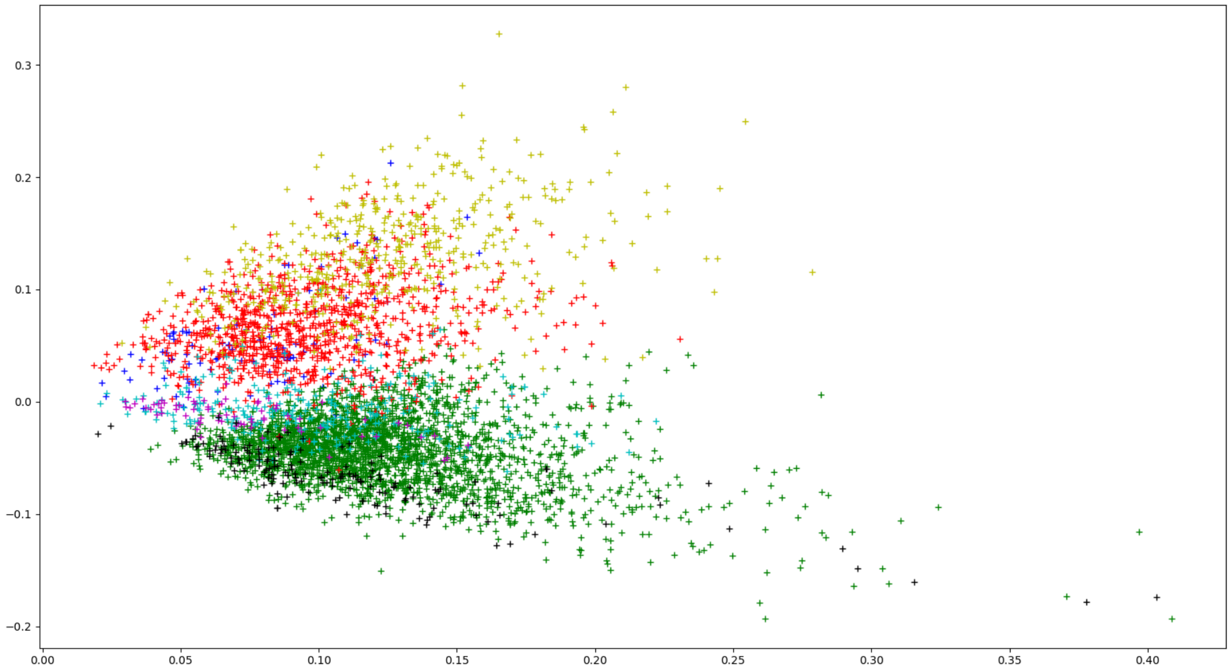
NMF Best Scores



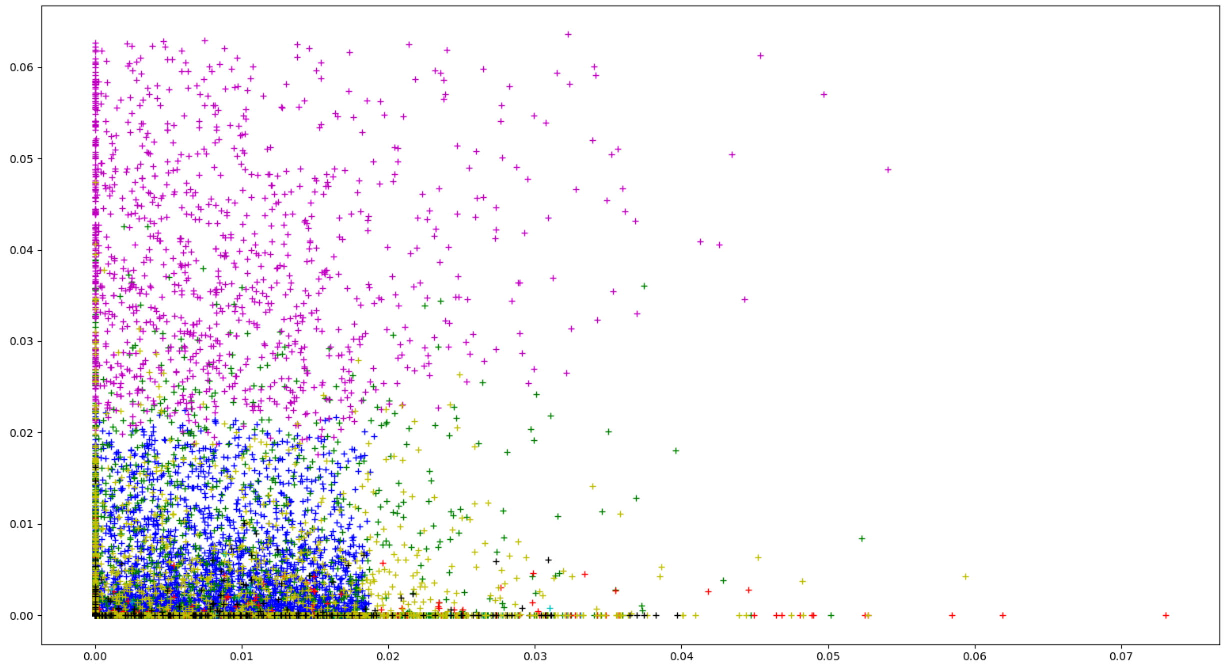
NMF Confusion Matrix

After choosing varies r values, which ranges from 2 to 300, and the plot and the confusion matrix shows the best r as 10.

Visualization of the Best Clustering Results

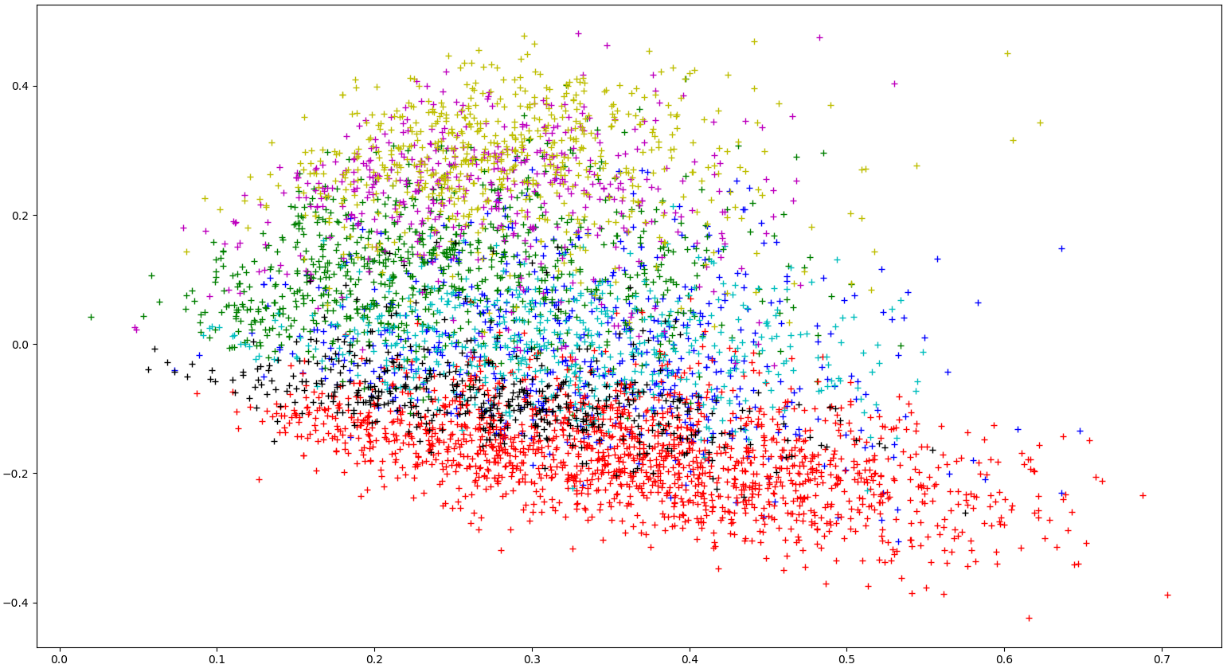


SLI visualization

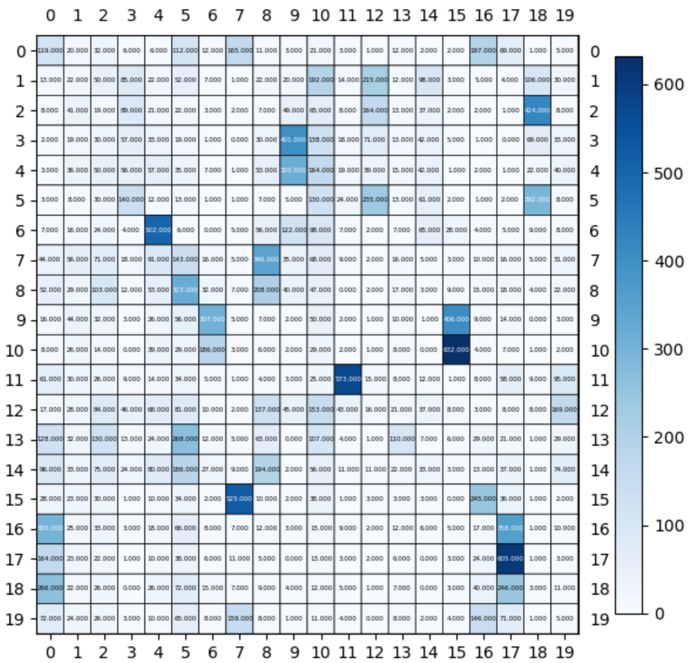


NMF visualization

SLI Normalization



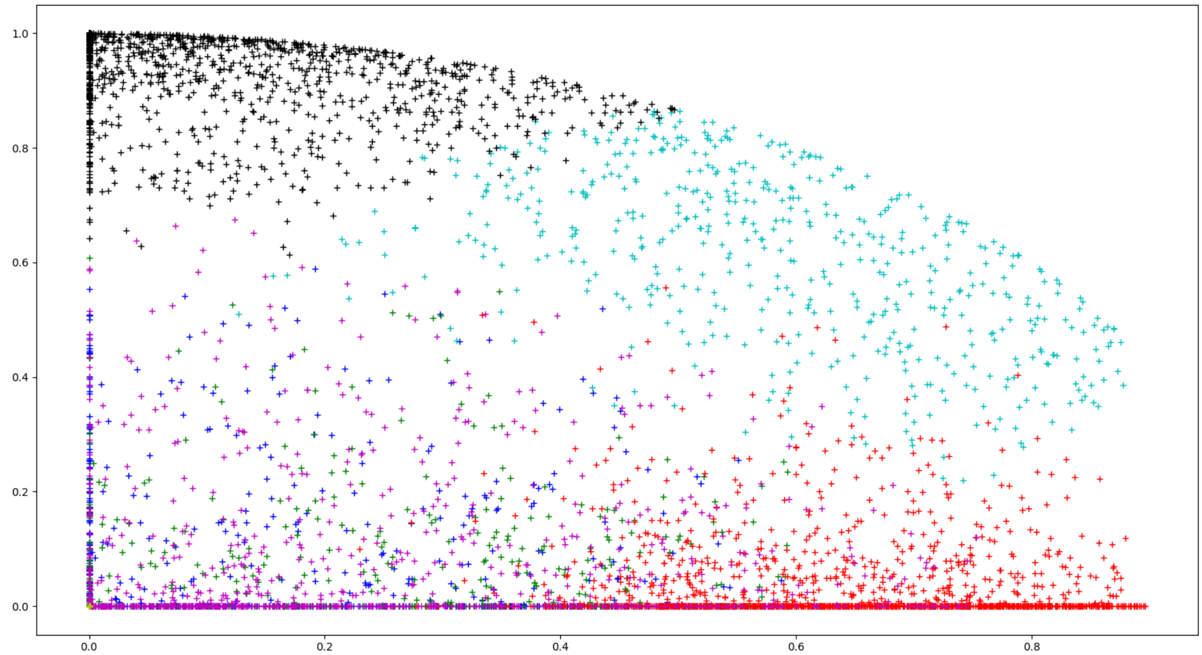
SLI Visualization after Normalization



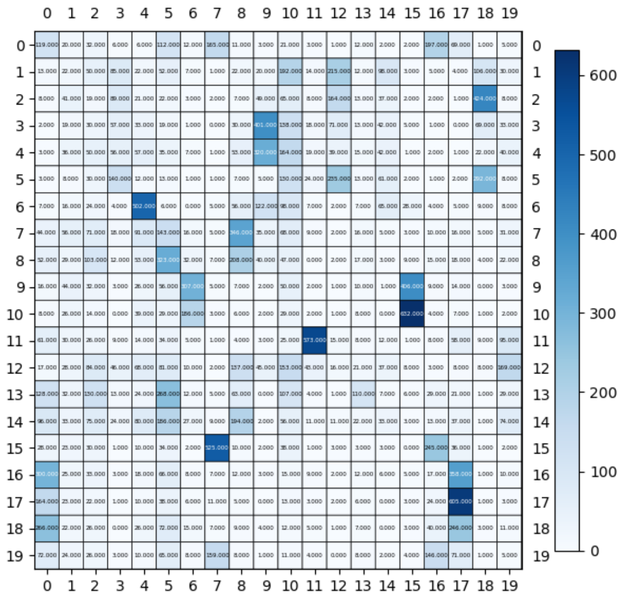
homogeneity	completeness	v-measure	adj rand index	adj mutual info
0.263167	0.383088	0.312001	0.057706	0.260739

Measurement Scores

NMF Normalization



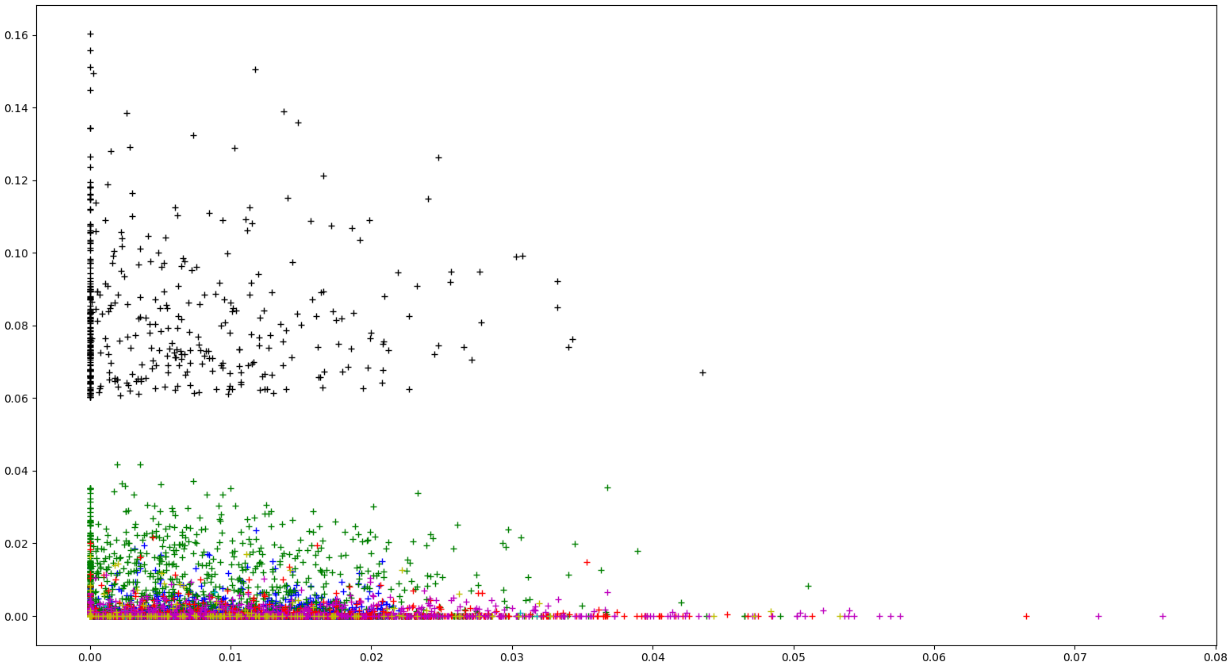
NMF Visualization after Normalization



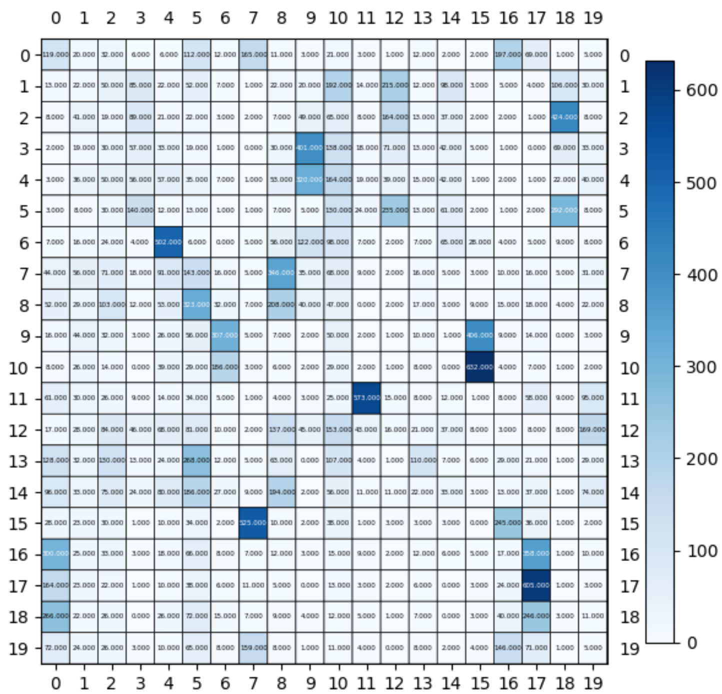
homogeneity	completeness	v-measure	adj rand index	adj mutual info
0.306838	0.314238	0.310494	0.172559	0.304600

Measurement Scores

NMF Logarithmic Transformation



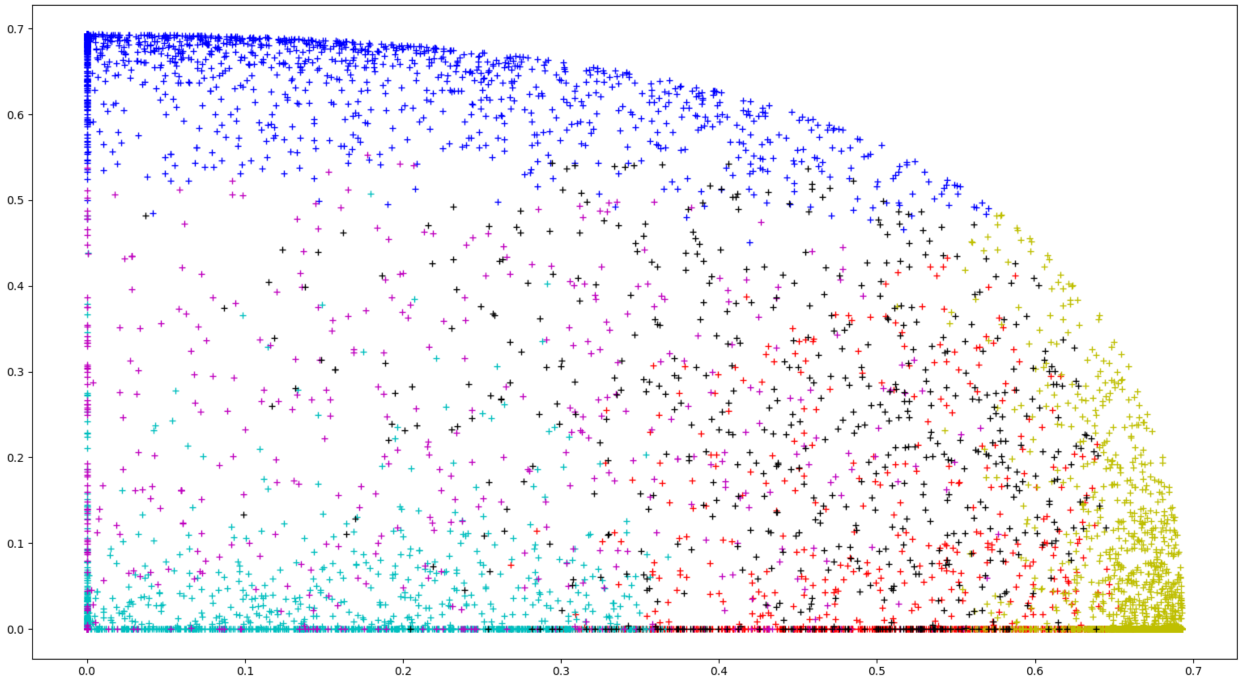
NMF Visualization after Logarithmic Transformation



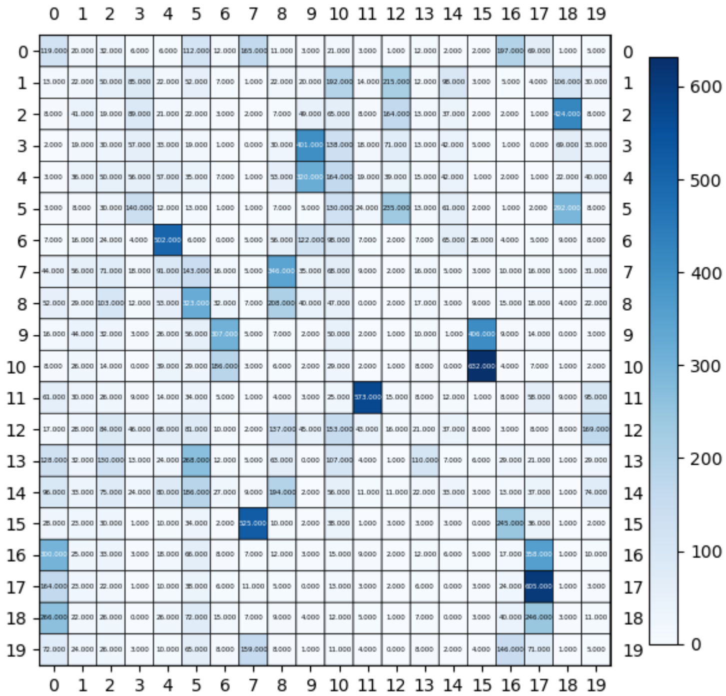
homogeneity	completeness	v-measure	adj rand index	adj mutual info
0.264368	**0.316948**	**0.288280**	**0.073057**	**0.261974**

Measurement Scores

NMF Normalization + Logarithmic Transformation



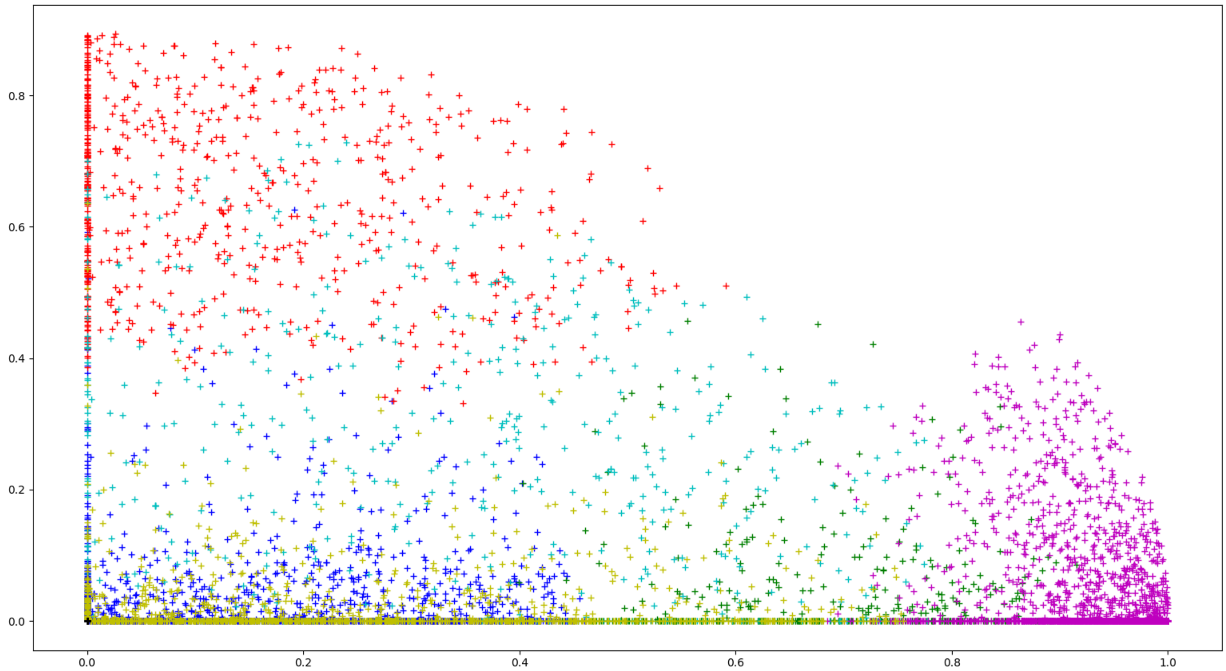
NMF Visualization after Normalization + Logarithmic Transformation



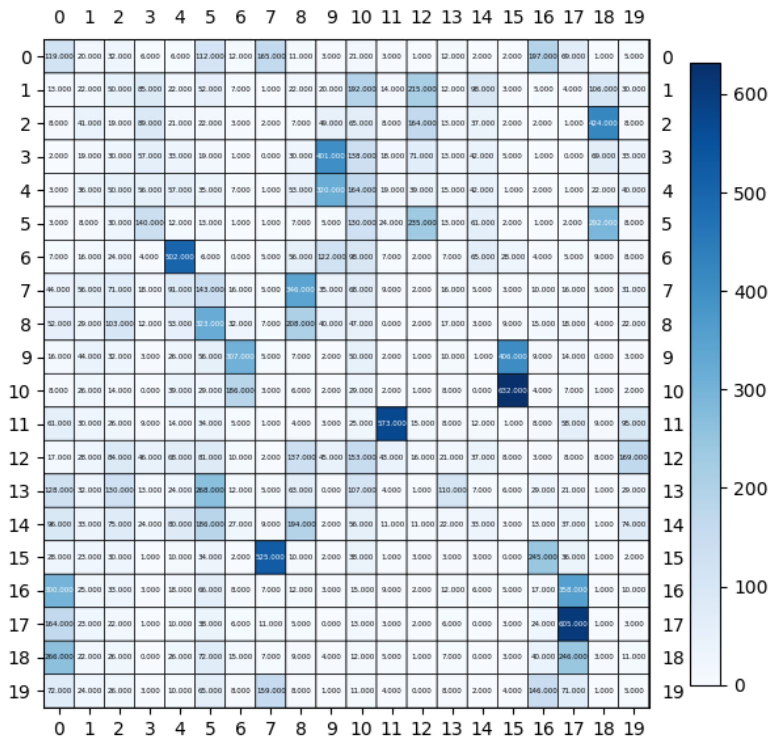
homogeneity	completeness	v-measure	adj rand index	adj mutual info
0.312089	**0.319704**	**0.315851**	**0.171869**	**0.309868**

Measurement Scores

NMF Logarithmic Transformation + Normalization



NMF Visualization Logarithmic Transformation + Normalization



homogeneity	completeness	v-measure	adj rand index	adj mutual info
0.307645	**0.318062**	**0.312767**	**0.169601**	**0.305409**

Measurement Score