ECE 219 Large-Scale Data Mining Project 2

605033865 Sheng Yung Tao
704945153 Hua-En Li

Results:

1. Build the TF-IDF matrix:

dimension = (7882, 27743) after applying TF-IDF vectorizer.

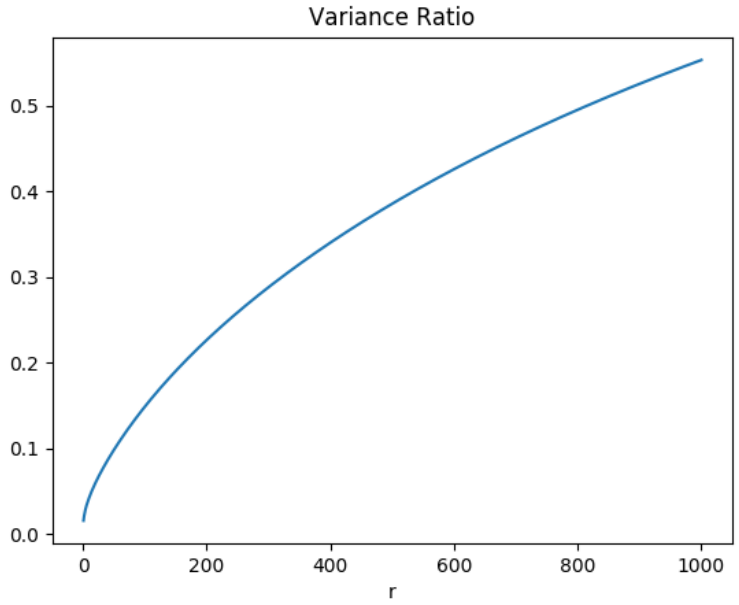
2. Apply KMeans with k=2 using the TF_IDF data:

859	3044
3965	14

Homogeneity score:	0.569
Completeness score:	0.591
V-measure:	0.581
Adjusted Rand Score:	0.606
Adjusted mutual info score:	0.569

3(a). Preprocess the data:

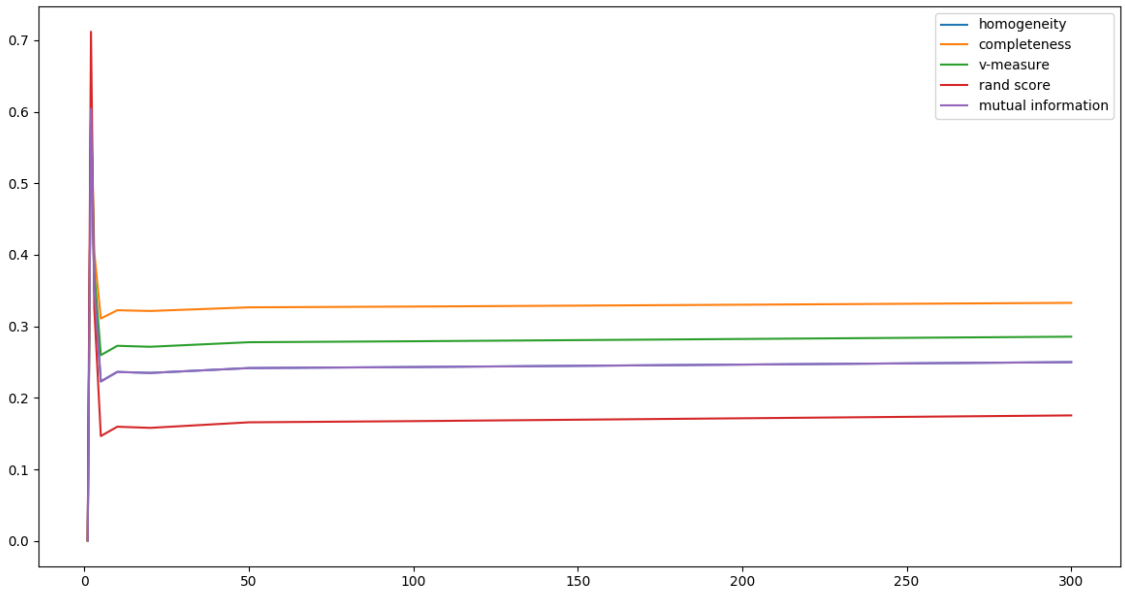
(i) Retained variance ratio with SVD dimensionality reduction:



(ii) Choose best r, SVD:

Columns below are: homogeneity/completeness/v-measure/adjusted rand score/adjusted mutual information score/contingency matrix

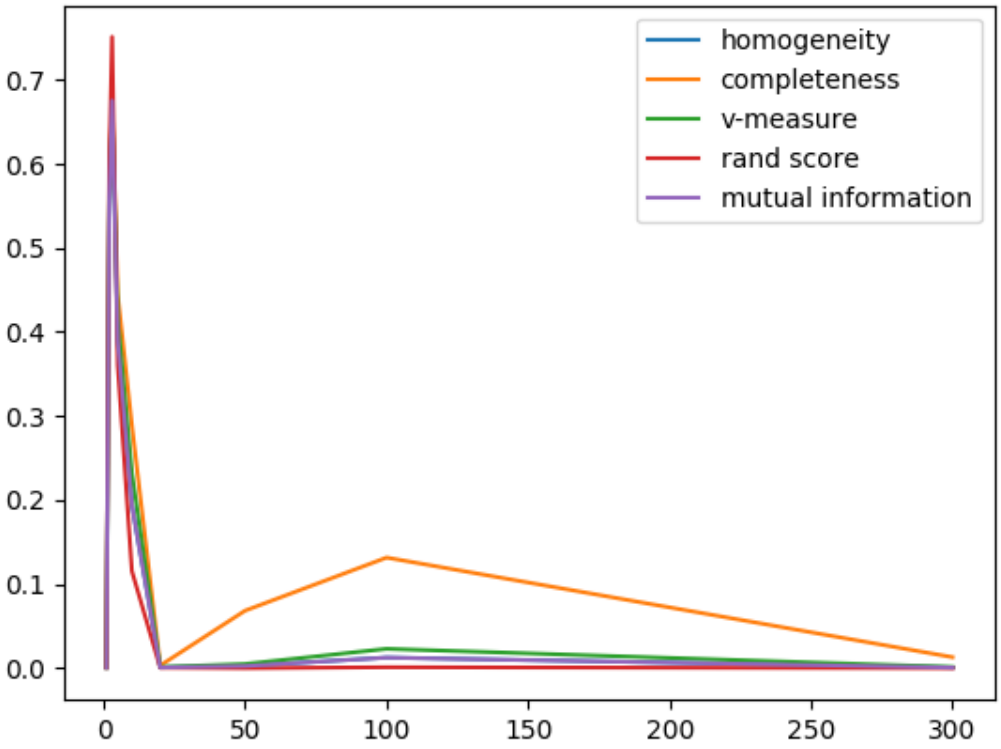
r	1	2	3	5	10	20	50	100	300
	0	0.604	0.355	0.223	0.263	0.235	0.242	0.243	0.250
	0	0.604	0.404	0.311	0.323	0.321	0.327	0.328	0.333
	0	0.604	0.378	0.260	0.273	0.272	0.278	0.279	0.286
	0	0.712	0.326	0.147	0.160	0.158	0.166	0.168	0.276
	0	0.604	0.354	0.223	0.236	0.235	0.242	0.243	0.250
	$\begin{bmatrix}2187 & 1755\\ 2320 & 1659\end{bmatrix}$	$\begin{bmatrix}3586 & 317\\ 299 & 3680\end{bmatrix}$	$\begin{bmatrix}25 & 3878\\ 2314 & 1665\end{bmatrix}$	$\begin{bmatrix}5 & 3898\\ 1553 & 2426\end{bmatrix}$	$\begin{bmatrix}3 & 3900\\ 1617 & 2362\end{bmatrix}$	$\begin{bmatrix}3 & 3900\\ 1609 & 2370\end{bmatrix}$	$\begin{bmatrix}3900 & 3\\ 2332 & 1647\end{bmatrix}$	$\begin{bmatrix}3900 & 3\\ 2324 & 1655\end{bmatrix}$	$\begin{bmatrix}3900 & 3\\ 2286 & 1693\end{bmatrix}$



According to above measures, the best r is 2.

Choose best r, NMF:

r	1	2	3	5	10	20	50	100	300
	0	0.558	0.675	0.389	0.194	0	0.002	0.012	0
	0	0.579	0.682	0.444	0.289	0.003	0.068	0.132	0.013
	0	0.568	0.678	0.415	0.232	0.001	0.004	0.023	0.001
	0	0.599	0.751	0.361	0.115	0	0	0.001	0
	0	0.558	0.675	0.389	0.194	0	0.002	0.001	0
	$\begin{bmatrix}2012 & 1891\\ 2164 & 1815\end{bmatrix}$	$\begin{bmatrix}3038 & 865\\ 25 & 3954\end{bmatrix}$	$\begin{bmatrix}472 & 3431\\ 3926 & 53\end{bmatrix}$	$\begin{bmatrix}1570 & 2333\\ 3975 & 4\end{bmatrix}$	$\begin{bmatrix}3898 & 5\\ 2596 & 1383\end{bmatrix}$	$\begin{bmatrix}3697 & 206\\ 3825 & 154\end{bmatrix}$	$\begin{bmatrix}3901 & 2\\ 3950 & 29\end{bmatrix}$	$\begin{bmatrix}3806 & 97\\ 3979 & 0\end{bmatrix}$	$\begin{bmatrix}3888 & 15\\ 3944 & 35\end{bmatrix}$



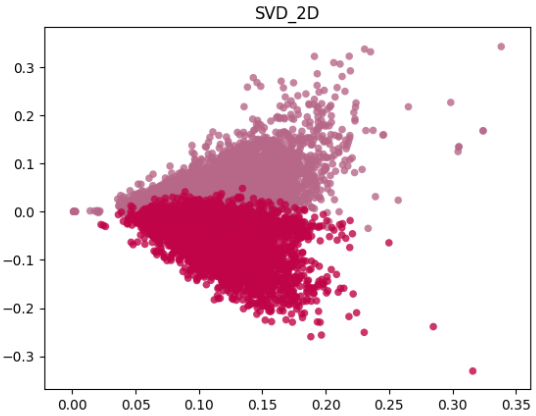
According to above measures, the best r is 3.

How do you explain the non-monotonic behavior of the measures as *r* increase?

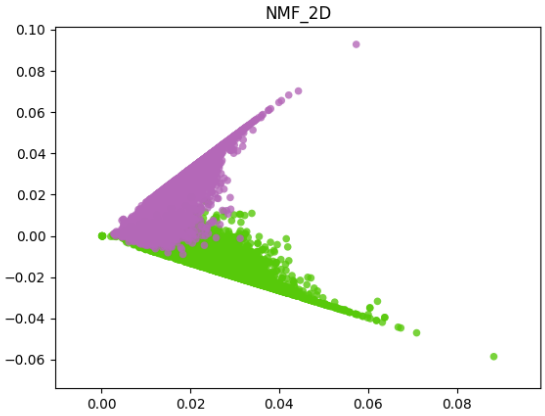
As r increases, the variance retained ratio increases, this means that more information is kept. However, this is a disadvantage for cluster, because points are more difficult to cluster together due to the high dimension space. The appendix also reveals that in high dimensional space, all Euclidean distances tend to be 1. That will be bad for cluster. As for bad performance when r=1, we found that while the variance of the first dimension is the biggest, the data are not well separated because of some outliers increase the variance.

4(a). Visualize the dimensionality deduction data using the best r.

SVD(r=2):



NMF (project r=3 to r=2):

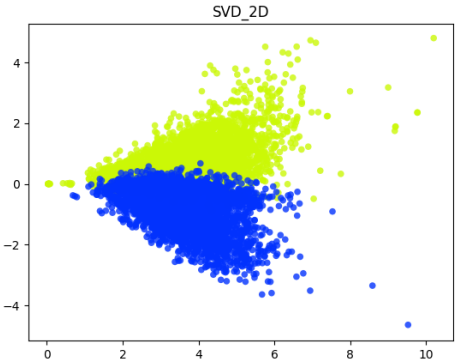


4(b).

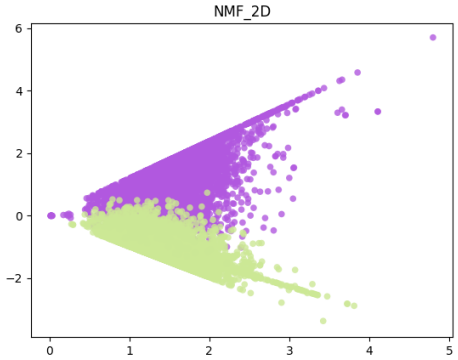
measure	Norm SVD	Norm NMF	Log NMF	Log->norm NMF	Norm->log NMF
1	0.219	0.658	0.678	0.689	0.043
2	0.249	0.666	0.678	0.689	0.050
3	0.233	0.662	0.678	0.689	0.046
4	0.235	0.729	0.779	0.788	0.049
5	0.219	0.657	0.678	0.689	0.043
Matrix	$\begin{bmatrix}1780 & 2123\\3731 & 248\end{bmatrix}$	$\begin{bmatrix}529 & 3374\\3933 & 46\end{bmatrix}$	$\begin{bmatrix}224 & 2679\\3742 & 237\end{bmatrix}$	$\begin{bmatrix}265 & 3638\\3802 & 177\end{bmatrix}$	$\begin{bmatrix}2376 & 1527\\3290 & 689\end{bmatrix}$

For data vs. ground truth:

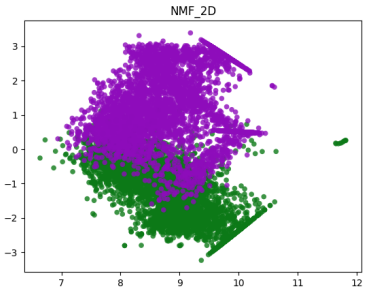
1.



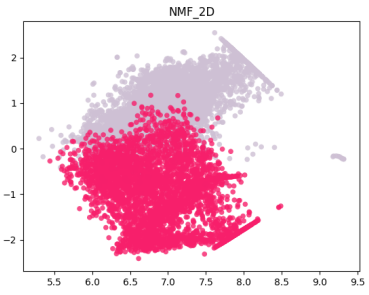
2.



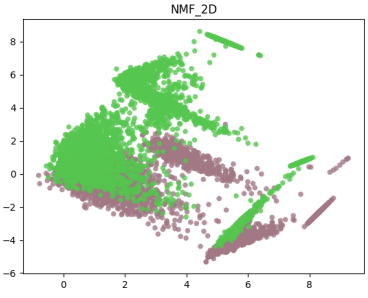
3.



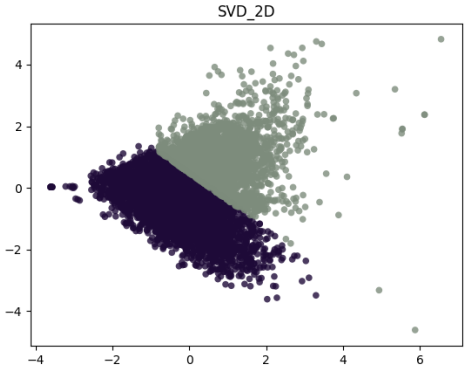
4.



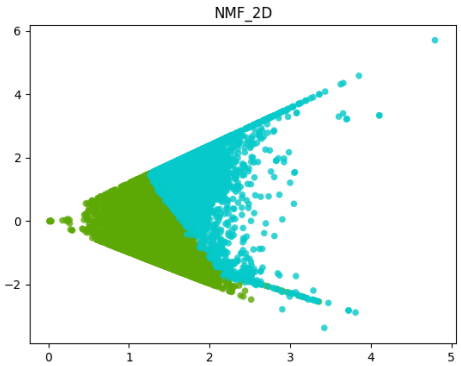
5.



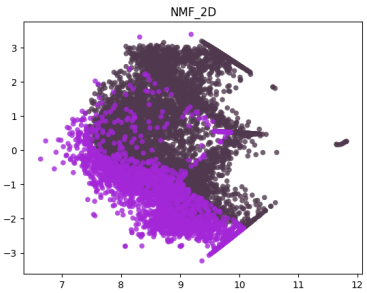
For data vs. prediction



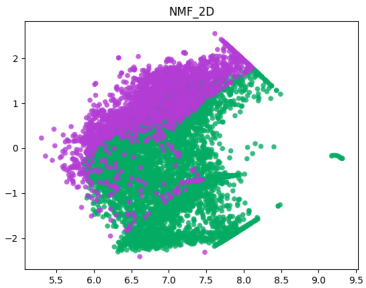
1.



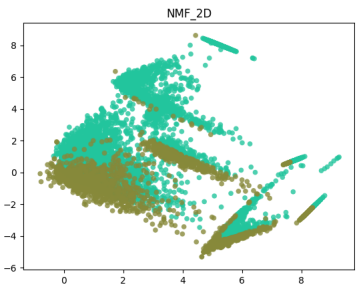
2.



3.



4.



5.

Observation:

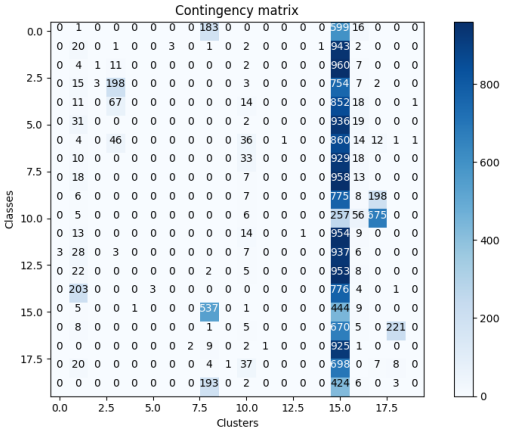
- Normalizing the SVD data has no change in data distributive shape. However, the prediction score is very poor. I printed the prediction and found that the prediction labels are far from the class ground truth. The reason why the performance is poor is probability because normalization units the variance, which conflicts what we do in SVD by keeping the largest variance.
- The log transform really increases the performance. I think the main reason is that logarithm can make extreme big values, which may be outliers, to be smaller. Log (10000) = 4. However, I also noticed that logarithm operation will transform 0 to –Inf. This is solved by adding small bias to zero values. But actually, we don't want these values to be too small, or it will again be outliers. Thus, I choose to add 0.001 to zero values. Which keeps the transformed zero values to be -3, not too far from zero. This is the trick to get good performance.

5. 20 categories:

Dimension after TF-IDF: (18846, 52268)

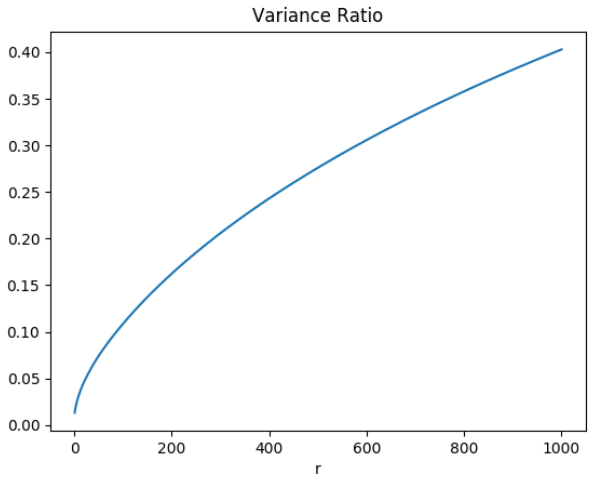
The score using TF-IDF data:

Homogeneity score:	0.112
Completeness score:	0.437
V-measure:	0.178
Adjusted Rand Score:	0.013
Adjusted mutual info score:	0.110



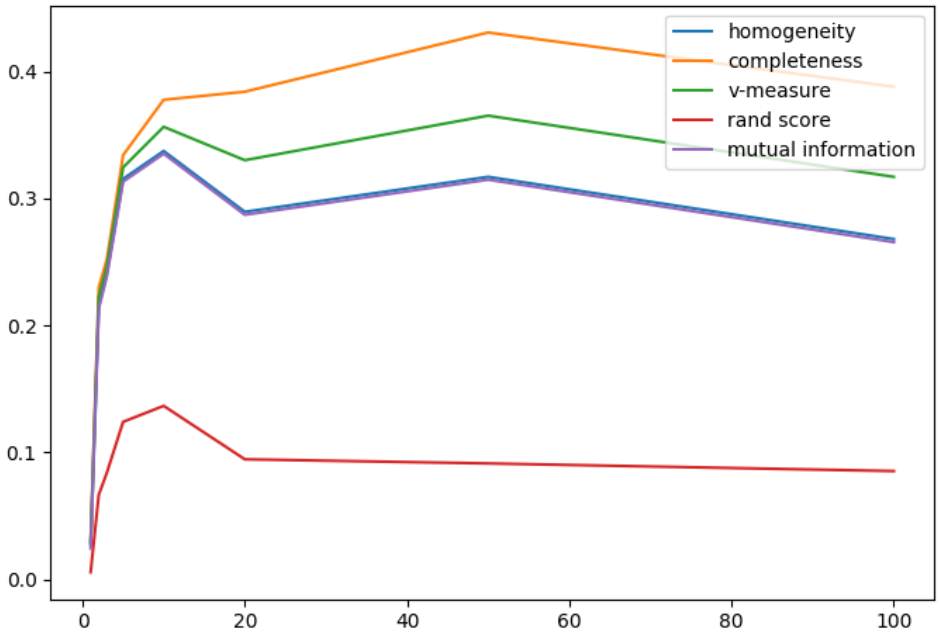
performance is really poor (high values on one column, not on diagonal)

Retained variance ratio:



Choose best r, SVD:

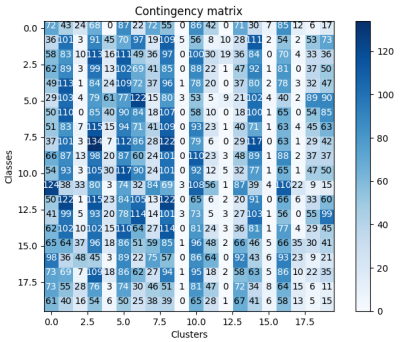
r	1	2	3	5	10	20	50	100	300
homogeneity	0.027	0.216	0.241	0.315	0.337	0.289	0.317	0.268	0.297
completeness	0.030	0.230	0.252	0.334	0.377	0.384	0.430	0.388	0.475
v-measure	0.029	0.223	0.246	0.324	0.356	0.330	0.365	0.317	0.365
rand score	0.005	0.066	0.084	0.124	0.136	0.094	0.091	0.085	0.075
mutual info	0.024	0.213	0.239	0.313	0.335	0.287	0.315	0.265	0.294

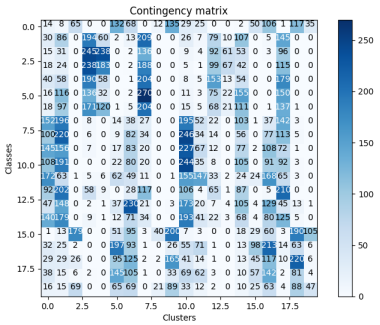


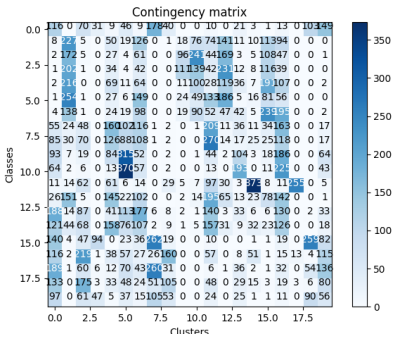
Best r=10, but r=50 also performs well. Generally speaking, none of them performs good because the measures do not have value greater than 0.5.

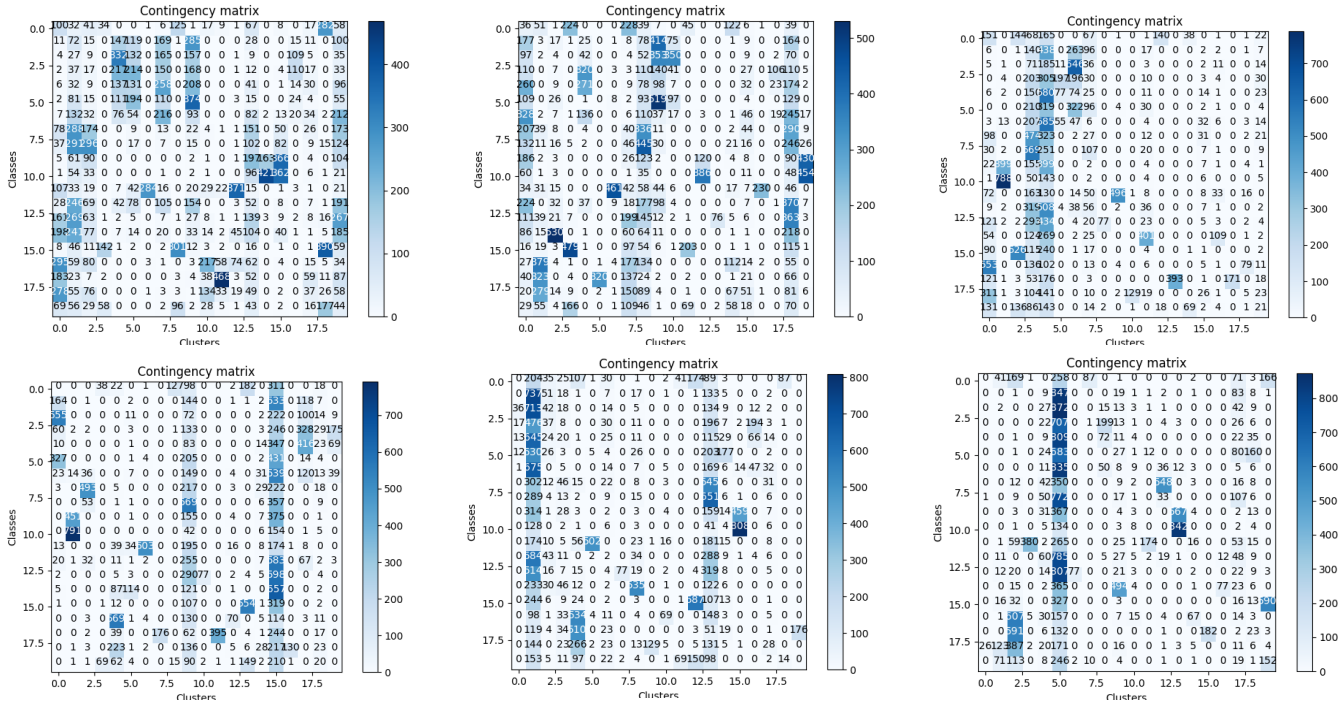
Contingency matrix:

R=1	R=2	R=3
R=5	R=10	R=20
R=50	R=100	R=300

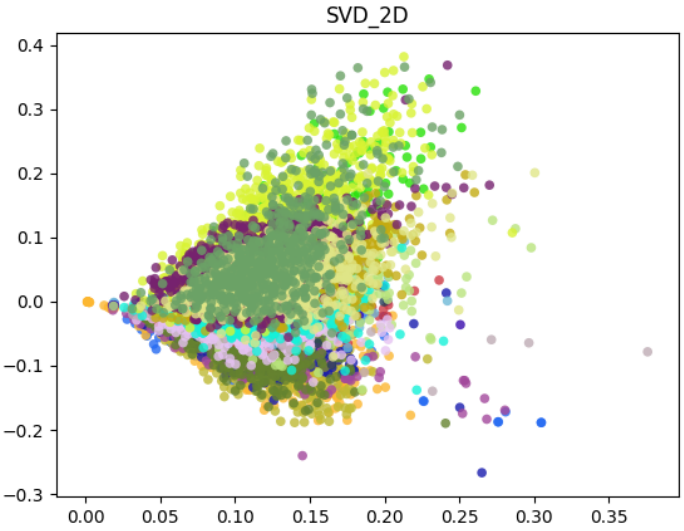






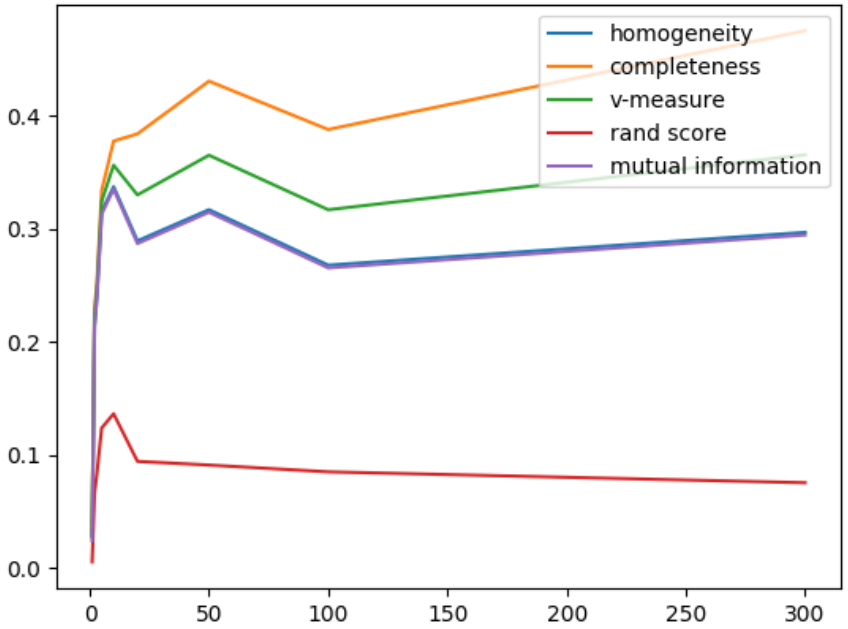


SVD best r visualization (project from r=10 to r=2):



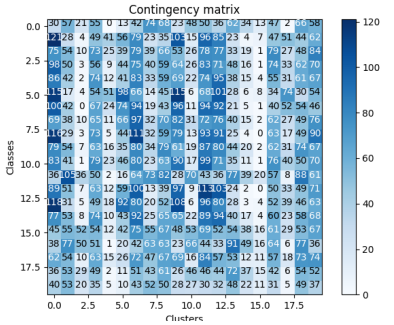
Choose best r, NMF:

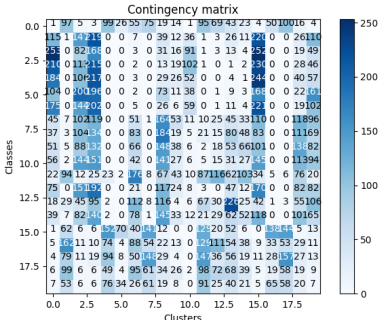
r	1	2	3	5	10	20	50	100	300
homogeneity	0.026	0.173	0.179	0.274	0.310	0.328	0.185	0.185	0.046
completeness	0.028	0.183	0.193	0.293	0.350	0.426	0.252	0.280	0.097
v-measure	0.027	0.178	0.186	0.283	0.329	0.370	0.214	0.222	0.063
rand score	0.005	0.049	0.052	0.098	0.127	0.094	0.038	0.030	0.004
mutual info	0.024	0.170	0.176	0.273	0.308	0.326	0.182	0.182	0.043

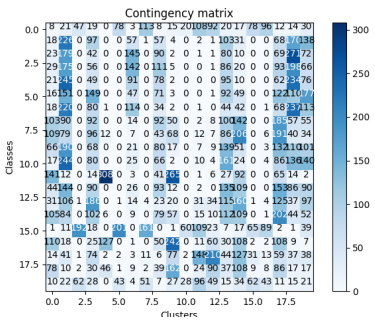


Best r is 20, but r=50 also has measure values.

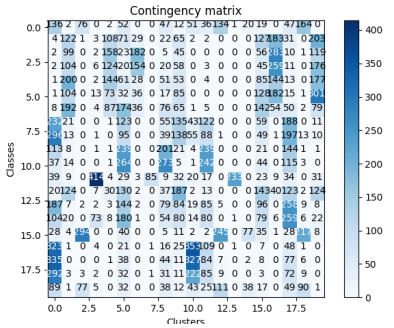
Contingency matrix: format as above

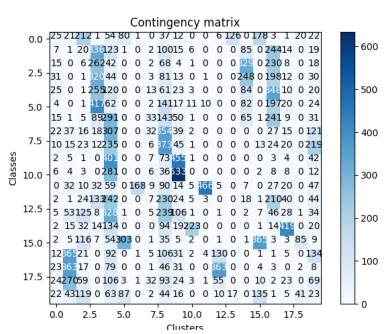


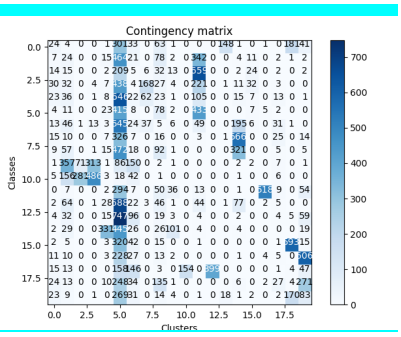


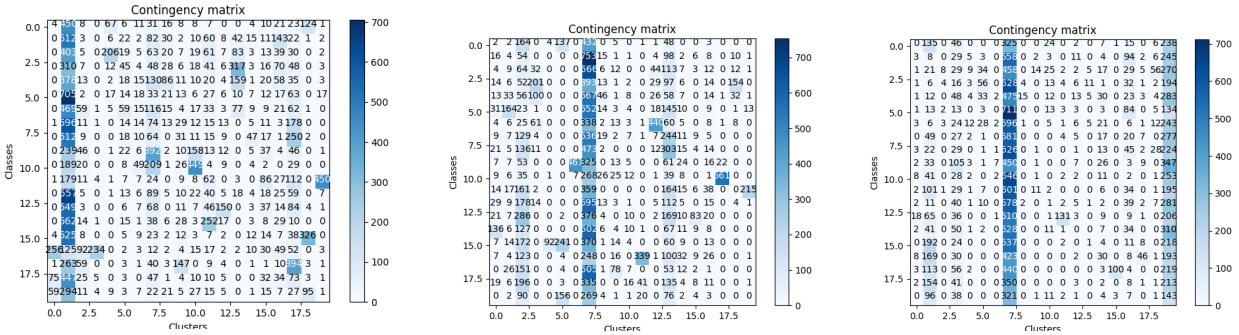


r=20, best (high diag vals)

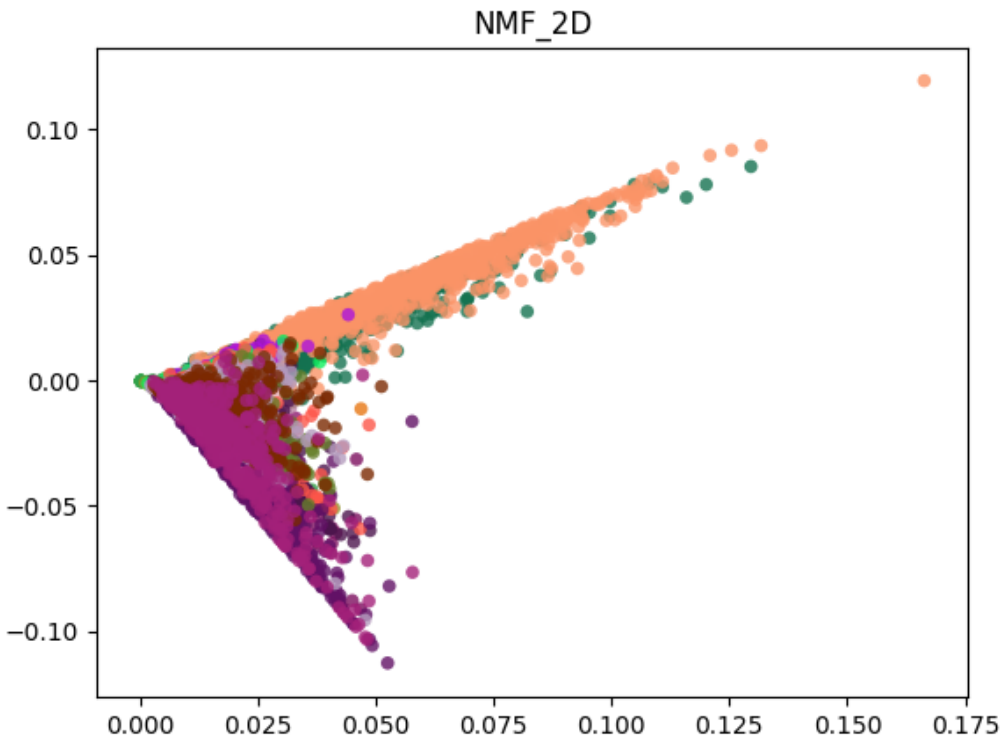








NMF best r visualization (project from r=20 to r=2):

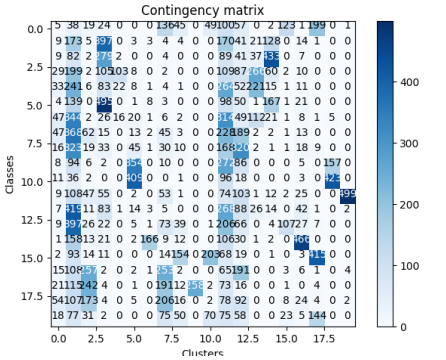


4. 5 transformations

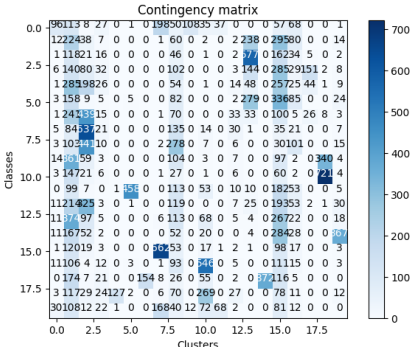
measure	Norm SVD	Norm NMF	Log NMF	Log->norm NMF	Norm->log NMF
1	0.307	0.301	0.398	0.381	0.258
2	0.348	0.359	0.412	0.388	0.259
3	0.326	0.327	0.405	0.385	0.259
4	0.111	0.127	0.235	0.249	0.124
5	0.305	0.299	0.396	0.379	0.256

Both normed NMF and log NMF outperform original SVD and NMF results.

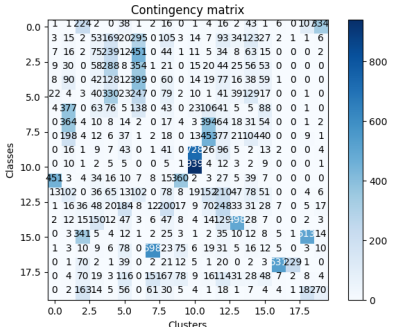
Contingency matrix:



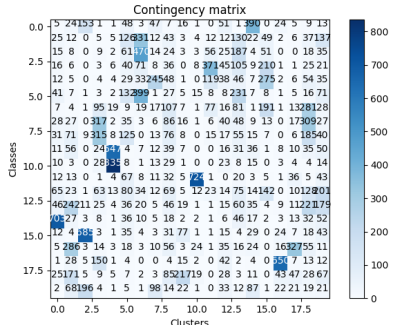
1.



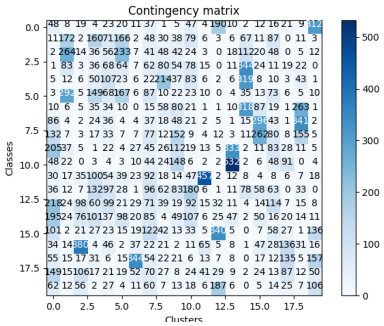
2.



3

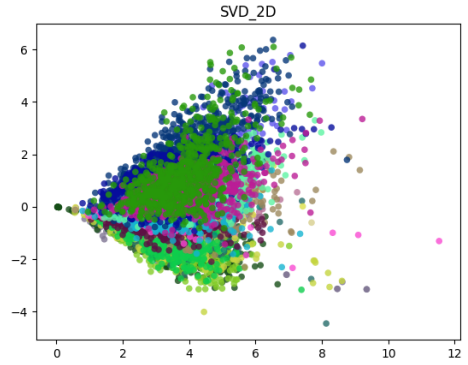


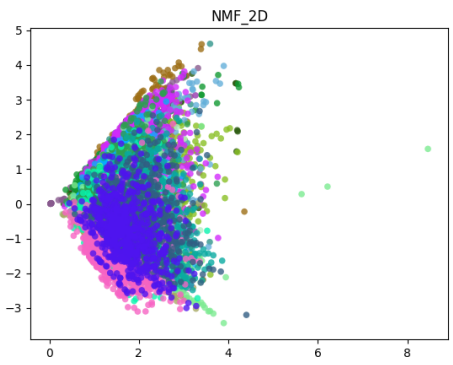
4

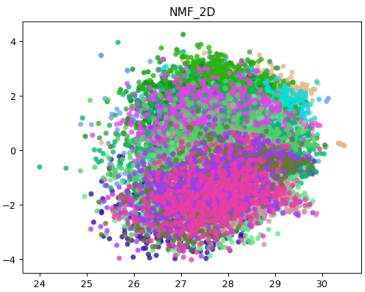


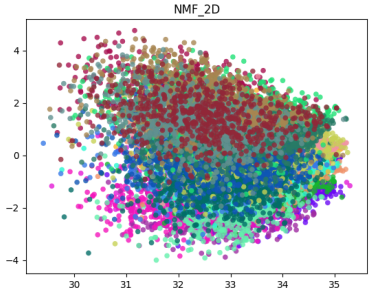
5

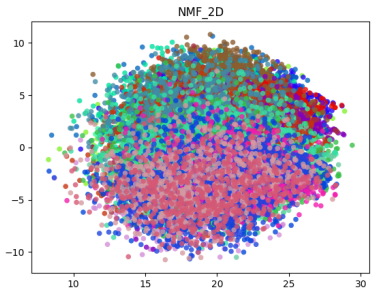
Visualization:











We can also found from visualization that normed NMF can split the data in 2D space better than other methods. And in general, we found that KMeans does bad in 20 cluster tasks. All 5 measures are below 0.5. This also shows that clustering of multiple classes is a difficult task.