EE 219 Project 4 - Report

March 8

Isha Verma	Heenal Doshi	Pranav Thulasiram Bhat
ishaverma@cs.ucla.edu	heenald@cs.ucla.edu	pranavtbhat@cs.ucla.edu
404761131	004758927	704741684

1 Introduction

In this project, we performed clustering on the 20 Newsgroup dataset which is a collection of approximately 20,000 documents partitioned into roughly 20 different groups each corresponding to a different topic. Clustering algorithms are unsupervised methods of learning aimed at finding groups of data points that are similar to each other. Clustering differs from classification in the sense that in case of clustering no a priori labelling or grouping of data is present.

In this project, we are using K means clustering for iteratively grouping data points into regions characterized by a set of cluster centroids and assigning each data point to the cluster with the nearest cluster centroid. We assume that the class labels are not available and our aim is to find the groups of the documents such that documents in same group are more similar to each other than to the documents in other groups. We also evaluate the performance of our task by comparing our results with the ground truth which is the actual class labels in the data set.

To start off, we use the documents in the two classes: computer technology and recreational activity. The categories in these two groups were the following:

- Computer Technology
- Recreational Activity

We later also apply clustering task on the data set with six classes and evaluate the performance on it.

2 Question 1

The clustering algorithm attempts to find groups of documents, without any a-priori knowledge about the data. In order to bring the document text-data into an analyzable and numeric format, we computed the TF-IDF representation of the data, by following the same procedure as Project-2:

- Remove invalid characters, punctuation and symbols
- Splitting the text on whitespaces, into an array of words
- Removing stop words
- Converting words into their stub forms using a Snowball stemmer or Lemmatizer.
- Computing the TF-IDF representation of the overall dataset using a Count vectorizer and a TF-IDF transformer.

For tokenizing, we used the StemTokenizer module form the nltk package. Words were split using a regex tokenizer splitting on whitespaces. A snowball stemmer was used to lemmatize the words and finally the Count Vectorizer was used to count word occurrences and nltk stopwords were removed. A pipeline was built for the workflow mentioned and the documents were processed.

To reduce the corpus size we removed the terms that appeared in large number of documents and infrequent terms by setting max df = 0.99 (Removing terms that appear in more than 99% of documents) and min df = 2 (Removing terms that appeared in a single document). To build the TF-IDF matrix the TF-IDF transform from Sklearn package was used.

Number of terms in our TF-IDF representation were 29862.

3 Question 2

In this question, we applied K-means clustering to the entire data set with k = 2. For this we used the KMeans module of Sklearn package. We generated the TF-IDF matrix as done in question 1 and for the data set with 8 categories mentioned above. The confusion matrix and the scores were reported. For calculating both of them, the actual labels had to be mapped to binary as we are doing clustering for two classes. So we divide the target labels of data by 2.

- The homogeneity score which is a measure of of how purely clusters contain only data points that belong to a single class was 0.272.
- A clustering result satisfies completeness if all of its clusters contain only data points that belong to a single class. The completeness score was 0.351.

- The Rand Index is similar to accuracy measure, which computes similarity between the clustering labels and ground truth labels. The adjusted rand score was 0.200.
- Finally the adjusted mutual score was 0.272 which measures mutual information between the cluster label distribution and the ground truth label distributions.

The confusion matrix obtained from running the K means clustering is as follows:

Class Labels	0	1
0	1	3902
1	1800	2179

Table 1: Confusion Matrix

We can imagine the ground truth labels to be on the first column and the predicted class labels on the first row of the matrix above. We can see that our clustering didnt do very well as for label 1 almost equal number of data points were grouped into two different classes i.e. 1800 documents were identified as one group and 2179 as other. Although for the other group, the result was good as 3902 documents were grouped in one cluster and only 1 in the other cluster.

A perfect clustering would result into a diagonal confusion matrix which denotes least confusion but the matrix obtained here is not diagonal and there is no permutation of rows which result in a diagonal matrix. We can also conclude from the low homogeneity, completeness, adjusted rand score and adjusted mutual info score that the clustering result obtained in this task did not perform well.

4 Question 3

We started by plotting (Figure 1) the singular values in descending order in order to find the initial guess for reduced dimensionality. We observed the elbow around 4, hence we varied dimension from 2 to 80 using SVD and NMF, calculating various metrics.

- Truncated SVD: We observed normalizing improved the score considerably. Without normalizing the homogeneity values were around 0.28 on average. However with normalizing, we observed it to be around 0.78.

  Figure 2 shows the values of various metrics for dimensions 2 to 80. The homogeneity and completeness scores were highest for reduced dimensions = 40. The homogeneity and completeness score stayed almost constant as after a certain dimension, indicating higher dimensions did not cause much change in clustering.

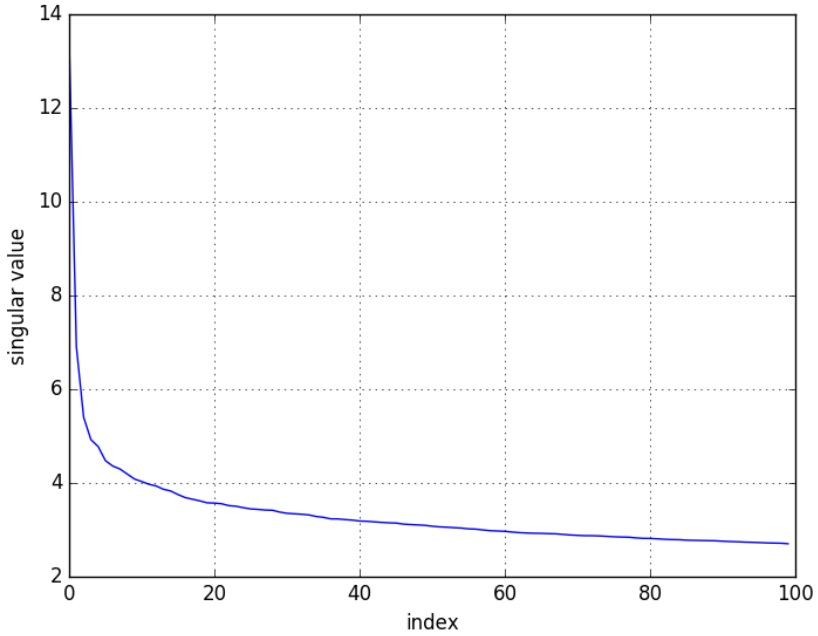

Figure 1: Singular values

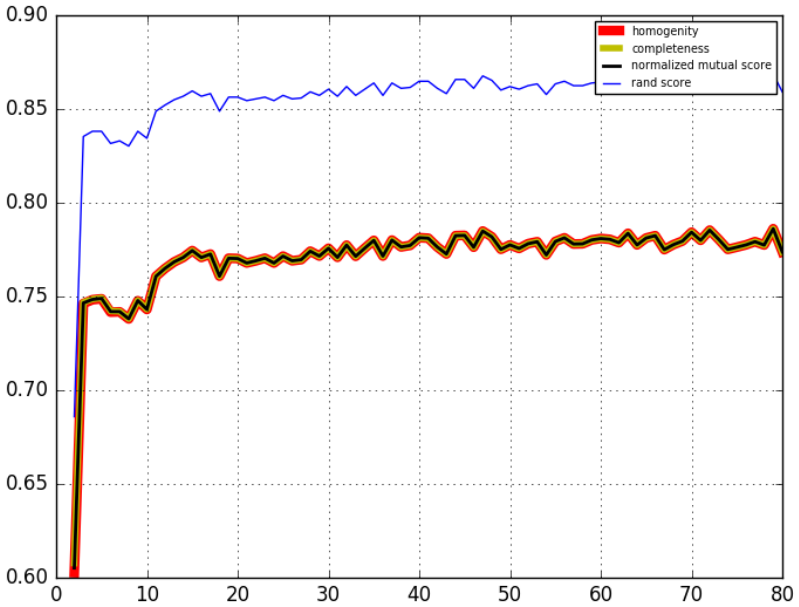

Figure 2: Metrics when using SVD with reduced dimensions (on X axis)

- NMF: We observed much lower metric values when using NMF, as seen in Figure 3.

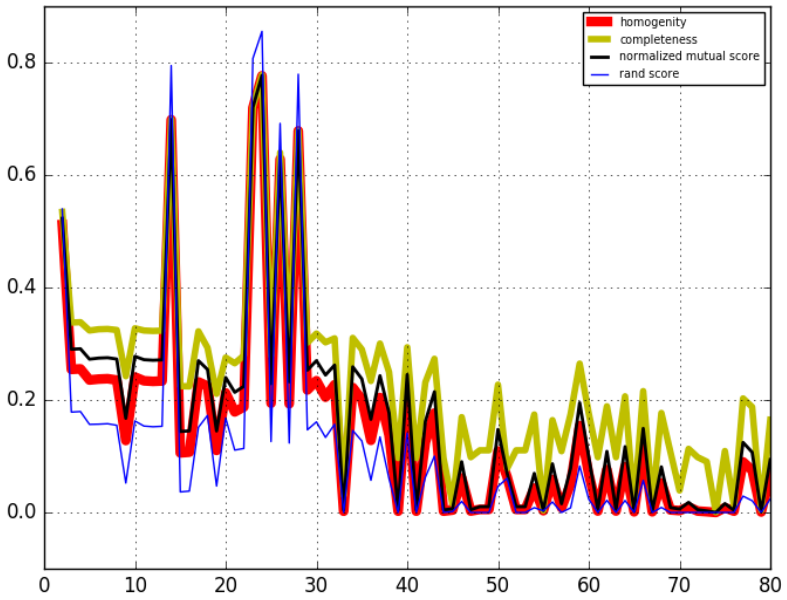

Figure 3: Metrics when using NMF with reduced dimensions (on X axis)

4.1 Non-linear Transformations

We also checked if adding non linear features helped clustering. We reduced dimensions to 40 (as selected above) to calculate these scores.

For polynomial of degree 2, the scores were as below,

- Homogeneity: 0.2639785941
- Completeness: 0.3448436437
- Normalized Mutual Info: 0.2990407859
- Adjusted Rand-Index: 0.1897194042
- Confusion Matrix:

3902	1
2223	1756

Table 2: Confusion Matrix

Thus the confusion matrix is not diagonal in the second row.

Natural log (numpy.log1p) features gave better results:

- Homogeneity: 0.7709171161
- Completeness: 0.7709986619
- Normalized Mutual Info: 0.7709578868
- Adjusted Rand-Index: 0.8568160283
- Confusion Matrix:

158	3745
3844	135

Table 3: Confusion Matrix

The confusion matrix is also comparatively sparse at non diagonal elements.

4.2 Visualizing NMF Embedding of Data

Following figure shows the TF_IDF data reduced to dimensions. As seen, the data is very close to 0 and not easily separable.

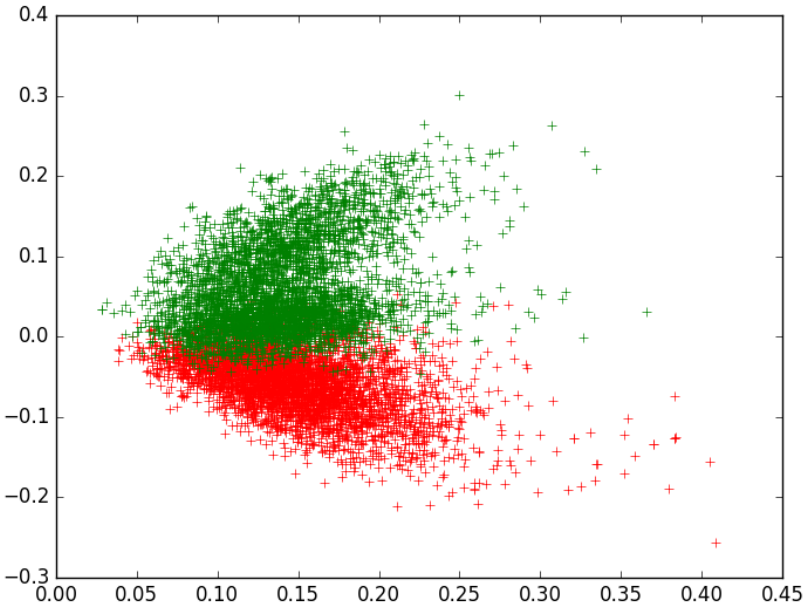

Figure 4: TF IDF Data in 2D

The data can be separated using log transformation as below,

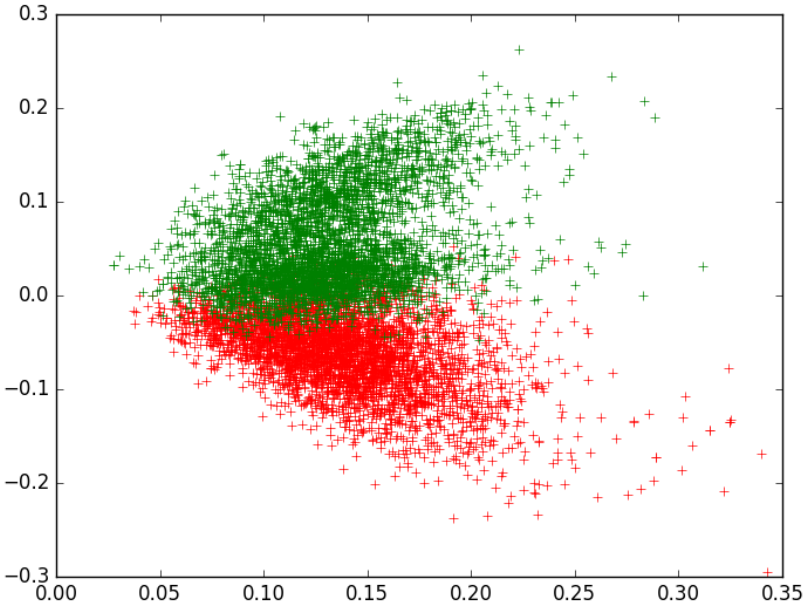

Figure 5: TF IDF Data in 2D with Log

5 Question 4

For this part of the problem, we re-projected the data from 40 dimensional space (identified as part of question 3 above) to a 2 dimensional space for visualization. The clusters we obtain with KMeans, plotted in 2d space are as seen in figure below

The confusion matrix for the same is,

3765	138
150	3829

Table 4: Confusion Matrix

Thus the matrix is pretty sparse on non diagonal elements, indicating lower mis-classifications. As can be seen from the figure the clusters are linearly separable. However if the points were not distributed in the manner as here, for example, if they were distributed as a log curve, non linear transform would be needed.

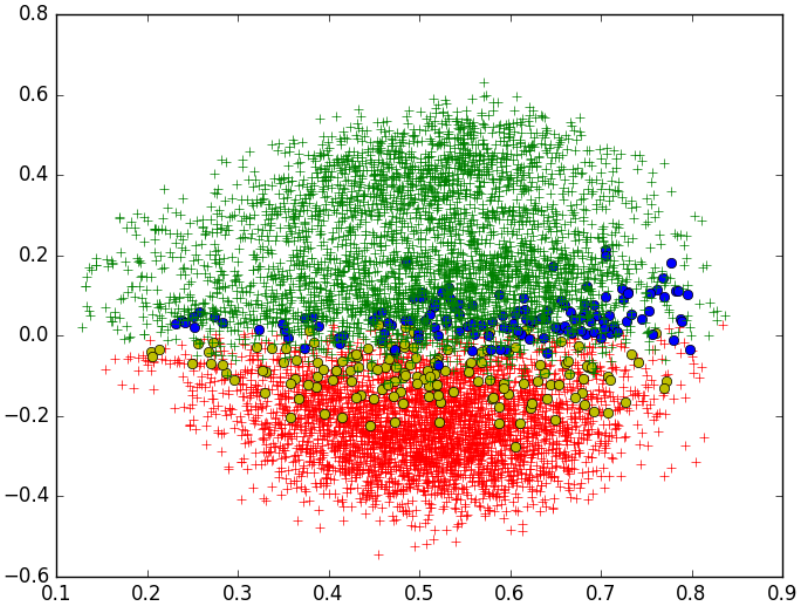

Figure 6: Kmeans - 2 clusters in 2D space

6 Question 5

In this section, we wanted to retrieve all 20 class labels. We included all the documents in the TF-IDF computation. In order to find the best parameters for clustering, we followed the following steps:

- We first computed an SVD of the TF-IDF obtained, and plot the singular values. We observed that the knee in the graph occurred at around d = 50.
- Next keeping k fixed at 20, we ran NMF and SVD for varying number of components(2 to 75), and plotted the resulting metrics.
- Keeping the number of dimensions fixed at 50, we varied the number of clusters between 4 and 31.

We found the best dimensions to be 50 and k value to be 23.

7 Question 6

In this section, we considered the problem of clustering with 6 super classes:

- Computer Technology
- Recreational activity

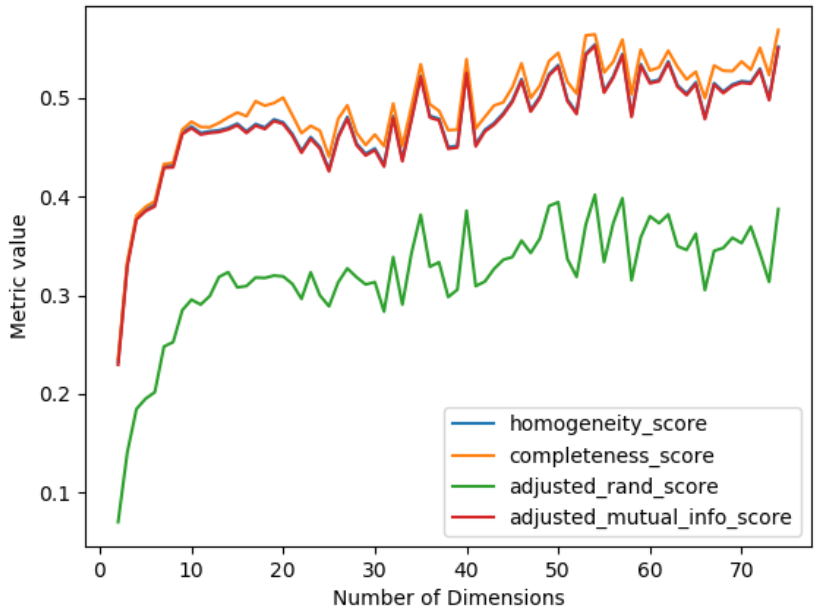

Figure 7: Performance Metrics of SVD with fixed k=20

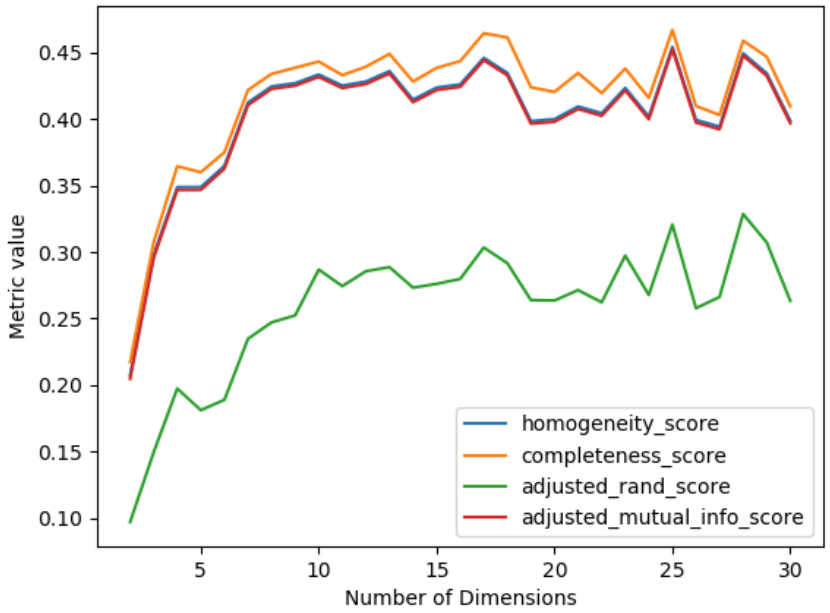

Figure 8: Performance Metrics of NMF with fixed k=20

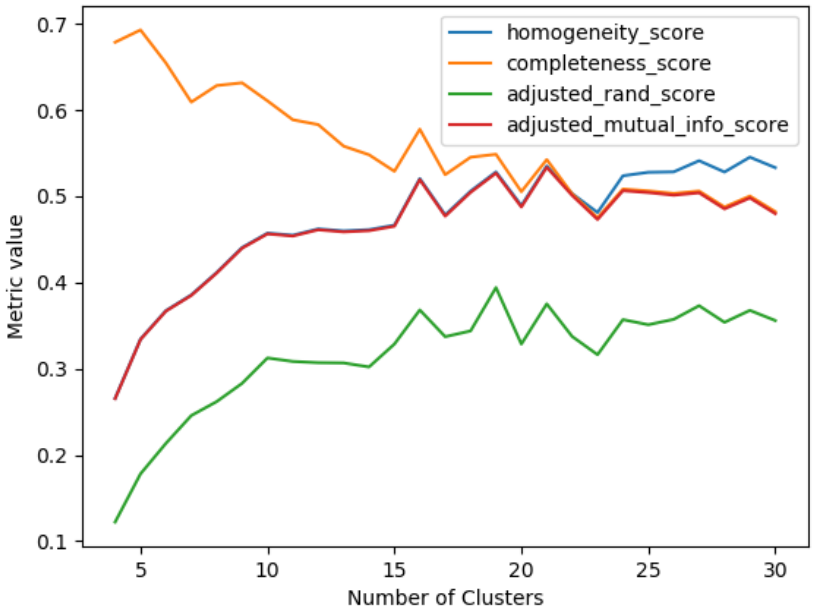

Figure 9: Performance Metrics of SVD with dimensions fixed at 50

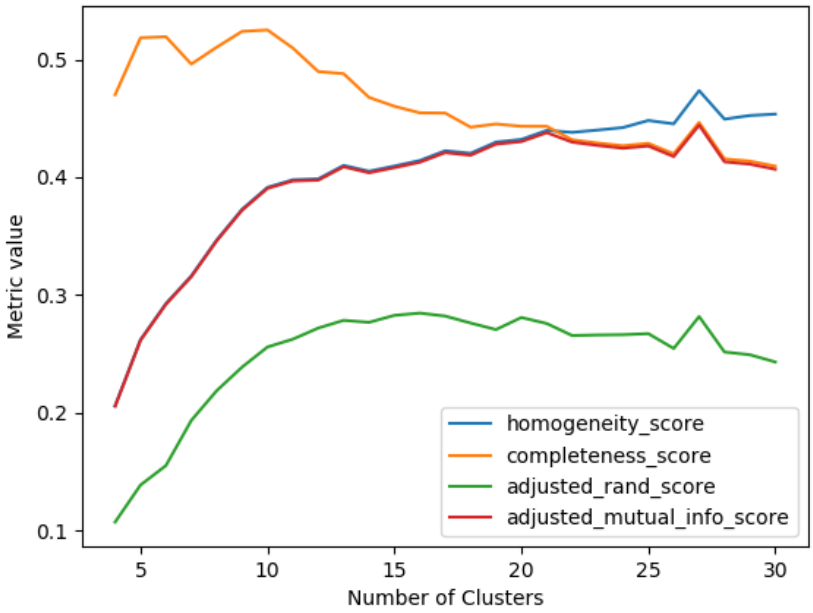

Figure 10: Performance Metrics of NMF with dimensions fixed at 50

- Science
- Miscellaneous
- Politics
- Religion

We used these super classes to relabel the dataset, and followed the following procedure:

- We first computed an SVD of the TF-IDF obtained, and plotted the singular values. We observed that the knee in the graph occurred at around d = 50.
- Next keeping k fixed at 6, we ran NMF and SVD for varying number of components(2 to 75), and plotted the resulting metrics.

At 2 dimensions, the performance metrics were as follows:

- Homogeneity: 0.208
- Completeness: 0.199
- Normalized Mutual Info: 0.125
- Adjusted Rand-Index: 0.198
- Confusion Matrix:

Class Labels	0	1	2	3	4	5
0	1535	474	259	167	1724	543
1	963	10	216	826	883	1051
2	498	34	994	1365	114	963
3	62	17	365	370	7	169
4	102	1022	813	626	15	316
5	17	1107	873	281	2	63

Table 5: Confusion Matrix

We observed that the best performance metrics were obtained at k = 6 and number of dimensions = 50.

- Homogeneity: 0.340
- Completeness: 0.350
- Normalized Mutual Info: 0.198
- Adjusted Rand-Index: 0.340
- Confusion Matrix:

Class Labels	0	1	2	3	4	5
0	382	1540	1	2077	126	576
1	0	884	9	1124	37	1895
2	3	137	1705	653	765	705
3	20	7	0	145	17	801
4	877	22	4	133	824	1034
5	394	4	5	100	1401	439

Table 6: Confusion Matrix

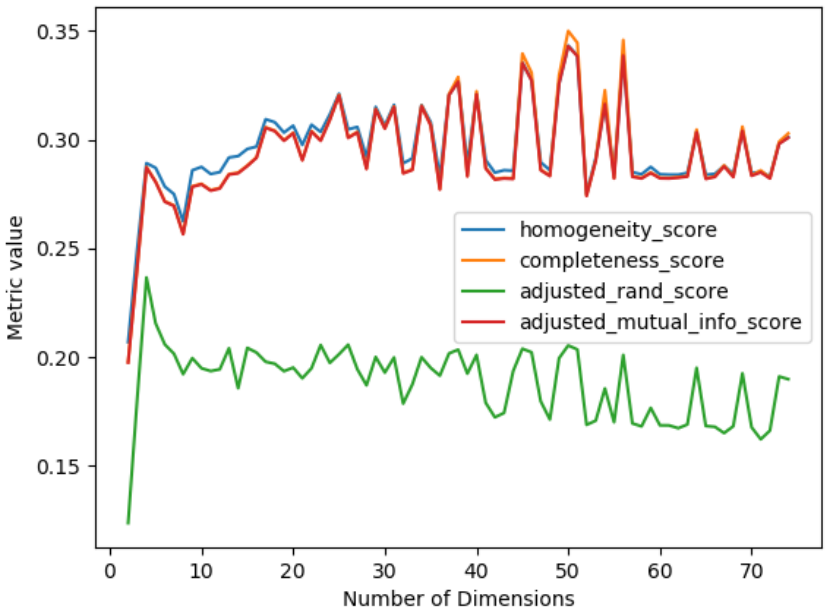

Figure 11: Variance of performance metrics with number of dimensions