ECE219 Large-Scale Data Mining: Models and Algorithms **(Due: Wed, 02/05/20)**

Project 2: Clustering

Instructor: Vwani Roychowdhury Zhaoxi Yu(005432230) & Yuhao Yin(104880239)

1 Introduction

Clustering algorithms are unsupervised methods for finding groups of data points that have similar representations in a feature space. Clustering differs from classification in that no *a priori* labeling (grouping) of the data points is available.

K-means clustering is a simple and popular clustering algorithm. Given a set of data points $\{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ in multidimensional space, it tries to find K clusters such that each data point belongs to exactly one cluster, and that the sum of the squares of the distances between each data point and the center of the cluster it belongs to is minimized. If we define $\boldsymbol{\mu}_k$ to be the "center" of the kth cluster, and

$$\gamma_{ik} = \begin{cases} 1, & \text{if } \mathbf{x}_i \text{ is assigned to cluster } k \\ 0, & \text{otherwise} \end{cases}, \quad i = 1, \cdots, N, \quad k = 1, \cdots, K.$$

Then our goal is to find optimal γ_{ik}'s and $\boldsymbol{\mu}_k$'s that minimize $J = \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{ik} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2$. The approach of K-means algorithm is to repeatedly perform the following two steps until convergence:

1. (Re)assign each data point to the cluster whose center is nearest to the data point.
2. (Re)calculate the position of the centers of the clusters: setting the center of the cluster to the mean of the data points that are currently within the cluster.

In this project, the goal includes:

1. To find proper representations of the data, s.t. the clustering is efficient and gives out reasonable results.
2. To perform K-means clustering on the dataset, and evaluate the result of the clustering.
3. To try different preprocessing methods which may increase the performance of the clustering.

2 Clustering of Text Data

2.1 Dataset

We work with "20 Newsgroups" dataset that we already explored in **Project 1**. It is a collection of approximately 20,000 documents, partitioned (nearly) evenly across 20 different newsgroups, each corresponding to a different category (topic).

In order to define the clustering task, 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 groups. We then use class labels as the ground truth to evaluate the performance of the clustering task.

To get started with a simple clustering task, we work with a well-separable portion of the data set and see if we can retrieve the known classes. Specifically, let us define two classes comprising of the following categories.

Class 1	comp.graphics	comp.os.ms-windows.misc	comp.sys.ibm.pc.hardware	comp.sys.mac.hardware
Class 2	rec.autos	rec.motorcycles	rec.sport.baseball	rec.sport.hockey

Table 1: Two Well-Separated Classes

We would like to evaluate how purely the *a priori* known classes can be reconstructed through clustering. That is, we take all the documents belonging to these two classes and perform unsupervised clustering into two clusters. Then we determine how pure each cluster is when we look at the labels of the documents belonging to each cluster.

2.2 Building the TF-IDF Matrix

Following the steps in Project 1, transform the documents into TF-IDF vectors. Use `min_df=3`, exclude the stopwords (no need to do stemming or lemmatization).

QUESTION 1. Report the dimensions of the TF-IDF matrix you get.

Remark. We read in this "20 Newsgroups" textual dataset with headers and footers, since classification accuracy in Project 1 using textual data without removing headers and footers tends to be higher than those removing headers and footers. The same situation might apply in unsupervised learning as well: textual data with headers and footers might have better clustering performance compared to those without. Then, we calculate the document-term frequency matrix using `CountVectorizer` with argument setting `min_df=3` and `stop_words='english'`. The dimensions of the TF-IDF matrix is **(7882, 27768)**.

2.3 K-means Clustering using the TF-IDF Data

Apply K-means clustering with $k = 2$ using the TF-IDF data. Note that the `KMeans` class in `sklearn` has parameters named `random_state`, `max_iter` and `n_init`. Please use `random_state=0`, `max_iter`$\geq$`1000` and `n_init`$\geq$`30`. Compare the clustering results with the known class labels.

Given the clustering result and ground truth labels, contingency table $\mathbf{A}$ is the matrix whose entries A_{ij} is the number of data points that belong to both the class C_i the cluster K_j.

QUESTION 2. Report the contingency table of your clustering result.

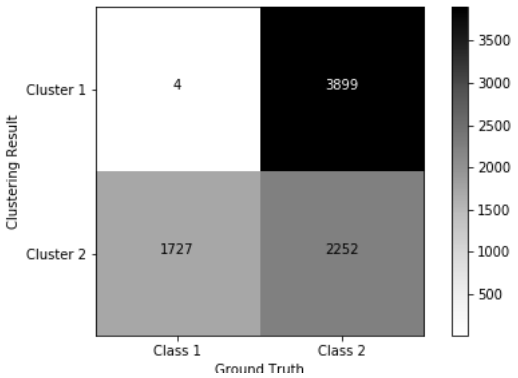


Figure 1: Contingency Table of Clustering Result

Remark. By comparing the clustering results between textual features excluding and including stopwords, we figure out that stopwords plays a significant role in terms of distinguishing two clusters. However, for the current contingency table reported above, certainly there is still plenty of room for further improvement.

In order to evaluate clustering results, there are various measures for a given partition of the data points with respect to the ground truth. We will use the measures **homogeneity score**, **completeness score**, **V-measure**, **adjusted Rand Index score** and **adjusted mutual info score**, all of which can be calculated by the corresponding functions provided in `sklearn.metrics`.

- **Homogeneity** is a measure of how "pure" the clusters are. If each cluster contains only data points from a single class, the homogeneity is satisfied.
- On the other hand, a clustering result satisfies **completeness** if all data points of a class are assigned to the same cluster. Both of these scores span between 0 and 1; where 1 stands for perfect clustering.

- The **V-measure** is then defined to be the harmonic average of homogeneity score and completeness score.
- The **adjusted 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, the **adjusted mutual information score** measures the mutual information between the cluster label distribution and the ground truth label distributions.

QUESTION 3. Report the 5 measures above for the K-means clustering results you get.

Homogeneity	0.255245
Completeness	0.336064
V-Measure	0.290132
Adjusted Rand Index	0.182710
Adjusted Mutual Info	0.290058

Table 2: Five Measures for K-means Clustering

Remark. The whole point of all these five different measures is to evaluate the clustering performance and we want all of them to be as close to 1 as possible. Again, there is still much work that needs to be done in terms of feature engineering, in order to facilitate the clustering result.

2.4 Dimensionality Reduction

One of the reasons that high dimensional sparse TF-IDF vectors do not yield a good clustering result is that in a high-dimensional space, the Euclidean distance is not a good metric anymore, in the sense that the distances between data points tends to be almost the same.

K-means clustering has other limitations. Since its objective is to minimize the sum of within-cluster L_2 distances, it implicitly assumes that the clusters are isotropically shaped, i.e. round-shaped. When the clusters are not round-shaped, K-means may fail to identify the clusters properly. Even when the clusters are round, K-means algorithm may also fail when the clusters have unequal variances.

In this part we try to find a "better" representation tailored to the way that K-means clustering algorithm works, by reducing the dimension of our data before clustering. We will use Singular Value Decomposition (SVD) and Non-negative Matrix Factorization (NMF) that you are already familiar with for dimensionality reduction.

First we want to 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. A guideline is to see what ratio of the variance of the original data is retained after the dimensionality reduction.

QUESTION 4. Report the plot of the percent of variance the top r principle components can retain v.s. r, for r = 1 to 1000.

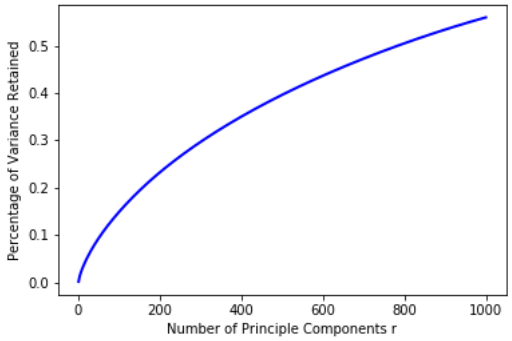


Figure 2: Significance of Top 1000 Singular Values

Remark. For the top 1000 out of more than 27,000 singular values, they are capable of retaining more than 56% of the total variance of the original TF-IDF data after the dimensionality reduction.

Now, use the following two methods to reduce the dimension of the data. Sweep over the dimension parameters for each method, and choose one that yields better results in terms of clustering purity metrics.

- Truncated SVD / PCA

  Note that you don't need to perform SVD multiple times: performing SVD with r = 1000 gives you the data projected on all the top 1000 principle components, so for smaller r's, you just need to exclude the least important features.
- NMF

QUESTION 5. Let r be the dimension that we want to reduce the data to (i.e. `n_components`). Try $r = 1, 2, 3, 5, 10, 20, 50, 100, 300$, and plot the 5 measure scores v.s. r for both SVD and NMF. Report a good choice of r for SVD and NMF respectively.

Remark. According to Table 3 in the next page, which summaries five clustering purity metrics for two dimensionality reduction methods: PCA and NMF, optimality of r across different measurement scores are surprisingly consistent.

Remark 2. For this particular textual dataset, the optimal r for both PCA and NMF is **2**. Note that there exists a universal trade-off between the information preservation and better performance of k-means in lower dimensions.

QUESTION 6. How do you explain the non-monotonic behavior of the measures as r increases?

Remark. When r is relatively large, the Euclidean distance is not a good metric in high-dimensional space due to the curse of dimensionality, in the sense that distances between data points tends to be almost same. Therefore, the k-means clustering algorithm, which groups data points based on their intermediate distances, is no longer effective. On the other hand, when r is relatively small, although k-means clustering could be working in this setup, loss of information turns out to be a much bigger issue in the process of projecting high-dimensional sparse TF-IDF vectors into lower space. These probably are the two main reasons causing non-monotonic behavior of the measures when r varies.

2.5 Visualization

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

- Ground truth class label
- Clustering label

respectively.

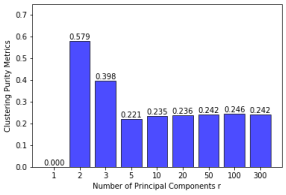
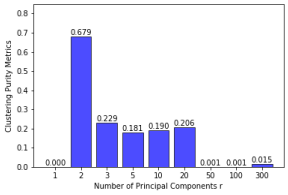
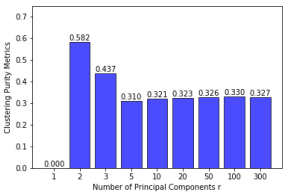
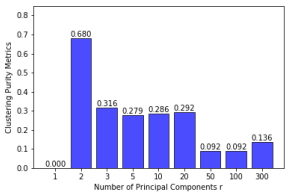
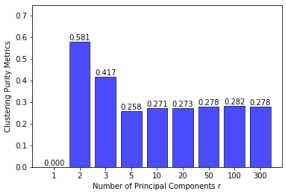
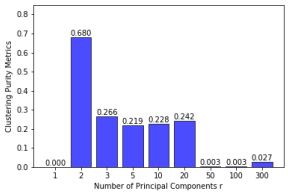
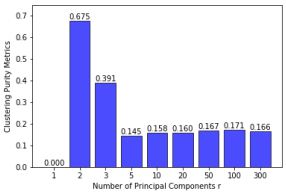
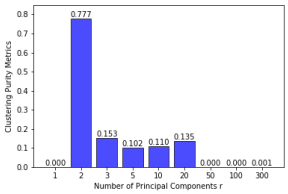
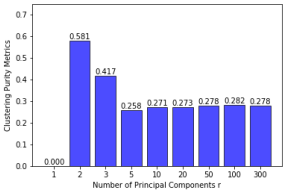
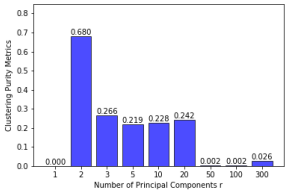
	PCA	NMF
Homogeneity		
Completeness		
V-Measure		
Adjusted Rand Index		
Adjusted Mutual Information		

Table 3: Respective Clustering Metrics for SVD and NMF

QUESTION 7. Visualize the clustering results for:

- PCA with your choice of r
- NMF with your choice of r

Based on the result from **QUESTION 5**, the optimal r, i.e., the optimal number of features, for both PCA and NMF dimensionality reduction methods are 2. Visualize the clustering results by coloring the points according to ground truth class label and clustering label respectively and evaluate the clustering performances:

Principle Component Analysis with optimal $r = 2$:

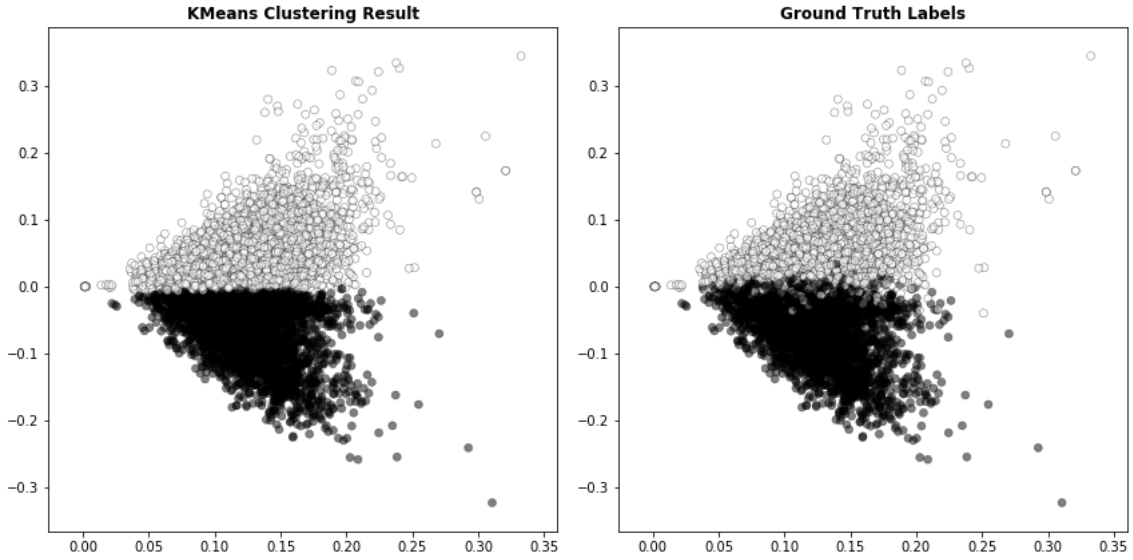


Figure 3: Visualization of Clustering Results using PCA

Non-Negative Matrix Factorization with optimal $r = 2$:

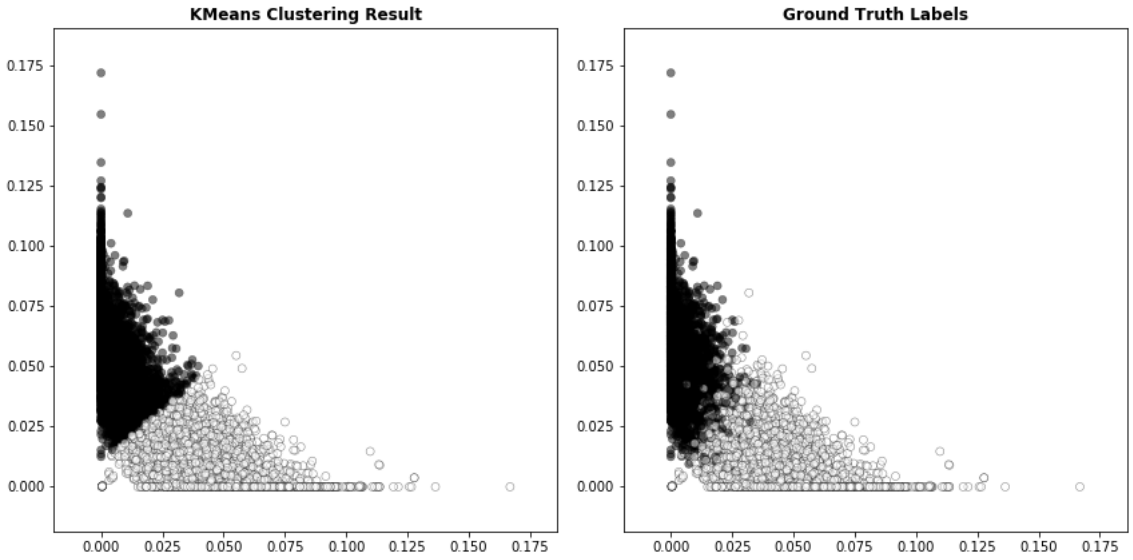


Figure 4: Visualization of Clustering Results using NMF

Remark. Comparing the K-means clustering results with ground truth labels, we conclude that dimensionalty reduction methods PCA and NMF combined with their respective optimal r achieve a reasonable clustering performance.

Now try the transformation methods below to see whether they increase the clustering performance. Perform transformation on SVD-reduced data and NMF-reduced data, respectively. Still use the best r we had in previous parts.

- Scaling features s.t. each feature has unit variance, i.e. each column of the reduced-dimensional data matrix has unit variance (if we use the convention that rows correspond to documents).
- Applying a logarithmic non-linear transformation to the data vectors for the case with NMF (non-negative features):

$$f(x) = \log(x + \epsilon), \quad 0 < \epsilon \ll 1$$

- Try combining the above transformations for the NMF case.

To sum up, try the SVD case w/ and w/o performing scaling (2 possibilities). Similarly, try different combinations of w/ and w/o performing scaling and non-linarity for the NMF case (4 possibilities).

QUESTION 8. Visualize the transformed data as in **QUESTION 7**.

Remark. For logarithmic non-linear transformation, we apply the grid search to figure out the optimal ϵ over $\{10^{-k} \mid k = 2, 3, \cdots, 10\}$ by means of optimizing clustering purity metrics of K-means clustering using NMF features with logarithmic transformation. The resulting optimal $\boldsymbol{\epsilon^* = 10^{-3}}$.

Default PCA:

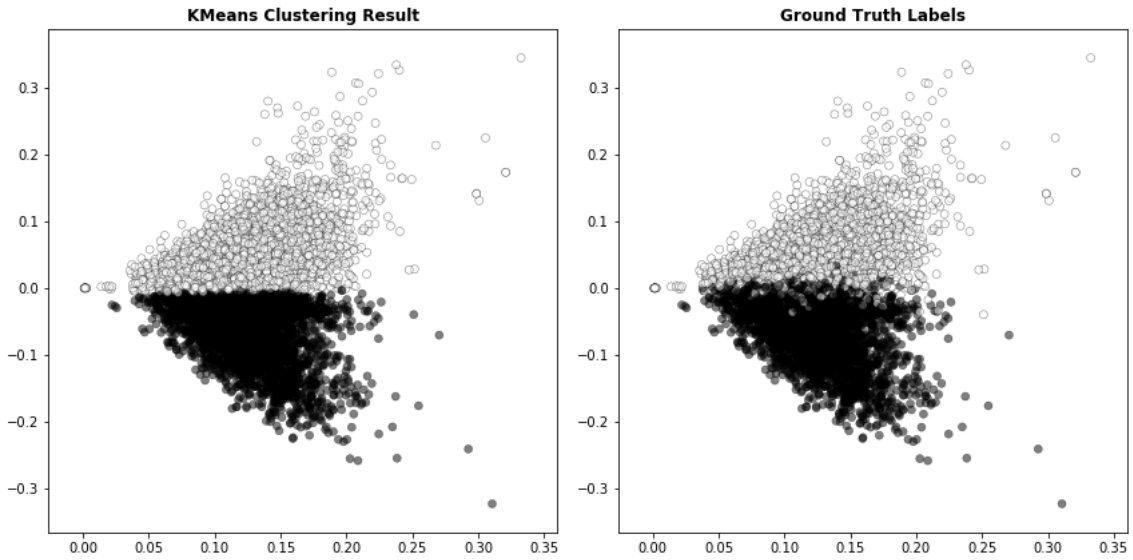


Figure 5: Visualization of Clustering Results using Default PCA

PCA with Feature Standardization (PCA_Standard):

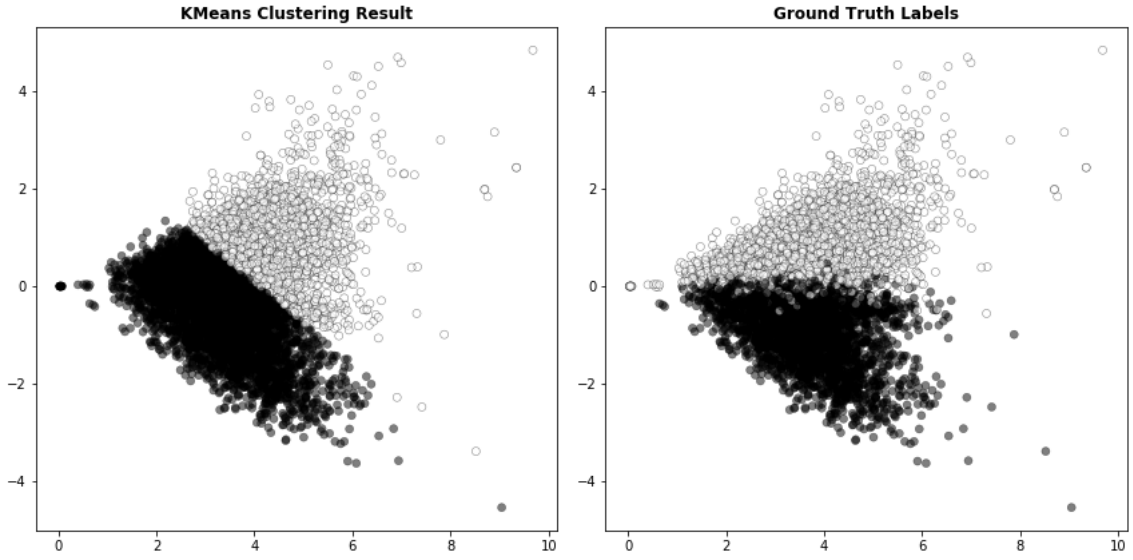


Figure 6: Visualization of Clustering Results using PCA_Standard

Default NMF:

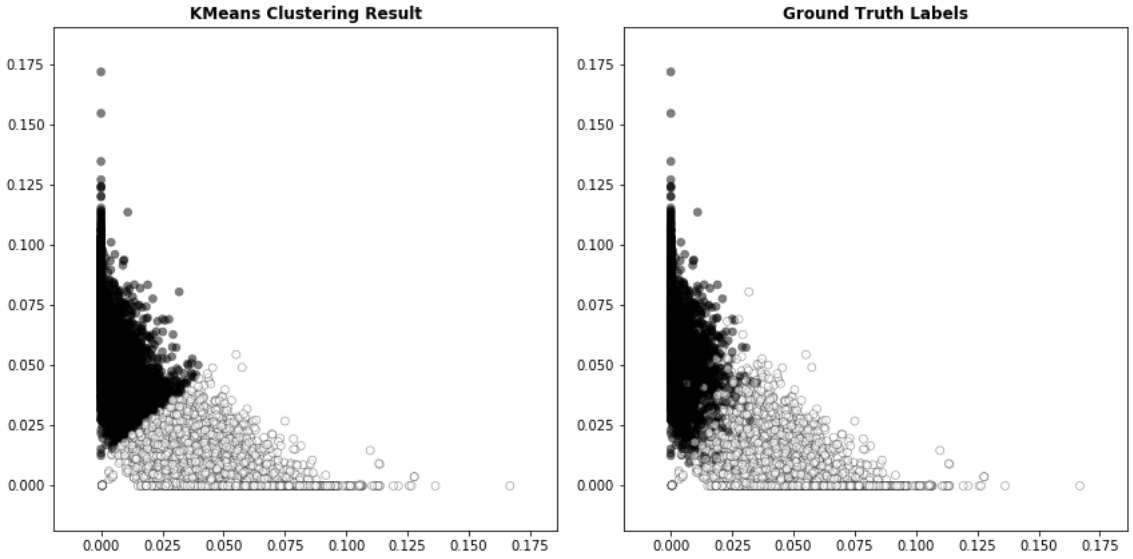


Figure 7: Visualization of Clustering Results using Default NMF

NMF with Feature Standardization (NMF_Standard):

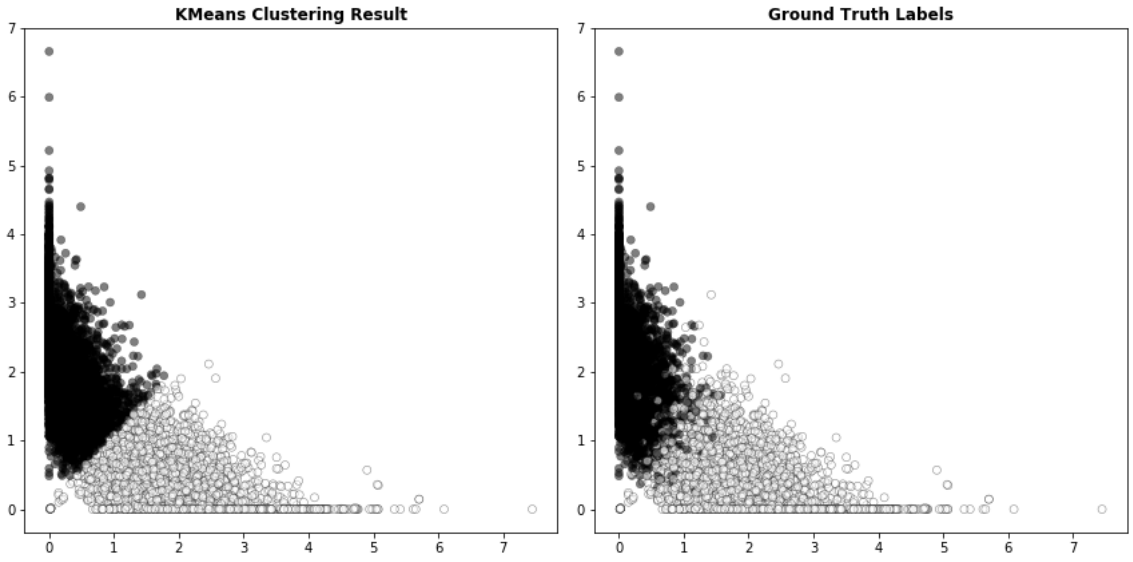


Figure 8: Visualization of Clustering Results using NMF_Standard

NMF with Feature Standardization followed by Logarithmic Transformation (NMF_Standard_Log):

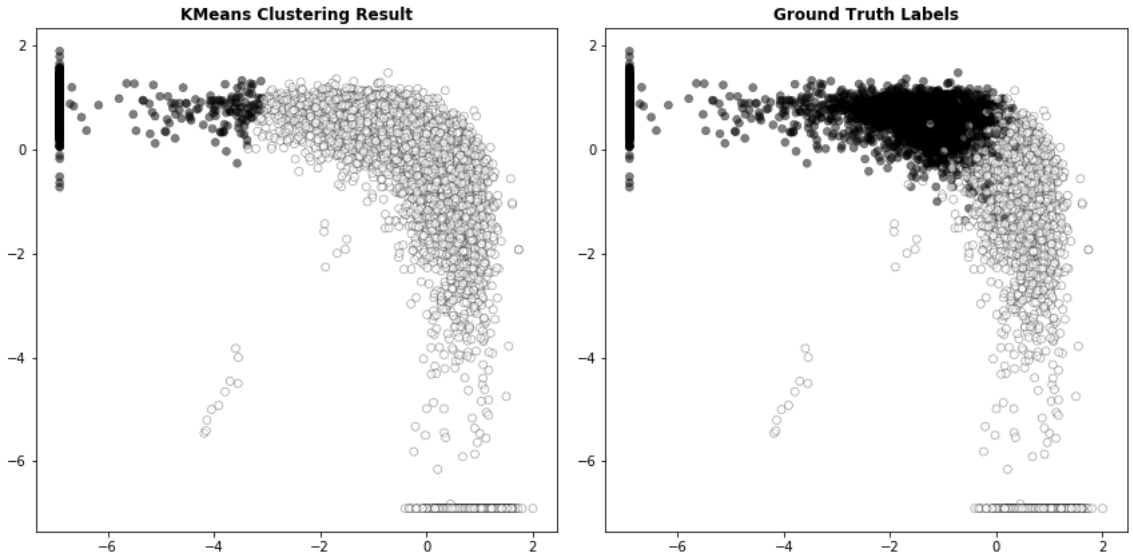


Figure 9: Visualization of Clustering Results using NMF_Standard_Log

NMF with Logarithmic Transformation (NMF_Log):

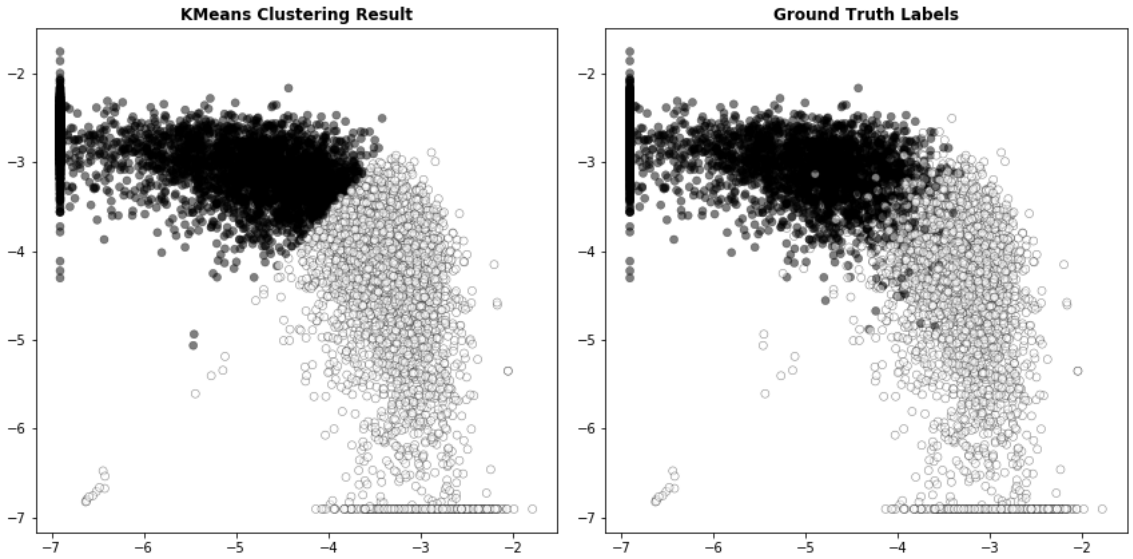


Figure 10: Visualization of Clustering Results using NMF_Log

NMF with Logarithmic Transformation followed by Feature Standardization (NMF_Log_Standard):

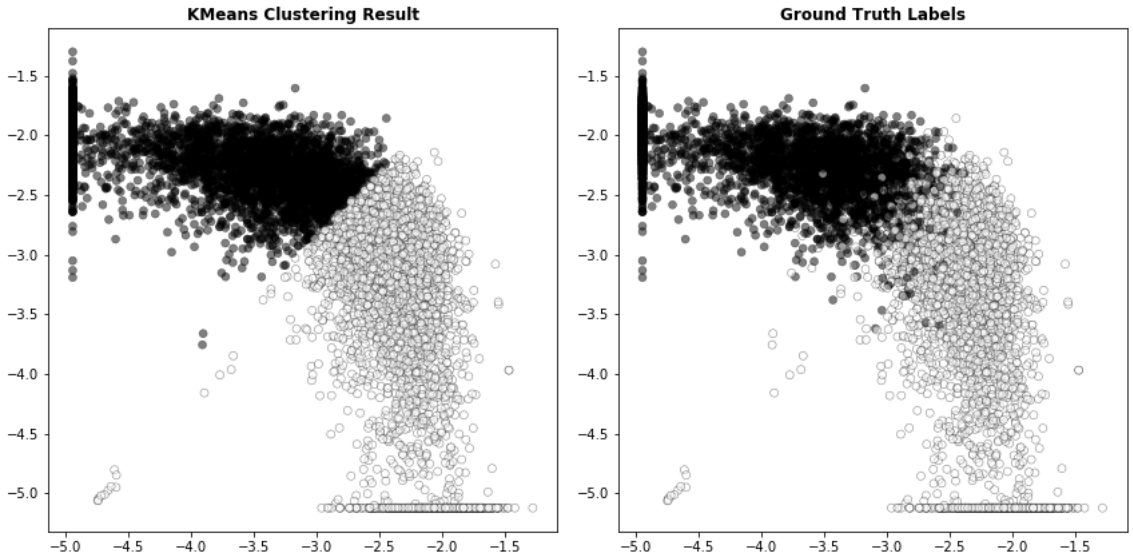


Figure 11: Visualization of Clustering Results using NMF_Log_Standard

QUESTION 9. Can you justify why the "logarithm transformation" may improve the clustering results?

Visualization results above validate the merit of logarithmic transformation in terms of improving the clustering results, by comparing Figure 7 with Figure 10.

Remark. Justification concerning this mainly lies in the fact that the influence of document-term frequency vectors on clustering performance may not be linear. Hence, taking the logarithmic transformation on the original dimension-reduced data tends to lower the influence of those high-frequency vocabularies, and thus improve the K-means clustering result.

Remark 2. On the other hand, we observe performance diminishing from Figure 5 to Figure 6, which indicates feature standardization might not be a good option here. Let's imagine some feature possessing relatively high variance, indicating its important role in distinguishing between two classes. Then, if we rescale it to have unit variance, the significance of the feature will be diminished, as is validated by the above plots.

QUESTION 10. Report the clustering measures (except for the contingency matrix) for the transformed data.

Remark. The clustering measures for the transformed data are summarized in Table 4, where **H** stands for homogeneity score, **C** stands for completeness score, **V** stands for V-measure score, **ARI** stands for adjusted Rand Index score, and last but not least, **AMI** stands for adjusted mutual information score.

Remark 2. These five clustering purity metrics justify that the logarithmic transformation may improve the clustering results and in contrast, the feature standardization may diminish the clustering performance to some extent.

	H	C	V	ARI	AMI
PCA	0.579310	0.581804	0.580555	0.675433	0.580516
PCA_Standard	0.235126	0.263639	0.248568	0.254369	0.248495
NMF	0.679048	0.680132	0.679590	0.777018	0.679560
NMF_Standard	0.684387	0.687121	0.685751	0.775229	0.685723
NMF_Standard_Log	0.193277	0.293192	0.232974	0.109886	0.232889
NMF_Log	0.712163	0.712273	0.712218	0.808652	0.712192
NMF_Log_Standard	0.710127	0.710852	0.710490	0.805004	0.710463

Table 4: Five Measures for Transformed Data

3 Clustering of Your Own Dataset

3.1 Dataset

In this part, we use "hand-written digits" dataset provided by `scikit-learn`, which is a collection of approximately 1800 images, each datapoint is an 8x8 image corresponding to a digit from 0 to 9.

3.2 Feature Extraction

This dataset is easy to use since the features can be extracted directly by `digits.images`, each of which is an 8x8 matrix containing the image information. For instance, the first image from this dataset is shown below, which corresponds to digit 0. Further more, in order to apply a classifier on this data, we need to use `digits.data` to flatten the image, to turn the data in a (samples, feature) matrix. The dimension of `digits.data` matrix is $\mathbf{(1797, 64)}$.

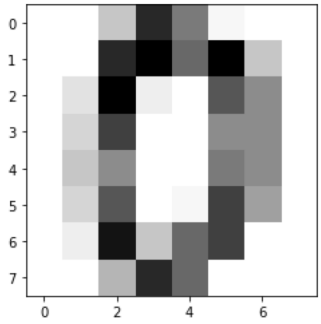


Figure 12: First Image of Digit 0 from Hand-written Digits Dataset

3.3 K-means Clustering using digits.data

Apply K-means clustering with $k = 10$ using `digits.data`. Note that the `KMeans` class in `sklearn` has parameters named `random_state`, `max_iter` and `n_init`. Please use `random_state=0`, `max_iter`$\geq$`1000` and `n_init`$\geq$`30`. Compare the clustering results with the known class labels.

Given the clustering result and ground truth labels, contingency table $\mathbf{A}$ is the matrix whose entries A_{ij} is the number of data points that belong to both the class C_i the cluster K_j.

QUESTION 11. Report the contingency table of your clustering result.

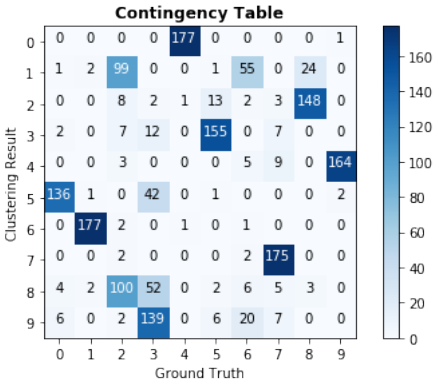


Figure 13: Contingency Table of Clustering Result

Similarly, in order to evaluate clustering results, we use the measures **homogeneity score**, **completeness score**, **V-measure**, **adjusted Rand Index score** and **adjusted mutual info score** for a given partition of the data points with respect to the ground truth.

QUESTION 12. Report the 5 measures above for the K-means clustering results you get.

Homogeneity	0.740193
Completeness	0.748857
V-Measure	0.744500
Adjusted Rand Index	0.668689
Adjusted Mutual Info	0.741927

Table 5: Five Measures for K-means Clustering

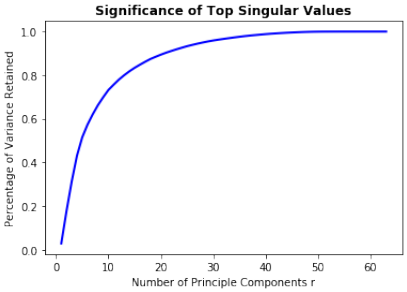
Remark. The whole point of all these five different measures is to evaluate the clustering performance and we want all of them to be as close to 1 as possible. Again, there is still much work that needs to be done in terms of feature engineering, in order to facilitate the clustering result.

3.4 Dimensionality Reduction

Similar to what we demonstrated in part 1, we also use SVD and NMF to reduce dimensionality in this part.

First, we want to find the effective dimension of the data through inspection of the singular values of the `digits.data` matrix. We calculate the ratio of the variance of the original data is retained after the dimensionality reduction.

QUESTION 13. Report the plot of the percent of variance the top r principle components can retain v.s. r, for r = 1 to 64.



Remark. For the top 40 out of 64 singular values, they are capable of retaining more than 98% of the total variance of the original `digits.data` after the dimensionality reduction.

Now, use the following two methods to reduce the dimension of the data. Sweep over the dimension parameter for each method, and choose one that yields better results in terms of clustering purity metrics.

- Truncated SVD / PCA
- NMF

QUESTION 14. Let r be the dimension that we want to reduce the data to (i.e. `n_components`). Try $r = 1, 2, 3, 5, 10, 20, 30, 40, 50, 64$, and plot the 5 measure scores v.s. r for both SVD and NMF. Report a good choice of r for SVD and NMF respectively.

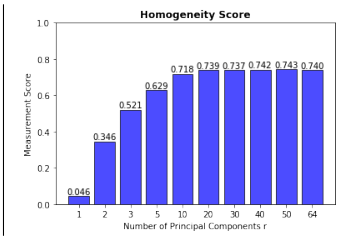
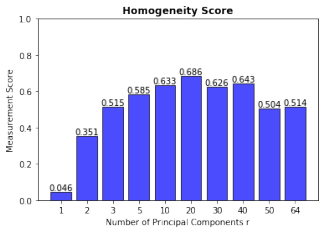
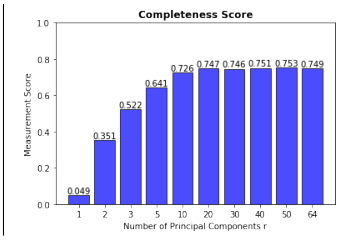
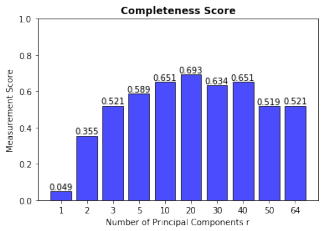
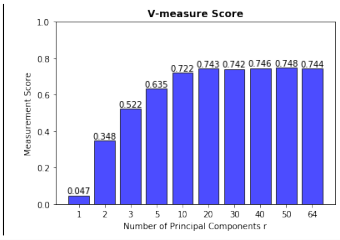
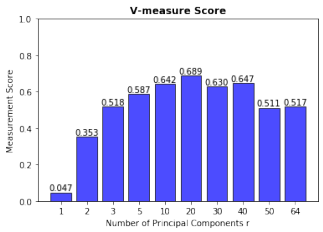
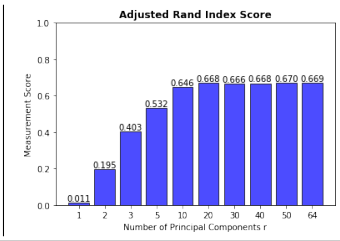
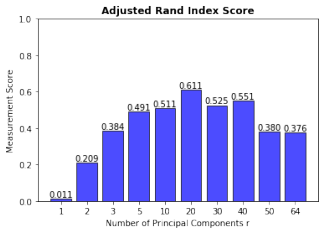
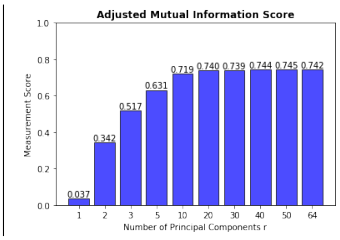
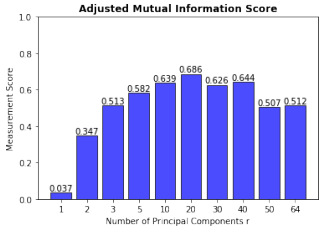
	PCA	NMF
Homogeneity		
Completeness		
V-Measure		
Adjusted Rand Index		
Adjusted Mutual Information		

Table 6: Respective Clustering Metrics for SVD and NMF

Remark. According to Table 6, which summaries five clustering purity metrics for two dimensionality reduction methods: PCA and NMF, the optimal r for PCA is **50** and for NMF is **20**.

3.5 Visualization

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

- Ground truth class label
- Clustering label

respectively.

QUESTION 15. Visualize the clustering results for:

- PCA with your choice of r
- NMF with your choice of r

Based on the result from **QUESTION 14**, the optimal r, i.e., the optimal number of features, for PCA is 50 and for NMF is 20. Visualize the clustering results by coloring the points according to ground truth class label and clustering label respectively and evaluate the clustering performances:

Principle Component Analysis with optimal $r = 50$:

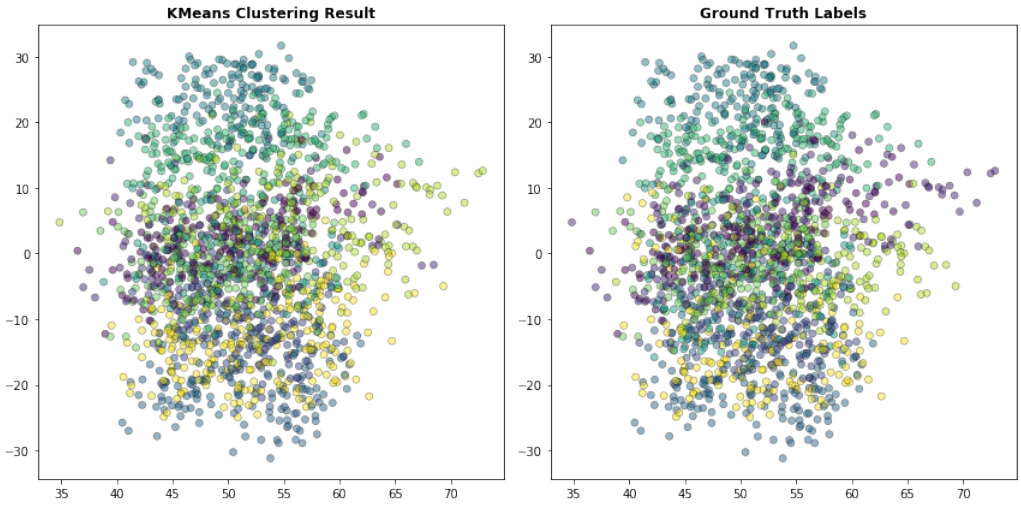


Figure 14: Visualization of Clustering Results using PCA

Non-Negative Matrix Factorization with optimal $r = 20$:

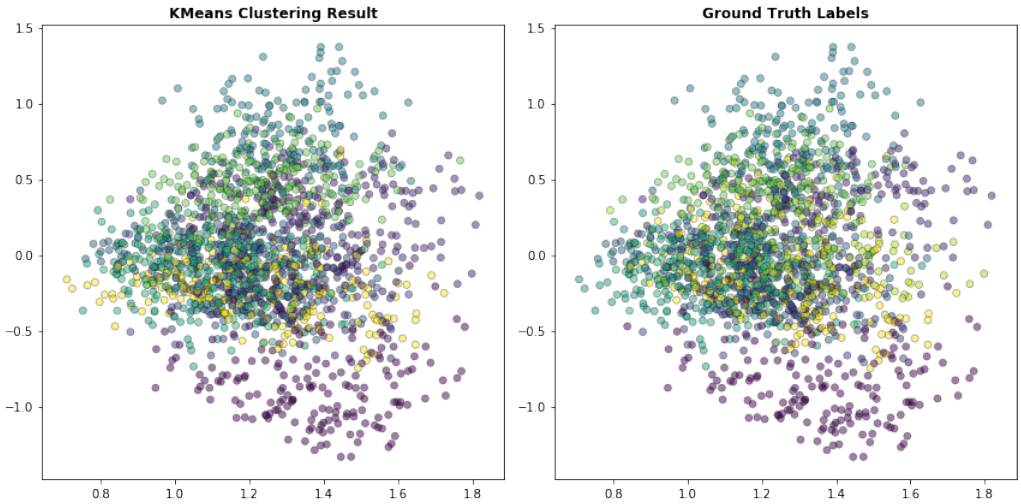


Figure 15: Visualization of Clustering Results using NMF

As we did in part 1, we try the transformation methods of scaling features, applying a logarithmic non-linear transformation and combining of them to see whether they increase the clustering performance.

To sum up, try the SVD case w/ and w/o performing scaling (2 possibilities). Similarly, try different combinations of w/ and w/o performing scaling and non-linarity for the NMF case (4 possibilities).

QUESTION 16. Visualize the transformed data as in **QUESTION 15**.

Remark. For logarithmic non-linear transformation, we apply the grid search to figure out the optimal ϵ over $\{10^{-k} \mid k = 2, 3, \cdots, 10\}$ by means of optimizing clustering purity metrics of K-means clustering using NMF features with logarithmic transformation. The resulting optimal $\boldsymbol{\epsilon^* = 10^{-2}}$.

Default PCA:

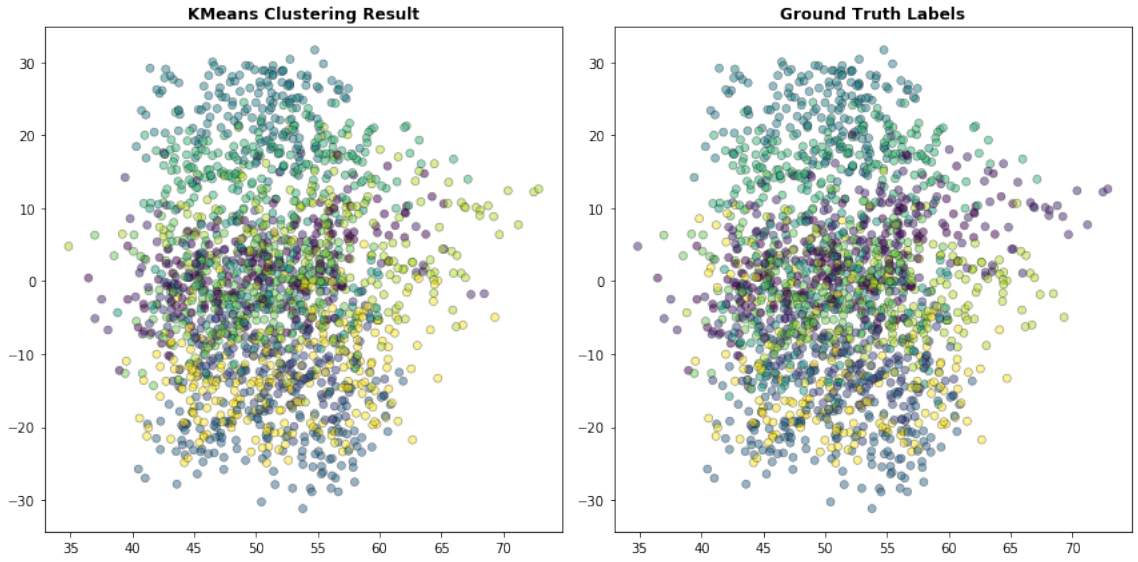


Figure 16: Visualization of Clustering Results using Default PCA

PCA with Feature Standardization (PCA_Standard):

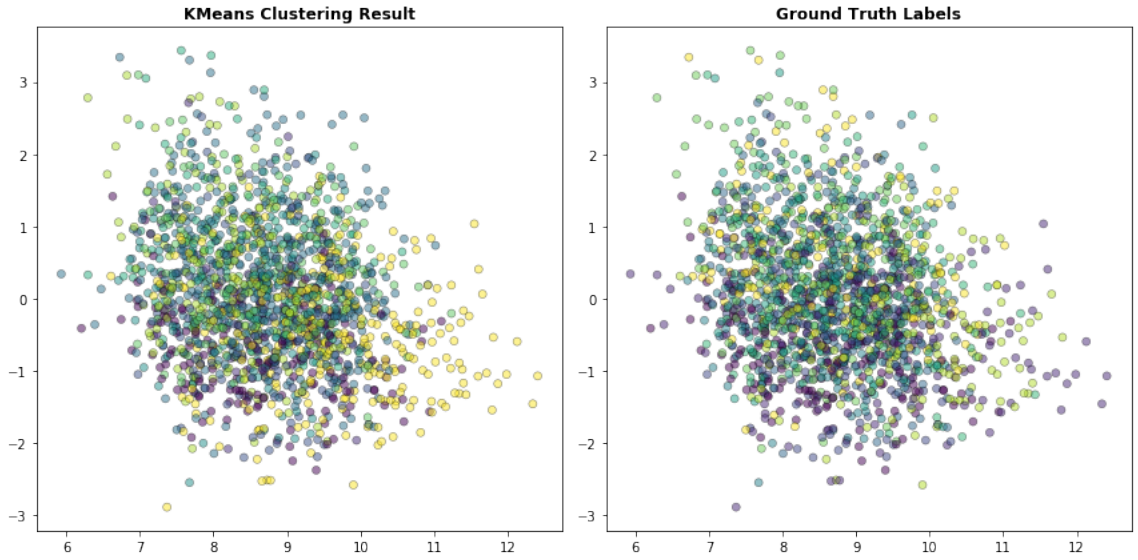


Figure 17: Visualization of Clustering Results using PCA_Standard

Default NMF:

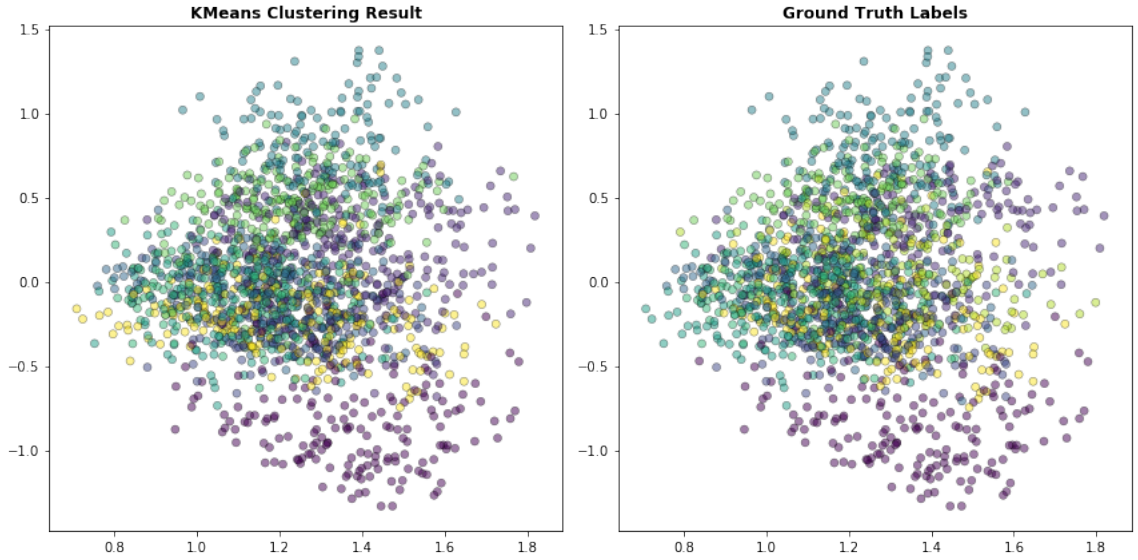


Figure 18: Visualization of Clustering Results using Default NMF

NMF with Feature Standardization (NMF_Standard):

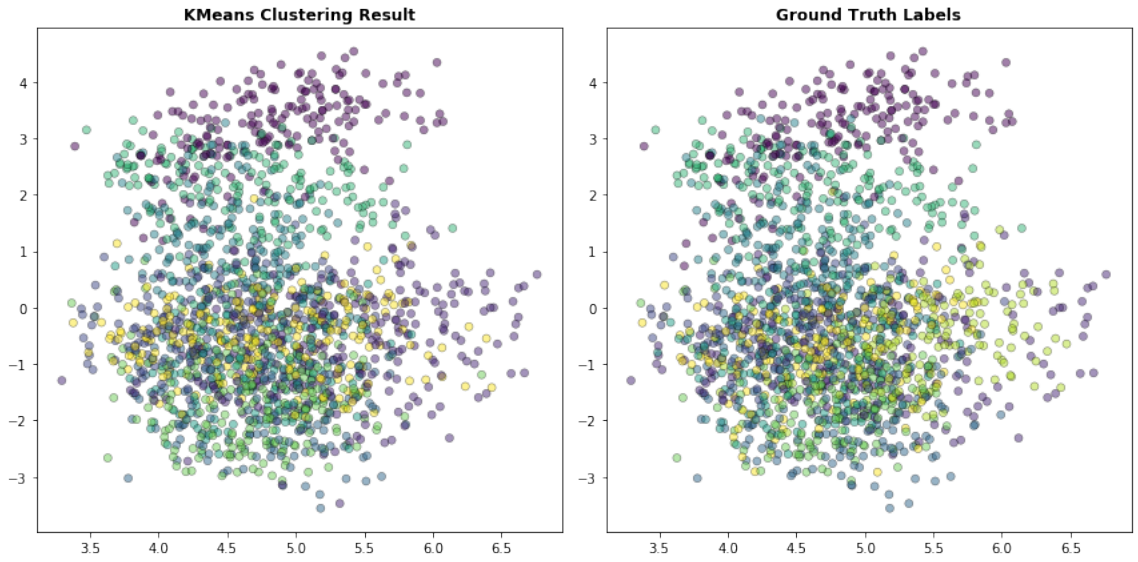


Figure 19: Visualization of Clustering Results using NMF_Standard

NMF with Feature Standardization followed by Logarithmic Transformation (NMF_Standard_Log):

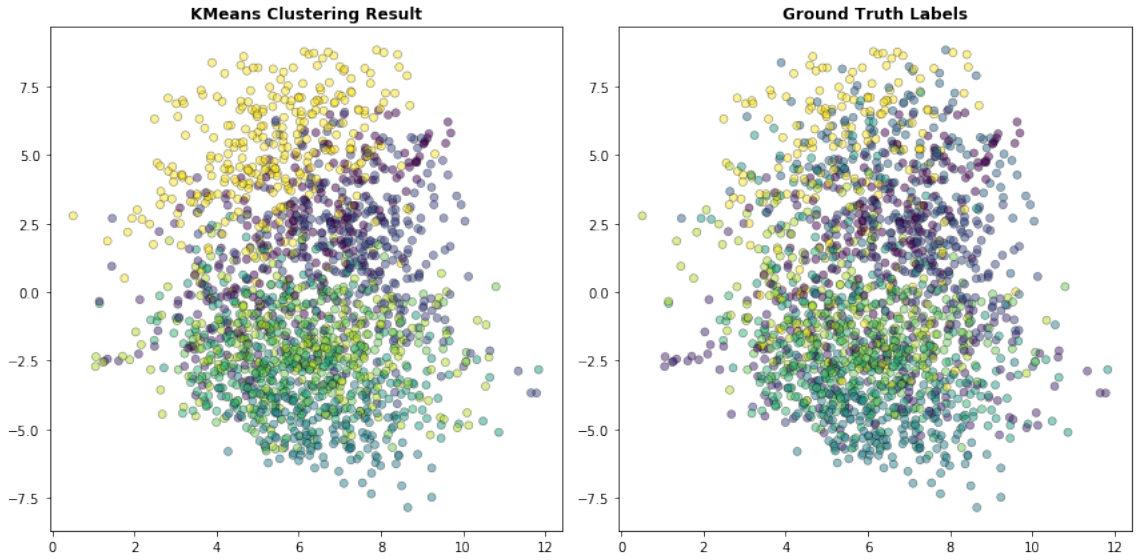


Figure 20: Visualization of Clustering Results using NMF_Standard_Log

NMF with Logarithmic Transformation (NMF_Log):

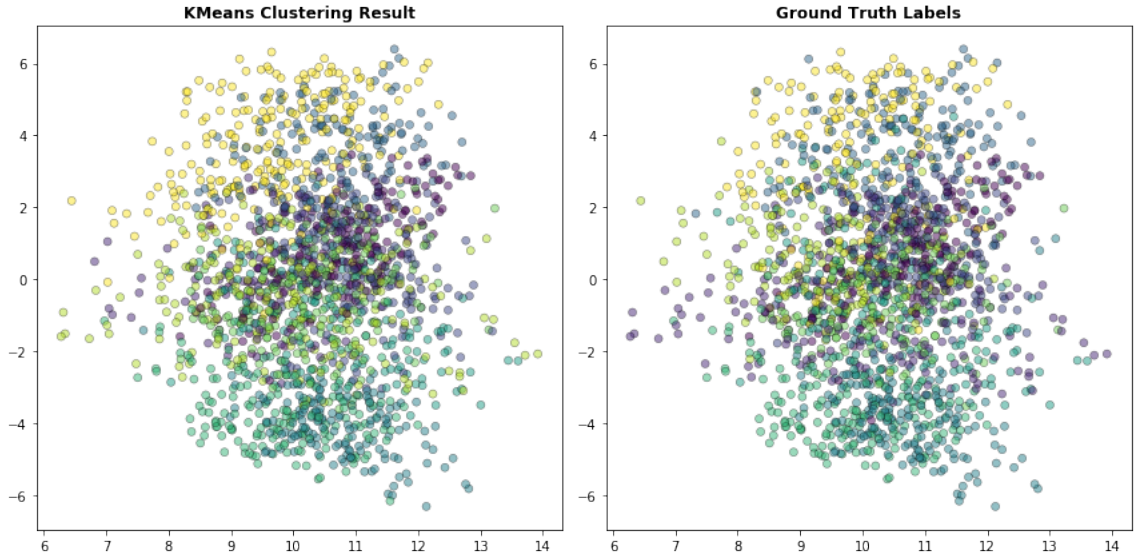


Figure 21: Visualization of Clustering Results using NMF_Log

NMF with Logarithmic Transformation followed by Feature Standardization (NMF_Log_Standard):

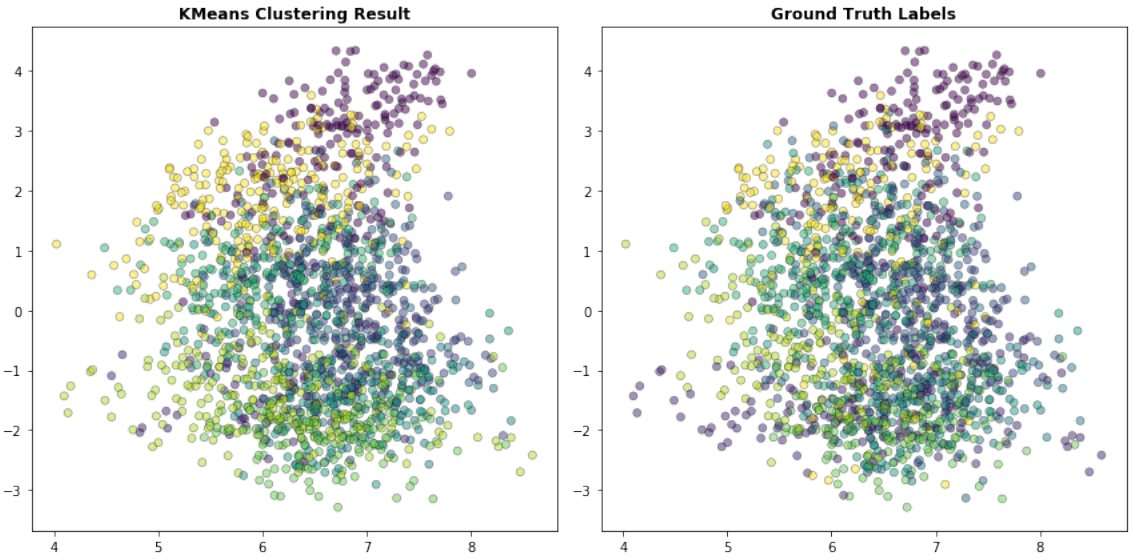


Figure 22: Visualization of Clustering Results using NMF_Log_Standard

QUESTION 17. Report the clustering measures (except for the contingency matrix) for the transformed data.

Remark. The clustering measures for the transformed data are summarized in Table 7, where **H** stands for homogeneity score, **C** stands for completeness score, **V** stands for V-measure score, **ARI** stands for adjusted Rand Index score, and last but not least, **AMI** stands for adjusted mutual information score.

Remark 2. These five clustering purity metrics justify that the feature standardization might not contribute a lot to the improvement of performance, and the logarithmic transformation actually worsen the performance. This is probably because features are not skewed for this specific hand-written digits dataset.

	H	**C**	**V**	**ARI**	**AMI**
PCA	0.743449	0.752538	0.747966	0.669565	0.745427
PCA_Standard	0.626784	0.697702	0.660344	0.515613	0.657149
NMF	0.680106	0.717197	0.698159	0.624762	0.695409
NMF_Standard	0.714956	0.755814	0.734818	0.653812	0.732398
NMF_Standard_Log	0.657063	0.703704	0.679584	0.566321	0.676640
NMF_Log	0.709976	0.718279	0.714103	0.634026	0.711224
NMF_Log_Standard	0.720051	0.727576	0.723794	0.644727	0.721014

Table 7: Five Measures for Transformed Data

4 Color Clustering - Image Segmentation

In this part we would like to perform "segmentation" on images based on color clustering. Choose an image of size $m \times n$ of a celebrity's face. Reshape your image to a matrix of size $mn \times 3$, where the size 3 stems from the RGB channels. Transform pixel RGB information to "normalized (r, g) space" where:

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}$$

Choose a small number of clusters, cluster the pixels according to their colors, and report your result.

QUESTION 18. BONUS - can you suggest a methodology to make an appropriate choice of k and initial seeds of cluster centers?

Remark. For image segmentation task, i.e., clustering RGB images to form object segments in the picture, there is no ground truth for the number of clusters, since for different categories of pictures, there may exist unequal number of appropriate partitions. The methodology we adopt here is to try out different number of clusters for the K-means clustering, and choose the best k based on the segmentation performance.

Remark 2. In terms of the methodology to appropriately choose the initial seeds of cluster centers, running the k-means clustering algorithm multiple times with random initialization of centroid seeds should render a relatively satisfying image segmentation.

Remark 3. The final image segmentation result is illustrated below. We choose the famous NBA player, Stephen Curry from Golden State Warriors, as the target image.

Figure 23: Original Image

Figure 24: Image Segmentation