EE 219: Large-Scale Data Mining

Project 2

Chinni Sarat Bhargava 805219546
Eden Haney 005221845
Osama Hassan 705221917
Shannon Sabino 905227598

February 4, 2019

Contents

Question 1 **3**

Question 2 **3**

Question 3 **4**

Question 4 **4**

Question 5 **6**
- Homogeneity . . . 6
- Completeness . . . 7
- Rand Score . . . 7
- V-Score . . . 8
- Mutual Info . . . 8

Question 6 **9**

Question 7 **9**
- Part A . . . 9
- Part B . . . 10

Question 8 **11**

Question 9 **13**

Question 10 **14**

Question 11 **15**

Question 12 **16**

Appendix **18**

References **18**

Question 1

We begin this project by importing the data set that we will be working with, the 20 newsgroups text data set. Eight document categories were imported from 20newsgroups data set and they were divided into two main classes as shown Table 1.

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.rec.sport.hockey

Table 1: The main two categories and the documents' subcategories in each.

Feature extraction is a key step in any classic machine learning task. In our project we need to obtain a feature vector representation of the document in a corpus. One simple approach is to use bag of words. In this method, we represent each document as count vector of the number of times each word occurred in the document.

Words that rarely occur do not add value in the classification task and are removed. In this project we removed words that occur less than 3 times.

Normalization step is done to the count of vocabulary words in each document. The purpose of this normalization step is to specify the importance of a word to document in the corpus. Term Frequency-Inverse Document Frequency (TF-IDF)" metric was utilized to evaluate this importance.

$tf(t, d)$ is the count a term in document d. $Idf(t)$ is inverse document frequency of a term. So the metric is penalizing frequently occurring terms and encouraging the less frequent terms.

The resulting shape of the TF-IDF matrices for the training and test data sets are reported in Table 2.

Dataset	TF-IDF Matrix Shape
Training Dataset	(7882, 27768)

Table 2: The shape of the TF-IDF matrices of the training subset.

Question 2

We then applied k-means clustering with k=2 using the TF-IDF data. The performance of the algorithm is evaluated by the contingency table. The contingency table is the matrix whose entries A_{ij} is the number of data points that belong to both the class C_i and the cluster K_j.

The resulting contingency table for this training is reported in Table 3. We can see that cluster A is much more homogeneous than cluster B as it only contains 4 documents from class 0 and the remaining 1718 documents are from class 1. On the other hand, cluster B is better in terms of completeness as it contains higher percentages of both class 0 and class 1 compared with cluster A. However, the overall performance of clustering is low as it did not succeed in separating Class 0 and Class 1 in relatively two independent clusters.

	Cluster A	Cluster B
Class 1	4	2936
Class 2	1718	3224

Table 3: Contingency Table.

Question 3

The performance of the k-means algorithm can be evaluated using several clustering evaluation metrics. Five metrics of interest were used to quantify our clustering algorithm which are homogeneity, completeness, V-measure, adjusted Rand Index and adjusted mutual information score.

Homogeneity quantifies the purity of the cluster. The ideal homogeneity condition is when only a single ground truth label exists in each cluster. Completeness is ideally satisfied if the algorithm assigns all objects that belong to the same category in the ground truth to the same cluster. Both, homogeneity and completeness metrics may take values from 0 to 1 where the larger, the better. To take both homogeneity and completeness into account, the V-measure is used which is the harmonic mean of homogeneity and completeness. Whereas the regular mean treats all values equally, the harmonic mean gives much more weight to low values. As a result, the classifier will only get a high V-measure score if both Homogeneity and Completeness are high. The adjusted Rand index (ARI) has the maximum value 1 with expected value of 0 when the clusters are random. A larger ARI means a higher agreement between two partitions. The adjusted mutual information score measures the mutual information between the cluster label distribution and the ground truth label distributions.

These metrics are reported in Table 4

Metric	Value
Homogeneity	0.176
Completeness	0.221
V-measure	0.196
Adjusted Rand-Index	0.012
Adjusted mutual info score	0.176

Table 4: Performance metrics of the k-means algorithm.

Since the clustered data exist in high-dimensional space and no dimensional reduction algorithm was applied prior to clustering, all the five metrics have low values and the clustering quality is degraded.

Question 4

High dimensional sparse TF-IDF vectors do not yield in good clustering result. This is mainly because the Euclidean distance tends to almost the same in high-dimensional space and fails to be a good performance metric.

To partially tackle this issue, a better representation is achieved by dimensional reduction. The singular value decomposition (SVD) was first used in our project for this purpose. To figure out the top singular values of the TF-IDF matrix and figure out the significant significant components,

the percent of variance for the top 1000 principle components were calculated and reported for individual components and in accumulative fashion (see Figure 1).

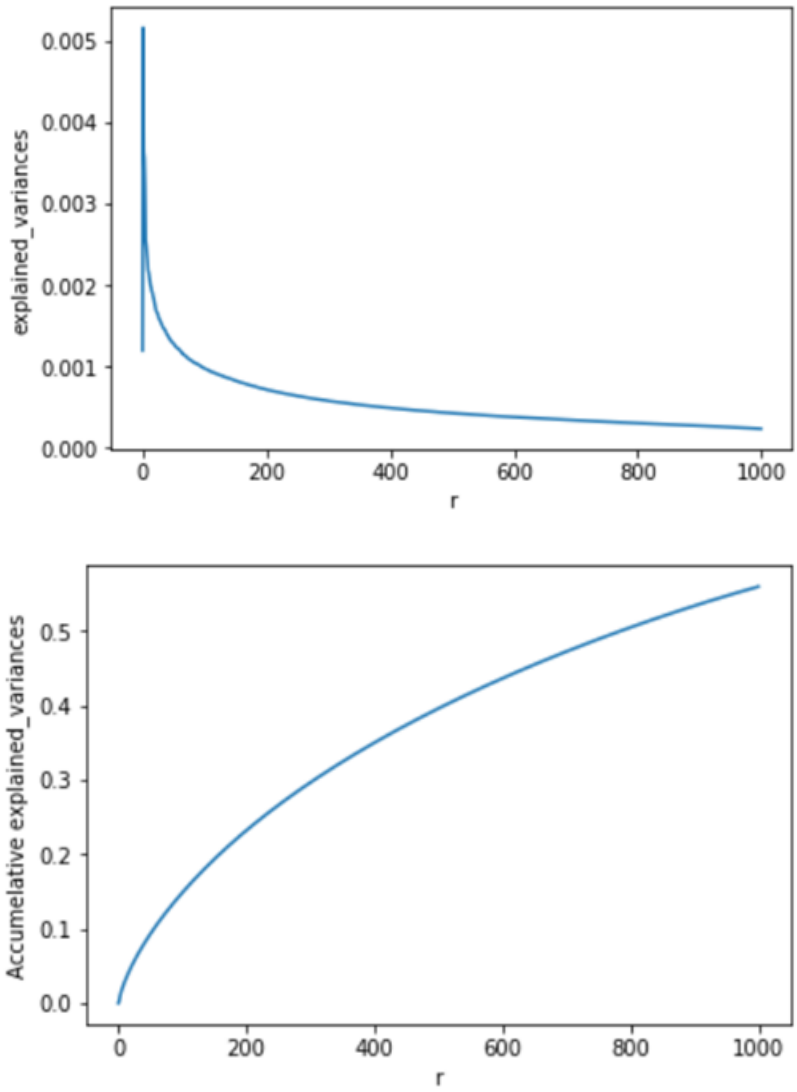


Figure 1: The percent of variance for the top 1000 principle components, denoted by r, were calculated and reported for individual components (Top) and in accumulative fashion (Bottom).

We can see that the variance increases very rapidly for the lowest number of dimensions before sharply slowing its growth. This is meaningful to us because we want to reduce the dimensions in the direction that preserves the most variance in the data. To explain why, consider Figure (2). If we were to reduce this example data down along line c_2, we can see that the reduced data will not be very representative of the original data, and very little information will be preserved. Compare this to the reduction onto line c_1. This dimension preserves the variance of the data, and is much more representative of the original data. This helps us to understand what Figure 1 is indicating to us — we only need to retain a few of the reduced dimensions to retain most of the information of the data.

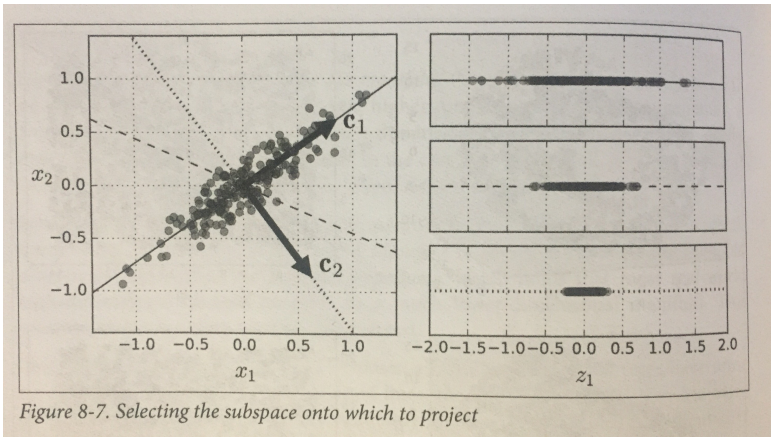
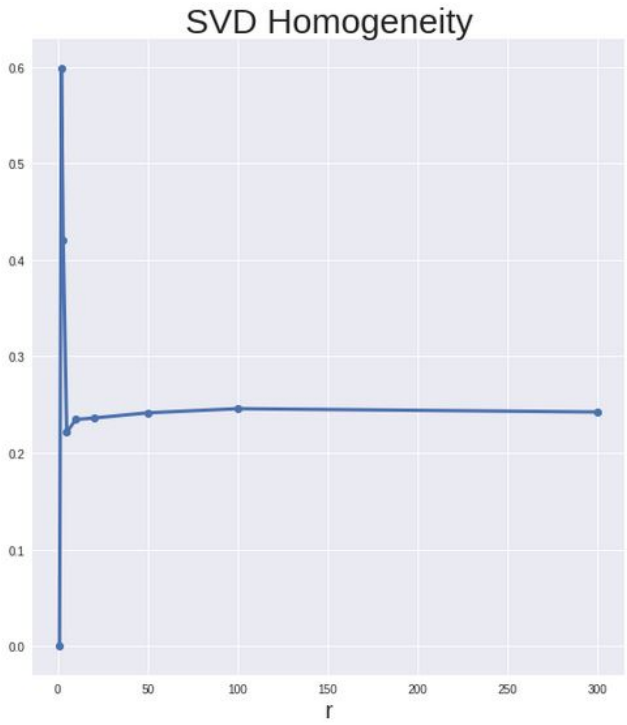


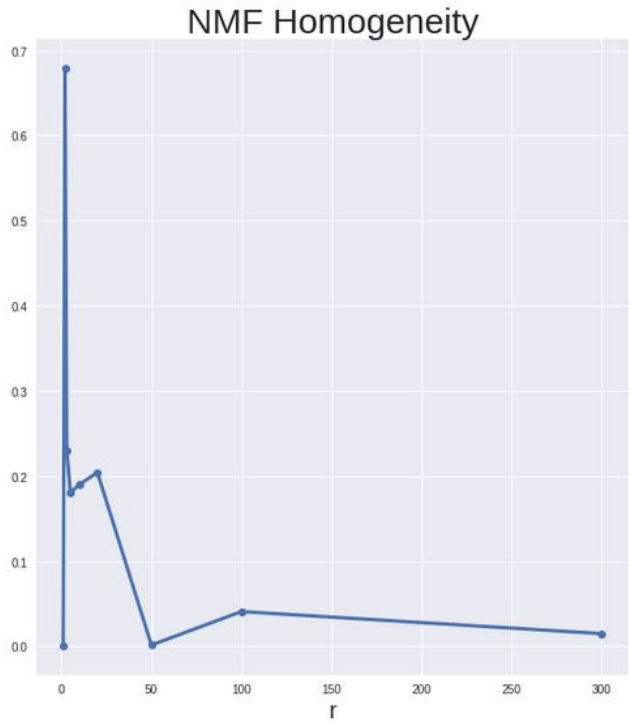
Figure 2: Image from [1].

Question 5

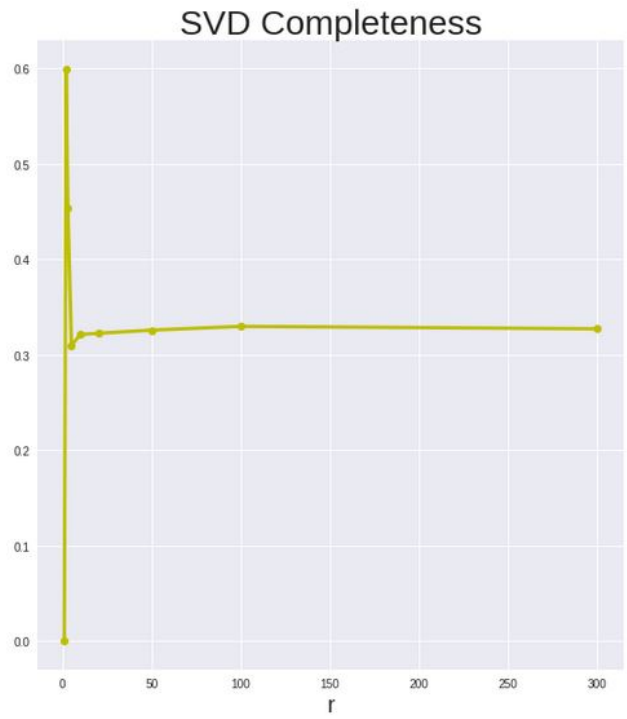
Let us compare the performance of SVD and NMF reductions of different sizes for k-mean clustering. We examine the same 5 metrics as we did in Question 3.

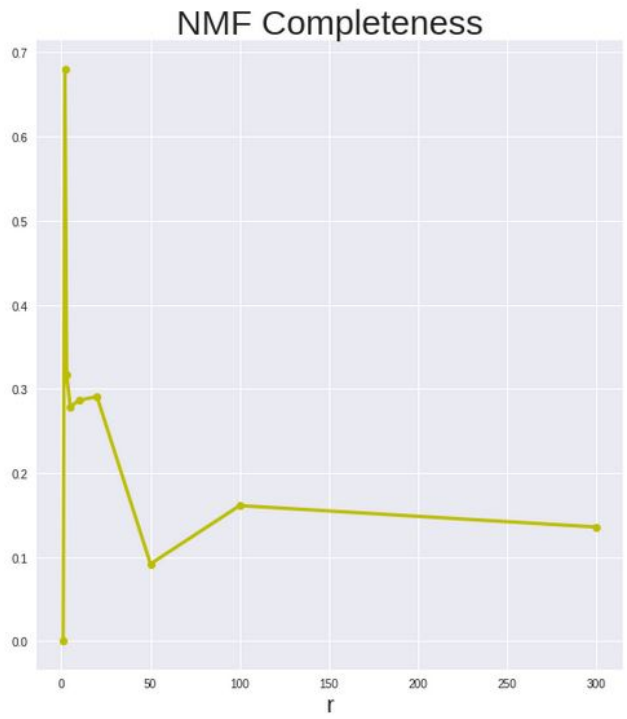
Homogeneity



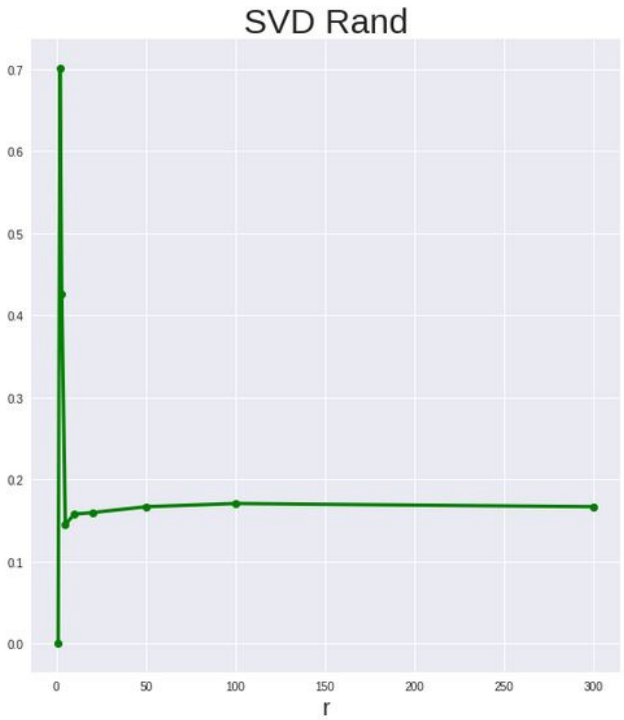


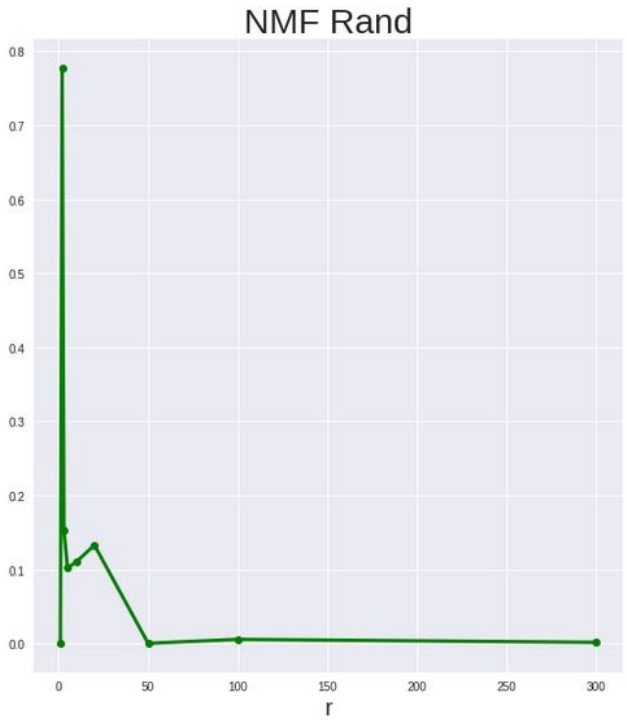
Completeness



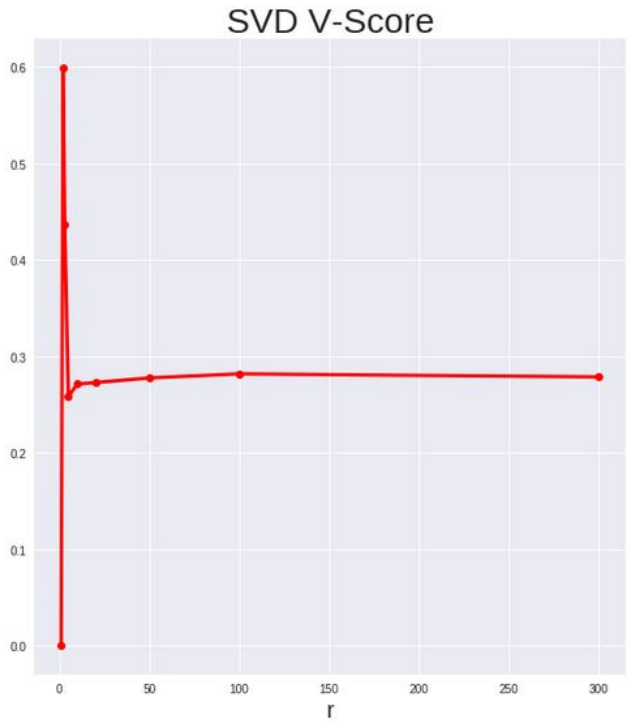


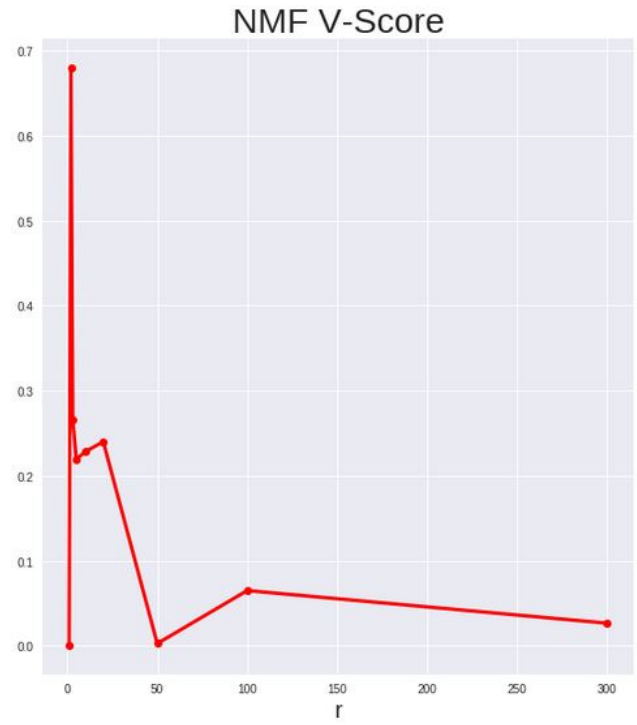
Rand Score



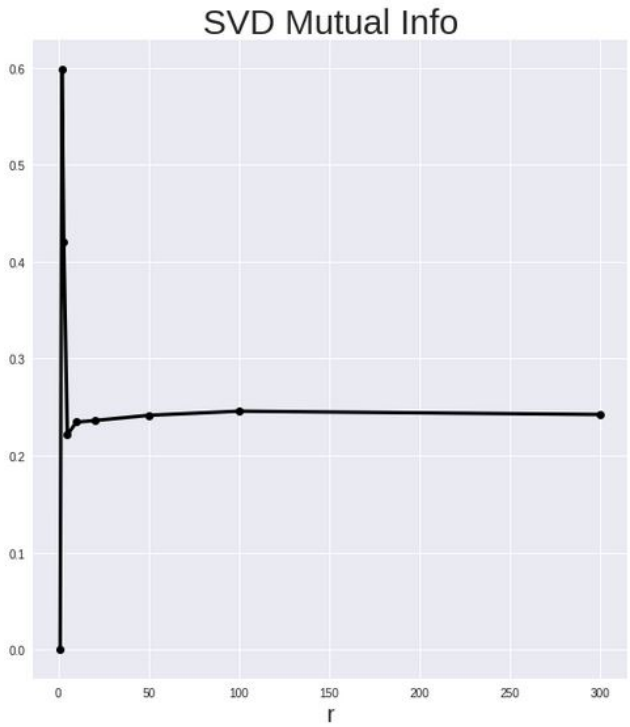


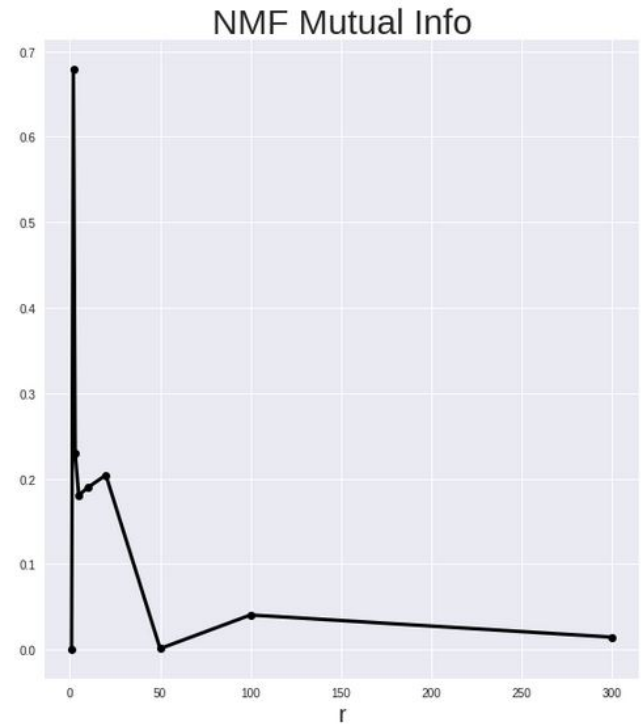
V-Score





Mutual Info





We see that the best dimension choice is the same across all metrics for both SVD and NMF reduction. However, we can also note that including more dimensions performs better in SVD, in contract to NMF's slightly higher peak performance.

Question 6

None of the prediction metrics increase monotonically. They instead all decrease very sharply after the second feature. Usually, a good way to pick the features we should keep would be to keep the highest variance features. As we can see in Question 4, the variance increases the fastest for the lowest number of dimensions. This does indicate to us that the lowest features will be the most useful. So in this sense, it is not surprising to see that only a few dimensions are necessary for good results.

To understand why more features act detrimentally to our clustering, it may be helpful to consider the type of data that we have. For the quality of our classification to decrease, the included features must blur the lines between our binary classes. For textual data, it is very easy to believe that there will be overlapping words between the two datasets that aren't meaningful to either classification, such as "going" or "fix". This means that these features are present in both of our datasets without being specifically relevant to either, and attempting to draw conclusions from their presence is misleading.

Question 7

Part A

We were asked to visualize the "best case" scenarios for both the SVD and NMF clustering dimensions from question 5. We can visualize the clustering results by projecting the dim-reduced data points onto 2-D plane with SVD, and coloring the points according to the true labels and the cluster-predicted labels.

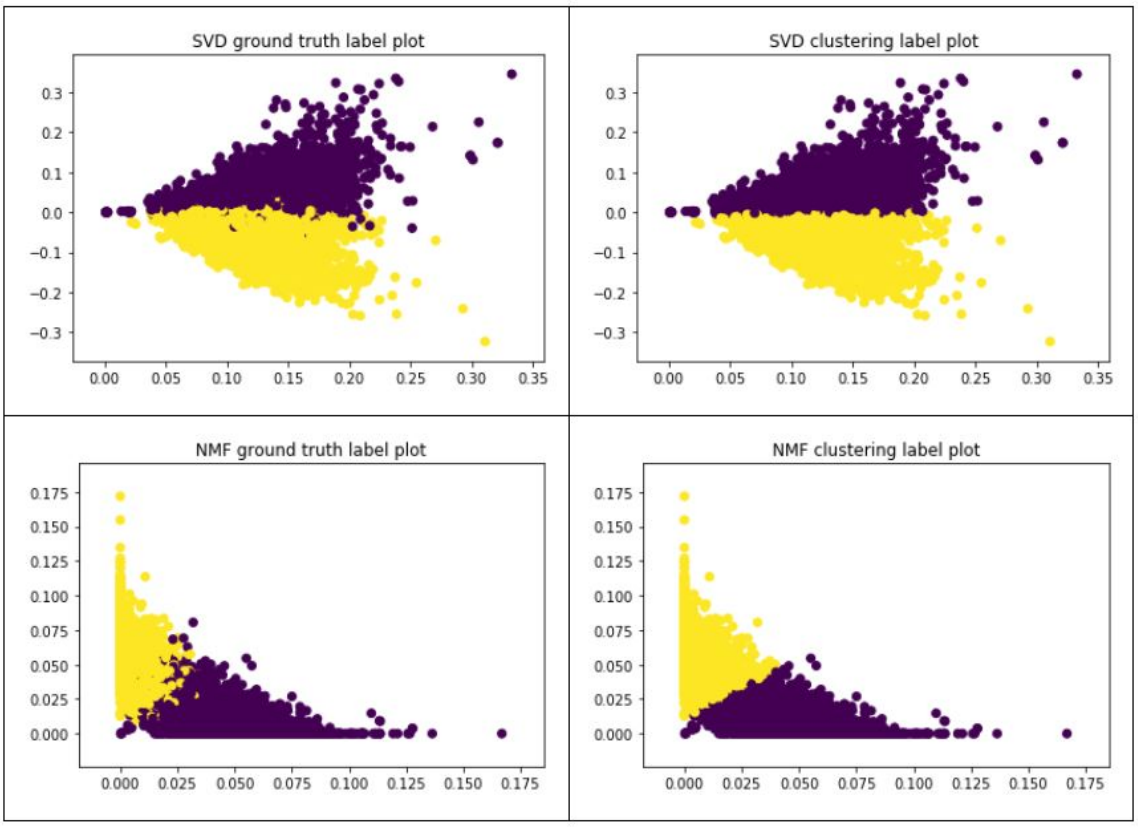


Figure 3: We notice the difference in how NMF and SVD data distributions compare, though it is not automatically apparent that NMF performs better than SVD, as the numbers show.

Part B

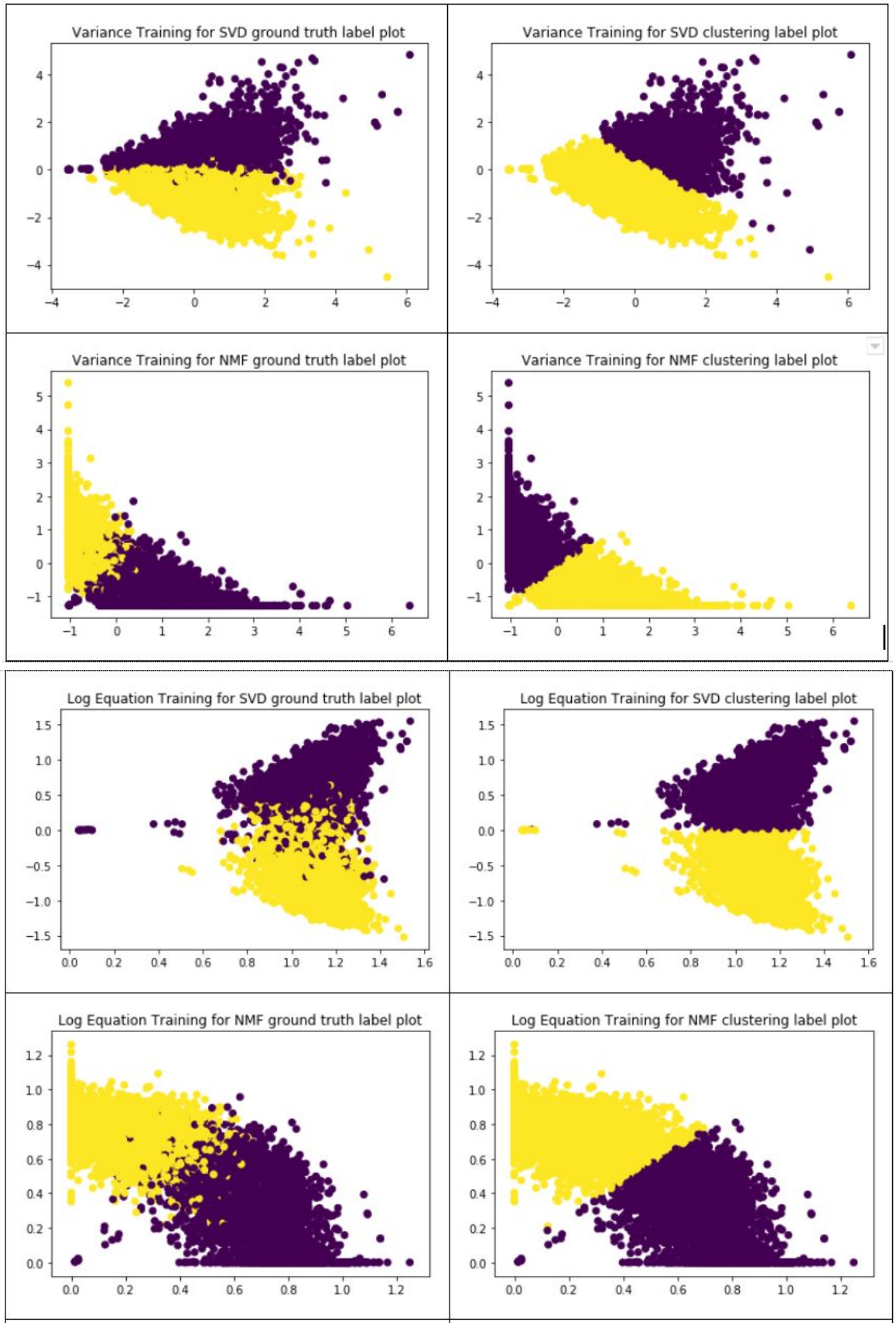
Next we applied and compared various training measures to the data for SVD and NMF clustering.

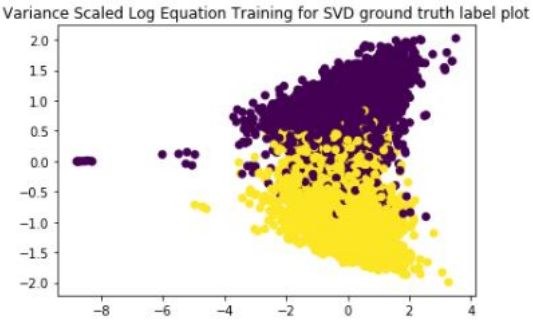
The two types of training we focused on were variance scaling, in which component-wise scaling is applied to transform the data to a unit variance, and equation (1).

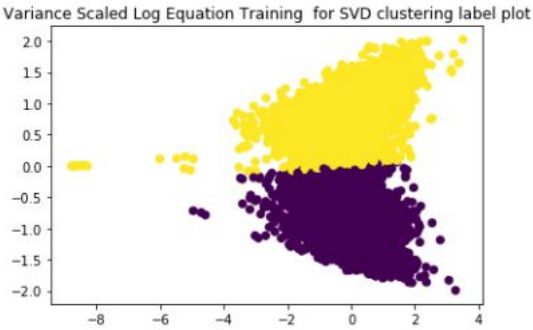
$$f(x) = \operatorname{sgn}(x) \cdot (\log(x + c) - \log c), \quad (\operatorname{sgn}(x))_i = \begin{cases} 1 & x_i > 0 \\ 0 & x_i = 0 \\ -1 & x_i < 0 \end{cases} \tag{1}$$

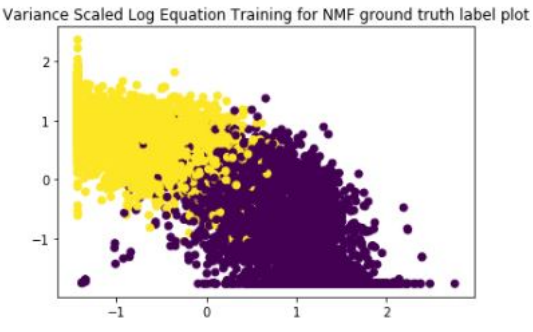
We applied each scaling property individually and then combined them to see how they work together.

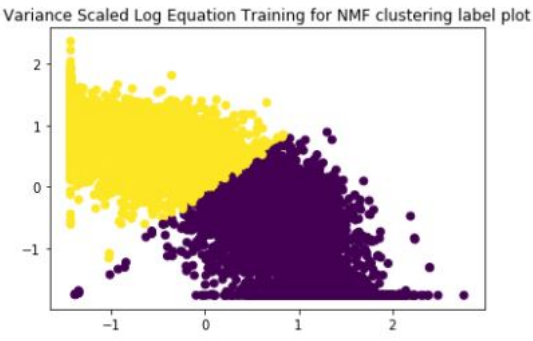
Question 8

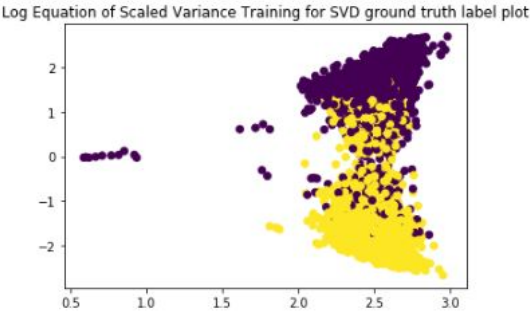

Variance Training for SVD ground truth label plot
Variance Training for SVD clustering label plot
Variance Training for NMF ground truth label plot
Variance Training for NMF clustering label plot
Log Equation Training for SVD ground truth label plot
Log Equation Training for SVD clustering label plot
Log Equation Training for NMF ground truth label plot
Log Equation Training for NMF clustering label plot

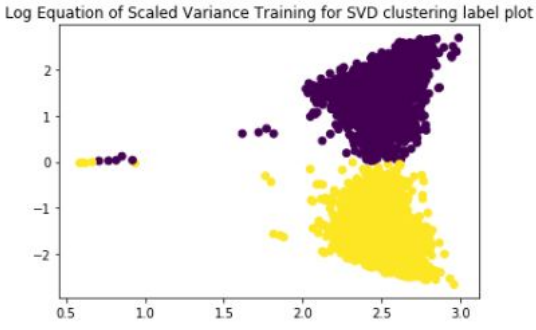

Variance Scaled Log Equation Training for SVD ground truth label plot

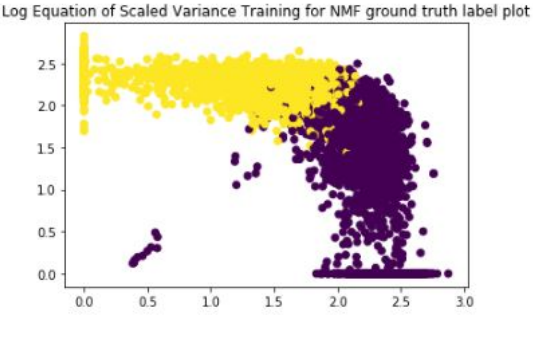

Variance Scaled Log Equation Training for SVD clustering label plot

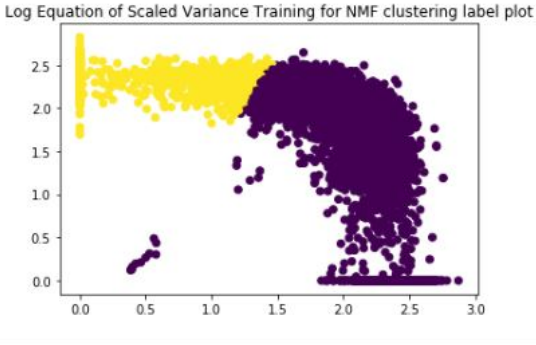

Variance Scaled Log Equation Training for NMF ground truth label plot

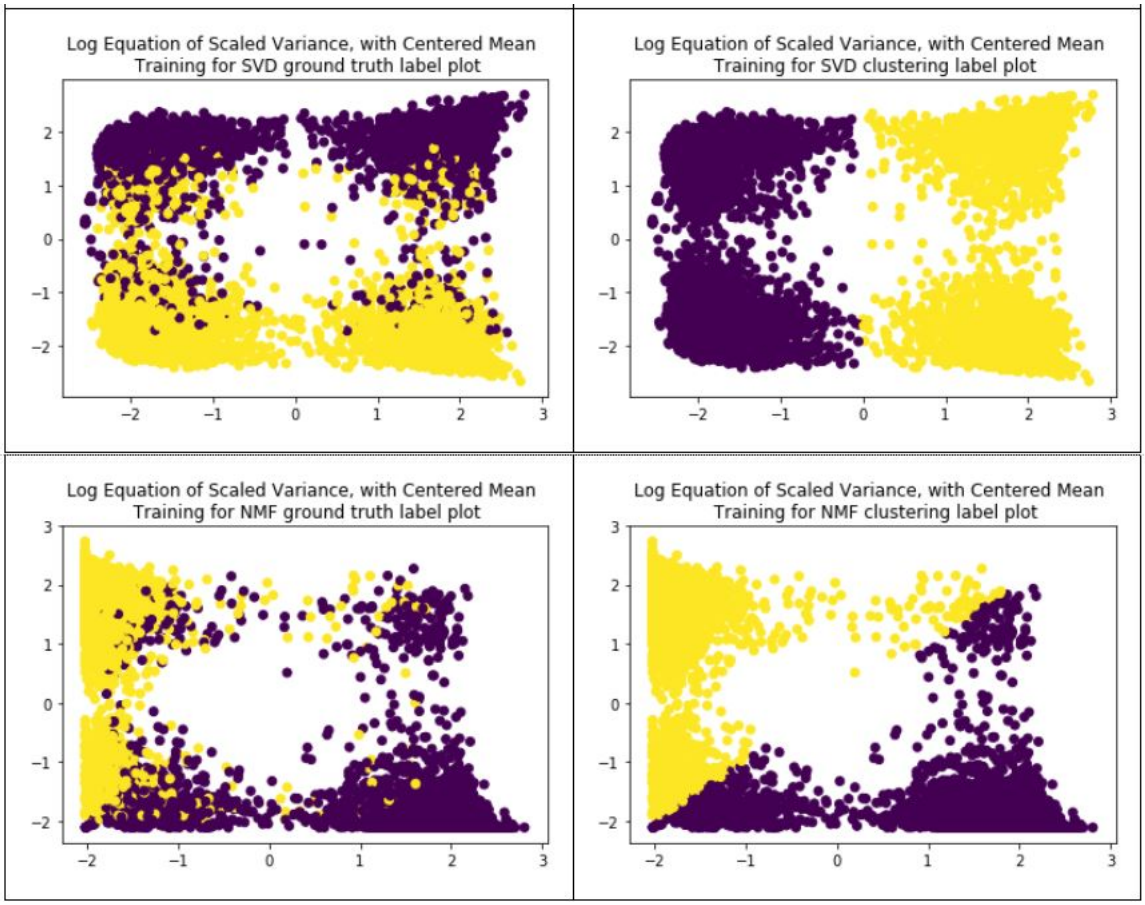

Variance Scaled Log Equation Training for NMF clustering label plot


Log Equation of Scaled Variance Training for SVD ground truth label plot


Log Equation of Scaled Variance Training for SVD clustering label plot


Log Equation of Scaled Variance Training for NMF ground truth label plot


Log Equation of Scaled Variance Training for NMF clustering label plot



Question 9

In figure 4 we examine the effect of the logarithmically transformed data. As we can see, taking the log of the data has a stronger effect on larger data values, somewhat pulling them in, while simultaneously spreading out smaller (negative) values. By doing this, the data closest to the divide between clusters is spread out, giving better clustering results.

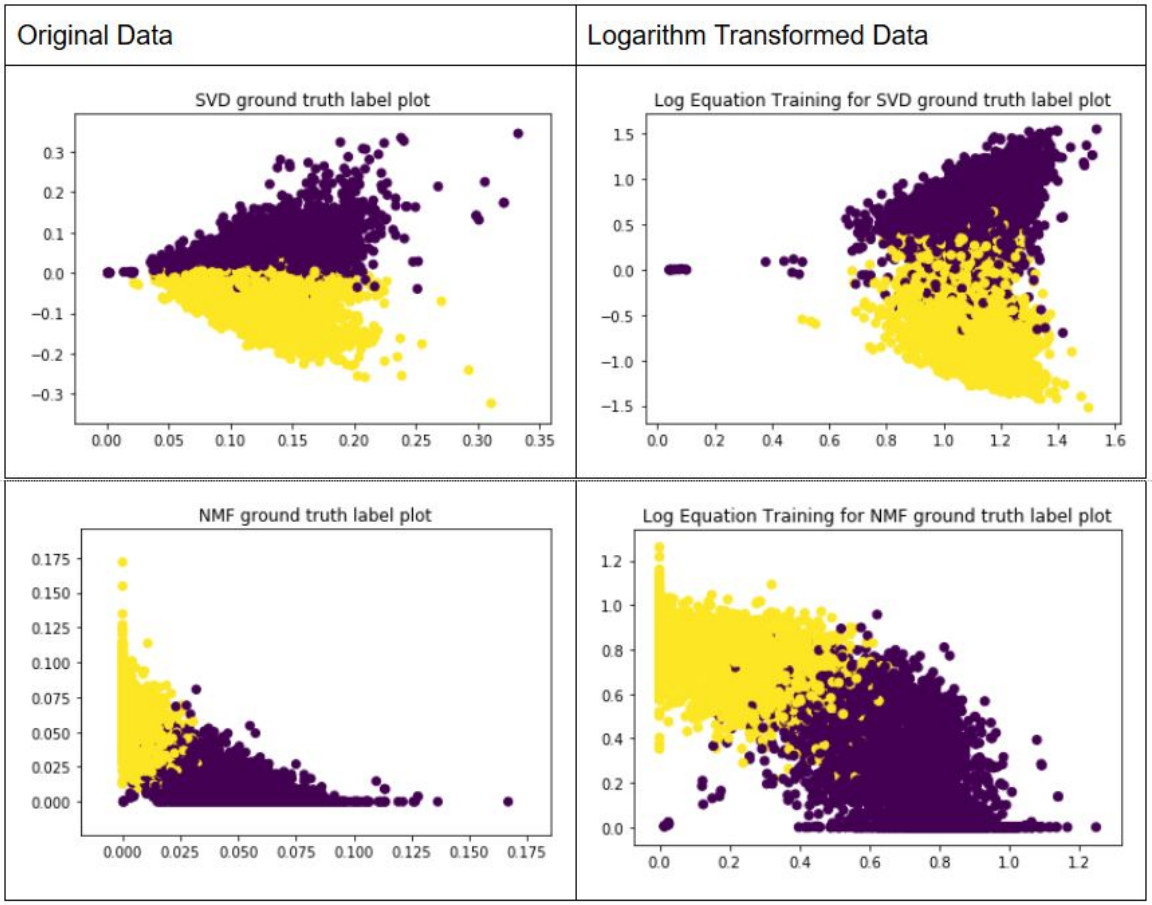


Figure 4: Question 9

Question 10

The results from the training combinations are show below.

	Homo-geneity	**Compl-eteness**	**V-measure**	**Adjusted Rand-Index**	**Adjusted Mutual Info**	**Average of Scores**
Variance Training						
SVD	0.235	0.263	0.248	0.255	0.235	0.247
NMF	0.683	0.686	0.684	0.773	0.683	0.702
Log Equation Training						
SVD	0.602	0.602	0.602	0.710	0.602	0.624
NMF	0.676	0.679	0.677	0.765	0.676	0.695
Variance Scaled Log Equation Training						
SVD	0.606	0.606	0.606	0.713	0.605	0.627
NMF	0.686	0.689	0.688	0.777	0.686	0.705
Log Equation of Scaled Variance Training						
	0.604	0.604	0.604	0.711	0.604	0.625
	0.313	0.383	0.345	0.249	0.313	0.320
Log Equation of Scaled Variance, with Centered Mean, Training						
SVD	0.000	0.000	0.000	0.000	0.000	0.000
NMF	0.696	0.696	0.696	0.793	0.696	0.715

Question 11

As in previous parts, we will first try using K-Means directly on the high dimensional TF-IDF data. We applied K-Means clustering algorithm on the entire data set with K=20 and the results of clustering are summarized in Table 6. In the next question, we will try various dimensionality reduction techniques and transformations to improve our results.

Dataset	TF-IDF Matrix Shape
Complete Dataset	(18846, 52295)

Table 5: TF-IDF matrices of the complete newsgroup data set

Metric	Value
Homogeneity	0.359
Completeness	0.451
V-measure	0.400
Adjusted Rand-Index	0.137
Adjusted mutual info score	0.357

Table 6: Performance metrics of the k-means algorithm.

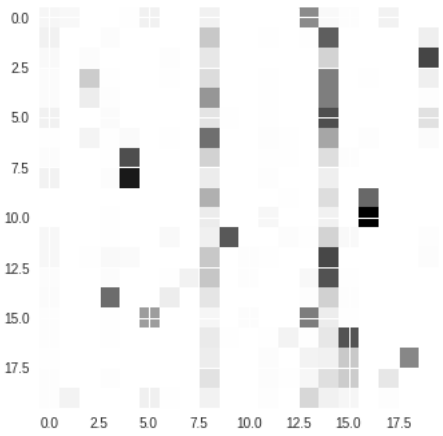


Figure 5: Contingency matrix of Newsgroup data set

Question 12

In this question we will try various dimensionality reduction and data transformation techniques as we did before to improve our clustering performance. In order to find the best possible setting we performed grid search. We performed grid search on the following parameters:

Parameters	**Values**
Dim Reduction	NMF, SVD
Dims	1,2,3,5,20,50,100,150,200,250,300
Transforms	Plain, Normalize, Log Transform, Normalize Log, Log Normalize

Table 7: Performance of K means clustering on News groups data set

The best result in terms of V-score is obtained when SVD is used for dimensionality reduction is with the settings r = 250 and Log Transformation. For NMF, the best result was with r=50 and Variance Scaling of Log Transformation. Over all NMF is the winner.

Metric	Value
Homogeneity	0.4688
Completeness	0.5263
V-measure	**0.4959**
Adjusted Rand-Index	0.2528
Adjusted mutual info score	0.4670

Table 8: **Performance metrics of the k-means with best settings on SVD.**

Metric	Value
Homogeneity	0.4742
Completeness	0.5549
V-measure	**0.5114**
Adjusted Rand-Index	0.2661
Adjusted mutual info score	0.4725

Table 9: **Performance metrics of the k-means with best settings on NMF.**

The performance of log transform and Variance scaling of log transformation (i.e. normalize log) did well compared to others as these transformations increase the gap between features holding small values, and decrease the gap between features holding large values. The performance in terms of V-Score increased with the r-value until a threshold is reached, and then it saturated or decreased due same explained in previous sections of the report. This trend can be seen in the Table 10. In the table below, "normalize log" refers to Variance scaling of Log Transformed data, and "log normalize" refers to Log Transformed Variance scaling.

Dimensionality Redn	r value	Transform	V-Score
svd	1	Nonlinear log	0.02920737791423757
svd	2	Nonlinear log	0.22137431035889787
svd	3	Nonlinear log	0.24556890227908298
svd	5	Nonlinear log	0.33620301926167195
svd	20	Nonlinear log	0.3780618791036021
svd	50	Nonlinear log	0.40744154838791874
svd	100	Nonlinear log	0.4278652555578429
svd	150	Nonlinear log	0.43932371643652257
svd	200	log normalize	0.42117212169141577
svd	**250**	**Nonlinear log**	**0.4959037050491699**
svd	300	Nonlinear log	0.4028692911432682
nmf	1	plain	0.028991012059297015
nmf	2	normalize log	0.18078926495899358
nmf	3	Nonlinear log	0.20889804017403682
nmf	5	Nonlinear log	0.31405437018215376
nmf	20	Nonlinear log	0.3831676576523215
nmf	**50**	**normalize log**	**0.5114307814724552**
nmf	100	normalize log	0.4782193561264264
nmf	150	normalize log	0.458026053925147
nmf	200	normalize log	0.4759353007156798
nmf	250	normalize log	0.48040482154957237
nmf	300	normalize log	0.4901858365152337

Table 10: Best Performance metrics of the k-means with r.

Based on the results, NMF performed slightly better than SVD that to using fewer dimensions, which translates as less computation during test time. So by using the dimensionality reduction and applying transforms on the dimensionality reduced data we are able to increase the V-Score from 0.4 to 0.511 which is 25% improvement. So from this analysis, it is in general better to use NMF

over SVD and to perform Variance Scaling of log data on dimensionality reduced data in order to obtain the best performance.

Therefore the recipe for good clustering performance is in general: dimensionality reduction using NMF (slightly better than SVD, but takes lot of training), applying Variance Scaled Log Transformation, and applying K-Means on this low dimensional transformed data.

Appendix

References

[1] Aurelien Geron. *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems.* O'Reilly Media, Sebastopol, CA, 2017.