CPSC 8420 Final Project, Fall 2024

Jacob Jeffries
Department of Materials Science and Engineering, Clemson University
Theoretical Division, Los Alamos National Laboratory
jwjeffr@clemson.edu

Abstract

The 21st century has been a very politically hot time period, with various military alliances forming between different nations and groups. In this work, I use the military alliances formed in this century to cluster the world's countries into 3 distinct groups, corresponding to nations aligned with The United States 🇺🇸, Russia 🇷🇺, and China 🇨🇳.

1 Data Collection

To gather military alliances made in the 21st century, I used a Wikipedia article on military alliances [1], and grabbed all currently valid military alliances from the year 2000-present. Note that some of these alliances are effectively defunct. An important example is the Islamic Republic of Afghanistan being replaced by the Islamic Emirate of Afghanistan (i.e. the Taliban's government), effectively ending any military alliance any nation had made with Afghanistan 🇦🇫.

Fortunately, the defunct alliances are color coded in white in the table on the article, so these were not included. Furthermore, any alliances made with the European Union 🇪🇺 as a diplomatic body were not included. Lastly, alliances made with militant groups that are not officially standing armies of their respective countries were not included, namely Hezbollah. The Syrian government also fell from the time I collected this data to now, so these alliances were included.

Some examples of relatively short lines of the resulting text file are below. This text file was created manually.

```
Australia, Japan, India, United States
Denmark, Finland, Iceland, Norway, Sweden
Azerbaijan, Turkey
France, United Kingdom
Austria, Croatia, Czech Republic, Hungary, Slovakia, Slovenia
Moldova, Romania
Israel, United States
Philippines, United States
Iran, Iraq, Russia, Syria
Tunisia, United States
```

The complete file can be viewed on GitHub [2]. We can visualize these relations in a graph, where each edge represents a bilateral relation:

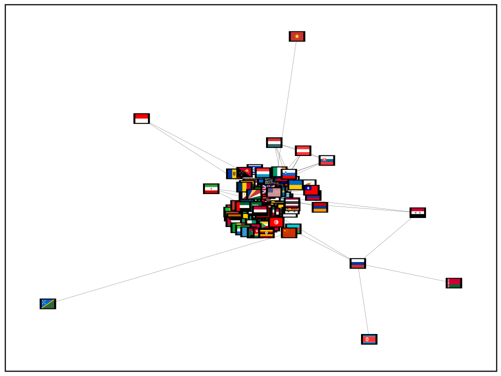

Figure 1: Militaristic connections visualized in a graph rendered with networkx [3], using the Kamada-Kawai [4] algorithm to layout nodes. Nodes represent countries with flags rendered by flagpy, and edges represent military alliances between countries.

Some communities here are clear, but not all. For example, the Russia 🇷🇺, Syria 🇸🇾, Belarus 🇧🇾, and North Korea 🇰🇵 community is clear, but the large cluster of countries in the middle (representing countries active in forming alliances in the 21st century) is not very clear. We can systematically group countries using spectral clustering, which will make the communities more clear.

2 Data Analysis

To group the countries, I created an adjacency matrix A, where $A_{ij} = 1$ if the countries i and j formed an alliance within the 21st century. Note that multiple alliances are counted, so $A_{ij} = 2$ if countries i and j formed two alliances, for example. Then, each row of A was normalized so that they sum up to unity. This is to ensure that large multilateral agreements (such as the Combined Maritime Forces) do not overtake the algorithm.

I then perform spectral clustering, by:

- Initializing the diagonal degree matrix $D_{ii} = \sum_j A_{ij}$
- Calculating the normalized random-walk Laplacian matrix $L_{\text{rw}} = I - D^{-1}A$
- Creating a matrix M from the first 3 eigenvectors of L_{rw}, corresponding to the 3 smallest eigenvalues
- Performing k-means clustering on the rows of M for 3 clusters

It is worth noting here that $D = I$ due to the way I normalized the adjacency matrix, so $D^{-1}A = A$, which speeds calculations up a bit. The implementation of this clustering is as follows:

```python
from dataclasses import dataclass

import numpy as np
from numpy.typing import NDArray


@dataclass
class KMeans:

    k: int
    seed: int = 123456789
    tolerance: float = 1.0e-14

    def fit(self, data: NDArray) -> NDArray:

        generator = np.random.default_rng(seed=self.seed)

        # choose k random points from X as centroids
        # assume shape of X is (N, d) where N is number of samples and d is dimension of each sample

        centroids = generator.choice(data, size=self.k, replace=False)
        error = np.inf

        categories = None

        while error >= self.tolerance:

            old_centroids = centroids.copy()
            distances_from_centroids = np.linalg.norm(
                centroids[:, np.newaxis, :] - data[np.newaxis, :, :], axis=2
            )
            categories = np.argmin(distances_from_centroids, axis=0)

            for i in range(self.k):
                centroids[i] = np.mean(data[categories == i])

            error = np.linalg.norm(old_centroids - centroids)

        if categories is not None:
            return categories
        raise ValueError


def spectral_clustering(adjacency_matrix: NDArray, num_categories: int) -> tuple[NDArray, NDArray]:

    rw_laplacian = np.eye(adjacency_matrix.shape[0]) - adjacency_matrix
    eigenvalues, eigenvectors = np.linalg.eig(rw_laplacian)

    eigenvectors = eigenvectors[:, np.argsort(eigenvalues)]
    M = eigenvectors[:, :num_categories]
    U, S, Vh = np.linalg.svd(M.T @ M)

    return M @ U, KMeans(k=num_categories).fit(M)
```

Figure 2: Python implementation of spectral clustering.

From this implementation, we can then perform spectral clustering on our adjacency matrix with a simple call `pca_vectors, categories = spectral_clustering(adjacency_matrix)`. This also lets us perform a PCA analysis on the embedded points in M.

Results

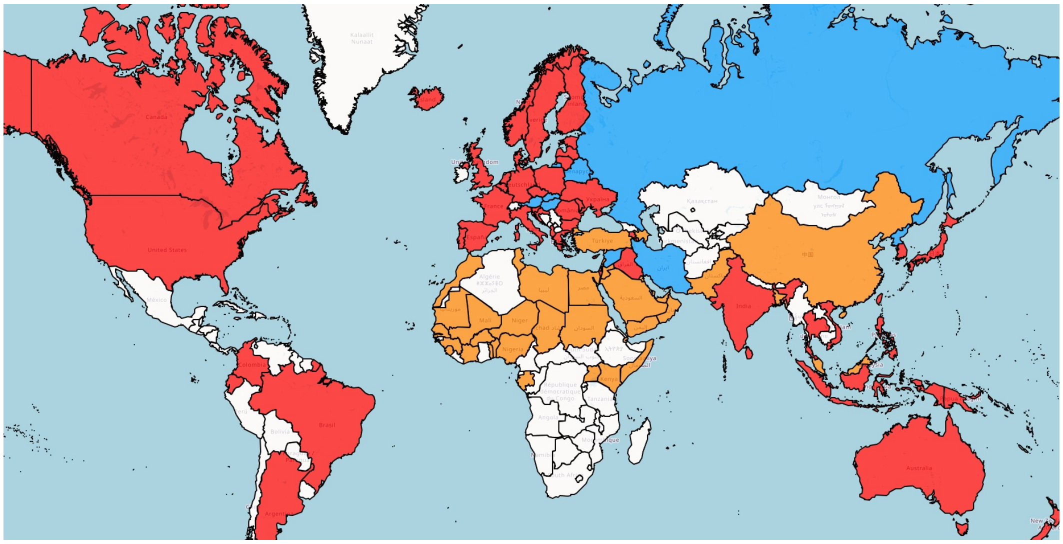

Figure 3: k-means clustering on countries. Map generated with the folium library [5]. White indicates a lack of data for that country, meaning no military alliances were formed with that country during the 21st century.

We see a US-aligned block in red, including most of NATO (e.g. The United Kingdom 🇬🇧, Germany 🇩🇪, etc.) and major non-NATO allies, and a Russia-aligned block in blue, including key Russian allies such as Syria 🇸🇾, and North Korea 🇰🇵, and Belarus 🇧🇾. Some US-aligned nations, namely Austria 🇦🇹, Hungary 🇭🇺, and Slovakia 🇸🇰 are clustered in the Russia-aligned block. However, this is not completely wrong; Hungary 🇭🇺 has grown closer to Russia 🇷🇺 as a result of Victor Orban's rule as prime minister, and both Austria 🇦🇹 and Slovakia 🇸🇰 have close historical and militaristic ties to Hungary 🇭🇺.

The ambiguous category here is the China-aligned block in orange, since China 🇨🇳 is not particularly politically aligned with the Middle East. However, this can be explained by China's investment in African countries, which naturally have more ties with the Middle East. The complete map is available as an interactive HTML file on GitHub [6]. This clustering is also very easy to see in a PCA analysis on the embedded points in M, as seen below:

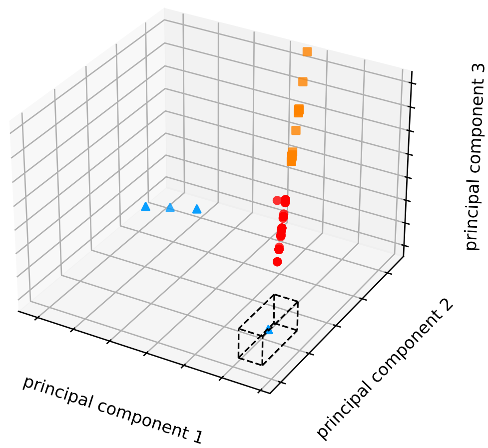

Figure 4: PCA on the embedded points in M. Cluster groups are separated by color and marker shape, with the same colors as on the world map. Austria 🇦🇹, Hungary 🇭🇺, and Slovakia 🇸🇰 are contained in the black-outlined rectangle.

We similarly see the aligned blocks matching the colors on the world map, with the clustering a bit easier to see geometrically. An ambiguous grouping is the 3 blue points that overlap each other (marked by a rectangle with black outlines), which are Austria 🇦🇹, Hungary 🇭🇺, and Slovakia 🇸🇰. These points could also reasonably be red, or US-aligned.

3 Conclusion

In this project, I have performed spectral clustering and PCA on a 21st century military alliance dataset. Both results are consistent with the qualitative evolution of military and diplomatic alliances, showcasing the closeness of nations with The United States 🇺🇸, Russia 🇷🇺, and China 🇨🇳. The full implementation and datasets are available on my GitHub [7].

References

[1] Wikipedia contributors. List of military alliances — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=List_of_military_alliances&oldid=1261635825`, 2024. [Online; accessed 7-December-2024].

[2] alliances.txt. `https://github.com/jwjeffr/spectral_alliances/blob/main/alliances.txt`.

[3] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

[4] Tomihisa Kamada and Satoru Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7–15, 1989.

[5] python visualization. Folium. `https://python-visualization.github.io/folium/`.

[6] map.html. `https://github.com/jwjeffr/spectral_alliances/blob/main/map.html`.

[7] spectral_alliances. `https://github.com/jwjeffr/spectral_alliances`.