Document Classification using K-Means Clustering

Data:

https://archive.ics.uci.edu/ml/datasets/Bag+of+Words

Algorithm Used :

1) Convert each bag of word to a matrix A, where every row represents a Word ID, colomn represents a Doc ID and so each element represents frequency of a Word in the Doc. For eg: A(m,n) shows the frequency of the word ID m in doc ID n.

2) Now, convert the matrix A to B such that B(m,n) = 0 (if A(m,n)=0) or 1(if A(m,n)>0). B(m,n) shows the whether the word ID m is present in doc ID n or not. This now changes the bag of words to set of words.

3) Each column of the matrix B is now a vector of dimension max (WordID), say m, in the bag of word. Plotting two such Docs in a m-dimensional space, we now measure the distance between them using Jaccard Similarity. (It is important to note here that Jaccard distance between two Docs will be 1-Jaccard Similarity, because more similar a pair of Docs are lesser should be the distance between them).

4) So, we now apply K-Means clustering on this B matrix. But unlike the Ordinary K-Means, here the distance between two points(two documents) will be Jaccard Distance.

5) K Means Algorithm : Randomise the k centroids as any k random documents. Use cdist function to measure Jaccard distance between a point with every centroid and repeat it for every point. Classify every point to that centroid cluster with which its Jaccard distance was least. Now(most important step), to update the centroids in a cluster, instead of using the mean which does not make sense, find Jaccard distance of every point with respect to every other point in the cluster and sum it up. The point for which the sum is least is the new centroid. Now, repeat this process till the specified max-iterations.

6) Now, as a measure of model evaluation, we define a new type of inertia. In the final clusters, find Jaccard distance of each point in a cluster with its cluster's centroid. Sum this for every cluster. The lesser this sum, the better the model is.

7) To find the best model, find inertia of the model for different cluster values. The value of number of clusters where the decrease in inertia becomes less (elbow is reached), can be taken as optimum.

Enron Data Set :

Since this data set is huge, the bag of words needs to be filtered. By grouping the bag of words with Word IDs and taking sum of frequencies in descending order, we tried to plot frequency of Words to find thresholds. The idea behind filtering based on frequency of Word IDs is that, words occurring more than a threshold can be considered as stop words and hence not useful in creating meaningful clustering of documents and similarly words occurring less than a threshold can be taken as very unique words again not helping in clustering. By error and trial, we conclude to remove words occurring less than 15 times (8000 words appx) and words occurring more than 50 times (12000 appx). But to execute this, an iteration was required on approx. 37 lakh rows for almost 8000 times, which could not be implemented. Although, after removing all the words based on the criteria mentioned the remaining process for clustering would be exactly similar as was for Kos and Nips.

Results :

1) The number of clusters for both Kos and Nips data set came out as 3 (graph showed in word file named "Optimal_Cluster_Check").

2) The enron dataset was left due to the large size of the dataset which could not be processed with our algorithm and systems.

3) PCA was not done to visualise because PCA again uses Euclidian distances and not Jaccard distances to project points in a higher dimension space to a lower dimension. Due to this no visualisation was done.

4) For each data set, two excel files were downloaded, one showing the final centroid values and the other showing the final cluster value for each Doc ID.

Data set	NIPS	KOS	Enron
Inertia	~1255	~616	-
Time Taken	50 mins	45 mins	-
Memory	-	-	-

Scope For Improvement :

1) The number of clusters could be more if the loop for inertia vs number of clusters world have run for more number of clusters, because a clear elbow is not achieved at 3 cluster value. This was avoided due to lack of resources (time and RAM).

2) The memory was not recorded since we forgot to note memory in the first go, and could not afford to run it again.

3) Visualisation of the clusters could be achieved by reducing the space to a lower dimension using some relevant Projection algorithm (other than PCA).

4) A better algorithm could be written where Jaccard Distance is not calculated but accessed everywhere in the K-Means algo. This could be done by calculating the complete Jaccard distance matrix first and then just accessing its elements.

5) The number of iterations in k means can be taken more than 30, if resources allow.