EE 219 Project 2
Clustering
Winter 2018

Xiaohan Wang

Yang Tang

Introduction

In this project, we did clustering algorithms on textual dataset "20 Newsgroup" to capture the dependencies among the documents and find proper representations of the data pretending not knowing the class labels of the documents. Then we use the class labels as ground truth to evaluate the performance of the clustering task. Besides, we tried different preprocess methods to improve the performance of the clustering algorithms.

In project 1, we performed classification on the same dataset. These two methods, classification and clustering, though, are not the same cases. Their goals are similar in that they both need to categorize the data. However, classification is a priori algorithm. We have some given categories first, then we need to train and extract the feature of those categories, and finally utilize those features to classify the testing data into proper given categories. Clustering makes a different story. Clustering algorithms are unsupervised methods for finding groups of data points that have similar representations in a proper space. It has no priori labeling(grouping) and we need to predict, test and correct the categories iteratively until all testing data are clustered into proper clusters in convergence.

In this project, we use K-means clustering algorithm. Its idea is similar to least squares method. We performed the following steps to minimize the least squares of distances between each data point and the center of the cluster

$$\mathrm{J} = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - \mu_k\|^2$$

Where μ_k is the kth cluster center, and r_{nk} denotes the nth data point is belonging to the kth cluster or not.

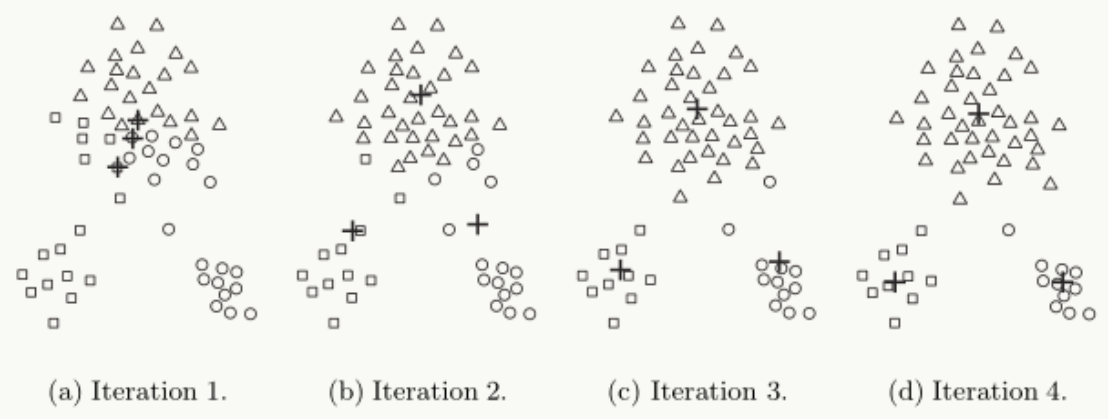
1) Initialize randomly k cluster center positions.

2) Assign each data point to the nearest cluster center.

3) Calculate the position of each center of cluster.

4) Set the center of the cluster to the mean of the data points that are currently within the cluster.

5) Iteratively repeat step 2) to 4) until all data points convergent to kth cluster centers.



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4.

Question 1

Similar to that of project 1, using the same 8 subclasses datasets as table 1, first we extract key terms of documents and quantify how significant a word is to a document by using TFxIDF.

Table 1. Data set classes

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`

The data process is as follow:

1. Remove the punctuations in original data. Punctuations are meaningless to the content of documents, while they can add much noise to feature extraction, so we need to remove those punctuations. We listed all possible punctuations referring to ASCII codes, and then split string using regex tokenizer.
2. Remove newsgroup headers, footer signatures and quotes. Those are irrelevant information to key meaning of the documents.
3. Exclude stop words. Stop words like "I", "she", etc. can appear in all texts and cannot contribute the feature to classify the text, so we need to remove them.
4. Create TFxIDF vector to get feature terms. By setting the min_df parameter, we can set the baseline of appearance that a candidate word need to have. If a word appears less time than min_df, we can ignore it as irrelevant word. The larger the min_df is, the less number of featured terms we will get.

Using all sets (train sets and test sets) data of 8 topics, the dimensions of the TFxIDF matrix is 7882×15464, which means there are 7882 documents with 15464 featured terms extracted.

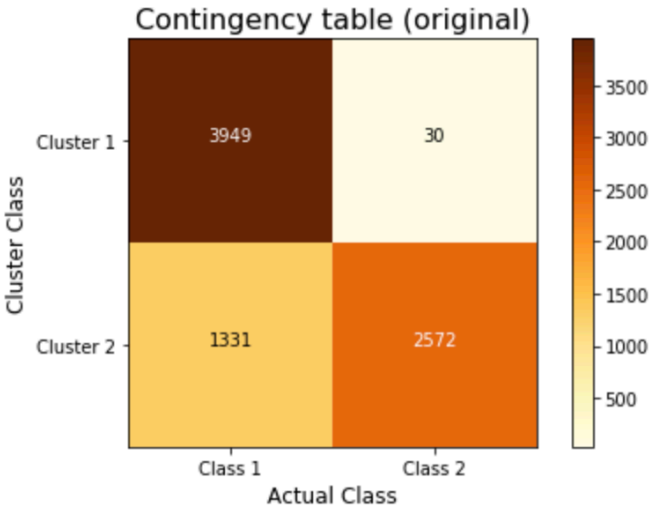
Question 2

In this part, we implemented K-means clustering method setting $\mathrm{k} = 2$. Then we visualized our clustering results using contingency table and evaluated the effects with five factors.

Question a)

In project 1, we used confusion matrix to visualize and evaluate our classification results. In this part, to visualize our clustering results, we need to use contingency table which is similar to confusion matrix. Confusion matrix is used to evaluate supervised data with two factors "actual classes" and "predicted classes". In contingency table, each element A_{ij} denotes the number of data points that are member of class c_i and elements of cluster k_j. Here, c_i can be viewed as "actual classes" and k_j can be viewed as "predicted classes" in confusion matrix. But the difference is that we don't

have given “actual classes” labels and all we need to do is to categorize the data into several clusters with no care of the clusters’ name. So setting with a 2x2 confusion matrix, we may get a diagonal-matrix or a vice-diagonal-matrix. Then we can adjust the matrix by finding a row permutation to get a diagonal-matrix. Below are our contingency table (diagonal-confusion matrix):



From the table above, we can see that through K-means clustering algorithm, it can hardly get to a diagonal matrix, which indicates that the clustering algorithm is not that ideal to purely cluster the data.

Question b)

In order to make a concrete comparison of different clustering results, we evaluated them through 5 measures: homogeneity score, completeness score, V-measure, adjusted rand score and adjusted mutual info score. Below are our results for K-means clustering results:

Table 2. 5 measures for K-means clustering results

Measures	Values
Homogeneity score	0.4243358018332555
Completeness score	0.4636994594803498
V-measure	0.4431452004668879
Adjusted rand score	0.428504440021173
Adjusted mutual info score	0.4242830978688626

From the table above, we can see that the 5 measures score are distributed in the range between 0.42 and 0.47 with the best value should be 1. The more the contingency table is like a diagonal matrix, the better our clustering algorithm clustered the data points purely and correctly. Clearly, our clustering algorithm now is not as good as possible, and it's not very stable with some fluctuations in each time's results.

Question 3

Question i

From the results in question 2, we can see that though the contingency table seems like a diagonal matrix, its performances is still not ideal. And this may be caused by the high dimension of TFxIDF matrix. Thus, in the next step, we need to reduce its dimension and find the effective dimension of the data through inspection of the top singular values, i.e. choosing the number of principal components.

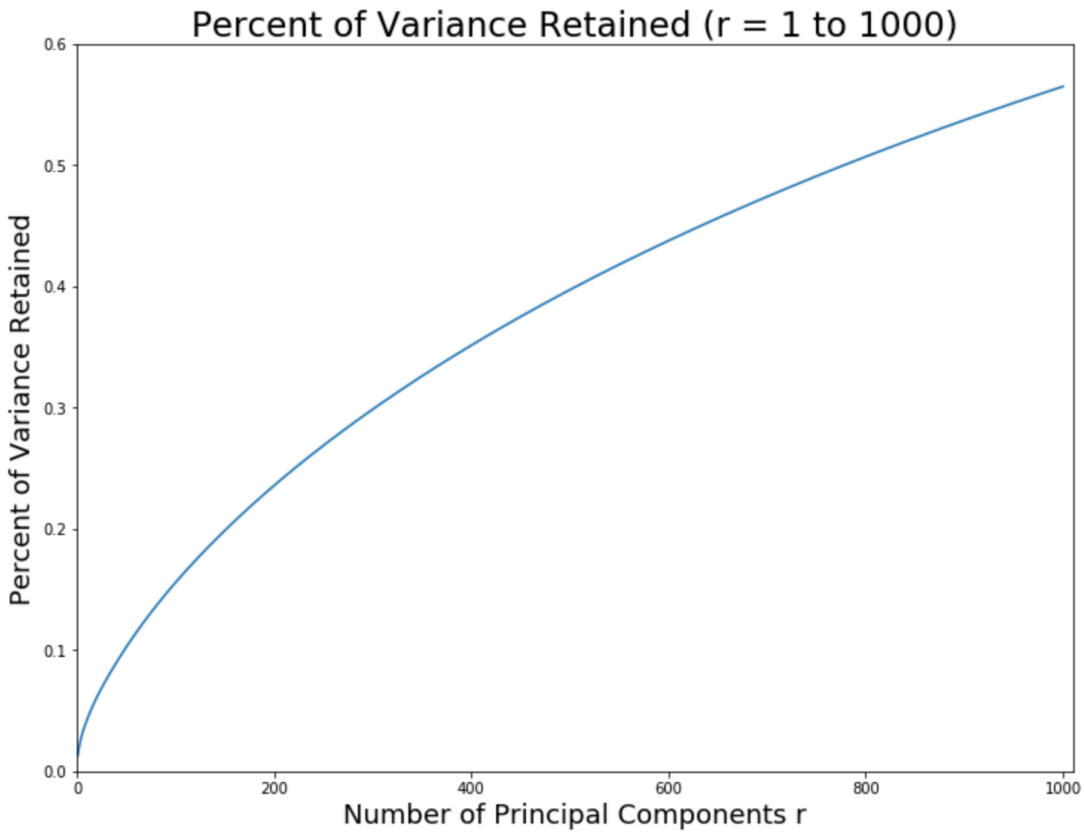
There is an equation in machine learning to evaluate the effects of dimension r

$$err = \frac{\frac{1}{m}\sum_1^m \left\| x_i - x_{i,approx} \right\|^2}{\frac{1}{m}\sum_1^m \|x_i\|^2}$$

if err is less than 0.01, than we can say that "99% of variance is retained". First we set $r = 1$, and calculate its err and check if it can satisfy the requirement. If not, then we increase r by 1 each time until finding the satisfied value. Here we can use truncated SVD to easily calculate err by

$$err = \frac{\sum_1^r S_{ii}}{\sum_1^n S_{ii}}$$

Below are our results with having r from 1 to 1000:



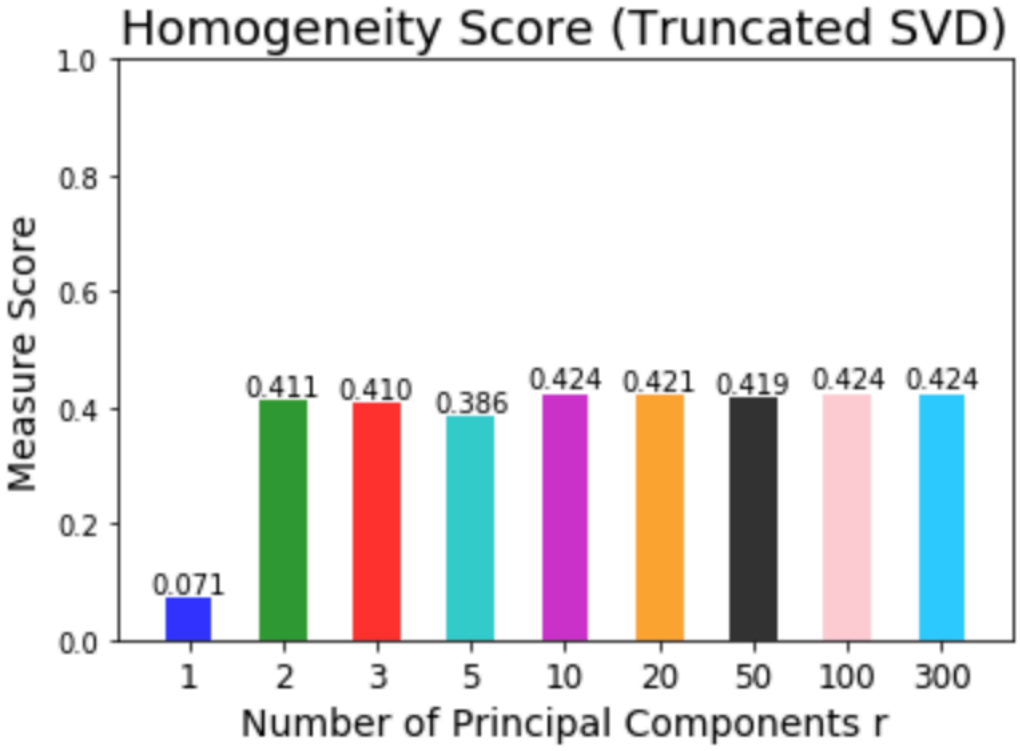
From the figure above, we find that as r increases, the percentage of variance the top r principal components can retain increases monotonically.

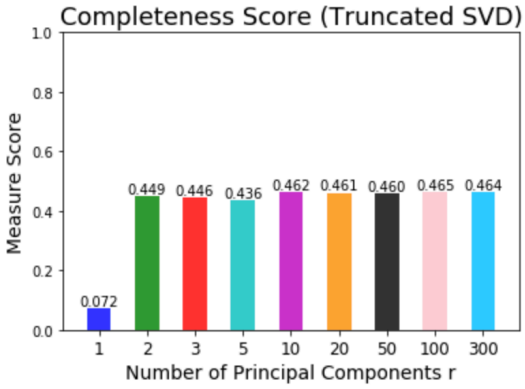
Question ii

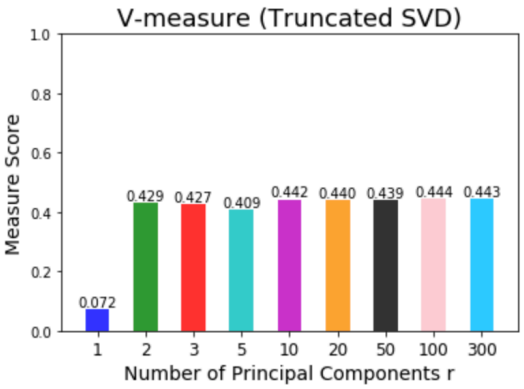
In this part, we use two methods to reduce dimension and compare their effects. One is Latent Semantic Indexing (LSI), the other is Non-Negative Matrix Factorization (NMF).

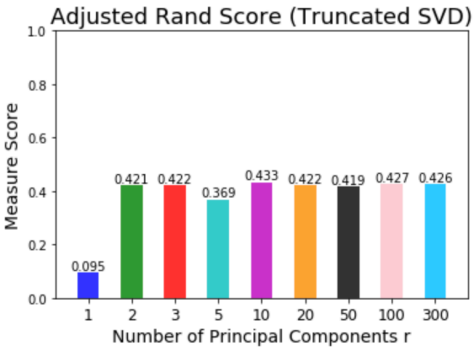
1. Truncated SVD (LSI) / PCA

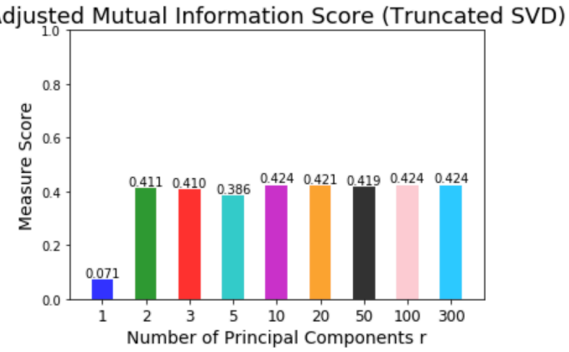
1) 5 measures scores v.s. r



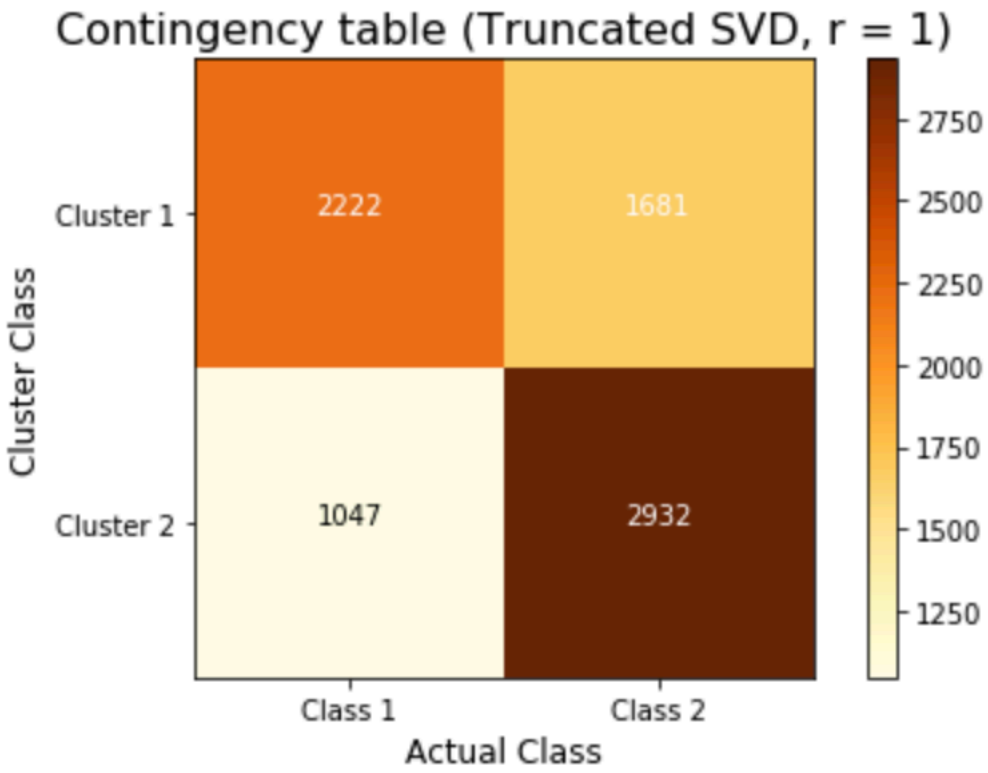


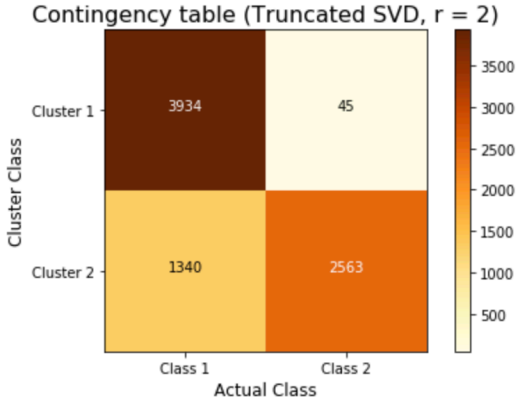


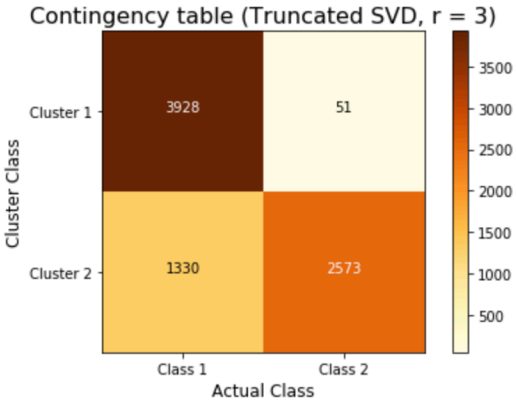


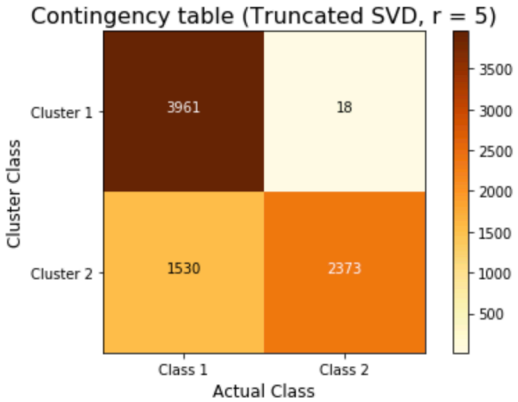


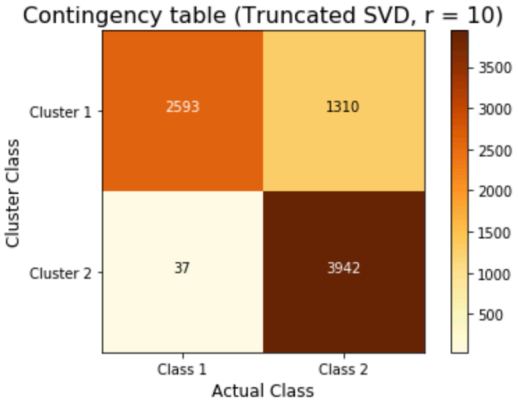
2) Contingency matrices

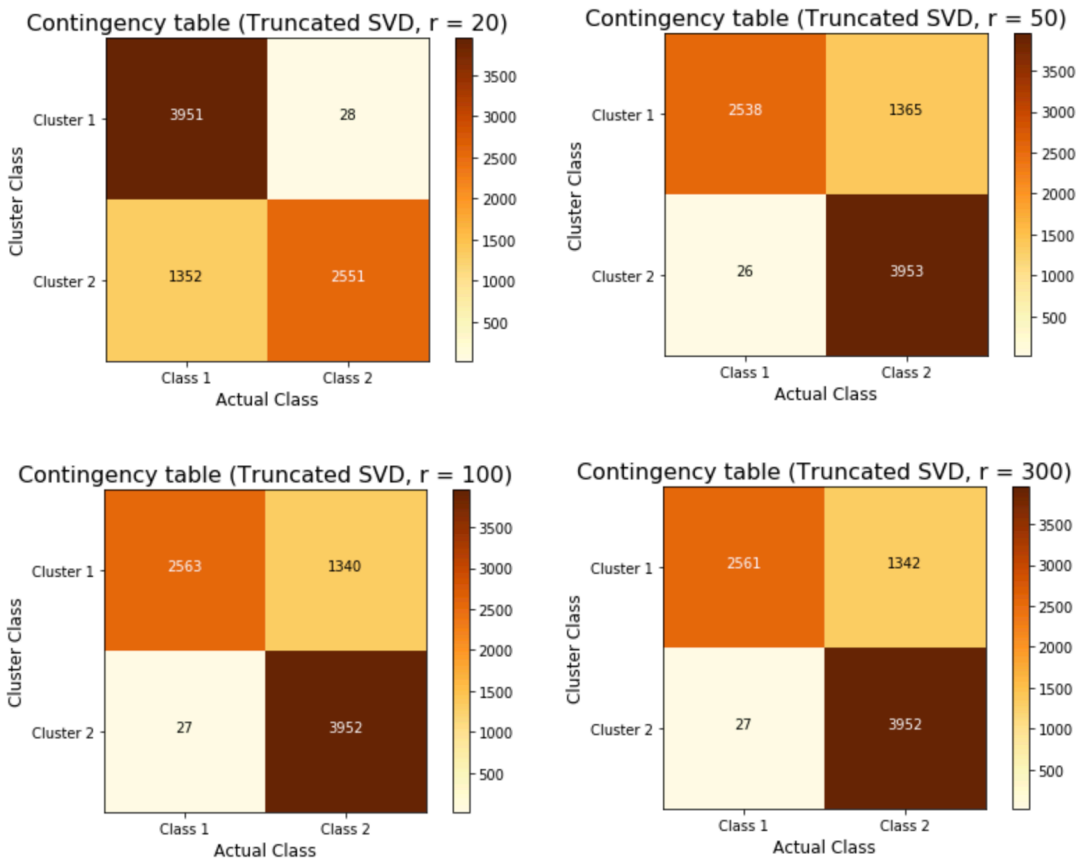








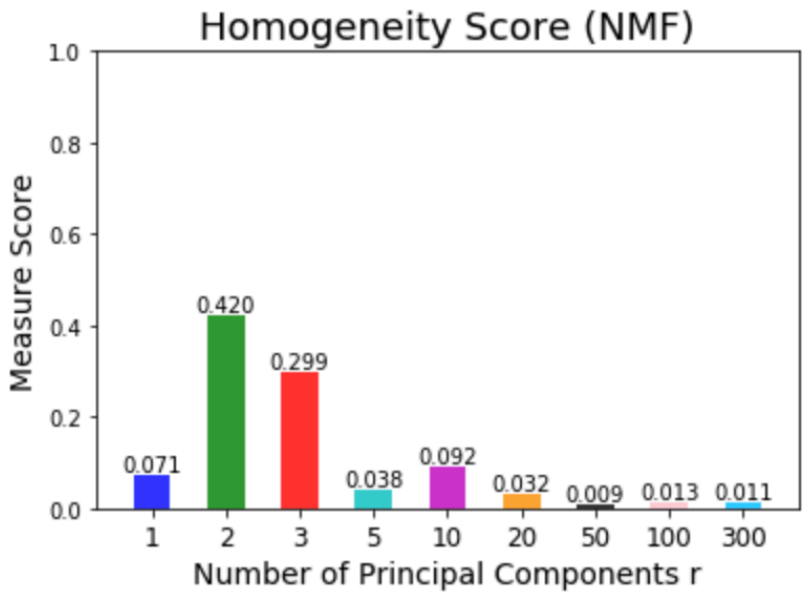


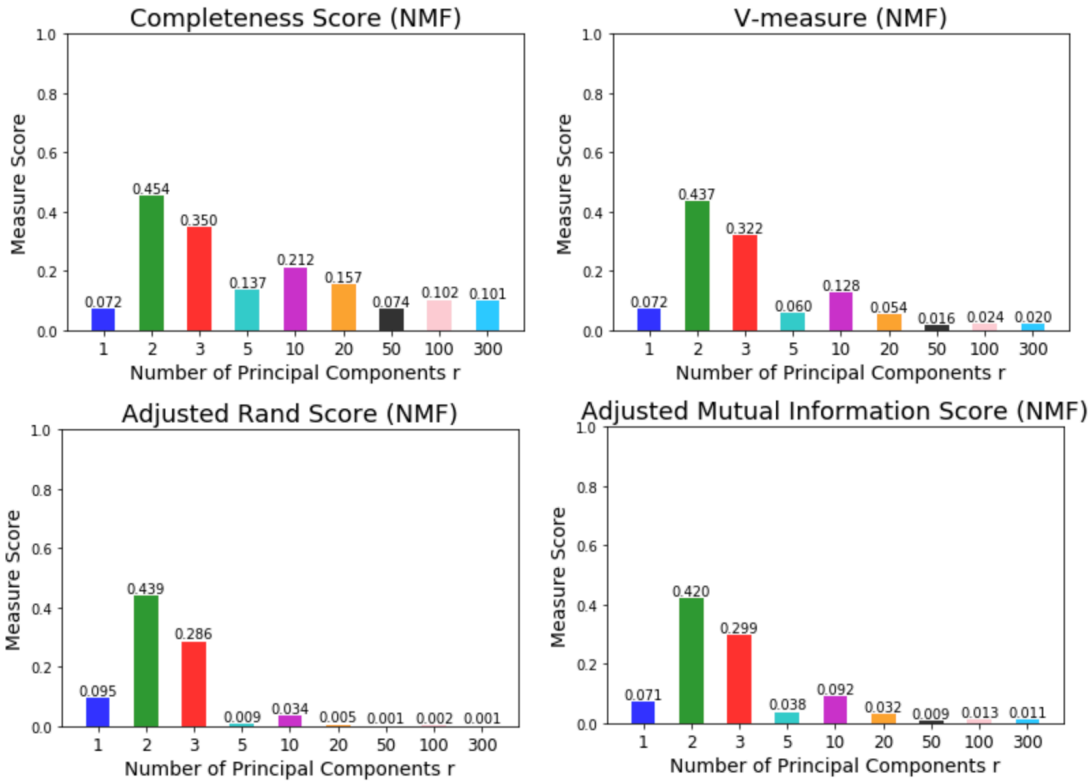


Comparing 5 measures and contingency tables comprehensively, we find that the best r choice for Truncated SVD is 10.

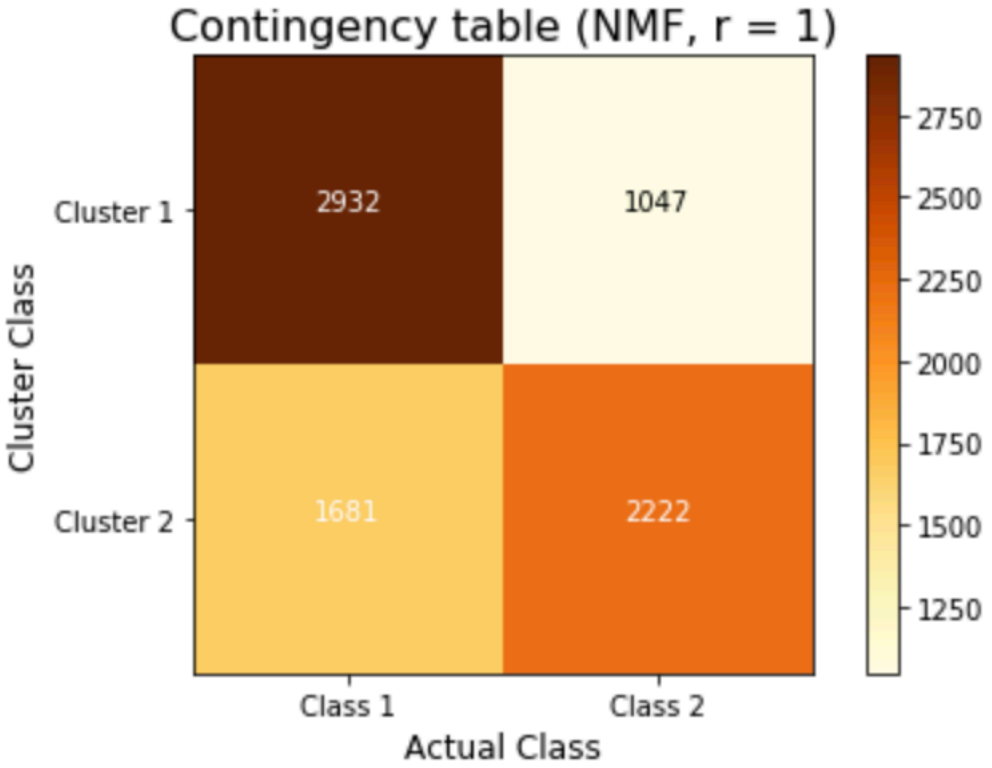
2. Non-Negative Matrix Factorization (NMF)

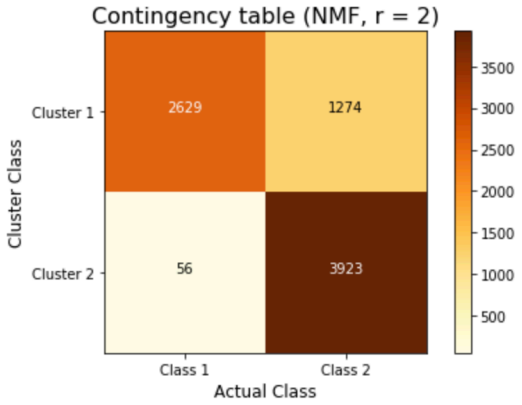
1) 5 measures scores v.s. r

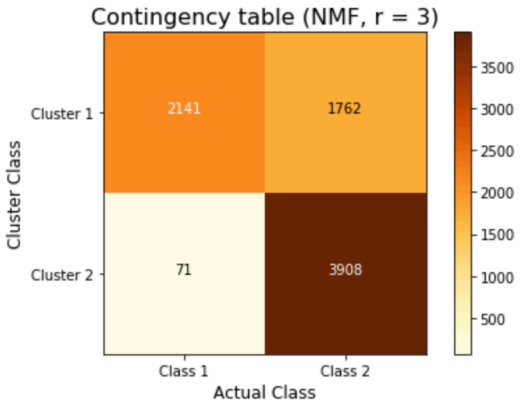


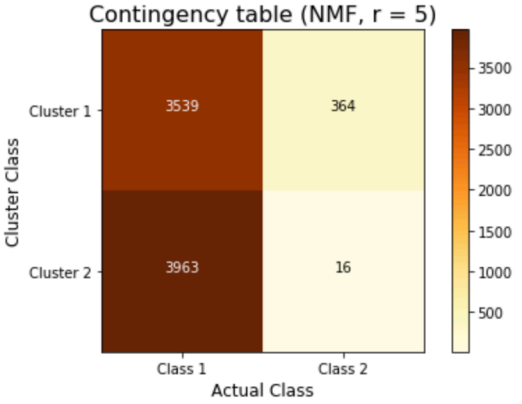


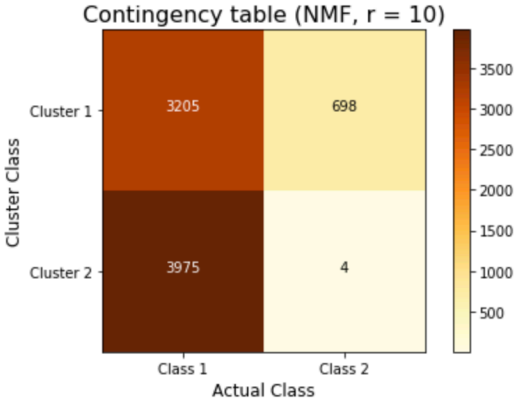
2) Contingency matrices

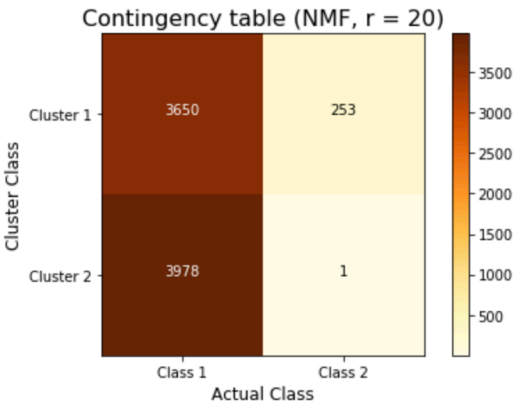


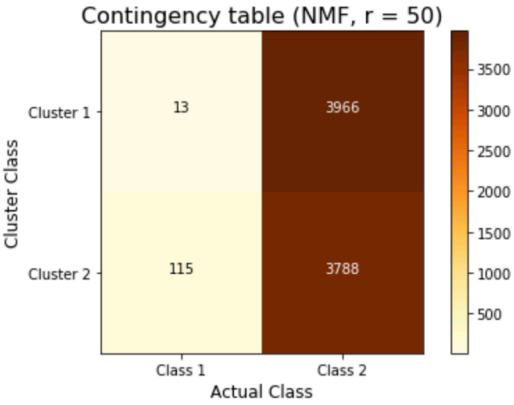

Contingency table (NMF, r = 2)
Cluster 1
Cluster 2
2629
1274
56
3923
Cluster Class
Class 1
Class 2
Actual Class
3500
3000
2500
2000
1500
1000
500

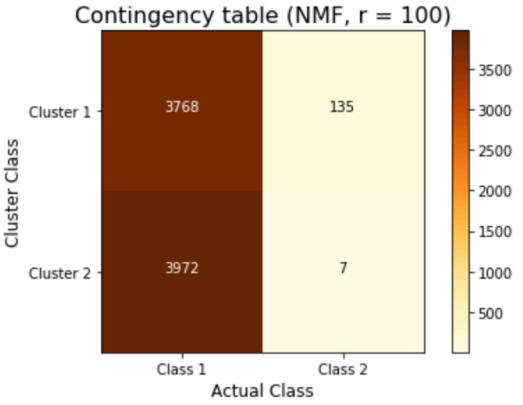

Contingency table (NMF, r = 3)
Cluster 1
Cluster 2
2141
1762
71
3908
Cluster Class
Class 1
Class 2
Actual Class
3500
3000
2500
2000
1500
1000
500

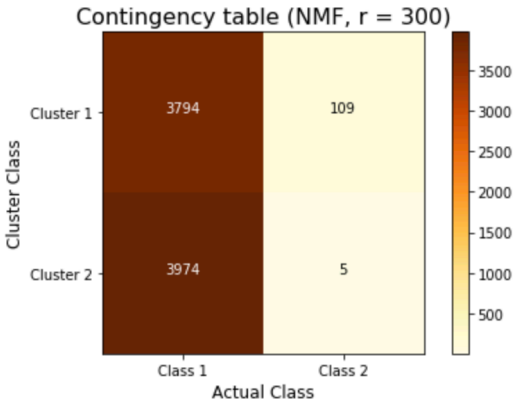

Contingency table (NMF, r = 5)
Cluster 1
Cluster 2
3539
364
3963
16
Cluster Class
Class 1
Class 2
Actual Class
3500
3000
2500
2000
1500
1000
500


Contingency table (NMF, r = 10)
Cluster 1
Cluster 2
3205
698
3975
4
Cluster Class
Class 1
Class 2
Actual Class
3500
3000
2500
2000
1500
1000
500


Contingency table (NMF, r = 20)
Cluster 1
Cluster 2
3650
253
3978
1
Cluster Class
Class 1
Class 2
Actual Class
3500
3000
2500
2000
1500
1000
500


Contingency table (NMF, r = 50)
Cluster 1
Cluster 2
13
3966
115
3788
Cluster Class
Class 1
Class 2
Actual Class
3500
3000
2500
2000
1500
1000
500


Contingency table (NMF, r = 100)
Cluster 1
Cluster 2
3768
135
3972
7
Cluster Class
Class 1
Class 2
Actual Class
3500
3000
2500
2000
1500
1000
500


Contingency table (NMF, r = 300)
Cluster 1
Cluster 2
3794
109
3974
5
Cluster Class
Class 1
Class 2
Actual Class
3500
3000
2500
2000
1500
1000
500

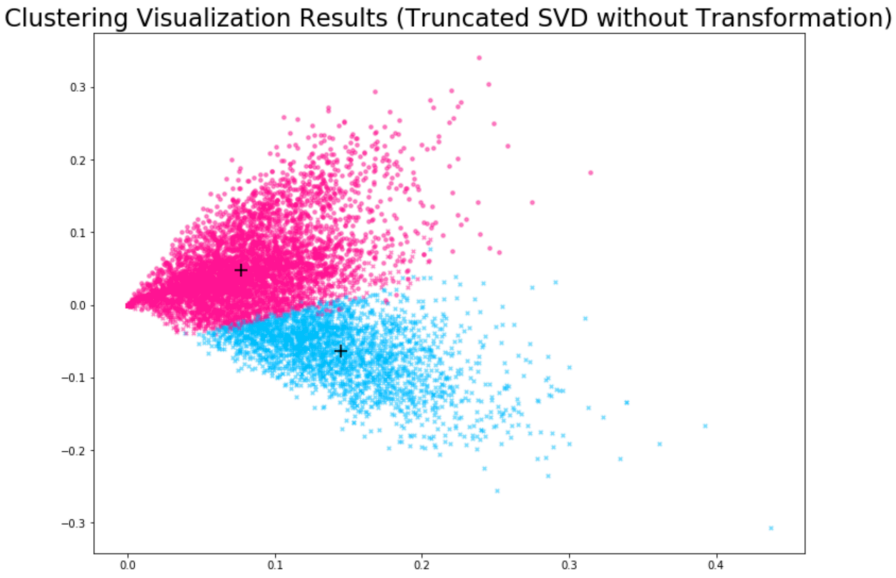
Comparing the 5 measure scores and contingency tables above, we can easily conclude that the best r choice for NMF is 2.

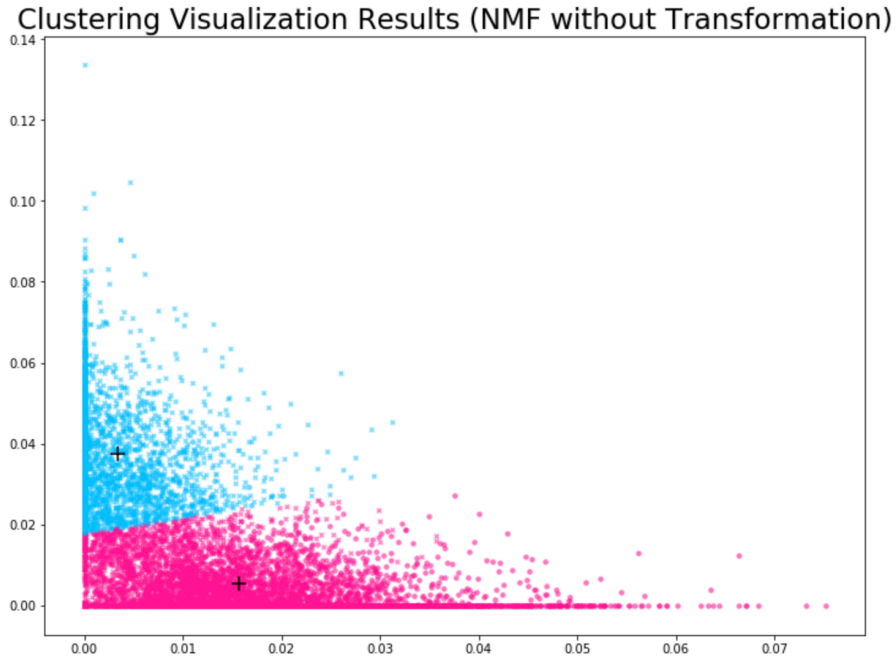
By observing the figures of 5 measures scores, we find that they are non-monotonic as r increases. This is because when r is at too low values, the information that the singular value contains is limited so that we cannot guarantee the correctness of the clustering, which caused the measure scores low at first. As r increases, it contains more information to help clustering accurately, which contributes to the increment of the measure scores. While when r raised up with too big value, it will create a high dimensional matrix. This brings the same problem as in question 2 that the Euclidean distance is not a good metric anymore, in the sense that the distances between data points tends to be almost the same. This leads to the decreasing clustering performance evaluated by 5 measures. Therefore, the 5 measure scores figures are not monotonic, and we need to find the best dimension r in different cases.

Question 4

Question a)

According to the results in question 3, we set the reduction dimension values as: Truncated SVD with $r = 10$, NMF with $r = 2$. Below are our clustering visualization results:

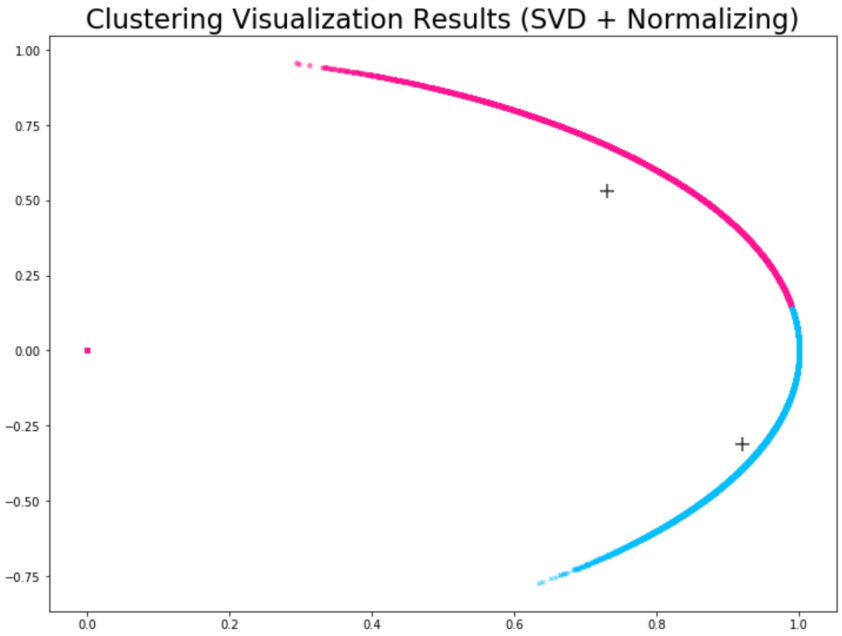


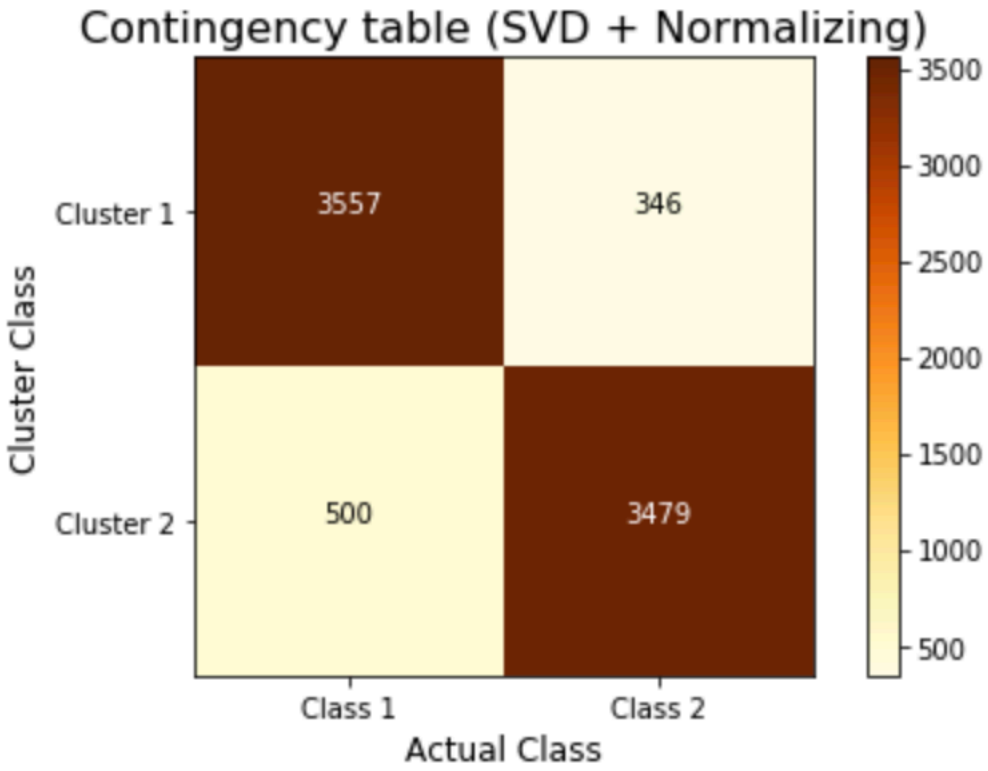


Question b)

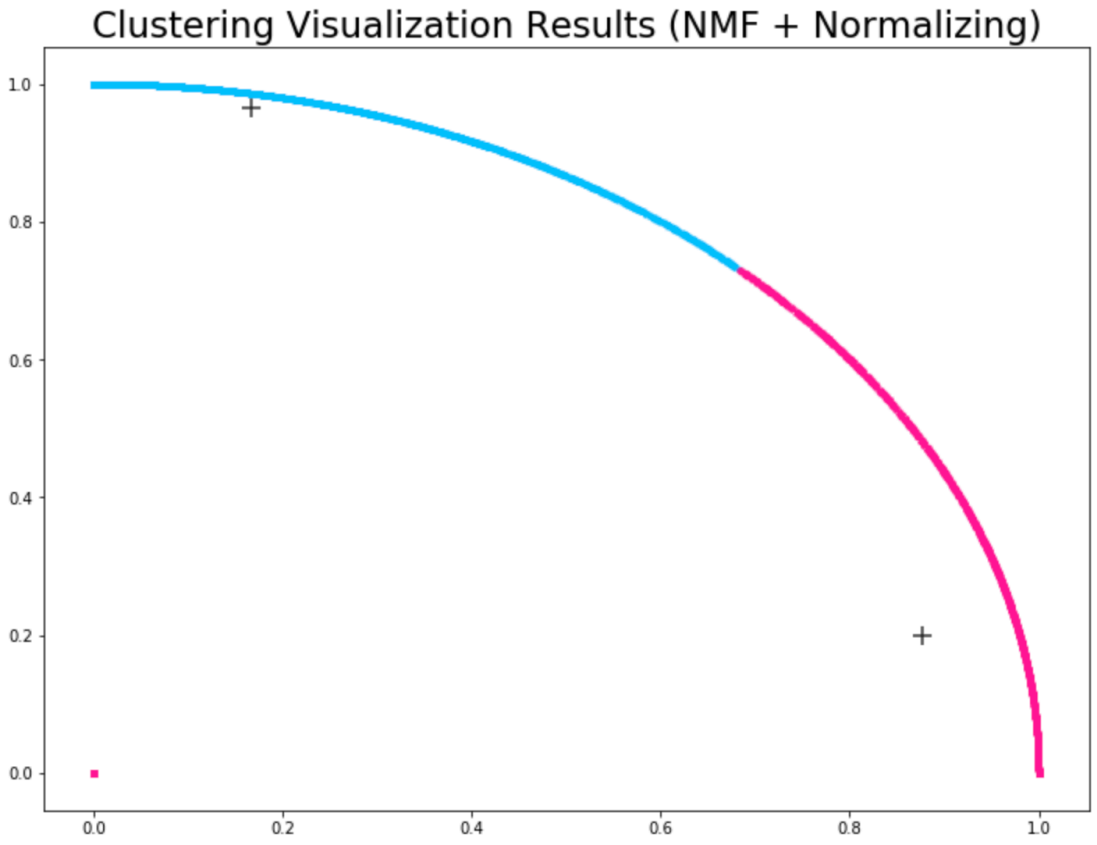
In this part, we tried three methods to improve clustering performance in question a. Same as that in last part, we set the reduction dimension values as Truncated SVD with $r = 10$, NMF with $r = 2$. Below are our clustering visualization results and corresponding contingency tables:

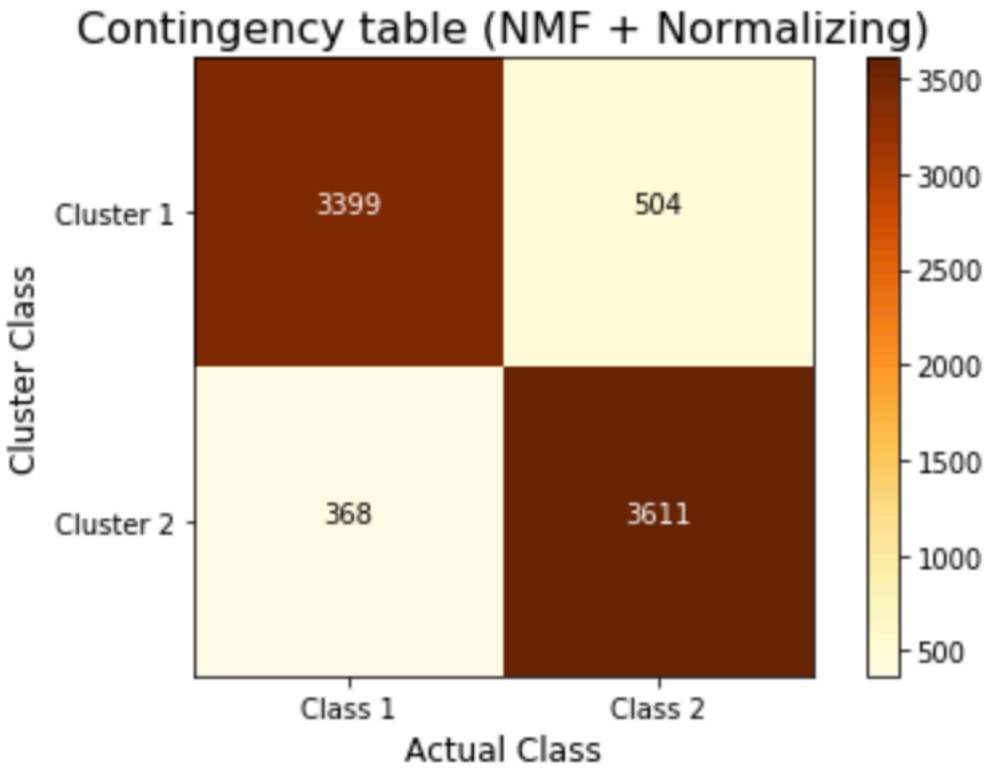
1) Truncated SVD + Normalizing



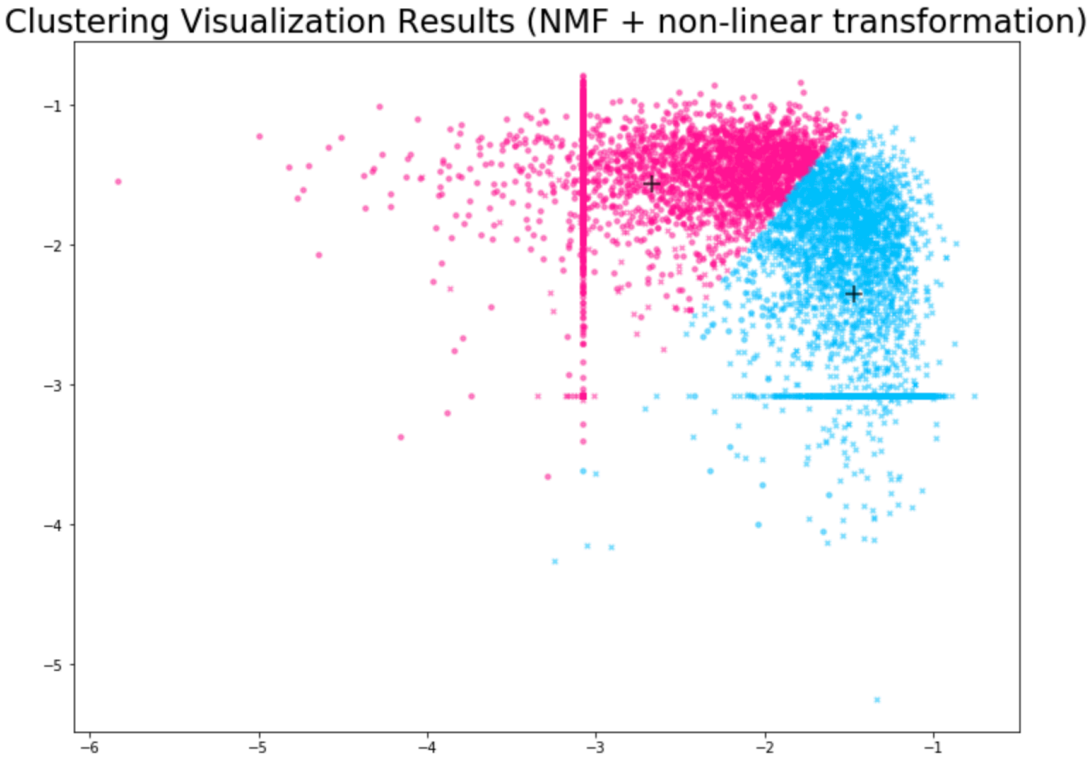


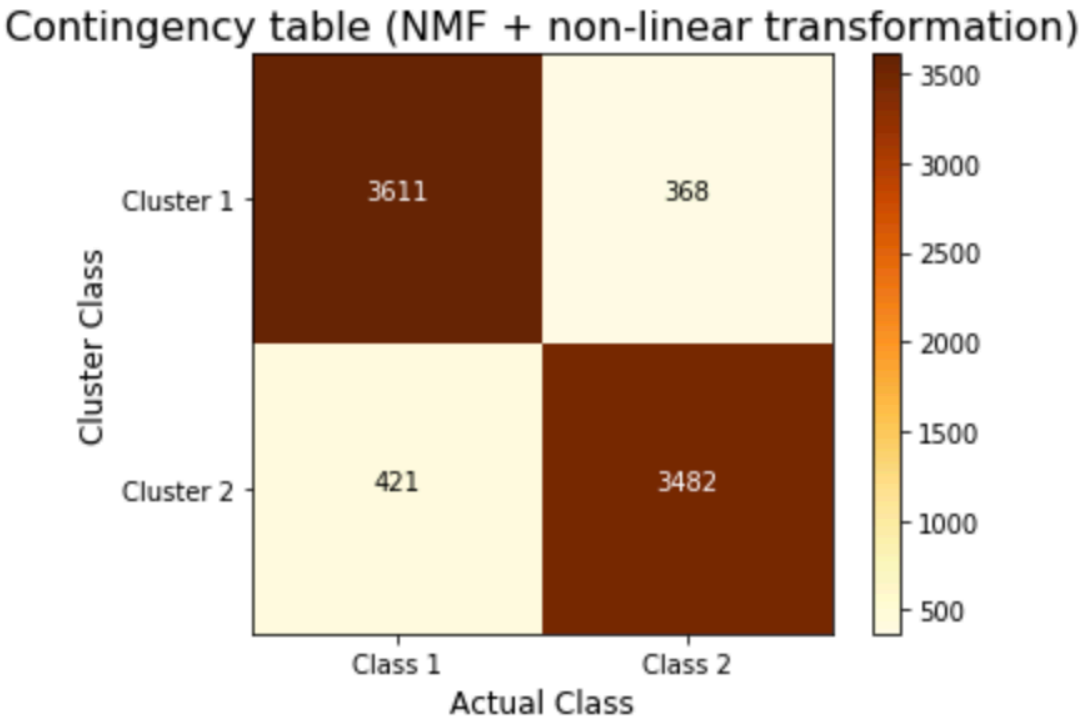
2) NMF + Normalizing



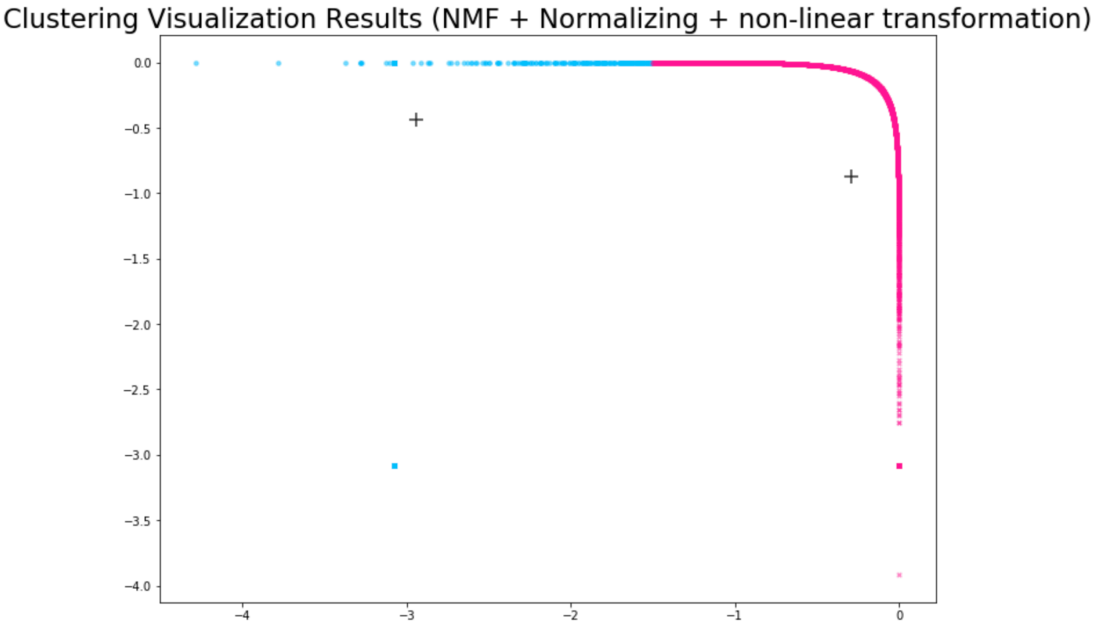


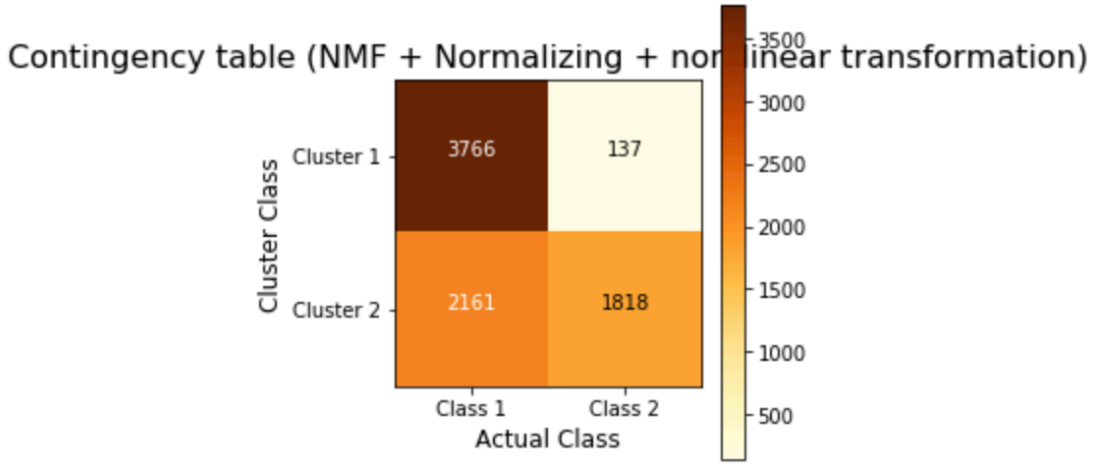
3) NMF + Non-linear Transformation





4) NMF + Normalizing + Non-linear Transformation





From the figures above, we can see that with non-linear (logarithm) transformation, the clustering performance is improved greatly. With only applying normalizing, we can also improve the performance. While if we combine both transformation methods with NMF, the clustering results appeared not ideal as we expect.

The reason why logarithm transformation may increase the clustering results is that the relevance of a term or a document is not proportional to term frequency. With sub-linear logarithm method can help eliminate the effect so that the influence of very large or very small values is amortized, which can contribute to a better performance of clustering.

Question 5

In the final part of project, we did the same dimension reduction then K-means clustering procedure to cluster the documents from 20 topics into 6 clusters (as indicated in table 3) based on the classes they belong to. We assigned label 0 to label 5 to represent actual clusters of original data. And we applied different dimension reduction methods which had been improved in the last part of the project on the dataset before K-means clustering. Also, we chose various numbers of terms to represent each document.

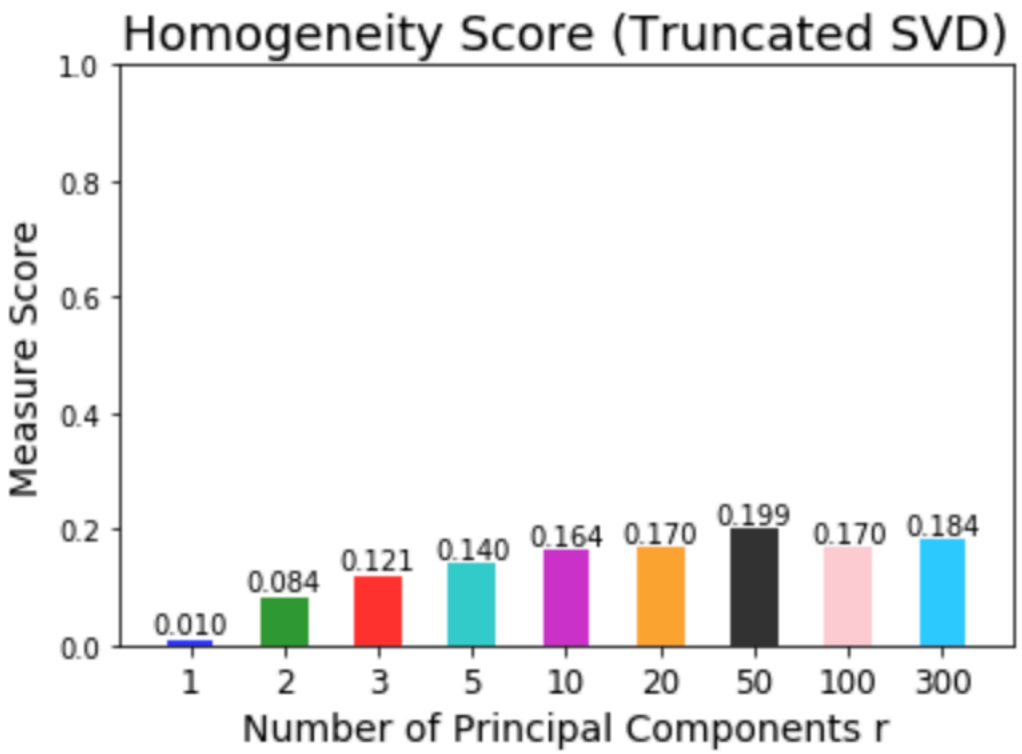
Table 3. All data sets classes

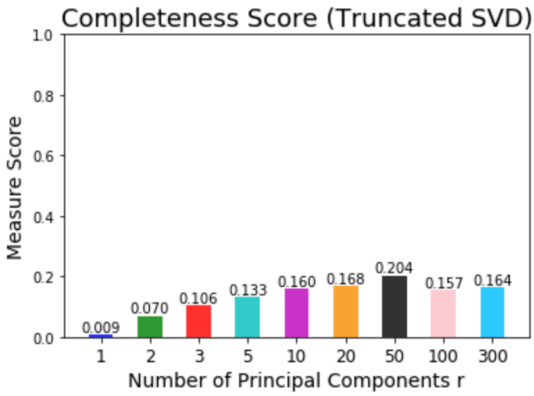
Class 1	Class 2	Class 3
comp.graphics	rec.autos	sci.crypt
comp.os.ms-windows.misc	rec.motorcycles	sci.electronics
comp.sys.ibm.pc.hardware	rec.sport.baseball	sci.med
comp.sys.mac.hardware	rec.sport.hockey	sci.space
Comp.windows.x		
Class 4	Class 5	Class 6
misc.forsale	talk.politics.misc	talk.religion.misc
	talk.politics.guns	alt.atheism
	talk.politics.mideast	soc.religion.christian

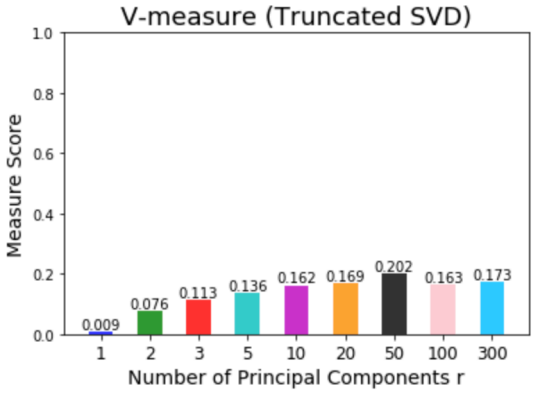
Below are our results:

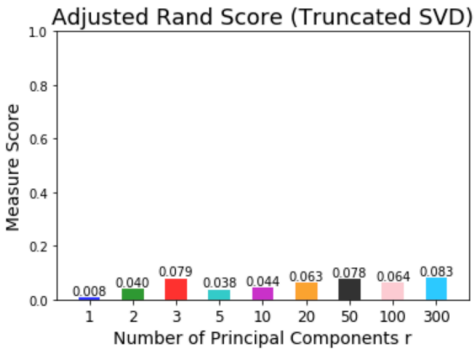
1. Truncated SVD + Normalizing

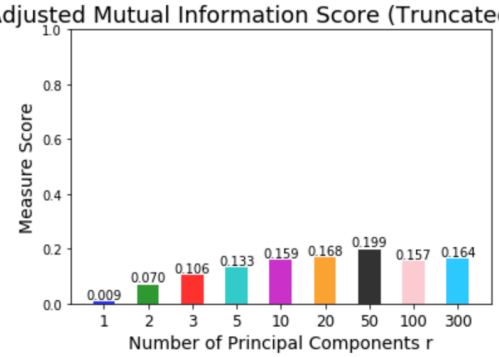
   1) 5 measures scores



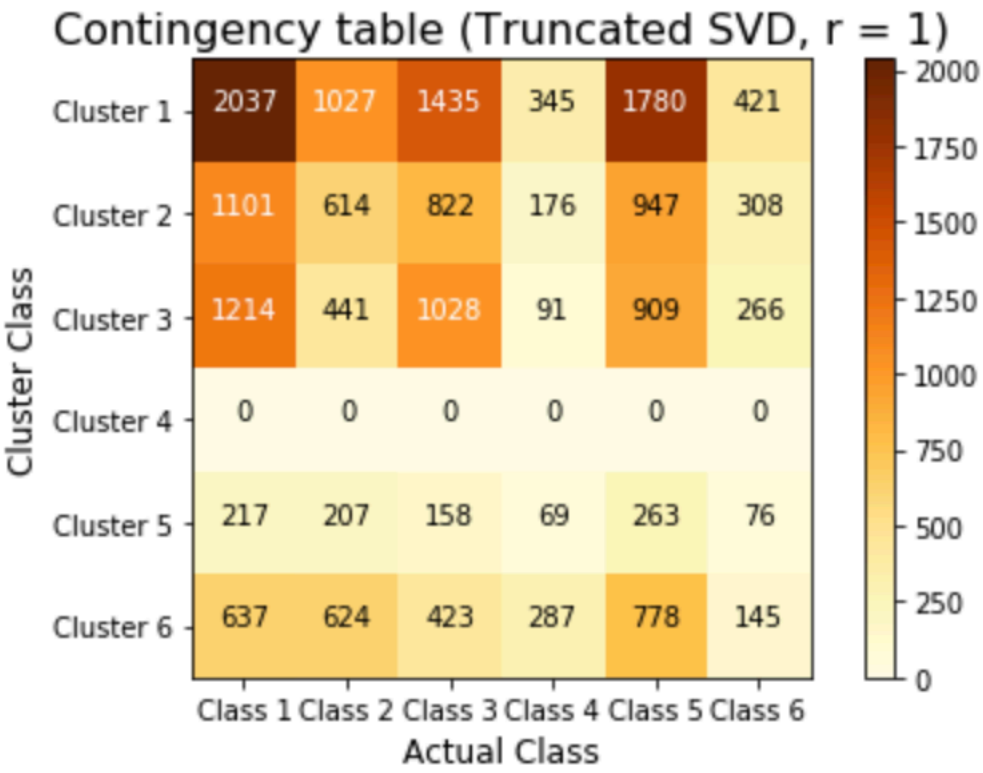


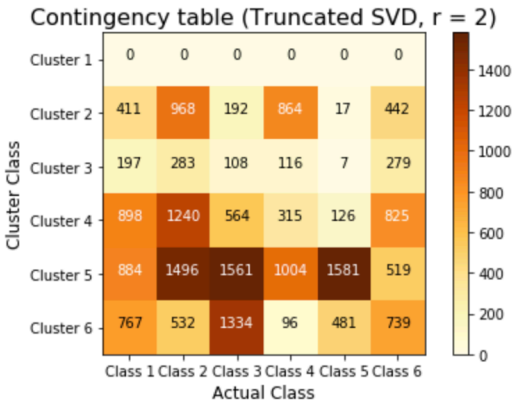


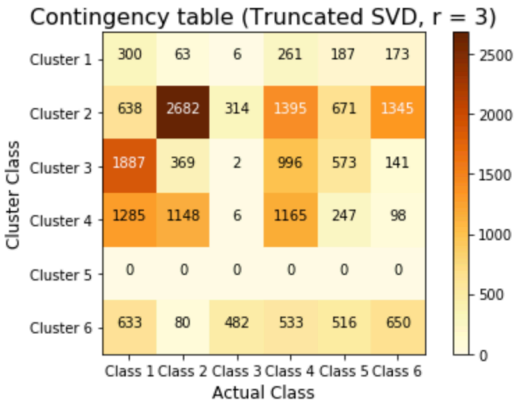


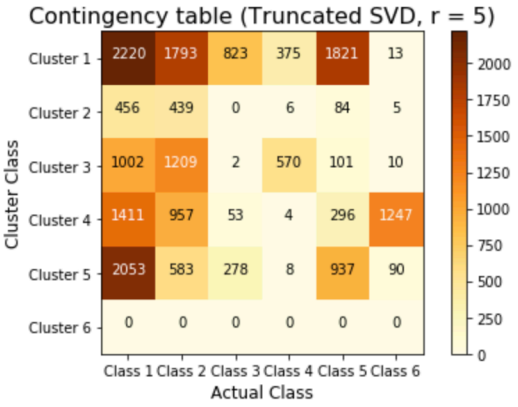


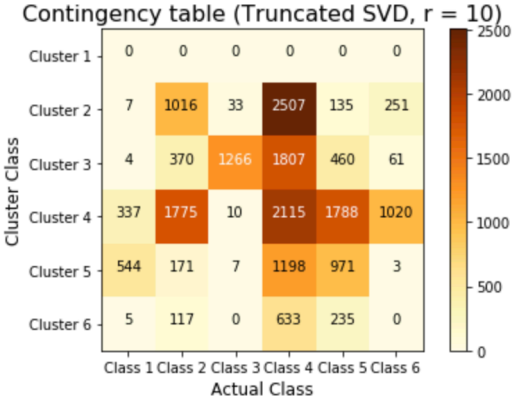
2) Contingency tables

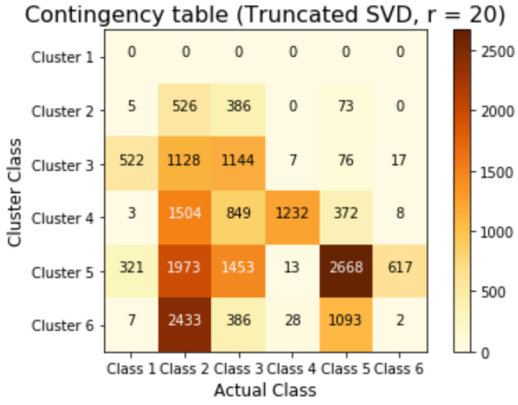


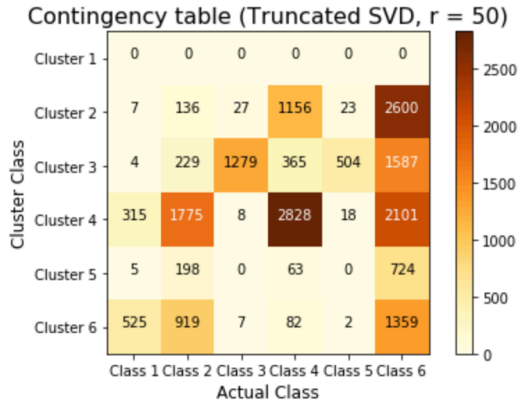

Contingency table (Truncated SVD, r = 2)
Cluster 1 0 0 0 0 0 0
Cluster 2 411 968 192 864 17 442
Cluster 3 197 283 108 116 7 279
Cluster 4 898 1240 564 315 126 825
Cluster 5 884 1496 1561 1004 1581 519
Cluster 6 767 532 1334 96 481 739
Cluster Class
Class 1 Class 2 Class 3 Class 4 Class 5 Class 6
Actual Class
1400 1200 1000 800 600 400 200 0

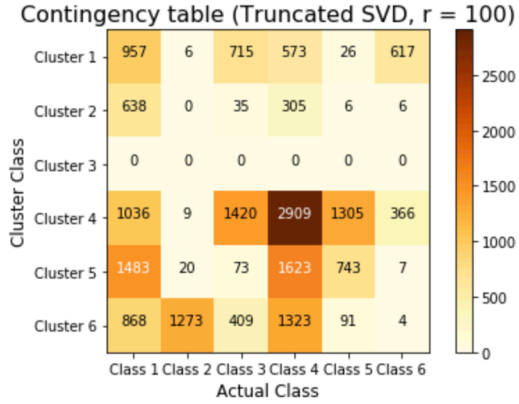

Contingency table (Truncated SVD, r = 3)
Cluster 1 300 63 6 261 187 173
Cluster 2 638 2682 314 1395 671 1345
Cluster 3 1887 369 2 996 573 141
Cluster 4 1285 1148 6 1165 247 98
Cluster 5 0 0 0 0 0 0
Cluster 6 633 80 482 533 516 650
Cluster Class
Class 1 Class 2 Class 3 Class 4 Class 5 Class 6
Actual Class
2500 2000 1500 1000 500 0

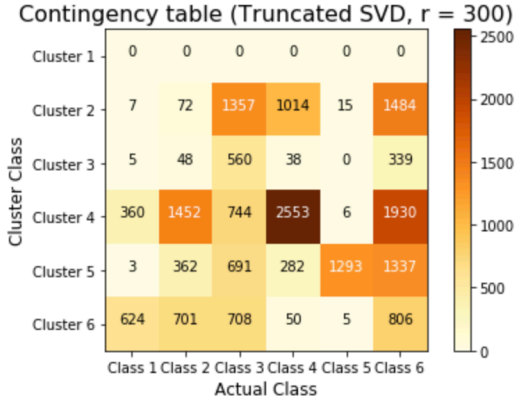

Contingency table (Truncated SVD, r = 5)
Cluster 1 2220 1793 823 375 1821 13
Cluster 2 456 439 0 6 84 5
Cluster 3 1002 1209 2 570 101 10
Cluster 4 1411 957 53 4 296 1247
Cluster 5 2053 583 278 8 937 90
Cluster 6 0 0 0 0 0 0
Cluster Class
Class 1 Class 2 Class 3 Class 4 Class 5 Class 6
Actual Class
2000 1750 1500 1250 1000 750 500 250 0


Contingency table (Truncated SVD, r = 10)
Cluster 1 0 0 0 0 0 0
Cluster 2 7 1016 33 2507 135 251
Cluster 3 4 370 1266 1807 460 61
Cluster 4 337 1775 10 2115 1788 1020
Cluster 5 544 171 7 1198 971 3
Cluster 6 5 117 0 633 235 0
Cluster Class
Class 1 Class 2 Class 3 Class 4 Class 5 Class 6
Actual Class
2500 2000 1500 1000 500 0

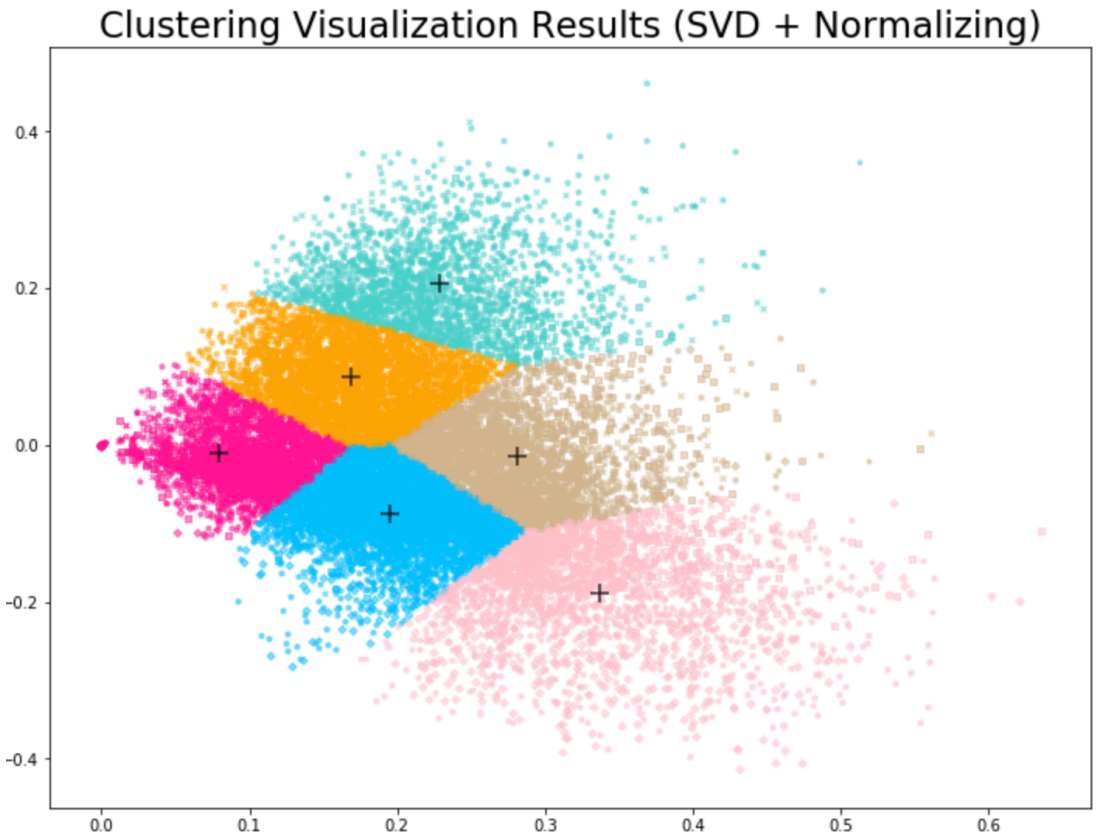

Contingency table (Truncated SVD, r = 20)
Cluster 1 0 0 0 0 0 0
Cluster 2 5 526 386 0 73 0
Cluster 3 522 1128 1144 7 76 17
Cluster 4 3 1504 849 1232 372 8
Cluster 5 321 1973 1453 13 2668 617
Cluster 6 7 2433 386 28 1093 2
Cluster Class
Class 1 Class 2 Class 3 Class 4 Class 5 Class 6
Actual Class
2500 2000 1500 1000 500 0


Contingency table (Truncated SVD, r = 50)
Cluster 1 0 0 0 0 0 0
Cluster 2 7 136 27 1156 23 2600
Cluster 3 4 229 1279 365 504 1587
Cluster 4 315 1775 8 2828 18 2101
Cluster 5 5 198 0 63 0 724
Cluster 6 525 919 7 82 2 1359
Cluster Class
Class 1 Class 2 Class 3 Class 4 Class 5 Class 6
Actual Class
2500 2000 1500 1000 500 0


Contingency table (Truncated SVD, r = 100)
Cluster 1 957 6 715 573 26 617
Cluster 2 638 0 35 305 6 6
Cluster 3 0 0 0 0 0 0
Cluster 4 1036 9 1420 2909 1305 366
Cluster 5 1483 20 73 1623 743 7
Cluster 6 868 1273 409 1323 91 4
Cluster Class
Class 1 Class 2 Class 3 Class 4 Class 5 Class 6
Actual Class
2500 2000 1500 1000 500 0

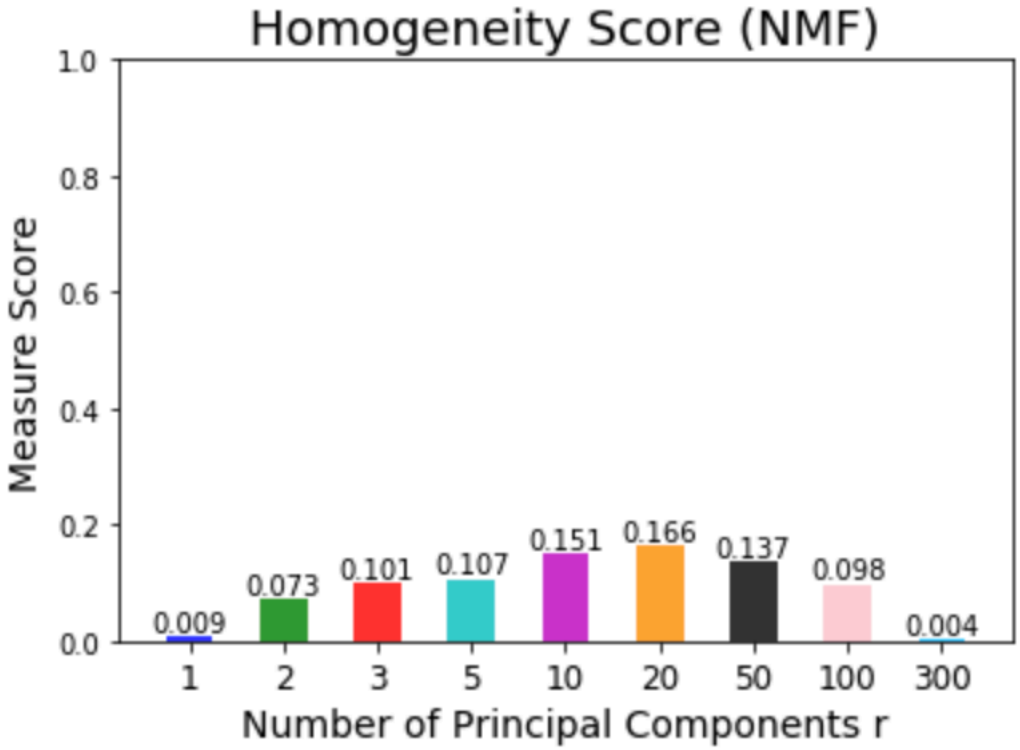

Contingency table (Truncated SVD, r = 300)
Cluster 1 0 0 0 0 0 0
Cluster 2 7 72 1357 1014 15 1484
Cluster 3 5 48 560 38 0 339
Cluster 4 360 1452 744 2553 6 1930
Cluster 5 3 362 691 282 1293 1337
Cluster 6 624 701 708 50 5 806
Cluster Class
Class 1 Class 2 Class 3 Class 4 Class 5 Class 6
Actual Class
2500 2000 1500 1000 500 0

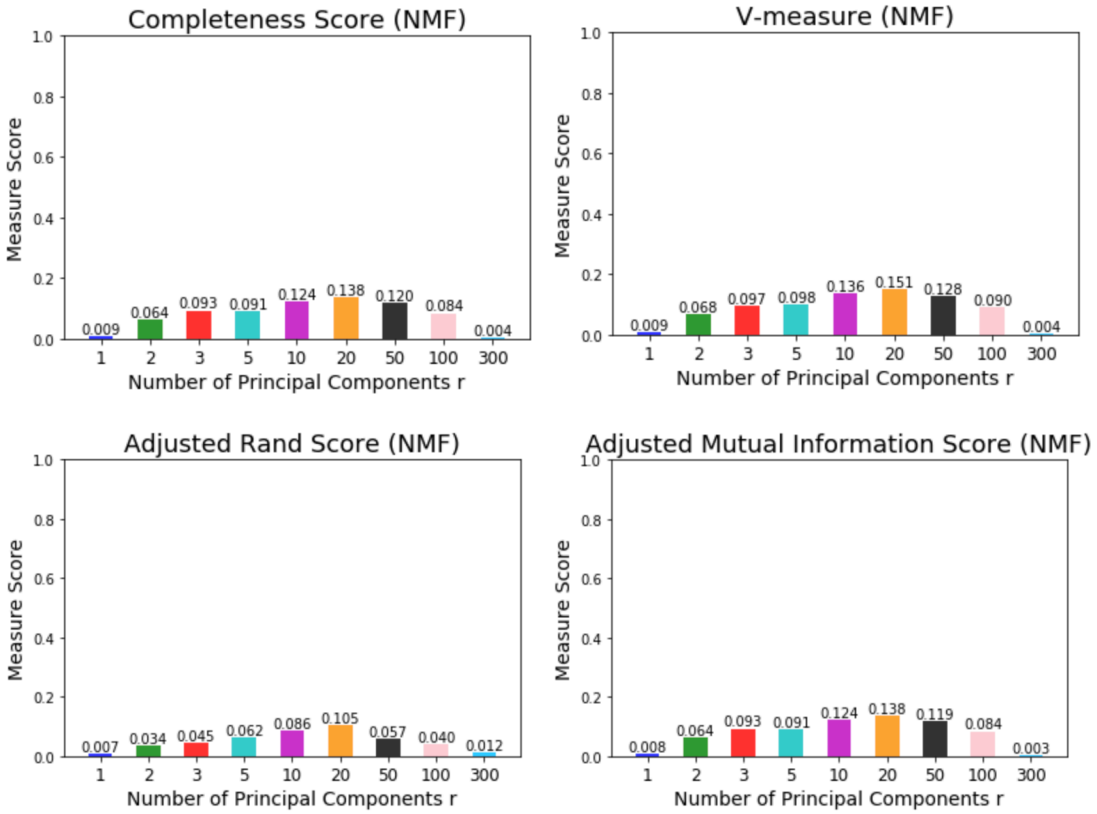
3）Clustering visualization results



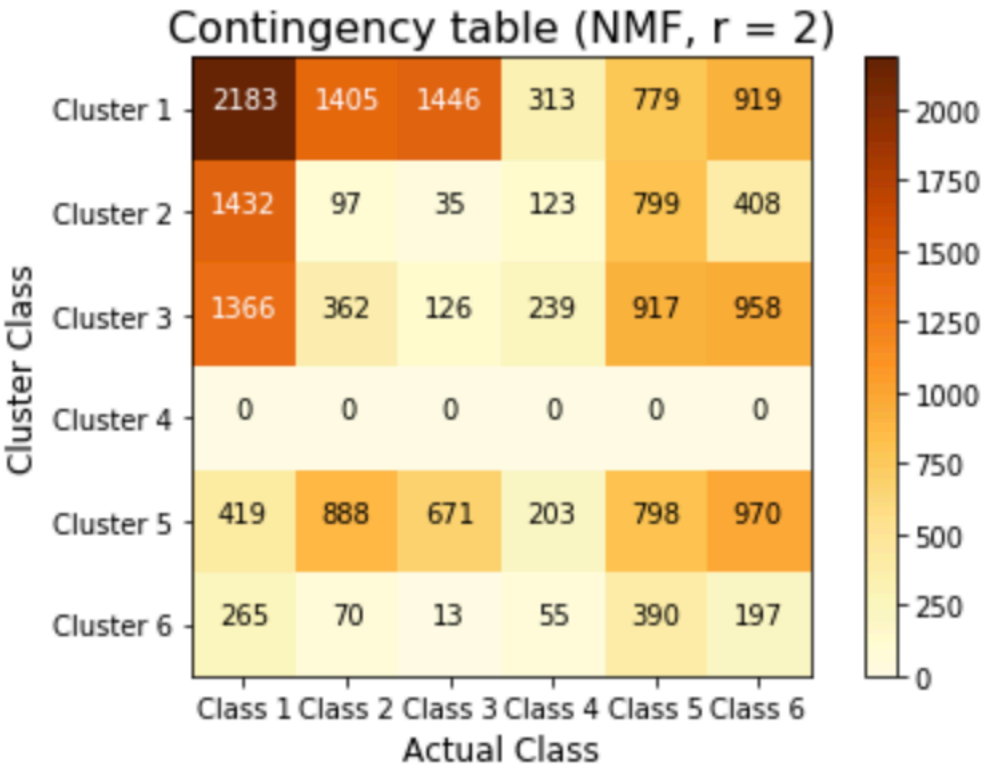
2. NMF + non-linear transformation

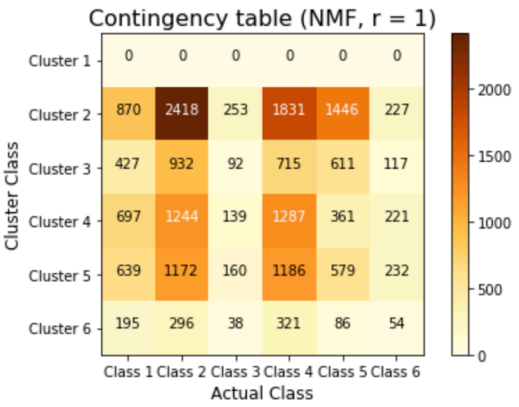
1）5 measures scores

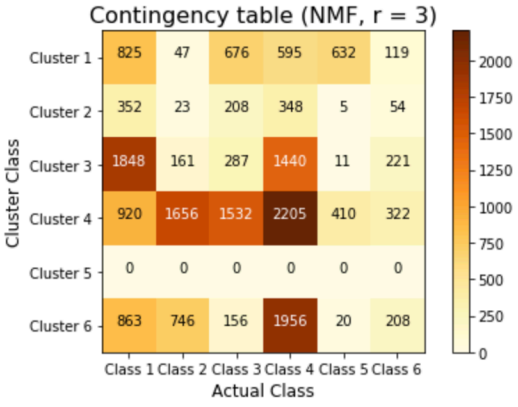


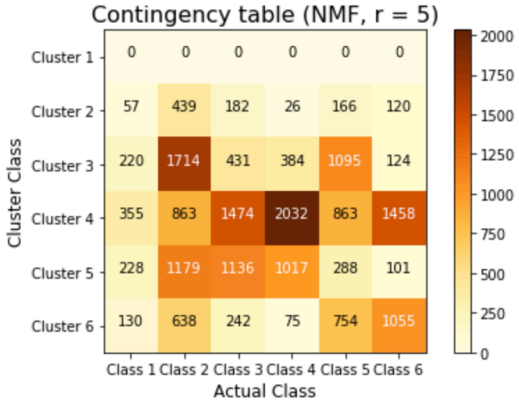


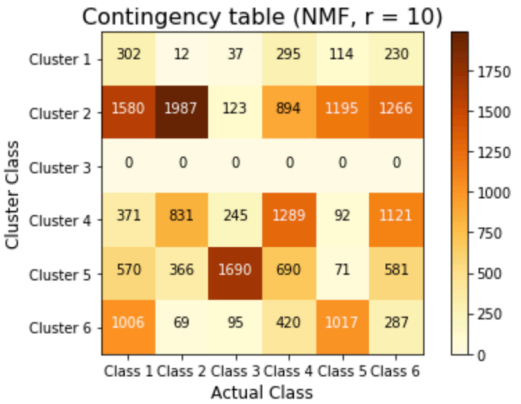
2） Contingency tables

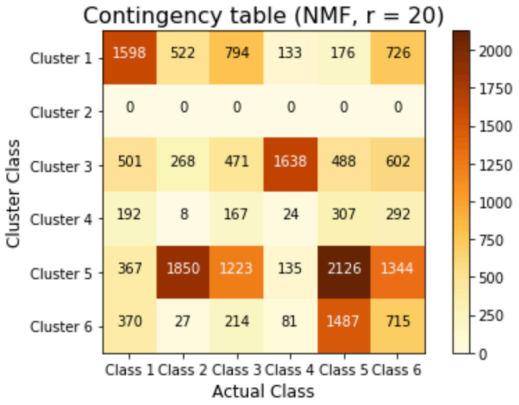


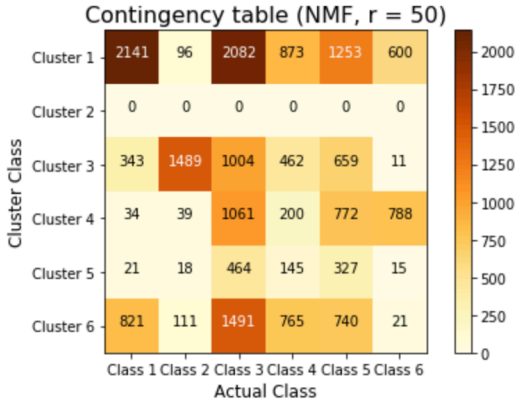

Contingency table (NMF, r = 1)
Cluster 1 0 0 0 0 0 0
Cluster 2 870 2418 253 1831 1446 227
Cluster 3 427 932 92 715 611 117
Cluster 4 697 1244 139 1287 361 221
Cluster 5 639 1172 160 1186 579 232
Cluster 6 195 296 38 321 86 54
Cluster Class
Class 1 Class 2 Class 3 Class 4 Class 5 Class 6
Actual Class
2000 1500 1000 500 0

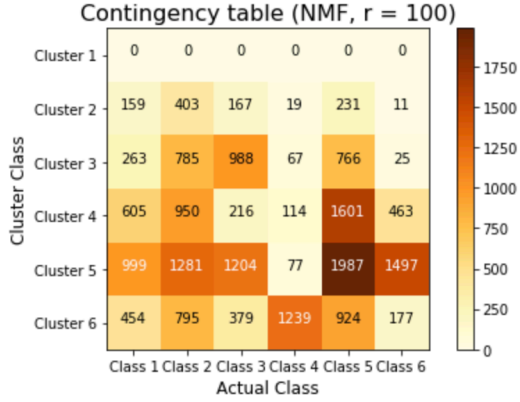

Contingency table (NMF, r = 3)
Cluster 1 825 47 676 595 632 119
Cluster 2 352 23 208 348 5 54
Cluster 3 1848 161 287 1440 11 221
Cluster 4 920 1656 1532 2205 410 322
Cluster 5 0 0 0 0 0 0
Cluster 6 863 746 156 1956 20 208
Cluster Class
Class 1 Class 2 Class 3 Class 4 Class 5 Class 6
Actual Class
2000 1750 1500 1250 1000 750 500 250 0

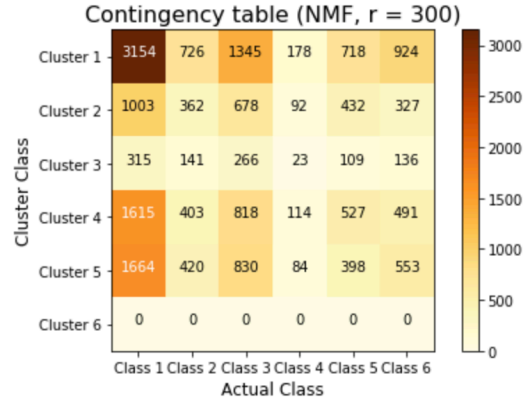

Contingency table (NMF, r = 5)
Cluster 1 0 0 0 0 0 0
Cluster 2 57 439 182 26 166 120
Cluster 3 220 1714 431 384 1095 124
Cluster 4 355 863 1474 2032 863 1458
Cluster 5 228 1179 1136 1017 288 101
Cluster 6 130 638 242 75 754 1055
Cluster Class
Class 1 Class 2 Class 3 Class 4 Class 5 Class 6
Actual Class
2000 1750 1500 1250 1000 750 500 250 0


Contingency table (NMF, r = 10)
Cluster 1 302 12 37 295 114 230
Cluster 2 1580 1987 123 894 1195 1266
Cluster 3 0 0 0 0 0 0
Cluster 4 371 831 245 1289 92 1121
Cluster 5 570 366 1690 690 71 581
Cluster 6 1006 69 95 420 1017 287
Cluster Class
Class 1 Class 2 Class 3 Class 4 Class 5 Class 6
Actual Class
1750 1500 1250 1000 750 500 250 0

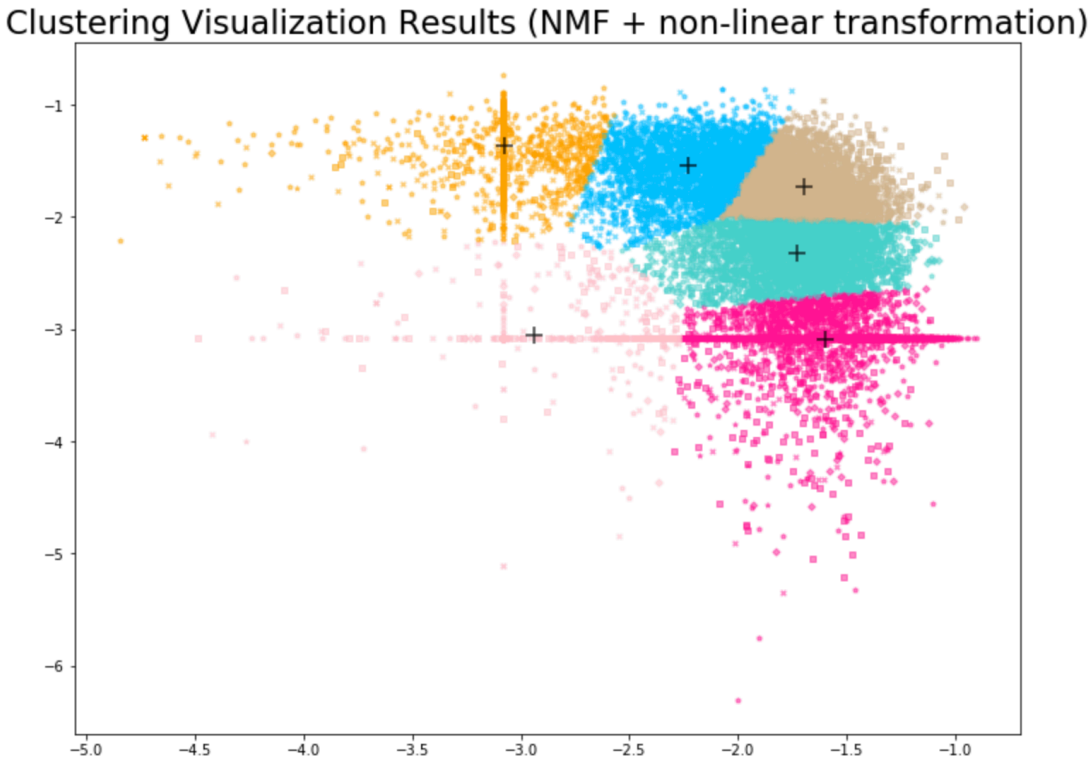

Contingency table (NMF, r = 20)
Cluster 1 1598 522 794 133 176 726
Cluster 2 0 0 0 0 0 0
Cluster 3 501 268 471 1638 488 602
Cluster 4 192 8 167 24 307 292
Cluster 5 367 1850 1223 135 2126 1344
Cluster 6 370 27 214 81 1487 715
Cluster Class
Class 1 Class 2 Class 3 Class 4 Class 5 Class 6
Actual Class
2000 1750 1500 1250 1000 750 500 250 0


Contingency table (NMF, r = 50)
Cluster 1 2141 96 2082 873 1253 600
Cluster 2 0 0 0 0 0 0
Cluster 3 343 1489 1004 462 659 11
Cluster 4 34 39 1061 200 772 788
Cluster 5 21 18 464 145 327 15
Cluster 6 821 111 1491 765 740 21
Cluster Class
Class 1 Class 2 Class 3 Class 4 Class 5 Class 6
Actual Class
2000 1750 1500 1250 1000 750 500 250 0


Contingency table (NMF, r = 100)
Cluster 1 0 0 0 0 0 0
Cluster 2 159 403 167 19 231 11
Cluster 3 263 785 988 67 766 25
Cluster 4 605 950 216 114 1601 463
Cluster 5 999 1281 1204 77 1987 1497
Cluster 6 454 795 379 1239 924 177
Cluster Class
Class 1 Class 2 Class 3 Class 4 Class 5 Class 6
Actual Class
1750 1500 1250 1000 750 500 250 0


Contingency table (NMF, r = 300)
Cluster 1 3154 726 1345 178 718 924
Cluster 2 1003 362 678 92 432 327
Cluster 3 315 141 266 23 109 136
Cluster 4 1615 403 818 114 527 491
Cluster 5 1664 420 830 84 398 553
Cluster 6 0 0 0 0 0 0
Cluster Class
Class 1 Class 2 Class 3 Class 4 Class 5 Class 6
Actual Class
3000 2500 2000 1500 1000 500 0

3） Clustering visualization results



We can see from the results above, that the optimized reduced dimensions for the two methods we tried were different from previous, the optimal dimension for truncated SVD is 50 and 20 for NMF. And there exists an all-zero rows in each contingency table because that class 4 only has one subclass and the number of documents may too small compared to other classes to cluster as a new class.