DATA MINING AND MACHINE LEARNING

Clustering Documents

using K-means and Jaccard index

Submitted by Hiba A P [MDS202326] and Nooh Ali [MDS202337]

Introduction

Understanding the relationships between documents is crucial. One common approach to achieve this is through clustering, where similar documents are grouped together based on their content. In this context, we'll explore the "Bag of Words" data set from the UCI Machine Learning Repository. Our task is to cluster a collection of text documents using the K-means algorithm. However, there's a twist: instead of the traditional "bag of words" representation, we'll treat each document as a set of words. This shift allows us to measure similarity using the Jaccard index, which quantifies the overlap of words between two documents.

Our Approach

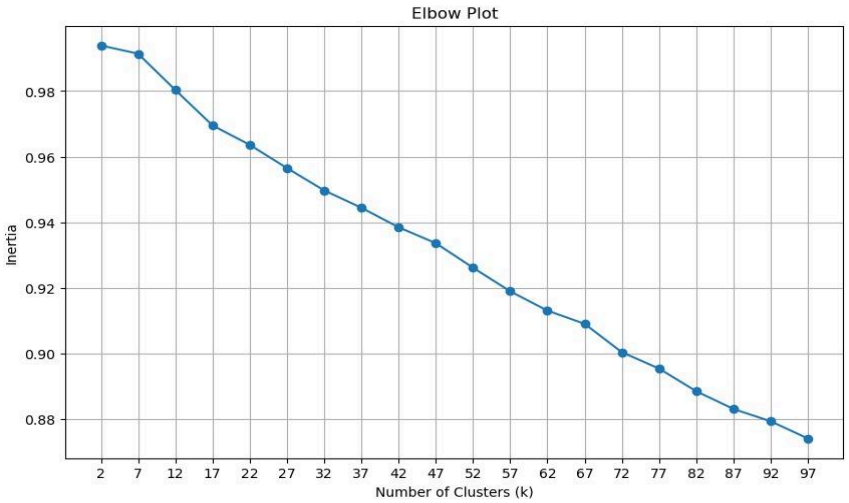
We began by analyzing each document, extracting the set of words used in it. For the purpose of clustering, we randomly selected K centroids. Next, we calculated the distance from each document to these centroids and assigned each document to the nearest centroid. Within each cluster, we computed a new centroid by taking the union of the word sets from all documents in that cluster. This new centroid then replaced the previous one, and we repeated this process iteratively (a total of 100 times). The

distance metric we used was one minus Jaccard similarity, where Jaccard similarity represents the intersection of word sets between two documents divided by their union.

1. **Selecting Initial Centroids**:
    - We randomly selected **K centroids** (where K is a user-defined parameter) as starting points for our clusters.
    - These centroids represent the center of each cluster.
2. **Assigning Documents to Nearest Centroid**:
    - For each document, we calculated its distance from each centroid. The distance metric we used was **one minus Jaccard similarity**.
    - Jaccard similarity measures the overlap between two sets (in our case, the sets of words in two documents).
    - We assigned each document to the nearest centroid based on this distance.
3. **Updating Centroids**:
    - After assigning documents to centroids, we recalculated the centroids.
    - The new centroid for each cluster was formed by taking the union of the sets of words from all documents in that cluster.
    - This updated centroid became the center for the next iteration.
4. **Iterative Process**:
    - We repeated the assignment and centroid update steps for a fixed number of iterations (in our case, 100 times).
    - With each iteration, the centroids moved closer to the true centers of their respective clusters.
5. **Exploring Different K Values**:
    - We experimented with different values of K to find the optimal number of clusters.
    - Each value of K resulted in a different partitioning of the documents.

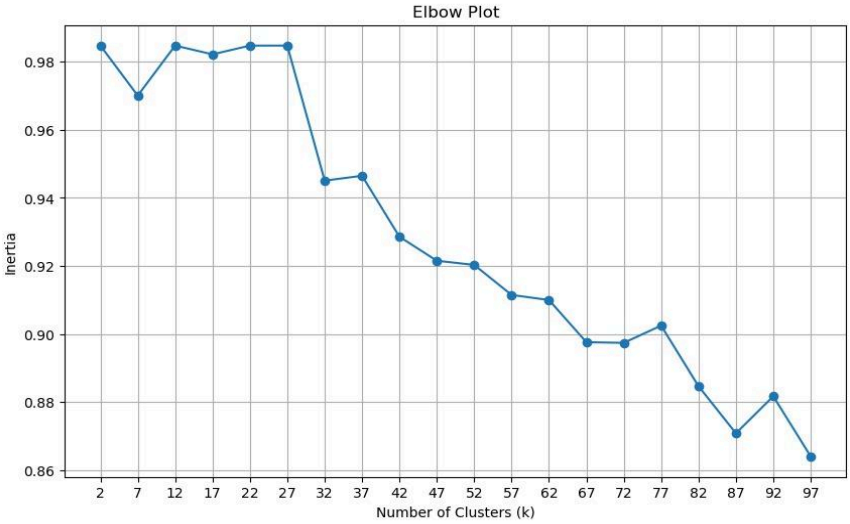
In summary, our process involved initializing centroids, assigning documents to the nearest centroid, updating centroids based on document sets, and iterating until convergence. The result was a set of clusters, each containing similar documents. The choice of K influenced the granularity of these clusters.

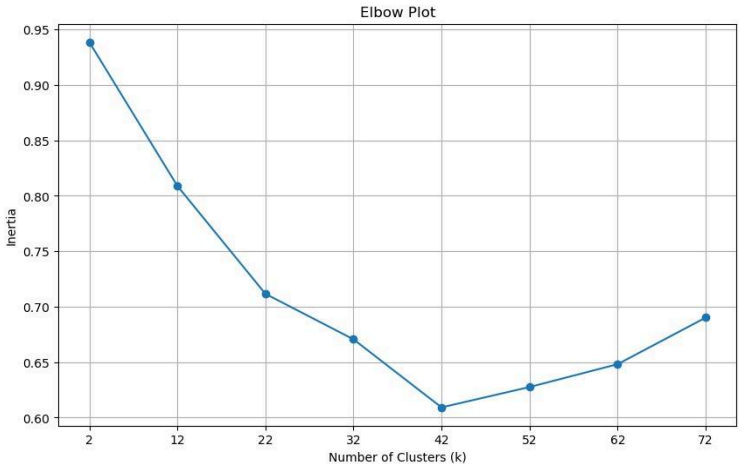
Conclusion



The left image displays the elbow plot for the Enron document collection. Based on the graph, we might opt for K = 17, which means dividing the entire set of documents into 17 clusters makes more sense.

The image on the right shows the elbow plot for the Kos document collection. After analyzing the graph, we feel that selecting K = 52 would be a sensible choice. This value implies dividing the entire set of documents into 52 clusters.





The elbow plot for the Kos document collection suggests that K = 42 is a reasonable choice. This value corresponds to dividing the entire set of documents into 42.