Wikipedia Clustering Competition (CS 4786)

Akshay Agarwal (aa2657), Abhishek Sarkar (as2765)

Problem Statement

For the purpose of the competition, 11039 Wikipedia articles belonging to *four* different categories namely *Actor*, *Movie*, *Math* and *Mathematician* are provided. The data is provided in the form of raw text files, bag of words and linkage matrix that connects various articles through hyperlinks. Using this information, the task is to use *unsupervised learning algorithms* and *dimensionality reduction techniques* to classify/ cluster the articles to the best possible accuracy.

Approach Overview

Three distinct approaches were used to cluster articles. Based on the data provided, two different data matrices were used, the **bag of words** and the **linkage matrix**. The following are the methods used:

- **Method I:** PCA performed bag of words to reduce dimension and then K-means performed to cluster data. (Accuracy 76%)
- **Method II:** Spectral Embedding performed on linkage matrix and then K-means performed to cluster data. (Best Accuracy 95.35%)
- **Method III:** PCA performed on bag of words to reduce dimension, spectral embedding performed on linkage matrix, followed by CCA performed to keep the redundant common information. (Accuracy 71%)

Method I

First, we only used text data to model our algorithm. We used the raw_data file and created our own input vector for it. We did the following methods:

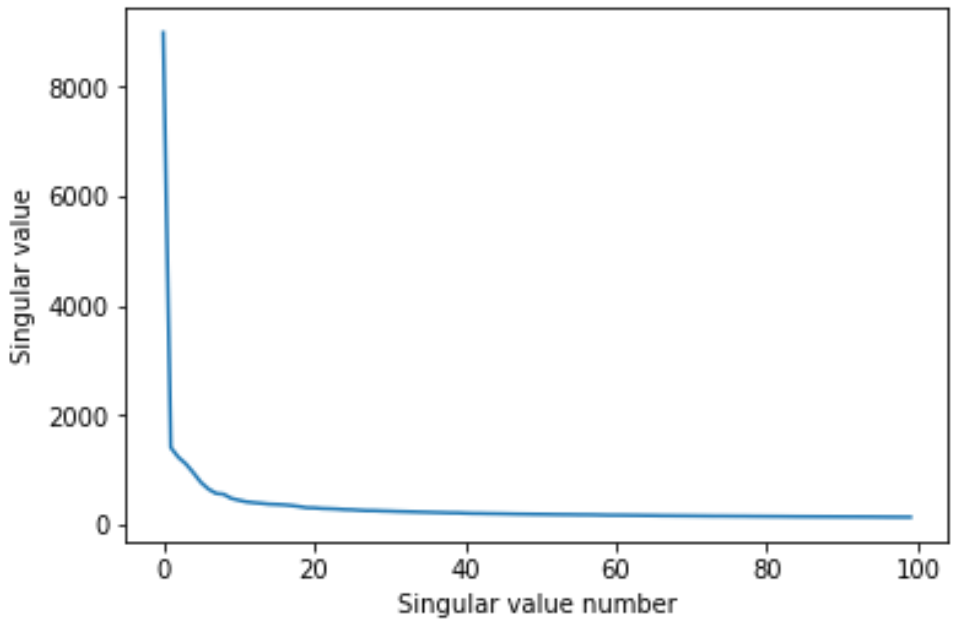
- tokenizing and stemming each article
- transforming the corpus into vector space using tf-idf
- clustering the documents using the k-means algorithm

We used NLTK package to tokenize and stem the articles. We then removed the stopwords from the text and reduced the dimensionality by tf-idf. We have set min_df as 0.002 and max_df as 0.95 which gave us around 4000 features.

Wikipedia Clustering Competition (CS 4786)

Akshay Agarwal (aa2657), Abhishek Sarkar (as2765)

In order to reduce the number of dimensions, we did SVD on the bag of words and plotted the singular value curve



We can see that 20 components explain almost all of the variance in the dataset and hence we performed SVD on *bag_of_words* using *20 components*.

We then performed *K-means* clustering on the dataset and achieved **76%** accuracy.

Method II

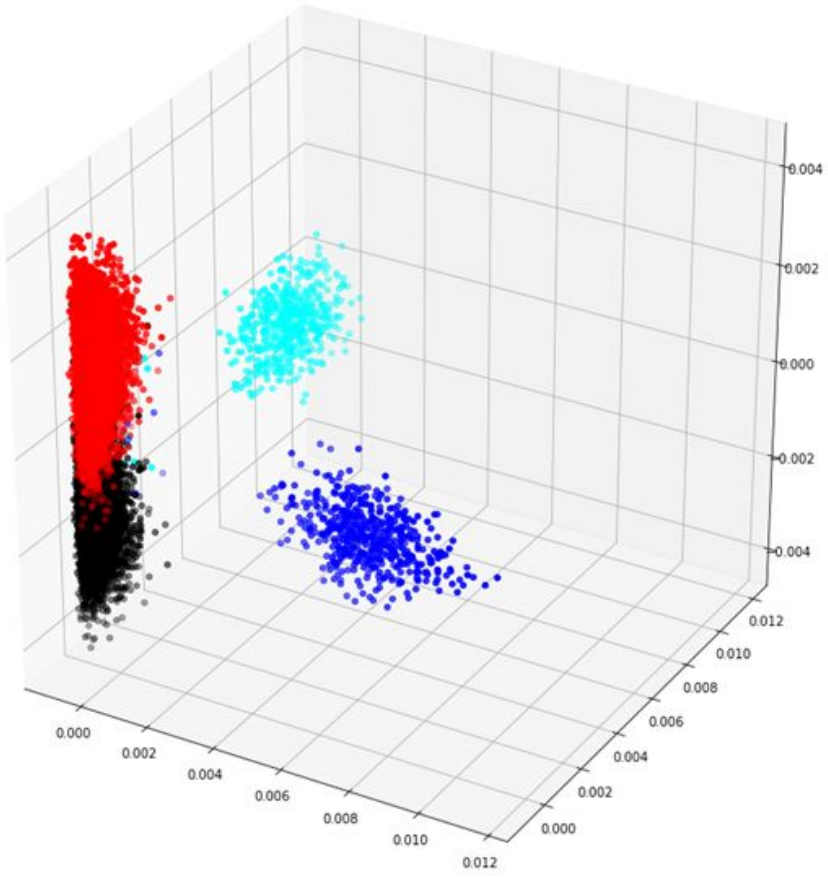
The first step was to convert the graph (link matrix) into the adjacency matrix. We achieved this using basic numpy operations and performed the following clustering algorithms on the link_dataset:

1. **Spectral Embedding followed by K-means:** We have hyper tuned parameters to get the best result when the *number of components* is 10. *Laplacian Norm* was used while computing the output. *Laplacian norm* means that we implemented Normalized Spectral Embedding and tried to push the articles with more links further apart. Although we used the first 10

Wikipedia Clustering Competition (CS 4786)

Akshay Agarwal (aa2657), Abhishek Sarkar (as2765)

eigen-vectors to perform K means, for the purpose of visualization, we plotted using the *first three* eigen-vectors (starting from the second smallest).

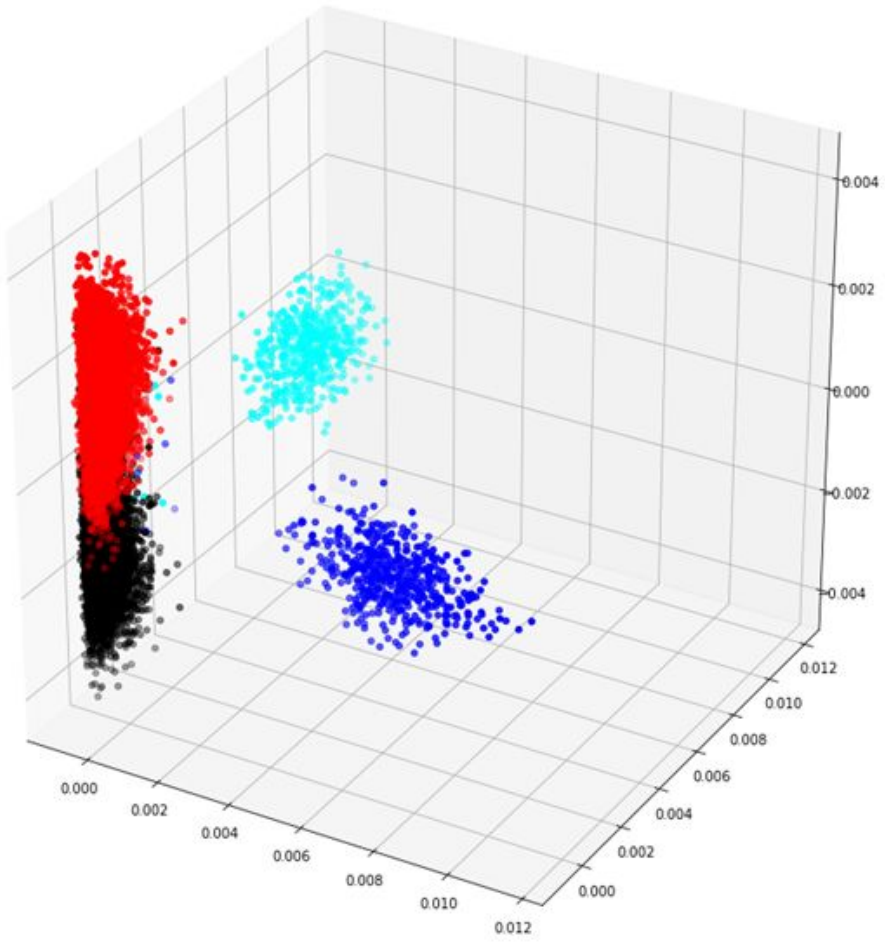


This followed by K-means gave us an **accuracy of 98.2%**.

2. **Using Affinity Propagation on Graph:** Affinity propagation (AP) is a clustering algorithm based on the concept of "message passing" between data points. Unlike clustering algorithms such as *k*-means or *k*-medoids, affinity propagation does not require the number of clusters to be determined or estimated before running the algorithm. Similar to *k*-medoids, affinity propagation finds "exemplars", members of the input set that are representative of clusters.

Wikipedia Clustering Competition (CS 4786)

Akshay Agarwal (aa2657), Abhishek Sarkar (as2765)



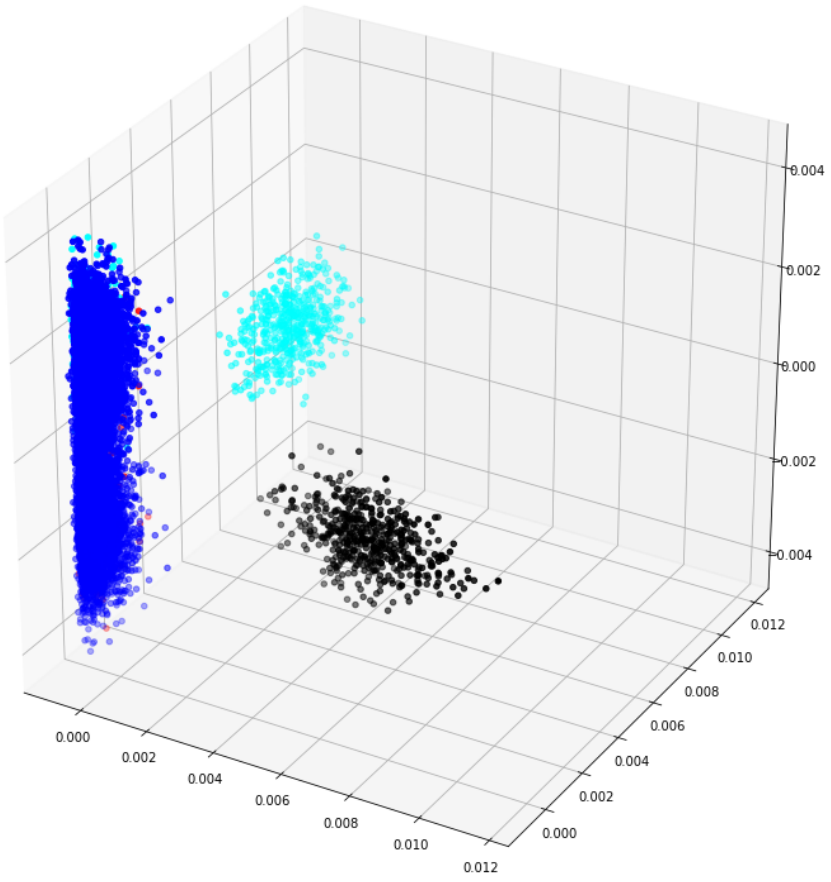
Accuracy was 90.75%.

3. Gaussian Mixture Models on Graph: We did GMM on graph using init_params as 'kmeans'. The reason GMM didn't work well is because the data was not Gaussian distributed, which is one of the prime assumptions of GMM's. As it can be seen, the clustering was not remotely as perfect as the other two methods.

Accuracy is less than the others and we got the following distribution of labels:

Wikipedia Clustering Competition (CS 4786)

Akshay Agarwal (aa2657), Abhishek Sarkar (as2765)



4. Spectral Embedding followed by K-means on weighted Graphs: This is a modification to the Normalised Spectral Embedding method. Here, we are taking into account the fact that two articles (say article i and j) can have more than one links. This results in the fact that the non-zero entries in the A matrix in many cases more than 1. We first computed the weighted adjacency matrix and then performed similar methods as in step 1.

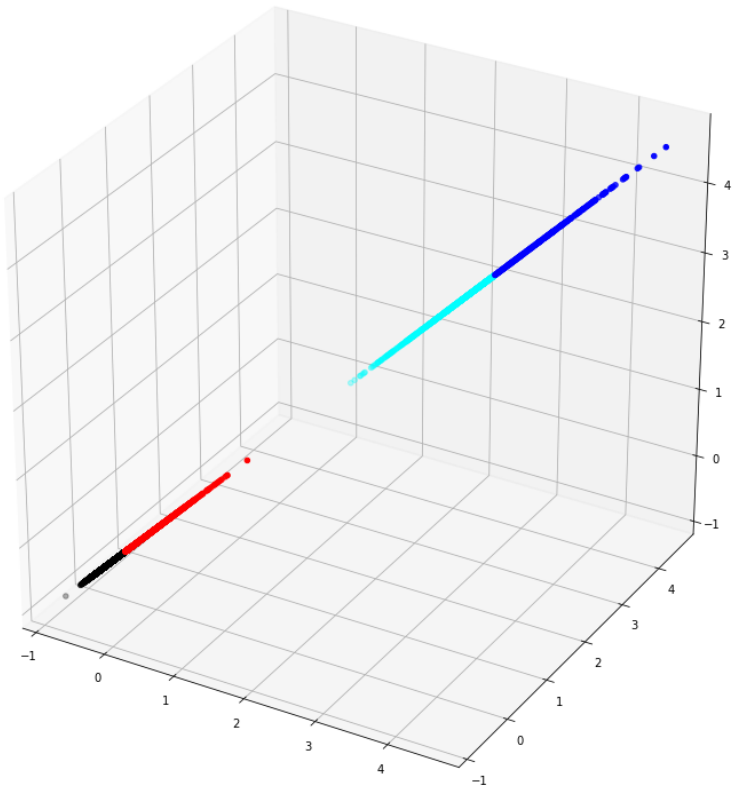
We got the **best accuracy here of 95.35%.**

Wikipedia Clustering Competition (CS 4786)

Akshay Agarwal (aa2657), Abhishek Sarkar (as2765)

Method III

We used CCA to get the most correlated 10 components between Adjacency matrix and bag_of_words matrix.
We performed CCA using inbuilt python function and **achieved an accuracy of 71%**.



Conclusion:

After performing various clustering techniques (K means and GMM) and different feature extraction methods, it is seen that *Normalized spectral embedding* followed by *K-means clustering* works best in classifying the wikipedia articles into one of the following categories. We achieved the best **accuracy of 98.2%**.
