**EE219 Project 4**

**Clustering**

**Winter 2017**

Xiongfeng Hu, 304753117

Yanming Zhang, 004761717

Cong Peng, 904760493



**Content**

Introduction 2

Problem 1: Transform the documents into tf-idf vectors 2

Problem 2: K-means Clustering with k = 2 3

Problem 3: K-means Clustering with dimension reduction 5

Problem 4: Visualize the performance of clustering 9

Problem 5: K-means clustering on 20 original subclasses 11

Problem 6: K-means clustering on 6 topic-wise classes 16

# Introduction

In this report we try to find proper representations of the data and evaluate the performance of clustering algorithms. Clustering algorithms are unsupervised methods for finding groups of data point that have similar representations in a proper space. Clustering differs from classification in that no a priori labeling (grouping) of the data points is available. As such, K-means clustering iteratively groups data points into regions characterized by a set of cluster centroids. Each data point is then assigned to the cluster with the nearest cluster centroid.

In this project, We work with “20 Newsgroups” dataset which is a collection of approximately 20,000 documents, partitioned (nearly) evenly across 20 different newsgroups, each corresponding to a different topic. Each topic can be viewed as a “class”. We pretend as if the class labels are not available and aim to find groupings of the documents, where documents in each group are more similar to each other than to those in other group. These clusters capture the dependencies among the documents that are known through class labels. We then use class labels as ground truth to evaluate the performance of the clustering task.

# Problem 1: Transform the documents into tf-idf vectors

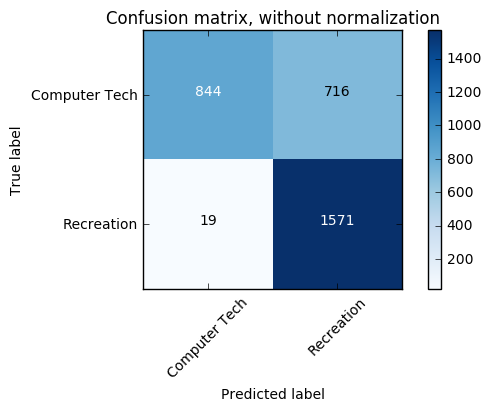
In this part, similar to what we have done in Project 2, we first constructed a tokenizer to trim the text into pure words. Then we implemented TFxIDF vectors for the selected data. During this part, we also split the data into two parts, train and test data.

*Table 1: Data shape*

|  |  |  |
| --- | --- | --- |
| TFxIDF vector | # of documents | # of features |
| Train | 4732 | 9993 |
| Test | 3150 | 9993 |

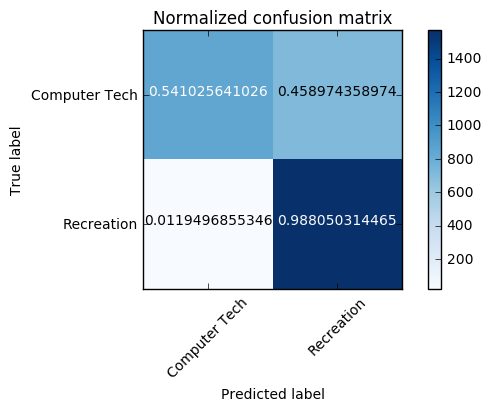
# Problem 2: K-means Clustering with k = 2

In this part, we implement K-means clustering with k = 2. First, we group subclasses into two main classes, ‘Computer Technology’ and ‘Recreation’. Similarly, we plot the confusion matrix of the result. The figures are shown as Figure 1 and Figure 2.



*Figure 1: confusion matrix without normalization of k = 2 clustering*

Below is the normalized confusion matrix required.



*Figure 2: confusion matrix with normalization of k = 2 clustering*

We can easily tell that by only implementing k-means algorithms, we can’t reach to an almost diagonal matrix.

In order to make a concrete comparison of different clustering results, there are various measures of purity a given partitioning of the data points with respect to the ground truth. The measures we examine in this project are homogeneity score, completeness score, adjusted rand score and the adjusted mutual info score. Homogeneity is a measure of how purely clusters contain only data points that belong to a single class. On the other hand, a clustering result satisfies completeness if all of its clusters contain only data points that belong to a single class. Both of these scores span between 0 and 1; where 1 stands for perfect clustering. The Rand Index is similar to accuracy measure, which computes similarity between the clustering labels and ground truth labels. This method counts all pairs of points that both fall either in the same cluster and the same class or in different clusters and different classes. Finally, adjusted mutual information score measures mutual information between the cluster label distribution and the ground truth label distributions.

*Table 2: Measures of Purity of Problem 2*

|  |  |
| --- | --- |
| Measures of Purity | Value |
| Homogeneity Score | 0.30711733196948748 |
| Completeness Score | 0.36252713800633013 |
| Adjusted Rand Score | 0.284250349488987 |
| Adjusted Mutual Info Score | 0.33367404362796677 |

# Problem 3: K-means Clustering with dimension reduction

As is discussed in the Problem(2), high dimensional sparse TF-IDF vectors do not yield a good clustering performance. We have to find a better representation tailored to how the clustering algorithm works.

Therefore, in this part, we use Latent Semantic Indexing (LSI) and Non-negative Matrix Factorization (NMF). In order to get a good initial guess for an appropriate dimensionality to feed in the K-means algorithm, find the effective dimension of the data through inspection of the top singular values of the TF-IDF matrix and see how many of them are significant in reconstructing the matrix with the truncated SVD representation.

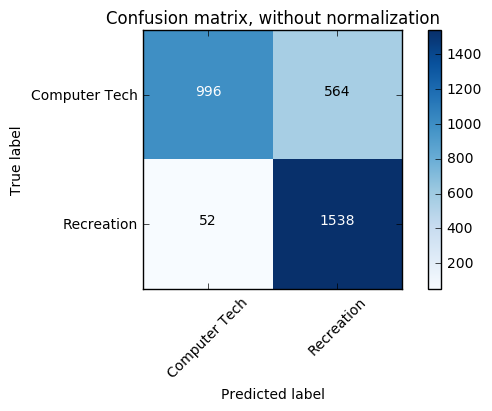
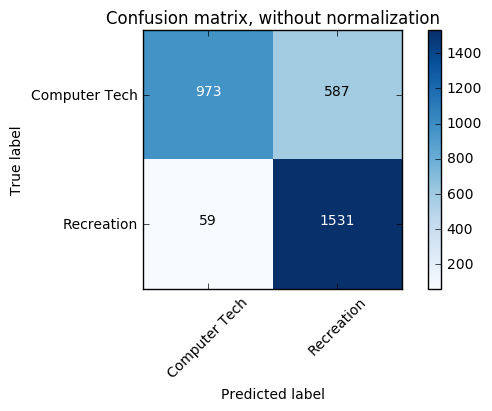
In order to make the clustering purity more satisfying, we apply normalization to the data as preprocessing.

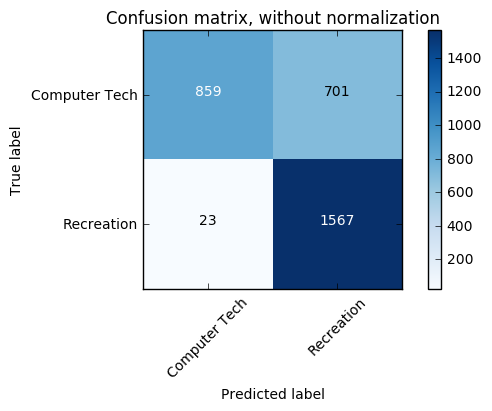
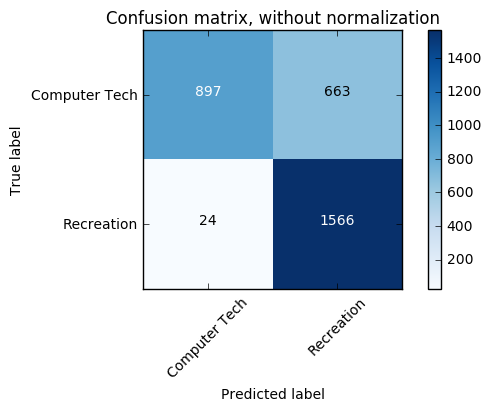
Firstly, we discuss LSI with the dimension starting from as low as 2 to 3 up to the effective dimension. The table 3 below displays the error probability over number of components.

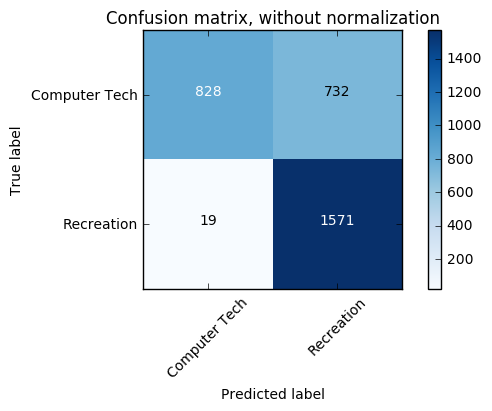
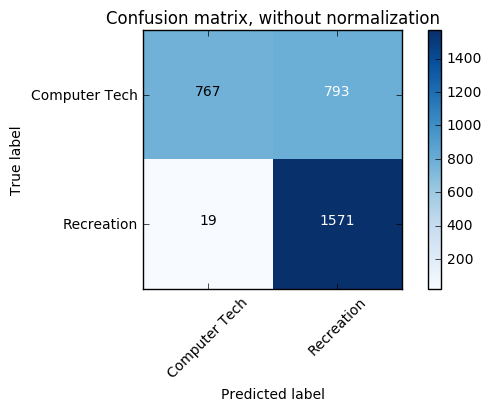
*Table 3: Error Probability over number of components(LSI)*

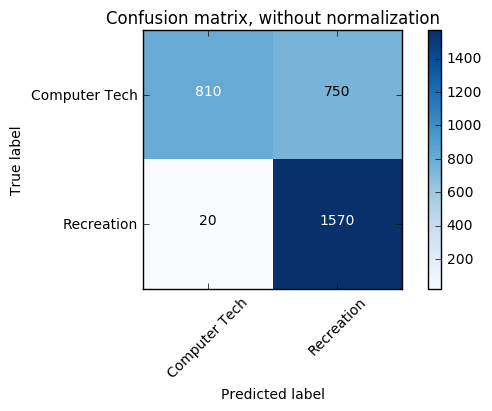
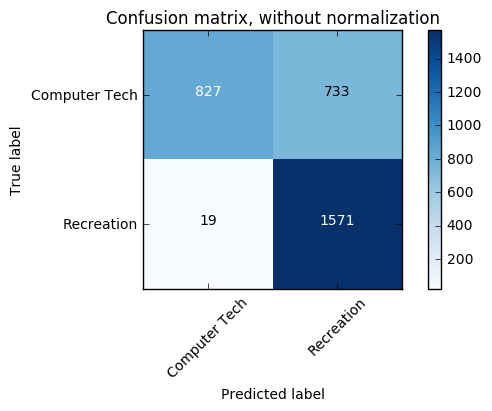
|  |  |
| --- | --- |
| N components | Error Probability (%) |
| 2 | 20.5 |
| 3 | 19.6 |
| 5 | 21.8 |
| 10 | 23.0 |
| 20 | 25.8 |
| 50 | 23.8 |
| 100 | 23.9 |
| 200 | 24.4 |

Below we plot the confusion matrices given n components under LSI (Figure 3)









*Figure 3: confusion matrices* *given n components under LSI*

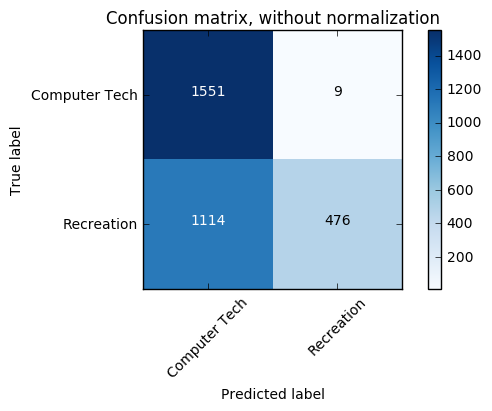
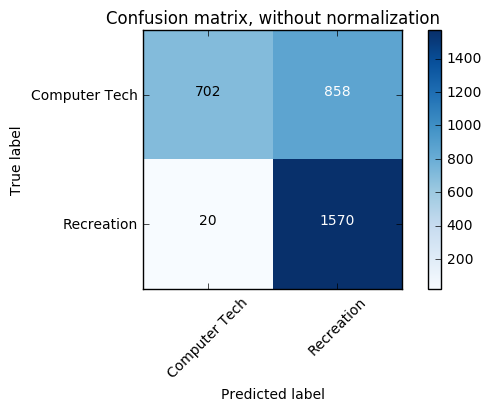
Hence, we conclude that the optimal value in Latent Semantic Indexing is 3.

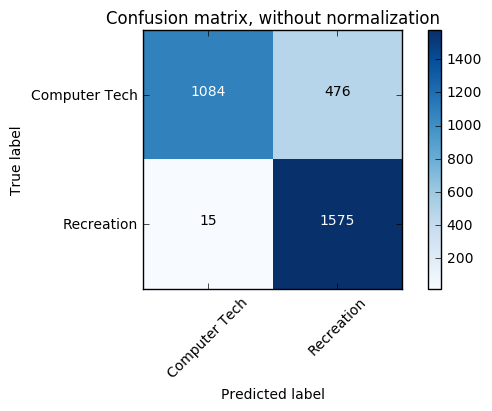
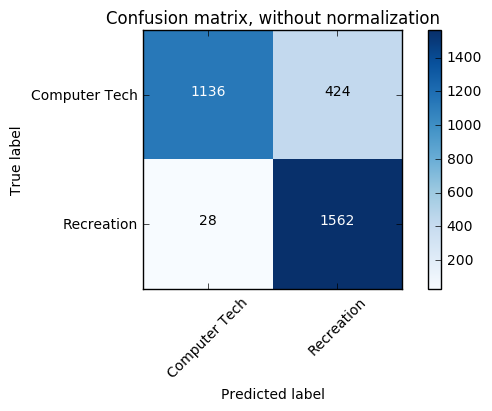
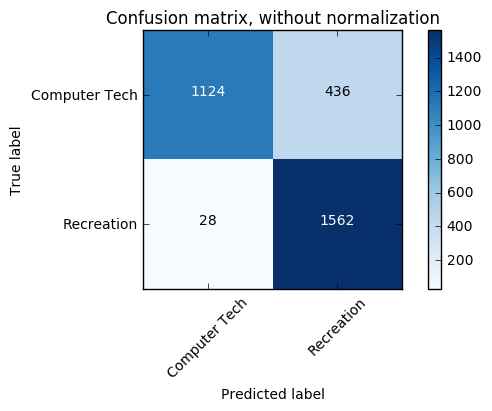
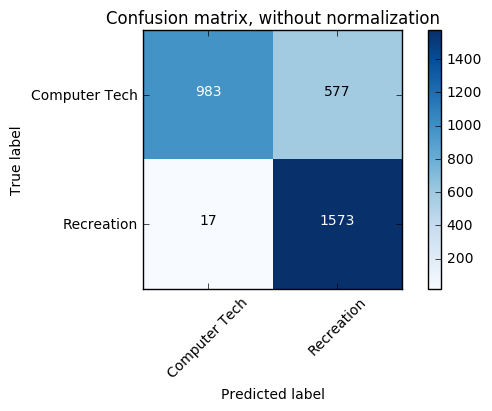
Next, we deal with the data in NMF. The table 4 below displays the error probability over number of components.

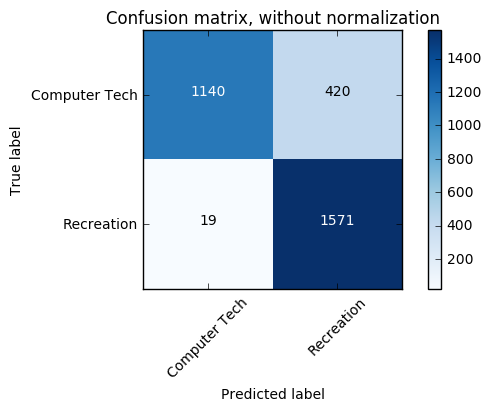
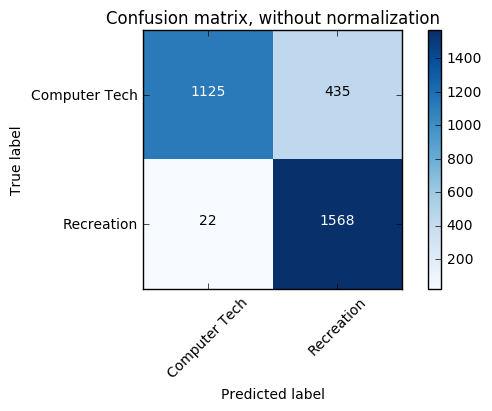
*Table 4: Error Probability over number of components (NMF)*

|  |  |
| --- | --- |
| N components | Error Probability (%) |
| 20 | 27.9 |
| 50 | 35.7 |
| 100 | 18.9 |
| 200 | 14.7 |
| 300 | 14.3 |
| 400 | 15.6 |
| 500 | 14.5 |
| 600 | 13.9 |

Below we plot the confusion matrices given n components under NMF (Figure 4).







*Figure 4: confusion matrices* *given n components under NMF*

Hence, we conclude that the optimal value in NMF is 600.

**Q: Can you justify why logarithm is a good candidate for your TFxIDF data?**

**A:** The aspect emphasized is that the relevance of a term or a document does not increase proportionally with term frequency. Using a sub-linear function therefore helps dumped down this effect. To that extend the influence of very large or very small values (e.g. very rare words) is also amortized. Finally as most people intuitively perceive scoring functions to be somewhat additive using logarithms will make probability of different independent terms from P(A,B)=P(A)P(B) to look more like log(P(A,B))=log(P(A))+log(P(B)).

Below, we report the best final data representation we use, which is NMF in this case, with 600 components. We list the measures of purity mentioned before in the table 5 below.

*Table 5: measures of purity with 600 components in NMF*

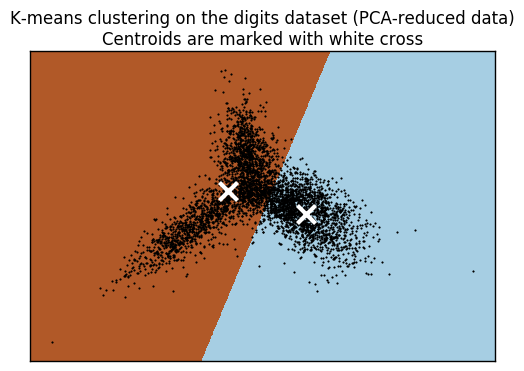
|  |  |
| --- | --- |
| Measures of Purity | Value |
| Homogeneity Score | 0.53613142177905266 |
| Completeness Score | 0.55258582267988761 |
| Adjusted Rand Score | 0.5591697286302122 |
| Adjusted Mutual Info Score | 0.53918340794056765 |

# Problem 4: Visualize the performance of clustering

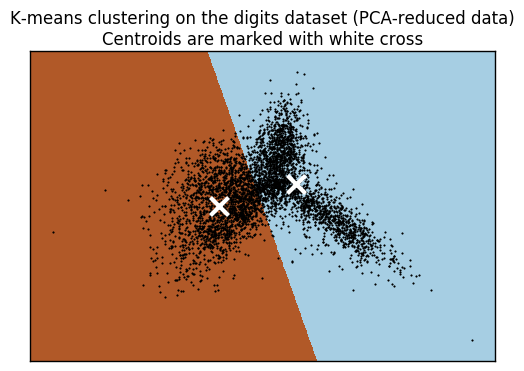
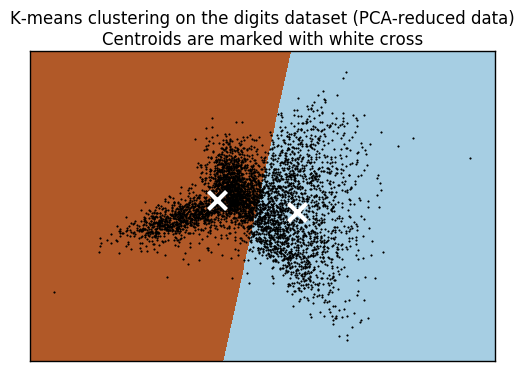
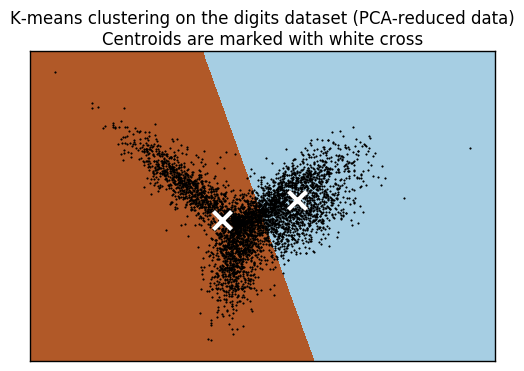
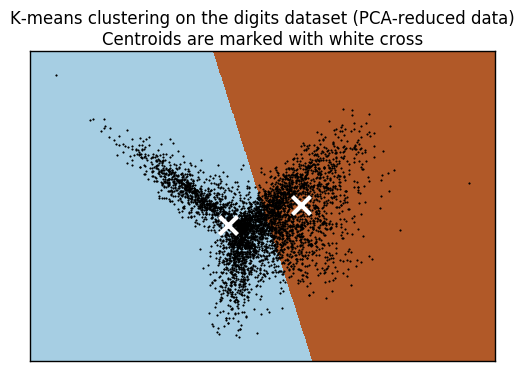
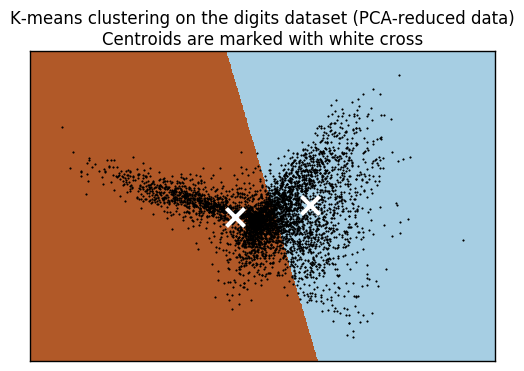
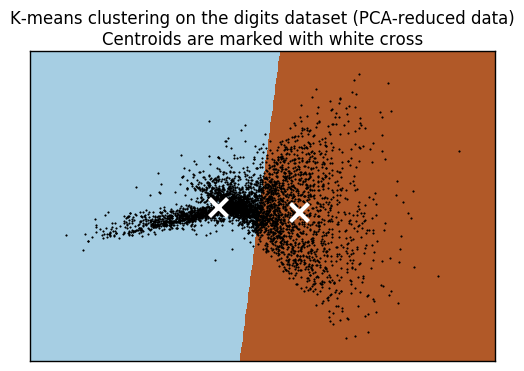
In this problem, to help understand the data more thoroughly, we visualize the performance of the clustering by projecting final data vectors onto 2 dimensions and color-coding the classes.

According to the package given from scikit-learn, we visualize the results on PCA-reduced data. Through this way, we project our data with a dimension of number of features onto two dimension.

Below we display several clustering figure (Figure 5) given different number of components. In first case, the number of components is 600.



N = 600



N = 50

N = 20

N = 200

N = 100

N = 400

N = 300

*Figure 5: clustering figures given different number of components.*

**Q: Can you justify why a non-linear transform is useful?**

**A:** In this dataset, each features is independent with each other, hence, it indicates that there is no relationship between these features, hence, we are actually dealing with non-linear transformation. By using non-linear transformation, we can easily solve a non-linear problem as a linear (straight-line) problem.

# Problem 5: K-means clustering on 20 original subclasses

In this problem, we examine how purely we can retrieve all the 20 original sub-class labels with clustering. Therefore, we include all the documents and the corresponding terms in the data matrix and find proper representation through reducing the dimension of the TF-IDF representation.

We first retrieve all the 20 original sub-class documents and generate the TF-IDF matrix like before. And then we cluster them without any dimension reduction. The purity measures are shown in Table 6.

*Table 6 Purity Measures of Problem 5 Without Dimension Reduction*

|  |  |  |  |
| --- | --- | --- | --- |
| Homogeneity | Completeness | Adjusted Rand-Index | Adjusted\_Mutual\_Info\_Score |
| 0.229 | 0.309 | 0.085 | 0.228 |

Then we apply Truncated SVD (LSI) to reduce the dimension of the TF-IDF representation and tune the parameter of effective ambient space dimension. The purity measures are shown in Table 7.

*Table 7 Purity Measures of Problem 5 With LSI*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n\_components | Homogeneity | Completeness | Adjusted Rand-Index | Adjusted\_Mutual\_Info\_Score |
| 2 | 0.231 | 0.138 | 0.06 | 0.137 |
| 3 | 0.286 | 0.177 | 0.079 | 0.176 |
| 5 | 0.303 | 0.195 | 0.081 | 0.194 |
| 10 | 0.313 | 0.214 | 0.079 | 0.213 |
| 20 | 0.293 | 0.201 | 0.072 | 0.2 |
| 50 | 0.272 | 0.208 | 0.053 | 0.207 |
| 75 | 0.287 | 0.224 | 0.059 | 0.223 |
| 100 | 0.303 | 0.235 | 0.079 | 0.234 |
| 200 | 0.3 | 0.239 | 0.076 | 0.238 |
| 300 | 0.285 | 0.248 | 0.056 | 0.247 |
| 400 | 0.272 | 0.234 | 0.073 | 0.233 |
| 500 | 0.261 | 0.209 | 0.053 | 0.208 |
| 600 | 0.265 | 0.224 | 0.056 | 0.223 |

We plot the relation between n\_components and different measures separately, as shown in Figure 6 – 9. Considering the time to run and each measure result, we choose n\_components = 10 as our optimal parameter.

*Figure 6 n\_ components – Homogeneity of Problem 5 With LSI*

*Figure 7 n\_ components – Completeness of Problem 5 With LSI*

*Figure 8 n\_ components – Adjusted Rand-Index of Problem 5 With LSI*

*Figure 9 n\_ components – Adjusted\_Mutual\_Info\_Score of Problem 5 With LSI*

Then we apply NMF to reduce the dimension of the TF-IDF representation and tune the parameter of effective ambient space dimension. The purity measures are shown in Table 8.

*Table 8 Purity Measures of Problem 5 With NMF*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n\_components | Homogeneity | Completeness | Adjusted Rand-Index | Adjusted\_Mutual\_Info\_Score |
| 2 | 0.221 | 0.133 | 0.059 | 0.132 |
| 3 | 0.269 | 0.17 | 0.078 | 0.169 |
| 5 | 0.29 | 0.196 | 0.081 | 0.195 |
| 10 | 0.279 | 0.199 | 0.062 | 0.199 |
| 20 | 0.272 | 0.206 | 0.06 | 0.205 |
| 50 | 0.27 | 0.217 | 0.057 | 0.216 |
| 75 | 0.275 | 0.226 | 0.065 | 0.225 |
| 100 | 0.273 | 0.229 | 0.058 | 0.228 |
| 200 | 0.28 | 0.224 | 0.072 | 0.223 |
| 300 | 0.29 | 0.261 | 0.052 | 0.26 |
| 400 | 0.314 | 0.26 | 0.08 | 0.259 |
| 500 | 0.31 | 0.255 | 0.086 | 0.254 |
| 700 | 0.336 | 0.267 | 0.081 | 0.266 |

We then plot the relation between n\_components and different measures separately, as shown in Figure 10 – 13. As we can see from the plots, the larger the n\_components is the better the result is, therefore, we choose n\_components = 700 as our optimal parameter.

*Figure 10 n\_ components – Homogeneity of Problem 5 With NMF*

*Figure 11 n\_ components – Completeness of Problem 5 With NMF*

*Figure 12 n\_ components – Adjusted Rand-Index of Problem 5 With NMF*

*Figure 13 n\_ components – Adjusted\_Mutual\_Info\_Score of Problem 5 With NMF*

# Problem 6: K-means clustering on 6 topic-wise classes

In this problem, we examine how purely we can retrieve the topic-wise classes labels with clustering.

We first retrieve all the 20 original sub-class documents and generate the TF-IDF matrix like before. The difference between this problem and problem 5) is that we need to further categorize the 20 subclasses into 6 topic-wise classes, so we first map the origin labels to the new labels. And then we cluster them without any dimension reduction. The purity measures are shown in Table 9.

*Table 9 Purity Measures of Problem 6 Without Dimension Reduction*

|  |  |  |  |
| --- | --- | --- | --- |
| Homogeneity | Completeness | Adjusted Rand-Index | Adjusted\_Mutual\_Info\_Score |
| 0.217 | 0.301 | 0.089 | 0.217 |

Then we apply Truncated SVD (LSI) to reduce the dimension of the TF-IDF representation and tune the parameter of effective ambient space dimension. The purity measures are shown in Table 10.

*Table 10 Purity Measures of Problem 6 With Truncated SVD (LSI)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n\_components | Homogeneity | Completeness | Adjusted Rand-Index | Adjusted\_Mutual\_Info\_Score |
| 2 | 0.18 | 0.187 | 0.093 | 0.18 |
| 3 | 0.216 | 0.231 | 0.118 | 0.216 |
| 5 | 0.201 | 0.239 | 0.083 | 0.201 |
| 10 | 0.187 | 0.282 | 0.072 | 0.072 |
| 20 | 0.207 | 0.293 | 0.073 | 0.207 |
| 50 | 0.206 | 0.296 | 0.075 | 0.205 |
| 75 | 0.206 | 0.294 | 0.07 | 0.206 |
| 100 | 0.189 | 0.274 | 0.07 | 0.189 |
| 300 | 0.226 | 0.313 | 0.092 | 0.226 |
| 400 | 0.14 | 0.274 | 0.005 | 0.139 |
| 600 | 0.191 | 0.322 | 0.05 | 0.191 |

We plot the relation between n\_components and different measures separately, as shown in Figure 14 – 17. Considering each measure result, we choose n\_components = 300 as our optimal parameter.

*Figure 14 n\_ components – Homogeneity of Problem 6 With LSI*

*Figure 15 n\_ components – Completeness of Problem 6 With LSI*

*Figure 16 n\_ components – Adjusted Rand-Index of Problem 6 With LSI*

*Figure 17 n\_ components – Adjusted\_Mutual\_Info\_Score of Problem 6 With LSI*

Then we apply NMF to reduce the dimension of the TF-IDF representation and tune the parameter of effective ambient space dimension. The purity measures are shown in Table 11.

*Table 11 Purity Measures of Problem 6 With NMF*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n\_components | Homogeneity | Completeness | Adjusted Rand-Index | Adjusted\_Mutual\_Info\_Score |
| 2 | 0.185 | 0.193 | 0.097 | 0.185 |
| 3 | 0.221 | 0.264 | 0.117 | 0.22 |
| 5 | 0.187 | 0.238 | 0.065 | 0.187 |
| 10 | 0.165 | 0.314 | 0.03 | 0.165 |
| 20 | 0.189 | 0.291 | 0.079 | 0.189 |
| 50 | 0.195 | 0.289 | 0.076 | 0.195 |
| 75 | 0.197 | 0.278 | 0.069 | 0.196 |
| 100 | 0.238 | 0.319 | 0.083 | 0.238 |
| 300 | 0.254 | 0.357 | 0.116 | 0.253 |
| 400 | 0.27 | 0.343 | 0.144 | 0.269 |
| 500 | 0.287 | 0.347 | 0.119 | 0.286 |
| 700 | 0.278 | 0.369 | 0.139 | 0.278 |

We then plot the relation between n\_components and different measures separately, as shown in Figure 18 – 21. As we can see from the plots, the larger the n\_components is the better the result is, therefore, we choose n\_components = 700 as our optimal parameter.

*Figure 18 n\_ components – Homogeneity of Problem 6 With NMF*

*Figure 19 n\_ components – Completeness of Problem 6 With NMF*

*Figure 20 n\_ components – Adjusted Rand-Index of Problem 6 With NMF*

*Figure 21 n\_ components – Adjusted\_Mutual\_Info\_Score of Problem 6 With NMF*