CSC 2611 Section 141-151 Lab 10

Document Clustering and Analysis using Custom K-means

For this lab, you will use a different subset of the 20 Newsgroup data set that you used in the lab 9. The subset for this dataset includes 2,500 documents (newsgroup posts), each belonging to one of 5 categories Windows (0), Crypt (1), Christian (2), Hockey (3), Forsale (4). The documents are represented by 9328 terms (stems). The dictionary (vocabulary) for the data set is given in the file "terms.txt" and the full term-by-document matrix is given in "matrix.txt" (comma separated values). The actual category labels for the documents are provided in the file "classes.txt". Your goal in this lab is to perform clustering on the documents and compare the clusters to the actual categories.

For this lab, DO NOT use the KMeans clustering function in scikit-learn. You may use Pandas and other modules from scikit-learn that you may need for preprocessing or evaluation. Your tasks in this lab are the following:

1. [20 points] Create your own distance function that, instead of using Euclidean distance, uses Cosine similarity. This is the distance function you will use to pass to the kMeans function in the included module (***Please use the module I provided for this lab***). Note: **you should NOT use external function for computing Cosine.** Write your own version that computes Cosine similarity between two *n*-dimensional vectors and returns the inverse as the distance between these vectors.

2. [15 points] Load the data set [Note: the data matrix provided has terms as rows and documents as columns. Since you will be clustering documents, you'll need to take the transpose of this matrix so that your main data matrix is a document x term matrix. In Numpy, you may use the ".T" operation to obtain the transpose.]. Then, use the *train_test_split* function (with *random_state* = 99) to perform a randomized split the data set (the document by term matrix) and set aside 20% for later use. **Use the 80% segment for clustering in the next part.** Next, as in the previous lab, perform TFxIDF transformation on these data sets.

3. [30 points] Perform K-means clustering on the transformed training data from part (2). Then, conduct a qualitative analysis of the clusters by examining the top features (terms) in each cluster to identify patterns in the data.

To facilitate your analysis of the clusters, write a function that displays the top N terms in each cluster, sorted by their average TF-IDF weight from the cluster centroid. For each top term in a cluster, your function should also display:

- **The mean TF-IDF weight** of the term in that cluster (this represents the term's importance).
- **The cluster DF count** for each term, which is the number of documents within that cluster that contain the term.
- **The cluster DF percentage**, calculated as the percentage of documents in the cluster that contain the term (i.e., if a cluster has 500 documents and the term "game" appears in 100 of those documents, then the cluster DF percentage for "game" in that cluster would be 20%).

Your output should also display the cluster size (the total number of documents in the cluster). Here is an example format to guide you:

```
Cluster  1 size = 396
-----------------------------------------------
           Freq   DF  % of Docs
window    10.90  281      70.96
file       7.35  153      38.64
do         4.89  129      32.58
driver     4.31   87      21.97
program    3.40  108      27.27
version    3.33   82      20.71
run        3.32  122      30.81
problem    2.63  108      27.27
microsoft  2.62   76      19.19
mous       2.59   34       8.59

Cluster  2 size = 392
-----------------------------------------------
        Freq   DF  % of Docs
game    8.32  203      51.79
team    6.86  184      46.94
plai    6.04  162      41.33
hockei  4.40  169      43.11
player  4.07  121      30.87
go      3.92  175      44.64
gm      3.47   16       4.08
goal    3.38   78      19.90
season  3.24   95      24.23
nhl     3.22  109      27.81

Cluster  3 size = 391
-----------------------------------------------
           Freq   DF  % of Docs
god       13.60  222      56.78
christian  8.52  180      46.04
sin        5.83   84      21.48
church     5.52   97      24.81
jesu       5.22  111      28.39
peopl      5.20  180      46.04
on         5.06  217      55.50
homosexu   4.63   36       9.21
believ     4.23  156      39.90
christ     4.19  113      28.90
```

Important Note for question 3: for this problem you should try several values of k for the number of clusters (try values of k from 4 through 8) and in each case try several runs to obtain clusters

that seem more meaningful. In some cases, you may find some small clusters containing noise documents, which is not unusual. The point is to experiment with different runs and cluster numbers until you find at least several clusters that seem to capture some of the key topics in the documents. **You do not need to provide the results of all your runs; you should only provide the results of your best clustering along with a brief discussion of your experimentation and your final observations.**

4. [20 points] **Evaluate Cluster Quality with Completeness and Homogeneity Scores**

   Using the cluster assignments from your K-means clustering on the training data and the true category labels provided in "classes.txt," assess how well your clusters match the actual categories by calculating the Completeness and Homogeneity scores.

   a. **Assign a Representative Label to Each Cluster:** For each cluster, determine its representative label based on a majority vote of the true labels of the training documents within that cluster. The representative label of a cluster is the most frequent category label among the documents in that cluster.
   b. **Calculate Completeness and Homogeneity:**
      i. Completeness score will indicate whether all documents in a given category are assigned to the same cluster (high completeness means that each cluster primarily contains documents from a single category).
      ii. Homogeneity score will indicate whether each cluster contains only documents that belong to a single category (high homogeneity means that clusters are not mixed with documents from different categories).
   c. **Use Best Clustering Run: Perform this analysis using the best clustering result you identified in the previous part (the optimal value of k and the best clustering run).**

5. [15 points] **Classify Test Documents Using Cluster Centroids.**

Using the final cluster centroids from your best K-means clustering (from the training data), classify each document in the 20% test set by assigning it to the closest cluster based on Cosine similarity.

a. Calculate Cosine Similarity: For each document in the test set, calculate the Cosine similarity between the document vector and each cluster centroid.
b. Assign Cluster Labels to Test Documents: For each test document, assign it to the cluster with the highest Cosine similarity score (i.e., the closest match). Use the representative label you determined in Question 4 as a pseudo-label for each cluster to interpret the results.
c. Display Results: For each test document, output:

1) The assigned cluster label based on the closest centroid.
2) The Cosine similarity score to the corresponding centroid, indicating how closely the test document matches the assigned cluster.

This process allows you to categorize new documents based on the clusters learned from the training data and interpret each test document's assigned cluster using the pseudo-labels from the training clusters.

Notes on Submission: **Please submit the notebook in both IPYNB and HTML formats (along with any auxiliary files).** Please organize your notebook and label sections so that it's clear **what parts of the notebook correspond to which problems in the lab** (submissions that are not well-organized, not well-documented, or are difficult to read will be penalized). Do not compress or Zip your submission files; each file should be submitted independently. Your assignment should be submitted via Canvas.