Assignment 4, CSE 474/574

Team Speaking Potatoes – Gaeun Seo, Saeyeon Hwang

Part 2.2 - Filtering target classes (4 points)

• **2.2.1.** Print the name of classes in your training set along with selected_targets you can use target_names attribute of newsgroups_train

comp.sys.mac.hardware
comp.sys.ibm.pc.hardware
rec.motorcycles
sci.space
alt.atheism
comp.graphics
rec.autos

Part 2.3 - Vectorizing documents (12 points)

• **2.3.1.** What does TF-IDF stand for?

TF is a short word for term frequency, which indicates how often a particular word appears in a document. Higher value indicates the importance of the word in the document.

• **2.3.2.** Why don't we only use term frequency of the words in a document as its feature vector? what is the benefit of adding inverse document frequency?

The importance of words decrease if the words are in other documents too. Therefore, we have to multiply TF and IDF(inverse document frequency) to calculate if words are not common in other documents and appears frequently in those documents.

• **2.3.3.** Calculate the tf-idf vectors of the following two documents, assuming this is the entire corpus:

Doc1 - 'a' = 0.12, 'sample' = 0.06
Doc2 - 'another' = 0.08, 'example' = 0.129

Part 3.1 - Sparsity (12 points)

In this section we will interpret the coefficients from the final model you trained on all of the training data.

• **3.1.1** Count the number of non-zeros in each row of the train_vec matrix.

1 row: 89,
2 row: 94,
3 row: 217,
4 row: 70,
5 row: 190,
...,
4077 row: 345,
4078 row: 258,
4079 row: 342,
4080 row: 205,
4081 row: 124

• **3.1.2** What is the average number non zero elements in each row?

170.56187209017398

• **3.1.3** On average what percentage of elements in each row have non-zero elements?

7.128486207033781

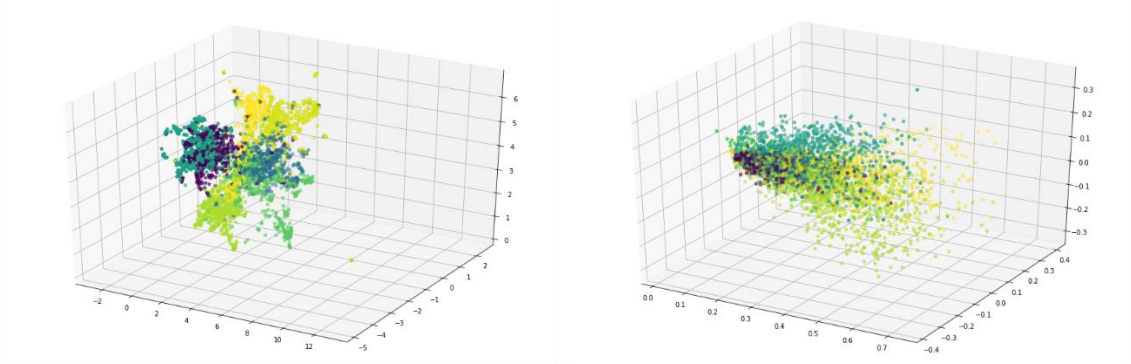
Part 3.2 - SVD (4 points)

• **3.2.1.** What portion of the variance in your dataset is explained by each of the SVD dimensions?

[0.01618638 0.00617073 0.00540306]

Part 3.4 - Visualization (8 points)

• **3.4.1.** Based on your observation, what is the difference between SVD and UMAP embeddings? 1-2 sentences should suffice.



The left one shows the scatter of UMAP embeddings, and the right one shows the scatter of SVD embeddings. The mini clusters are being separated better in UMAP than SVD.

• **3.4.2.** Which one do you prefer to use for a classification task? why? 1-2 sentences should suffice

UMAP is to be preferred for a classification task. For classification task, clarified boundaries make classification easier, and UMAP has clearer boundaries than SVD.

Part 4.1 - Clustering and evaluation (16 points)

• **4.1.1** What is the range of possible values of silhouette coefficients?

The range of possible values of silhouette coefficients is from -1 to 1.

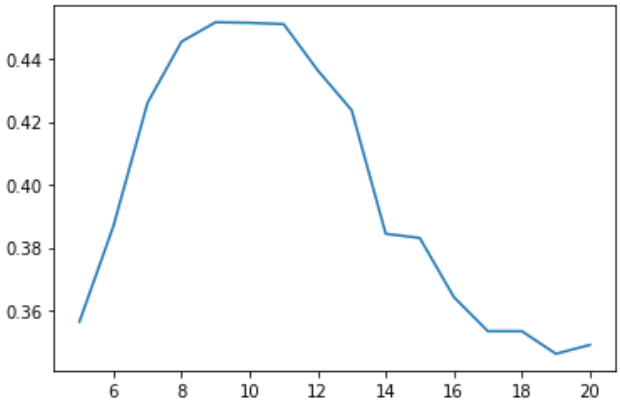
• **4.1.2** Describe what a silhouette score of -1 and 1 mean?

If silhouette score is close to -1, it means the number of clusters is too many or few. On the other hand, if it is close to 1, it means the number of clusters is proper and it is similar with the original one.

• **4.1.3.** Use silhouette score and KMeans from sklearn library to find the optimum number of clusters in your train_umap. Don't forget to use SEED as your kmeans random_seed. In order to do this try different values of cluster numbers from 5 to 20. Choose the one that results in the best score.

Done in ipynb file. The number of clusters that results in the best score was 9

• **4.1.4.** Plot silhouette score for different values of n_clusters (a plot with n_clusters on the x-axis and silhouette score on the y-axis). Don't forget to put the plot in your report.



n_clusters(5~20) on the x-axis, silhouette score on the y-axis

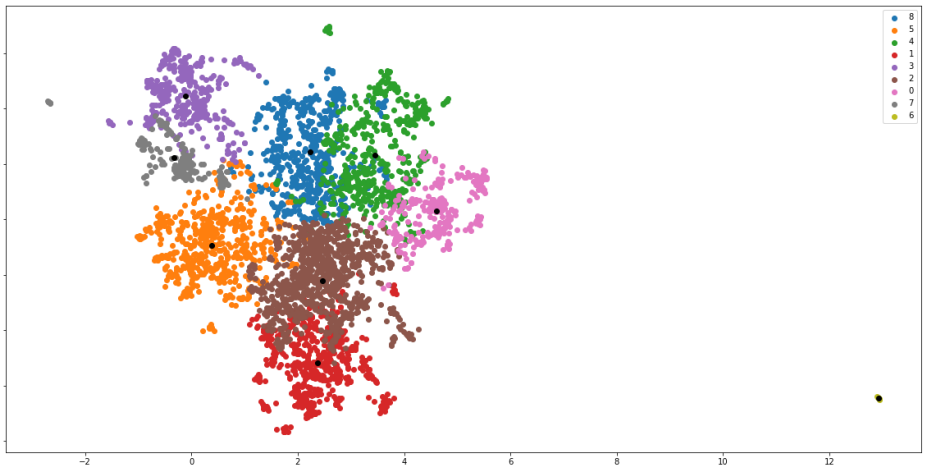
Part 4.2 - Making a Kmeans classifier (4 points)

• **4.2.1** show your mapping (resulted dictionary) inside your project report.

{0: 13, 1: 10, 2: 15, 3: 17, 4: 16, 5: 1, 6: 15, 7: 17, 8: 7}

Part 4.3 - Analyzing clusters (12 points)

• **4.3.1.** Are there any two clusters in your clustering output with the same original label (for example, are there two clusters which both have same training label)? Use your visualizations and describe why?



Yes, it's because the best number of clusters (9) was higher than original number of clusters (7). As you can see in the above figure, the data which labeled as 6 is separated from general data. Moreover, the data is too close from one another, so that it's hard to be separated. These problems make the error in clustering.

• **4.3.2.** Write the function bellow that returns nearest samples to a cluster center. Use this function and explain why there are overlaps in your labels?

Cluster_id=3, 7 were classified as 17, and cluster_id = 2, 6 were classified as 15. For instance, the most central samples of cluster id 3 were
[[-0.12699232, 1.2860556 , 4.6326923],
[-0.14816086, 1.3478087 , 4.8102107],
[-1.5482715 , 0.73476696, 6.273572]].
The most central samples of cluster id 7 were
[[-0.30121598, 0.10842459, 1.7587736],
[-0.3798152 , 0.09612282, 1.8144052],
[-0.06344778, -0.06665345, 1.6088219]].
Because their central samples are too similar, the overlaps in labels occur in clustering.

• **4.3.3.** Can you infere the overlapping label(s) by checking out most central samples? Check with original labels.

Like we explained in 4.3.2., the most central samples of the overlapping labels have similar values from one another. So we can infer overlapping labels by checking out most central samples.

Part 4.4 - Evaluate your Kmeans model on test dataset (12 points)

• **4.4.1.** Using the generated mapping, and your clustering model, predict the labels of test dataset (you can use the embeddings of test data that you generated by umap test_umap)

Done in the ipynb file

• **4.4.2.** Calculate the accuracy of model

accuracy score: 0.7133922001471671

• **4.4.3.** Calculate both micro and macro values of precision, recall and F1 score

micro precision_score: 0.7133922001471671
macro precision_score: 0.7134572762589674
micro recall_score: 0.7133922001471671
macro recall_score: 0.7876341421262192
micro f1_score: 0.713392200147167
macro f1_score: 0.708034731948376

Contribution Statement (Minus 15 points if you do not submit this)

Each group member contributes to the assignment equally, Saeyeon Hwang did part 2, Gaeun Seo did part3, and helped each other when problems happen. Also, both of members did part4 together.