Assignment 4, CSE 474/574, Gravity

Part 2.2 – filtering target classes

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`. Make sure you include this output in your PDF report.

Ans)

target	target_names
1	comp.graphics
7	rec.autos
10	rec.sport.hockey
13	sci.med
15	soc.religion.christian
16	talk.politics.guns
17	talk.politics.mideast

Part 2.3 - Vectorizing documents (12 points)

2.3.1. What does TF-IDF stand for?

Ans) TF-IDF stands for Term Frequency and Inverse Document Frequency. We use TF-IDF to convert sentences into vectors.

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

Ans) We don't get semantic meaning from the feature vector; we cannot get the importance of the word in a sentence when we only use the term frequency of the words in a document as its feature vector. When we use TD*IDF to construct the feature matrix, it downweighs the most common words in the collection of sentences that add no information. In this way, TF-IDF provides automatic screening for the most common words. The values in the feature matrix constructed by using the TF-IDF tell the importance of the word in a sentence. It would be useful for selecting the 'most distinguishing' words of a given document.

2.3.3. Calculate the TF-IDF vectors of the following two documents, assuming this is the entire corpus?

Ans) The TF-IDF vector for the training set is [[0. 0. 0. ... 0. 0. 0.] [0. 0. 0. ... 0. 0. 0.] [0. 0. 0. ... 0. 0. 0.] ... [0. 0. 0. ... 0. 0. 0.] [0. 0. 0. ... 0. 0. 0.] [0. 0. 0. ... 0. 0. 0.]] which is of size (4081,22610). The TF-IDF vector for the test set is [[0. 0. 0. ... 0. 0. 0.] [0. 0. 0. ... 0. 0. 0.] [0. 0. 0. ... 0. 0. 0.] ... [0. 0. 0. ... 0. 0. 0.] [0. 0. 0. ... 0. 0. 0.] [0. 0. 0. ... 0. 0. 0.]] which is of size (2718,22610).

Part 3.1 - Sparsity (12 points)

3.1.1 Count he number of non-zeros in each row of the train-matrix.
Ans) [46, 33, 39, 39, 91, 416, 41, 55, 108, 51, 80….] which is a list of size 4081.

3.1.2 What is the average number non zero elements in each row?
Ans) 80.6522

3.1.3 On average what percentage of elements in each row have non-zero elements?
Ans) [0.2034498009730208, 0.145953118089341….] which is a list of size 4081.

Part 3.2 - SVD (4 points)

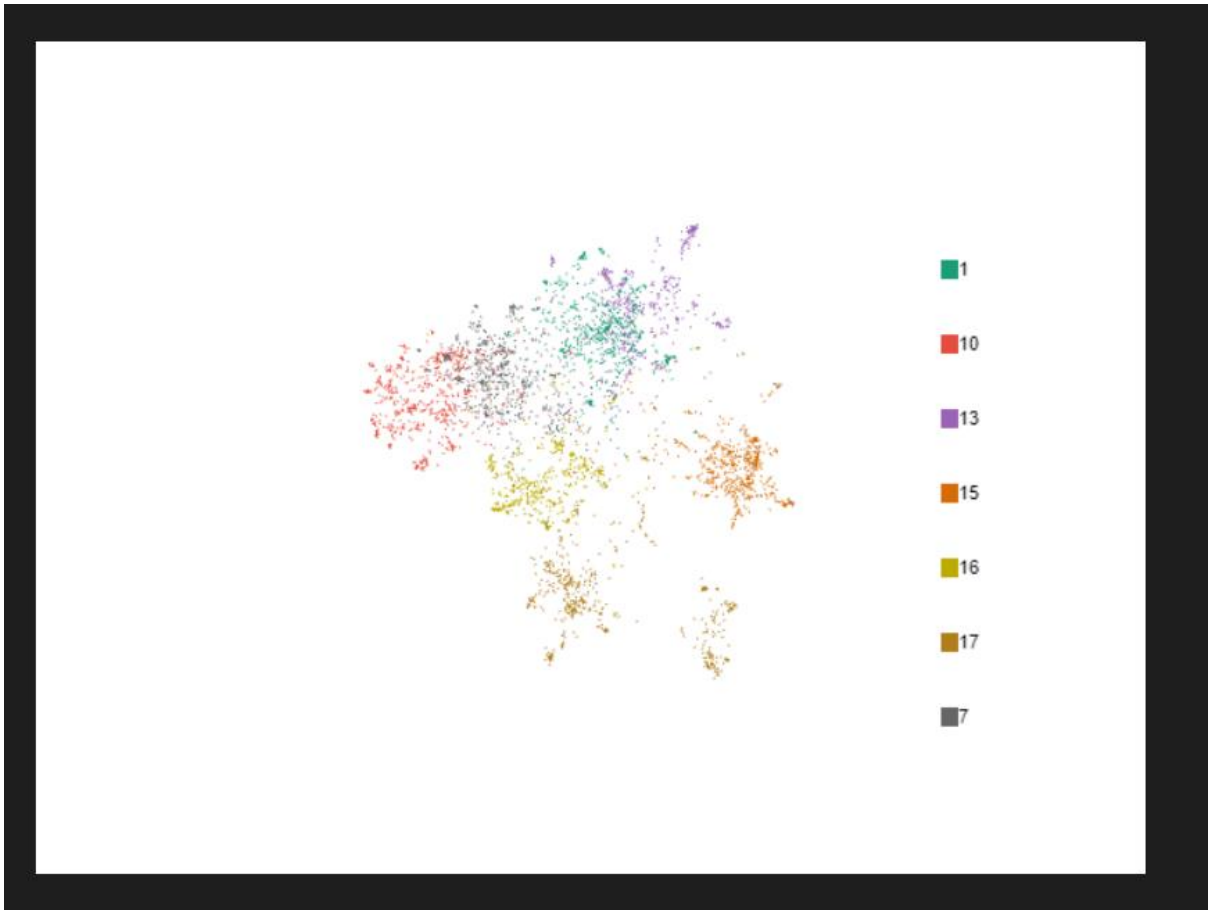
3.2.1. What portion of the variance in your dataset is explained by each of the SVD dimensions?
Ans) The portion of the variance explained by each of the three SVD dimensions is (0.00166477,0.00782159,0.00513792).

Part 3.4 - Visualization (8 points)

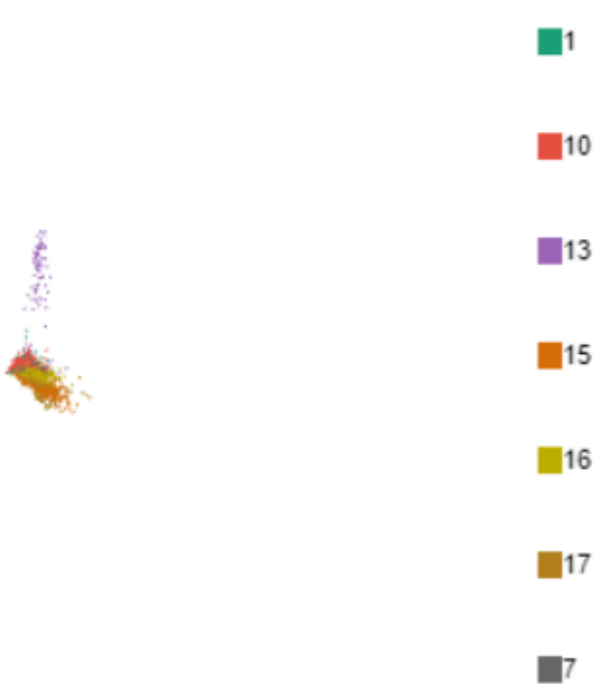
3.4.1. Based on your observation, what is the difference between SVD and UMAP embeddings?

Ans) The UMAP exhibits strong local clustering and groups the similar categories together

much more clearly than the SVD.



(Above figure represents plot for umap embeddings)



(Above figure represents plot for svd embeddings)

3.4.2. Which one do you prefer to use for a classification task? why?

Ans) The UMAP is preferred over the SVD for the classification task because the UMAP does the clustering and groups the similar categories much better than the SVD.

Part 4.1 - Clustering and evaluation (16 points)

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

Ans) The range of possible values of silhouette coefficients is [-1, 1].

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

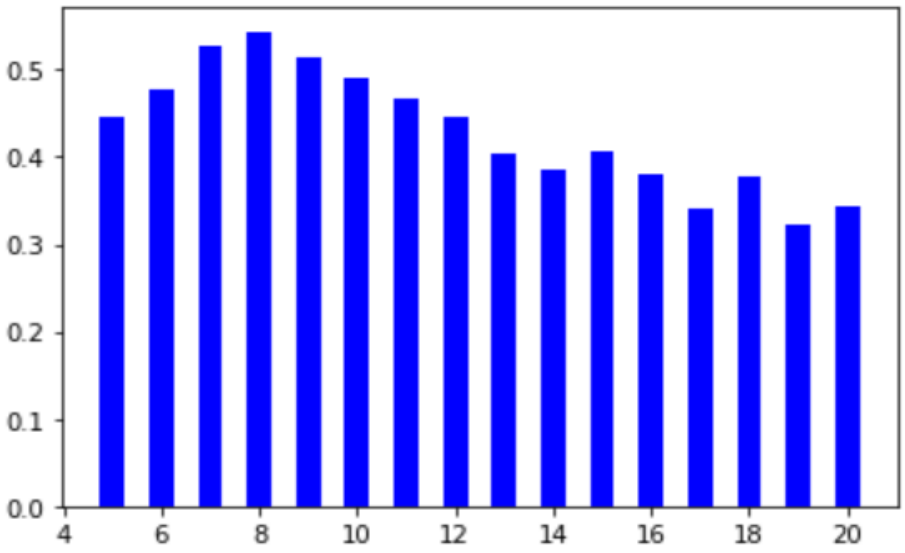
Ans) 1 means clusters are well apart from each other and clearly distinguished. -1 means clusters are assigned in the wrong way.

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.

Ans) The best silhouette score is at n = 8

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).

Ans)



Part 4.2 - Making a Kmeans classifier (4 points)

4.2.1 show your mapping (resulted dictionary).

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

4.3. Analyzing clusters

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

Ans) In our visualization we have two clusters 0 and 6 which have a target value of 17. After dimension reduction the values of the embeddings divides the output into 8 clusters. Since the data of 7 labels has 8 clusters, the centroid point of 2 clusters will be very close which in our case is 17.

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

Ans) After writing the above functions and getting the k most center points we see that centers of clusters 0 and 6 are very similar which are ([[0.67278194], [6.14275985], [4.92964412]]) and ([[2.9785536], [7.0123149], [5.83581698]]). As the k most center points have close values the clusters are very close and thus, we will have them as overlapping.

4.3.3. Can you infer the overlapping label(s) by checking out most central samples? check with original labels.

Ans) Yes. As mentioned in the above answers the k most central values of 0 and 6 are very similar and thus, we will have the overlapping labels there.

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

4.4.2. Calculate the accuracy of the model

Ans) 0.8612950699043415

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

Ans) The micro and macro values of precision are 0.8612950699043415,0.866399681506863.
The micro and macro values of the recall are 0.8612950699043415,0.861630324952893.
The micro and macro values of the F1 score are 0.8612950699043415 0.8598106141169206.

574 ONLY Part 5.1 - KNN classification (16 points)

5.1.2. Evaluate your model on test data (test_umap and test_svd). Which model performs better? Why?

Ans) The UMAP performs better because the clustering and grouping are much more clearly

seen in UMAP than in SVD. In SVD the clustering is not as clear as in UMAP and

the grouping and clustering in SVD are tightly packed.

5.1.3. Calculate macro and micro precision,recall and fscore for (test_umap). Which one of the two do you prefer for evaluating your model? why?

Ans) The macro and micro precision,recall and fscore for (test_umap) are

(0.862766740250184, 0.8665715198799745),(0.862766740250184,0.8630674437434205)

(0.862766740250184,0.8616588496000984) respectively. We would chose macro based metrics as we are not assigning greater importance to any one class and it is not global which will give us a better intuition if our particular label is failing or not. That's why we think macro based metrics would give yield us a more useful result.

5.1.4. Shortly describe why the two sets of values (macro and micro) are so similar in this case.

Ans) The two sets of values are very similar as in our model, the accuracy of each classes are very similar. Thus averaging them would give us a similar value as the one compared to the global value.

-Filtering target classes, Vectorizing documents, Making a Kmeans classifier,and Documentation of work done by Sai Krishna, reviewed by other teammates.

-Sparsity, SVD, Visualization done by Sukesh, reviewed by other teammates.

- Clustering and evaluation, Analyzing clusters, Evaluating Kmeans model on test dataset, KNN classification done by Sagar, reviewed by other teammates.