DMML - Assignment 2

Alena Maria Thomas - MDS202303

April 4, 2024

Overview

The ”Bag of Words” data set from the UCI Machine Learning Repository contains five text collections in the form of bags-of-words. The link for the UCI repository is here.

In each of the text collections, each document is summarized as a bag (multiset) of words. The individual documents are identified by document IDs and the words are identified by word IDs. After some cleaning up, in each collection the vocabulary of unique words has been truncated to only keep words that occurred more than ten times overall in that collection.

Task

Your task is to cluster the documents in these datasets via K-means clustering for different values of K and determine an optimum value of K. As a similarity measure, use Jaccard index, that measures similarity between two documents based on the overlap of words present in both documents. Note that this changes the underlying model from ”bag of words” to ”set of words”.

The datasets are of different sizes. Report your results on the three smaller datasets (Enron emails, NIPS blog entries, KOS blog entries).

Methodology

We use **Jaccard Index** as a similarity measure between two documents based on the overlap of words present in them.

$$J(D_1, D_2) = \frac{|D_1 \cap D_2|}{|D_1 \cup D_2|}$$

For K-Means Clustering, **Jaccard distance** is used as a measure of distance between a point and a centroid.

$$D(D_1, D_2) = 1 - J(D_1, D_2)$$

The optimum value of k is identified using the **elbow method** where we minimise the **inertia** (the sum of squared distances of samples to their closest cluster centroid).

Implementation

1. Uploaded the .txt files (skipping the first 3 lines) to create a dataframe with column names docID, wordID, and Count.

   - Since the **Enron** dataset was very large, picked out the documents with more than 200 distinct words, and worked on it.

2. Defined the following functions:

   (a) `iter_Centroids(prev_centroids, clusters, n_clusters, data)`: Recomputes the centroids based on the Jaccard distance between words within each cluster.

   (b) `iter_Clusters(centroids, n_clusters, data)`: Assigns data points to clusters based on Jaccard distance from a given set of centroids

   (c) `inertia_calc(centroids, clusters, n_clusters, data)`: Calculates the inertia, which is a measure of how compact the clusters are.

(d) `Jacc_KMeans(data, n_clusters=3, max_iter=100, tol=1e-4, random_state=69)`: Performs KMeans clustering using Jaccard distance and returns a list of centroids and a list of clusters

(e) `elbow_plot(data, max_clusters, data_name="")`: Plots an elbow plot for the given dataset

3. Created the sparse matrix where each column represents the presence or absence of a particular word in each of the documents, ignoring the multiplicity of the word (assigning 1 if the word is present in the document, 0 otherwise)

4. Used the sparse matrix to the Jaccard distance matrix between any two documents.

5. Applied K-Means Clustering from scratch using Jaccard Distance.

   (a) Initialised k centroids (one for each cluster) randomly.

   (b) Performed the following steps iteratively for some maximum number of assigned iterations.

      i. Assign the documents to the cluster whose centroid it is closest (based on the Jaccard distance).

      ii. Reassigned the centroids with the cluster element that has min dist with all others in the same cluster.

6. Plotted the graph of inertia vs number of cluster points to find a sharp elbow and hence get the optimum value of k.

Results

Measure	**KOS**	**NIPS**	**Enron**
Optimum no. of clusters (k)	2	2	2
Time to compute Jaccard matrix (s)	482	139	4509
Time to plot elbow graph	28.2	9.45	27.2
Space used (MiB)	416.24	179.77	1055.8