

**PRESENTATION**

**Clustering**

**Advance Topics in Data Mining (COME515/1)**

**Present by**

Muhammad Ahmed MAJEED

Registration No:

214308915

Master of Computer Engineering

Presented To

Associate Prof. Dr. Serhat OZEKES

Istanbul, Turkey

2021-2022

**CONTENT**

1. Introduction.
2. Data-Set.
3. Problem Statement.
4. Task 1- Report the dimensions of the TF-IDF matrix you get.
5. Task 2- Report the contingency table of your clustering result.
6. Task 3- Report the 5 measures above for the K-means clustering results you get.
7. Task 4- Report the Plot of the percent of variance the top r principle components can retain v.s.r, for r = 1 to 1000.
8. Task 5- Measure scores v.s. r for both SVD and NMF.
9. Task 6- How do you explain the non-monotonic behavior of the measures as increases?
10. Task 7- Visualize the clustering results for.
11. Task 8- Visualize the transformed data as in part (a):
12. Task 9- Can you justify why the “logarithm transformation” may improve the clustering results?
13. Task 10- Report the new clustering measures (except for the contingency matrix) for the clustering results of the transformed data.
14. Task 11- Repeat the following for 20 categories using the same parameters as in 2-class case.
15. Task 12- Try diﬀerent dimensions for both truncated SVD and NMF.

**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 {x1,...,xN} in multidimensional space, it tries to ﬁnd K clusters s.t. each data point belongs to one and only one cluster, and 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 µk to be the “center” of the kth cluster, and then our goal is to ﬁnd rnk’s and µk’s that minimize J = N X n=1 K X k=1 rnkkxn −µkk2. 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.
3. The center positions may be initialized randomly. In this project, the goal includes:
4. To ﬁnd proper representations of the data, s.t. the clustering is eﬃcient and gives out reasonable results.
5. To perform K-means clustering on the dataset, and evaluate the performance of the clustering.
6. To try diﬀerent preprocessing methods which may increase the performance of the clustering.

**Data-Set:**

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). Each topic can be viewed as a “class”.  
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 that we used in Project 1, and see if we can retrieve the known classes. Speciﬁcally, let us deﬁne two classes comprising of the following categories.

**Two well-separated classes:**

* 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

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.  
Use the settings as in the following code to load the data:

categories = ['comp.sys.ibm.pc.hardware', 'comp.graphics', 'comp.sys.mac.hardware', 'comp.os.ms-windows.misc', 'rec.autos', 'rec.motorcycles', 'rec.sport.baseball', 'rec.sport.hockey']

dataset = fetch\_20 newsgroups (subset='all', categories=categories, shuffle=True, random\_state=42)

**Problem Statement:**

* 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).

**Task 1 - Report the dimensions of the TF-IDF matrix you get:**

* 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 ≥ 1000 and n init ≥ 301. Compare the clustering results with the known class labels. (you can refer to sklearn - Clustering text documents using k-means for a basic work ﬂow)
* Given the clustering result and ground truth labels, contingency table A is the matrix whose entries Aij is the number of data points that are members of class ci and elements of cluster kj.

**Task 2 - Report the contingency table of your clustering result:**

* 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 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 stand 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.

**Task 3 - Report the 5 measures above for the K-means clustering results you get:**

* Dimensionality Reduction As you may have observed, high dimensional sparse TF-IDF vectors do not yield a good clustering result. One of the reasons 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 (see [1]). K-means clustering has other limitations. Since its objective is to minimize the sum of within-cluster l2 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. A direct visualization for these problems can be found at sklearn - Demonstration of k-means assumptions.

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.

1. 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.

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

Hint: explained variance ratio of TruncatedSVD objects. See sklearn document

1. 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

**Task 5 - Measure scores v.s. r for both SVD and NMF:**

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 the best r choice for SVD and NMF respectively. Note: what is “best” after all? What if some measures contradict with each other? Here you are faced with this challenge that you need to decide which measure you value the most, and design your own standard of “best”. Please explain your standard and justify it.

**Task 6 - How do you explain the non-monotonic behavior of the measures as increases?**

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.

**Task 7 - Visualize the clustering results for:**

* SVD with its best r
* NMF with its best r

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 non-linear transformation to the data vectors. Here we use logarithm transformation below as an example (try c = 0.01):
* Try combining both transformations (in both orders). To sum up, you are asked to try 2× (2 + 2) = 8 combinations.

**Task 8 - Visualize the transformed data as in part (a):**

**Task 9 - Can you justify why the “logarithm transformation” may improve the clustering results?**

**Task 10 - Report the new clustering measures (except for the contingency matrix) for the clustering results of the transformed data:**

* Expand Dataset into 20 categories in this part we want to examine how purely we can retrieve all 20 original sub-class labels with clustering. Therefore, we need to include all the documents and the corresponding terms in the data matrix and ﬁnd proper representation through dimensionality reduction of the TF-IDF representation.

**Task 11 - Repeat the following for 20 categories using the same parameters as in 2-class case:**

* Transform corpus to TF-IDF matrix;
* Directly perform K-means and report the 5 measures and the contingency matrix;

**Task 12 - Try diﬀerent dimensions for both truncated SVD and NMF:**

Try Different dimensions for both truncated SVD and NMF dimensionality reduction techniques and the diﬀerent transformations of the obtained feature vectors as outlined in above parts. You don’t need to report everything you tried, which will be tediously long. You are asked, however, to report your best combination, and quantitatively report how much better it is compared to other combinations. You should also include typical combinations showing what choices are desirable (or undesirable):