EE 219 Project 2

Clustering

Sneha Shankar – 404946026

Devanshi Patel - 504945601

**Table of Contents**

[**I. Introduction** 2](#_Toc506222738)

[**The K-means Clustering Algorithm:** 2](#_Toc506222739)

[**Dataset:** 2](#_Toc506222740)

[**II. Problem Statement** 2](#_Toc506222741)

[**1. Building the TF-IDF matrix:** 3](#_Toc506222742)

[**2. Basic K-means Clustering** 4](#_Toc506222743)

[**3. Pre-process the data** 5](#_Toc506222744)

[**4. Visualization of clusters** 13](#_Toc506222745)

[**5. Expand Dataset into 20 categories** 19](#_Toc506222746)

# **I. Introduction**

This project focuses on performing K-means clustering on the dataset and thereby evaluate the performance of the algorithm. The performance is monitored by using different pre-processing methods. Representing the data in a way which depicts the most efficient clusters is also one of the aims of this project.

As opposed to classification which is a supervised technique for grouping data, **clustering** algorithms are **unsupervised** methods for grouping data points with similar representation in a proper space. In this method, the a priori labelling of data points is not available.

## **The K-means Clustering Algorithm:**

K-means clustering is a simple and widely used clustering algorithm. This algorithm groups the data into k clusters or groups based on the features that are provided. It uses an iterative refinement technique to produce a final output which has each data point assigned to only one cluster. This assignment is such that the sum of the squares of the distance between each data point and the centroid of the cluster to which it belongs to is minimized.

The algorithm initially estimates the k centroids which are either generated randomly or selected randomly from the data set itself. It then iterates between the following two steps until it reaches convergence:

1. *Assigning of data point to a cluster:*

In this step, one data point is assigned/reassigned to the nearest cluster. The nearest distance is based on the squared Euclidean distance between the data point and the centroid of the cluster.

1. *Updating the centroid*

Once a data point is assigned to a cluster, all centroids are computed again. This is achieved by taking the mean of all the data points which are assigned to that centroid’s cluster.

## **Dataset:**

The dataset we have at our disposal is the ‘20 Newsgroups’ dataset which is a collection of approximately 20,000 documents. These documents are segregated nearly evenly into 20 classes i.e. 20 newsgroups.

# **II. Problem Statement**

## **1. Building the TF-IDF matrix:**

Since clustering is an unsupervised technique of grouping similar data points, the class labels are not available to the algorithm. Here, we consider that the documents in each group are similar to each other than the documents in other groups/clusters. We then use the actual labels to evaluate the performance of our clustering. As such, feature extraction with the help of the TFIDF technique plays an important role here. Before applying this technique for feature extraction, we tokenize the documents into words and then exclude the stop words from them. Stop words are the words which are very commonly present in all the documents and thereby add no importance. We have used Python’s NLTK library for the same.

The TFIDF measure determines the relevance of a word within a particular document. Here, TF stands for ‘Term Frequency’ which denotes the number of times a particular term occurs within a given document. This is used to determine the frequent words. In this project we set the threshold for frequent words using the min\_df parameter. IDF stands for ‘Inverse Document Frequency’ which is used to specify the relevance of rare words. TFIDF of a particular word is the multiplication of the TF and IDF of that word.

After applying TFIDF transformation on the training dataset, we obtain a document – term matrix in which each row represents a document and each column a term. We have made the following observation of this TFIDF matrix by setting min\_df as 3.

|  |  |
| --- | --- |
| **Number of documents** | **Number of terms extracted** |
| 7882 | 27805 |

## **2. Basic K-means Clustering**

In this part, we apply K-means clustering with k=2 on the TFIDF data we received from the previous step.

1. We use the contingency matrix to depict how many data points belong to each cluster. Here, since we use k=2, we are grouping our data only into the broad classes of ‘Computer Technology’ and ‘Recreational Activity’. Following is the contingency matrix observed for basic K-means clustering.

|  |  |  |
| --- | --- | --- |
| **Actual** | **Predicted** | |
| Computer Technology | Recreational Activity |
| Computer Technology | 6 | 3897 |
| Recreational Activity | 1686 | 2293 |

From the above matrix, it can be clearly observed that there are a lot of incorrect assignments of data points to clusters. Once we tune the parameters and pre-process the data, the performance improves.

1. Now, we use different measures to examine the purity of the clusters with respect to the ground truth. *Homogeneity* measures the level purity of clusters. This score ranges from 0 to 1. A score of 1 signifies that each cluster contains data point from only a single class i.e. perfect clustering. *Completeness* tells us whether all data points of a particular class are indeed assigned to the same cluster. Again, this score varies from 0 to 1. *V-measure* is the harmonic mean of homogeneity and completeness. The *Adjusted Rand Index* is similar to the accuracy measure. It computes the similarity between the clustering labels and the ground truth labels. This counts all the pairs of data points that fall either in the same cluster and the same class or in different clusters and different classes. Lastly, we have the *Adjusted Mutual Information score* which measures the mutual information between the label information of our clusters and the ground truth label distributions. The measures we observed are tabulated as follows:

|  |  |
| --- | --- |
| **Measure** | **Value** |
| Homogeneity | 0.246 |
| Completeness | 0.328 |
| V-measure | 0.281 |
| Adjusted Rand Score | 0.173 |
| Adjusted Mutual Info Score | 0.246 |

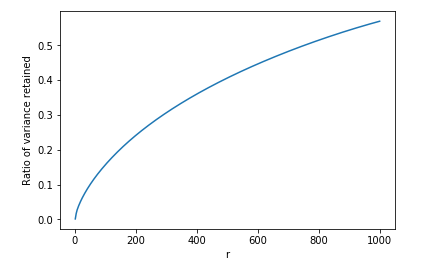
## **3. Pre-process the data**

As observed from the above table, the metrics are not good enough when K-means is applied to a high dimensional TFIDF matrix. One of the reason for this dip in the metrics is that the Euclidean distance which is used in K-means is not a good measure in the high dimensional space. Also, K-means might fail to identity the correct clusters to which the data points belong to since it implicitly assumes that the clusters are round shaped. This is because it tries to minimize the sum of within-cluster l^2 distances. To improve the above metrics and thereby to get a better representation which suits the working of K-means algorithm, we pre-process the data before clustering.

Here, we are using the Latent Semantic Indexing (LSI) and the Non- Negative Matrix Factorization (NMF) techniques for dimensionality reduction.

**i) Ratio of variance retained**

Before reduction, firstly, we find the effective dimension of the data through inspection of the top singular values of the TFIDF matrix and check how many of them are significant in reconstructing the matrix with the truncated SVD representation. To do this, we sweep though different values of r from 1 to 1000 where r stands for the top principle components. And for each value of r, we plot the percent of variance it can retain after dimensionality reduction. The following plot shows the same:



**ii) Reduction of data:**

We now use Truncated SVD and NMF to perform reduction on our TFIDF data. Here, we sweep over the dimension parameters and thereby get the best value of dimensional reduction. The following table denotes the different metrics we observe with different values of r (all values rounded to 3 digits after decimal point).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Dimension Parameter** | **Type** | **Homogeneity** | **Completeness** | **V-measure** | **Adjusted Rand Score** | **Adjusted Mutual Info Score** |
| 1 | LSI | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| NMF | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 2 | LSI | 0.043 | 0.081 | 0.056 | 0.026 | 0.043 |
| NMF | 0.003 | 0.023 | 0.005 | 0.000 | 0.003 |
| 3 | LSI | 0.499 | 0.500 | 0.499 | 0.605 | 0.499 |
| NMF | 0.001 | 0.009 | 0.002 | 0.000 | 0.001 |
| 5 | LSI | 0.001 | 0.014 | 0.002 | 0.000 | 0.001 |
| NMF | 0.001 | 0.015 | 0.002 | 0.000 | 0.001 |
| 10 | LSI | 0.001 | 0.012 | 0.002 | 0.000 | 0.001 |
| NMF | 0.212 | 0.299 | 0.249 | 0.139 | 0.212 |
| 20 | LSI | 0.001 | 0.013 | 0.002 | 0.000 | 0.001 |
| NMF | 0.079 | 0.194 | 0.112 | 0.022 | 0.079 |
| 50 | LSI | 0.229 | 0.314 | 0.265 | 0.155 | 0.229 |
| NMF | 0.077 | 0.202 | 0.111 | 0.024 | 0.077 |
| 75 | LSI | 0.001 | 0.013 | 0.002 | 0.000 | 0.001 |
| NMF | 0.075 | 0.200 | 0.109 | 0.023 | 0.075 |
| 100 | LSI | 0.001 | 0.013 | 0.002 | 0.000 | 0.001 |
| NMF | 0.067 | 0.191 | 0.099 | 0.020 | 0.067 |
| 300 | LSI | 0.001 | 0.013 | 0.002 | 0.000 | 0.001 |
| NMF | 0.065 | 0.187 | 0.097 | 0.019 | 0.065 |

**Thus, the best r value for LSI is 3, and that for NMF is 10.** This can also be observed from the graphs shown below. The values of the five metrics were first exported to excel and then plotted.

The 3D version of the graph was plotted to clearly depict the values of different metrics overlapping at the same score for the same value of r.

The contingency matrices for each of the rows of the above table are as follows:

* r=1; LSI

|  |  |  |
| --- | --- | --- |
| **Actual** | **Predicted** | |
| Computer Technology | Recreational Activity |
| Computer Technology | 2472 | 1431 |
| Recreational Activity | 2525 | 1454 |

* r=1; NMF

|  |  |  |
| --- | --- | --- |
| **Actual** | **Predicted** | |
| Computer Technology | Recreational Activity |
| Computer Technology | 2472 | 1431 |
| Recreational Activity | 2525 | 1454 |

* r=2; LSI

|  |  |  |
| --- | --- | --- |
| **Actual** | **Predicted** | |
| Computer Technology | Recreational Activity |
| Computer Technology | 3128 | 775 |
| Recreational Activity | 3802 | 177 |

* r=2; NMF

|  |  |  |
| --- | --- | --- |
| **Actual** | **Predicted** | |
| Computer Technology | Recreational Activity |
| Computer Technology | 3871 | 32 |
| Recreational Activity | 3887 | 92 |

* r=3; LSI

|  |  |  |
| --- | --- | --- |
| **Actual** | **Predicted** | |
| Computer Technology | Recreational Activity |
| Computer Technology | 519 | 3384 |
| Recreational Activity | 3623 | 356 |

* r=3; NMF

|  |  |  |
| --- | --- | --- |
| **Actual** | **Predicted** | |
| Computer Technology | Recreational Activity |
| Computer Technology | 3843 | 60 |
| Recreational Activity | 3867 | 112 |

* r=5; LSI

|  |  |  |
| --- | --- | --- |
| **Actual** | **Predicted** | |
| Computer Technology | Recreational Activity |
| Computer Technology | 3854 | 49 |
| Recreational Activity | 3957 | 22 |

* r=5; NMF

|  |  |  |
| --- | --- | --- |
| **Actual** | **Predicted** | |
| Computer Technology | Recreational Activity |
| Computer Technology | 3854 | 49 |
| Recreational Activity | 3958 | 21 |

* r=10; LSI

|  |  |  |
| --- | --- | --- |
| **Actual** | **Predicted** | |
| Computer Technology | Recreational Activity |
| Computer Technology | 3852 | 51 |
| Recreational Activity | 3955 | 24 |

* r=10; NMF

|  |  |  |
| --- | --- | --- |
| **Actual** | **Predicted** | |
| Computer Technology | Recreational Activity |
| Computer Technology | 10 | 3893 |
| Recreational Activity | 1517 | 2462 |

* r=20; LSI

|  |  |  |
| --- | --- | --- |
| **Actual** | **Predicted** | |
| Computer Technology | Recreational Activity |
| Computer Technology | 3853 | 50 |
| Recreational Activity | 3956 | 23 |

* r=20; NMF

|  |  |  |
| --- | --- | --- |
| **Actual** | **Predicted** | |
| Computer Technology | Recreational Activity |
| Computer Technology | 3897 | 6 |
| Recreational Activity | 3346 | 633 |

* r=50; LSI

|  |  |  |
| --- | --- | --- |
| **Actual** | **Predicted** | |
| Computer Technology | Recreational Activity |
| Computer Technology | 3896 | 7 |
| Recreational Activity | 2381 | 1598 |

* r=50; NMF

|  |  |  |
| --- | --- | --- |
| **Actual** | **Predicted** | |
| Computer Technology | Recreational Activity |
| Computer Technology | 3322 | 581 |
| Recreational Activity | 3977 | 2 |

* r=75; LSI

|  |  |  |
| --- | --- | --- |
| **Actual** | **Predicted** |  |
|  | Computer Technology | Recreational Activity |
| Computer Technology | 50 | 3853 |
| Recreational Activity | 23 | 3956 |

* r=75; NMF

|  |  |  |
| --- | --- | --- |
| **Actual** | **Predicted** |  |
|  | Computer Technology | Recreational Activity |
| Computer Technology | 3335 | 568 |
| Recreational Activity | 3977 | 2 |

* r=100; LSI

|  |  |  |
| --- | --- | --- |
| **Actual** | **Predicted** |  |
|  | Computer Technology | Recreational Activity |
| Computer Technology | 50 | 3853 |
| Recreational Activity | 23 | 3956 |

* r=100; NMF

|  |  |  |
| --- | --- | --- |
| **Actual** | **Predicted** |  |
|  | Computer Technology | Recreational Activity |
| Computer Technology | 3386 | 517 |
| Recreational Activity | 3976 | 3 |

* r=300; LSI

|  |  |  |
| --- | --- | --- |
| **Actual** | **Predicted** |  |
|  | Computer Technology | Recreational Activity |
| Computer Technology | 50 | 3853 |
| Recreational Activity | 23 | 3956 |

* r=300; NMF

|  |  |  |
| --- | --- | --- |
| **Actual** | **Predicted** |  |
|  | Computer Technology | Recreational Activity |
| Computer Technology | 3393 | 510 |
| Recreational Activity | 3975 | 4 |

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

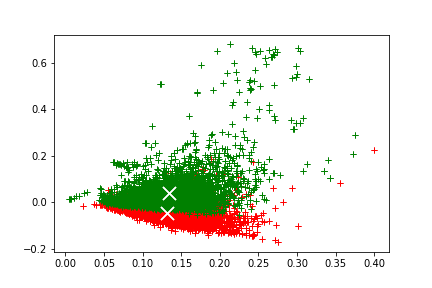
From the plots above, we observe that as the value of r increases, the non-monotonicity in the behavior of the metrics becomes evident. The reason for this is two things that balance each other’s effect. One is information retention which happens at high dimensions i.e. high value of r and the other is performance of k-means algorithm. It is widely known that k-means performs bad in high dimensions. This is the reason that we don’t get the best values of the measures at the highest value of r. Thus, as we increase r, the performance improves up to a certain point after which it starts degrading. This justifies the non-monotonic behavior of the measures.

## **4. Visualization of clusters**

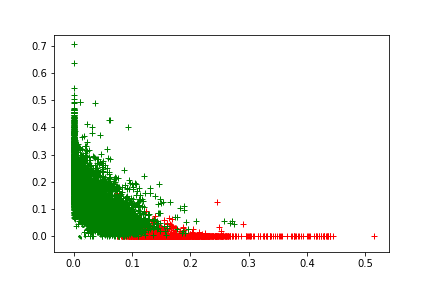
**a) Visualizing in 2D with best r**

We now use the best value of r as received from the above part to visualize the clusters. Since our **best value r is 3 and 10 for LSI and NMF** respectively and we need to project the final data vectors onto a 2D plane, we convert the matrix obtained after reduction into a two dimensional matrix. We use **SVD** to do so.

* After LSI decomposition



* After NMF Decomposition

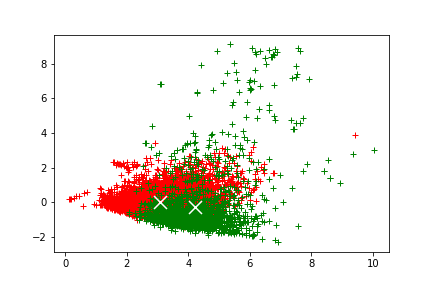


**b) Visualizing in 2D with best r value after applying transformations**

Now, we try different methods as below and test whether they increase the clustering performance (with the same best values of r). These visualizations are also on a 2D plane after applying SVD to the LSI and NMF matrices respectively.

1. ***Normalizing the features***

* *After LSI Decomposition*

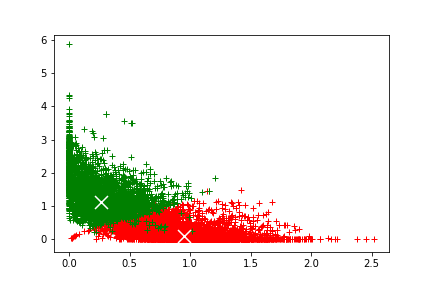


|  |  |
| --- | --- |
| **Metric** | **Value** |
| Homogeneity | 0.164 |
| Completeness | 0.214 |
| V-measure | 0.186 |
| Adjusted Rand Index | 0.135 |
| Adjusted Mutual Info Score | 0.164 |

|  |  |  |
| --- | --- | --- |
|  | **Predicted** |  |
| **Actual** | Computer Technology | Recreational Activity |
| Computer Technology | 3761 | 142 |
| Recreational Activity | 2350 | 1629 |

As seen from the above measures, we notice that normalizing the LSI matrix, degrades its purity. That is normalizing features does not really help in improving the performance after LSI decomposition.

* *After NMF Decomposition*

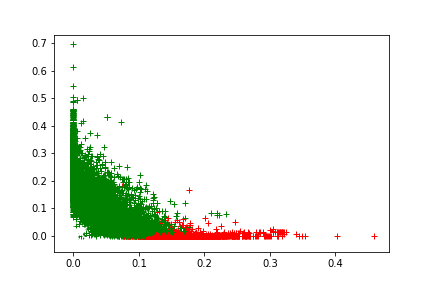


|  |  |
| --- | --- |
| **Metric** | **Value** |
| Homogeneity | 0.490 |
| Completeness | 0.505 |
| V-measure | 0.497 |
| Adjusted Rand Index | 0.558 |
| Adjusted Mutual Info Score | 0.490 |

|  |  |  |
| --- | --- | --- |
|  | **Predicted** |  |
| **Actual** | Computer Technology | Recreational Activity |
| Computer Technology | 875 | 3028 |
| Recreational Activity | 3856 | 123 |

We can see from the above statistics that there is a significant improvement in the performance after we normalize the features and feed it to NMF decomposition.

1. ***Non-linear (logarithmic) transformation after NMF***



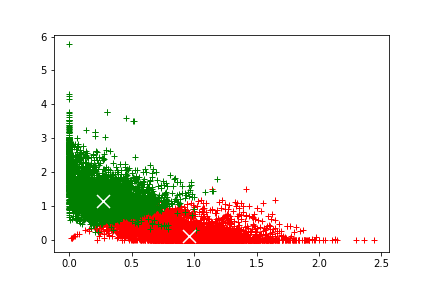
|  |  |
| --- | --- |
| **Metric** | **Value** |
| Homogeneity | 0.218 |
| Completeness | 0.305 |
| V-measure | 0.254 |
| Adjusted Rand Index | 0.142 |
| Adjusted Mutual Info Score | 0.218 |

|  |  |  |
| --- | --- | --- |
| **Actual** | **Predicted** |  |
|  | Computer Technology | Recreational Activity |
| Computer Technology | 7 | 3896 |
| Recreational Activity | 1532 | 2447 |

**Question: Can you justify why logarithm transformation may increase the clustering results?**

It should be noted that as the frequency of a term increases, it does not indicate that the relevance of the term should increase by an equivalent amount. To mitigate this issue, we apply non-linear transformation to our data. We particularly use logarithmic function because it does a great job of amortizing the influence of rare or frequent words thus, assigning appropriate weights to each term and improving the clustering performance.

1. ***Logarithmic Transformation + Normalization on NMF***

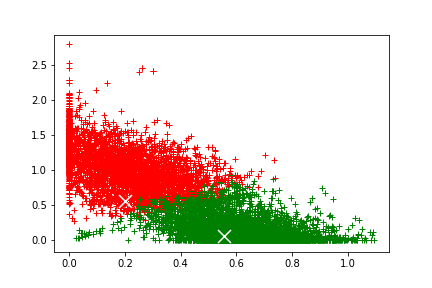


|  |  |
| --- | --- |
| **Metric** | **Value** |
| Homogeneity | 0.493 |
| Completeness | 0.507 |
| V-measure | 0.500 |
| Adjusted Rand Index | 0.561 |
| Adjusted Mutual Info Score | 0.493 |

|  |  |  |
| --- | --- | --- |
| **Actual** | **Predicted** |  |
|  | Computer Technology | Recreational Activity |
| Computer Technology | 866 | 3037 |
| Recreational Activity | 3856 | 123 |

We can see from the above statistics that there is a pretty good improvement in the performance if we apply logarithmic transformation and then normalize the features for NMF.

1. ***Normalization + Logarithmic Transformation on NMF***



|  |  |
| --- | --- |
| **Metric** | **Value** |
| Homogeneity | 0.649 |
| Completeness | 0.653 |
| V-measure | 0.651 |
| Adjusted Rand Index | 0.739 |
| Adjusted Mutual Info Score | 0.649 |

|  |  |  |
| --- | --- | --- |
| **Actual** | **Predicted** |  |
|  | Computer Technology | Recreational Activity |
| Computer Technology | 3460 | 443 |
| Recreational Activity | 110 | 3869 |

We can see from the above statistics that the **best improvement in the** performance is obtained after we normalize the features and then apply logarithmic transformation to the NMF reduced data.

## **5. Expand Dataset into 20 categories**

In this part, we have grouped the data-points into 20 clusters which correspond to the 20 original classes in the 20 Newsgroups dataset. We then examine the purity of the clusters using the same 5 measures we used in the previous parts, namely: Homogeneity, Completeness, V-Measure, Adjusted Rand Index and Adjusted Mutual Information Score. The steps taken are as follows:

* Apply TFIDF Transformation
* Reduce the dimensions of TFIDF matrix using Truncated SVD and NMF. The results have been compared for both these reduction techniques.
* Sweep through different values of r (dimension parameter) and record the best value for LSI and NMF decomposition respectively.
* Apply different transformations such as Normalization on LSI and NMF decomposed data, Logarithmic Transformation on only NFM decomposed data. We have also tried the combinations of Normalization + Logarithmic Transformation as well as Logarithmic Transformation + Normalization on NMF decomposed data.
* Visualize the 20 clusters in 2D plane after applying the above mentioned transformations.

**a) Apply TFIDF Transformation**

After applying the TFIDF transformation for 20 classes on the entire dataset, we get the following results:

|  |  |
| --- | --- |
| **Number of documents** | **Number of terms extracted** |
| 18846 | 52333 |

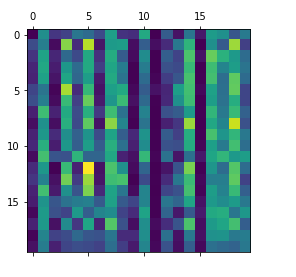
These results were obtained by applying the same parameters as in part 1 of this project.

**b) Choosing Best Value of dimension parameter (r)**

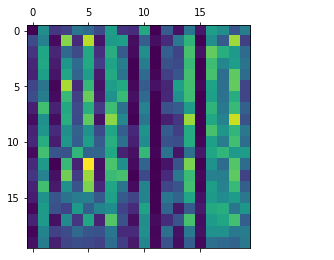
We decomposed the TFIDF matrix as received above using LSI and NMF decomposition. A comparison was drawn for the 5 measures as described above. It is stated in the following table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Dimension Parameter** | **Decomposition** | **Homogeneity** | **Completeness** | **V-measure** | **Adj Rand Index** | **Adj Mutual Info Score** |
| 1 | LSI | 0.022 | 0.023 | 0.022 | 0.005 | 0.018 |
| 1 | NMF | 0.022 | 0.023 | 0.022 | 0.004 | 0.018 |
| 2 | LSI | 0.158 | 0.172 | 0.165 | 0.051 | 0.155 |
| 2 | NMF | 0.144 | 0.154 | 0.149 | 0.045 | 0.142 |
| 3 | LSI | 0.160 | 0.186 | 0.172 | 0.054 | 0.158 |
| 3 | NMF | 0.144 | 0.173 | 0.157 | 0.052 | 0.141 |
| 5 | LSI | 0.204 | 0.227 | 0.215 | 0.074 | 0.201 |
| 5 | NMF | 0.180 | 0.209 | 0.193 | 0.067 | 0.177 |
| 10 | LSI | 0.281 | 0.319 | 0.299 | 0.114 | 0.279 |
| 10 | NMF | 0.297 | 0.345 | 0.319 | 0.111 | 0.294 |
| 20 | LSI | 0.285 | 0.373 | 0.323 | 0.095 | 0.282 |
| 20 | NMF | 0.280 | 0.344 | 0.321 | 0.082 | 0.277 |
| 50 | LSI | 0.275 | 0.380 | 0.319 | 0.088 | 0.273 |
| 50 | NMF | 0.124 | 0.212 | 0.157 | 0.025 | 0.122 |
| 75 | LSI | 0.279 | 0.393 | 0.336 | 0.077 | 0.277 |
| 75 | NMF | 0.155 | 0.309 | 0.207 | 0.025 | 0.152 |
| 100 | LSI | 0.292 | 0.401 | 0.338 | 0.098 | 0.290 |
| 100 | NMF | 0.183 | 0.297 | 0.226 | 0.034 | 0.180 |
| 300 | LSI | 0.272 | 0.400 | 0.329 | 0.079 | 0.270 |
| 300 | NMF | 0.070 | 0.097 | 0.081 | 0.014 | 0.067 |

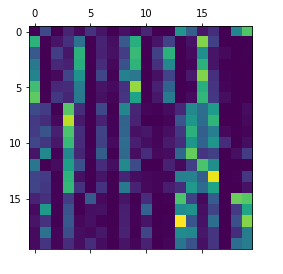
As observed from the table above, it is clear that **the best value for r is 100 for LSI decomposition and is 10 for NMF decomposition.** This can be seen from the plot of all the measures for different values of r which is as follows:

We have visualized the contingency matrices as follows:

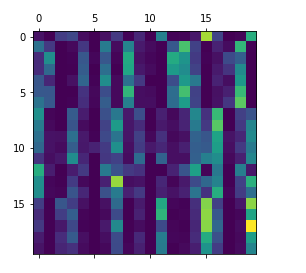
* **r=1; with LSI decomposition**



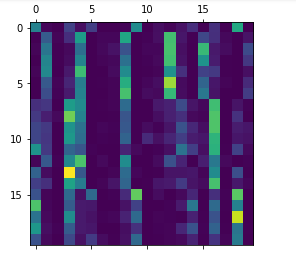
* **r=1; with NMF decomposition**



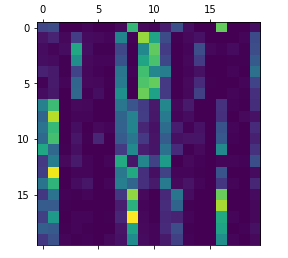
* r=2; **with LSI decomposition**



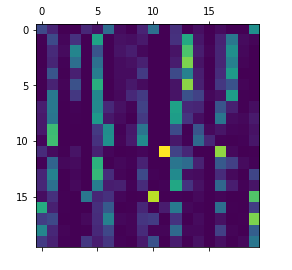
* r=2; **with NMF decomposition**



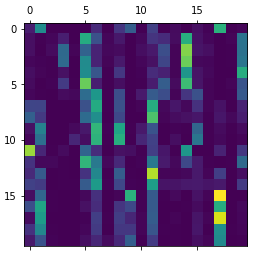
* r=3; **with LSI decomposition**



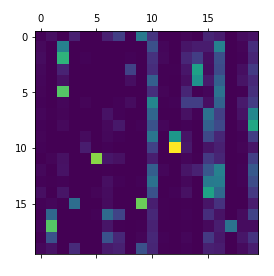
* r=3; **with NMF decomposition**



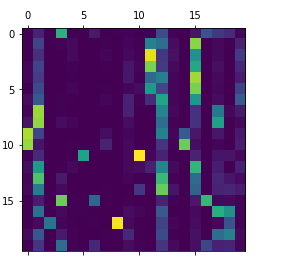
* r=5; **with LSI decomposition**



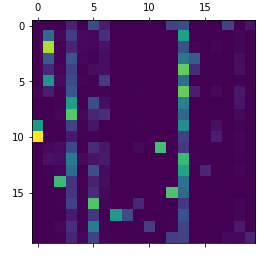
* r=5; **with NMF decomposition**



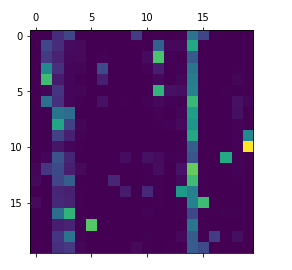
* r=10; **with LSI decomposition**



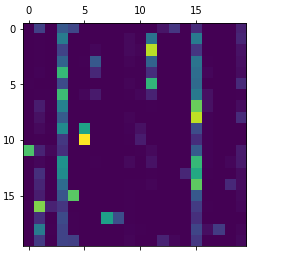
* r=10; **with NMF decomposition**



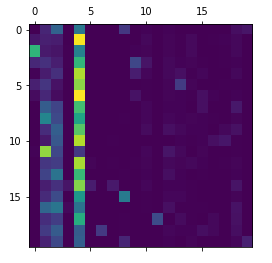
* r=20; **with LSI decomposition**



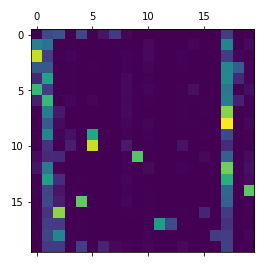
* r=20; **with NMF decomposition**



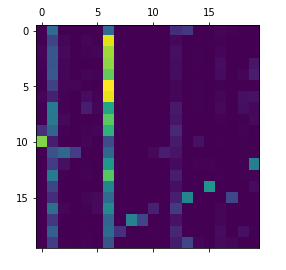
* r=50; **with LSI decomposition**



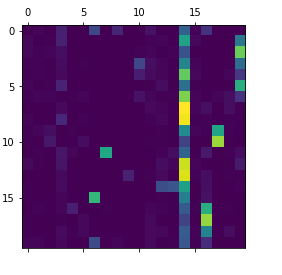
* r=50; **with NMF decomposition**



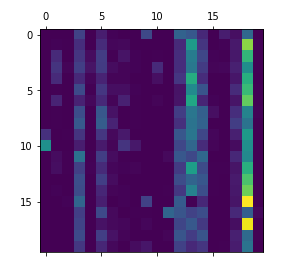
* r=100; **with LSI decomposition**



* r=100; **with NMF decomposition**



* r=300; **with LSI decomposition**

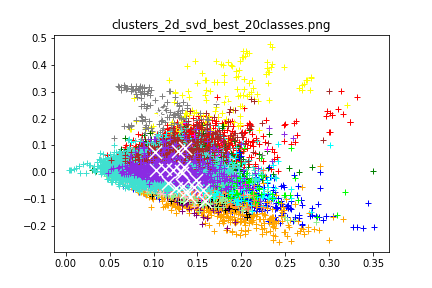


* r=300; **with NMF decomposition**

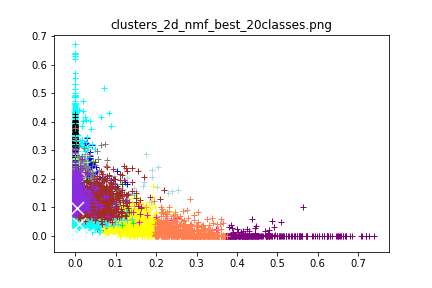
**c) Visualization in 2D with best r:**

We then visualized both the LSI reduced and the NMF reduced matrix in 2D after applying K-means. For visualizing, we used SVD to convert the matrix with best r in two dimensional and then plot the clusters

* **After LSI Decomposition**



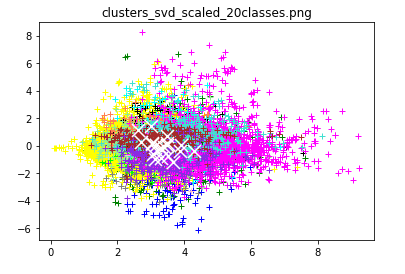
* **After NMF Decomposition**

****

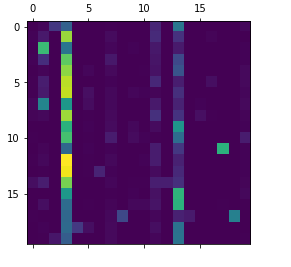
**d) Applying transformations and visualizing in 2D with best r:**

Now, we have applied some transformations on the LSI and NMF decomposed data as in part 4b. Following section shows the visualization and the contingency matrix for for LSI decomposed data (r=100) and NMF decomposed data (r=10) after applying transformations.

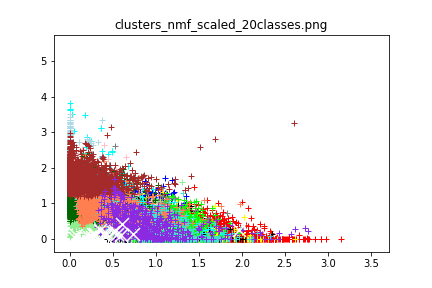
* ***Normalization on LSI Decomposed Matrix***



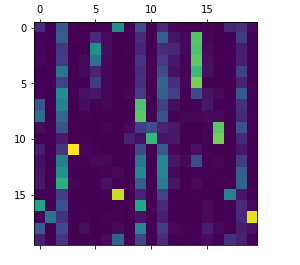
|  |  |
| --- | --- |
| **Metric** | **Value** |
| Homogeneity | 0.148 |
| Completeness | 0.270 |
| V-measure | 0.191 |
| Adjusted Rand Index | 0.031 |
| Adjusted Mutual Info Score | 0.145 |



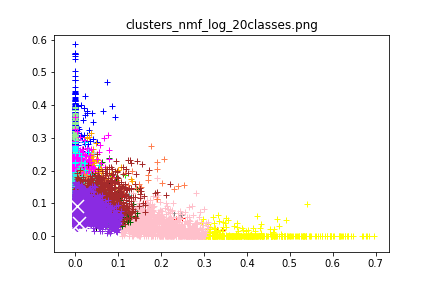
* ***Normalization on NMF Decomposed Matrix***

******

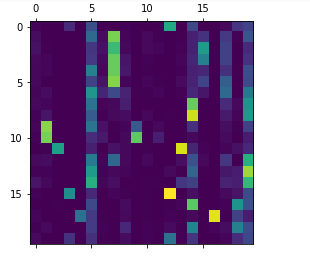
|  |  |
| --- | --- |
| **Metric** | **Value** |
| Homogeneity | 0.260 |
| Completeness | 0.302 |
| V-measure | 0.279 |
| Adjusted Rand Index | 0.099 |
| Adjusted Mutual Info Score | 0.257 |



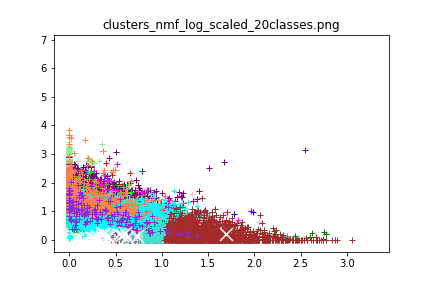
* ***Non-linear (logarithmic) transformation on NMF Decomposed matrix***

******

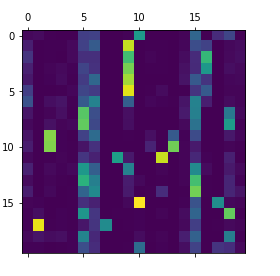
|  |  |
| --- | --- |
| **Metric** | **Value** |
| Homogeneity | 0.294 |
| Completeness | 0.342 |
| V-measure | 0.316 |
| Adjusted Rand Index | 0.114 |
| Adjusted Mutual Info Score | 0.292 |



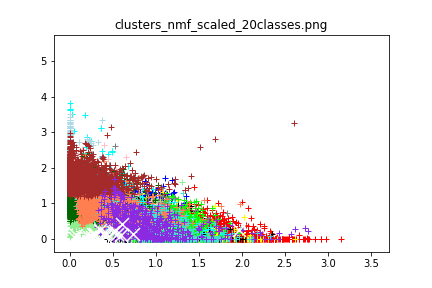
* ***Non-linear (logarithmic) transformation + Normalization on NMF Decomposed matrix***



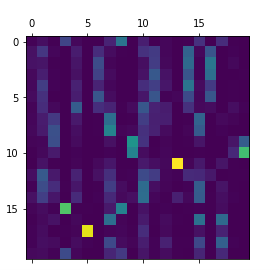
|  |  |
| --- | --- |
| **Metric** | **Value** |
| Homogeneity | 0.276 |
| Completeness | 0.323 |
| V-measure | 0.297 |
| Adjusted Rand Index | 0.105 |
| Adjusted Mutual Info Score | 0.273 |



* ***Normalization + Non-linear (logarithmic) transformation on NMF Decomposed matrix***



|  |  |
| --- | --- |
| **Metric** | **Value** |
| Homogeneity | 0.318 |
| Completeness | 0.330 |
| V-measure | 0.324 |
| Adjusted Rand Index | 0.151 |
| Adjusted Mutual Info Score | 0.316 |



Thus, we see that normalization to LSI decomposed data does not increase the purity. While, for NMF decomposed data, the purity is increased by first normalizing and then applying logarithmic transformation.