PART III:
POSITIONAL ANALYSIS

Assuming a person's social life, influence, behaviour can be described by their role or position in society, then that person could be fairly characterised or grouped (atleast to some extent) by their regular interactions and patterns of relations.

Unlike community detection which also uses clustering techniques, positional analysis does not require that node *i* be adjacent (linked) to node *j* for node *i* and node *j* to occupy the same position. Rather, it tries to define logical types of roles and examine how these relate to each other. For example: the role of a parent vs a child, a teacher vs a student, a husband vs a wife, etc. In other words, being a child doesn't require that each node in the position of child be connected to other nodes in the same position, or for the employees of a certain employer to be directly connected to other employees.

For positional analysis, the 'who' of a specific node is less important because whoever holds the role, holds the responsibility. What makes this type of analysis interesting is that the elements or expectations of a role are somewhat fixed or constant, such as the:

1. Rights and obligations with respect to other people or classes of people.
2. Requirement of a 'role compliment', such as another person who the role-occupant acts with respect to.

Methods

There are three notable strategies for identifying roles and positions in networks, namely, **structural equivalence**, **blockmodelling** and **feature analysis**. For the purposes of this analysis, I will only focus on the latter.

> ***What are Feature Based Strategies***
> Feature based strategies are a form of unsupervised learning technique used in computer science to capture features of nodes and classify them into types of roles and positions depending on a particular feature set. For example, we could measure nodes' centralities, the properties of their local neighborhoods (i.e. triangles, dyadic ties), their average distance from other nodes, etc., and then cluster them according to those features. In a sense, this assumes the least about the data and also is most feasible (computationally) for large graphs. It is performed using the k-means algorithm.
>
> ***What Is K-Means Algorithm***
> K-means algorithm is an iterative algorithm that divides a group of *n* datasets into *k* subgroups (clusters) based on the similarity of their features and their mean distance from the centroid of that particular subgroup (cluster) formed.

Steps in this module are as follows:

1. **Feature Selection** -> to choose notable features to include in the model.
2. **Cluster Detection** -> to determine the optimal number of clusters to detect.
3. **Cluster Classification** -> to describe clusters based on their unique features.
4. **Cluster Evaluation** -> to assess whether the nodes assigned within each cluster fit.

1 SELECTION

Feature selection is the first stage of implementing most clustering techniques. When working with large data, it is important to know that not every column (feature) in the dataset is going to have an impact on clustering. A feature set must be chosen such that nodes in the network are well described and correlated with the features. Adding unnecessary or irrelevant features in the model can compromise the quality and fit.

The algorithm also needs to consider all features on an even playing field. That means the values for all features must be transformed to the same scale, otherwise one measure (in meters) would be considered more important than another measure (in hectares) only because the values are larger and have higher variability. The process of transforming numerical features to use the same scale is known as feature scaling. Another important consideration is if the feature data has missing values. To treat missing data, omitted values can either be imputed, replaced with mean values, or excluded from the dataset completely.

2 DETECTION

A pre-requirement of k-means clustering is that the number of clusters to be detected by the model be declared prior to running the model. In other words, you need to tell the model how many clusters to group the data in. The challenge, however, is in knowing the right number of clusters to specify. Fortunately, there are two popular techniques to assist in detecting the optimal number of clusters to fit the data. These are the *Elbow Method* and *Silhouette Method*.

2.1 The Elbow Method

What is The Elbow Method
The elbow method is most commonly used in finding an optimum number of clusters. This method uses within clusters sum of squares (WCSS) which accounts for the total variation within a cluster. Since this is a measure of error, the objective of k-means is to try to minimize this value. As the number of clusters increases, the variance decreases.
The results of this method are shown by plotting the Elbow curve, where the x-axis represents the number of clusters and the y-axis an evaluation metric like inertia or in this case WCSS. The idea is to specify a range of clusters and evaluate at which point the decrease in inertia starts to slow or become constant. In other words, the point at which the curve starts to flatten.

2.2 The Silhouette Method

What is The Silhouette Method
The silhouette coefficient is a measure of cluster cohesion and separation. It quantifies how well a data point fits into its assigned cluster based on two factors: 1. How close the data point is to other points in the cluster. 2. How far away the data point is from points in other clusters. Silhouette coefficient values range between -1 and 1. Larger numbers indicate that samples are closer to their clusters than they are to other clusters.

3 CLASSIFICATION

Ideally, the aim of the algorithm is to detect distinct clusters such that nodes can be uniquely identified by their feature set. By observing the features of nodes in one cluster versus those in another, it should be

possible to classify clusters based on the common features shared amongst a group of nodes. *Mean Score Plots* are a simple tool to help classify nodes within a cluster. Just by looking at the mean score plots of each cluster, it should be easier to observe which features significantly deviate from the mean and are therefore unique to the nodes in that cluster.

> ***What Are Mean Score Plots***
> In a mean score plot, each horizontal bar plots the variation (standard deviation) of a specific feature from its mean for each cluster in the model. For reference, blue bars indicate a positive variation (value greater than mean) and red bars indicate a negative variation (value smaller than mean).

4 EVALUATION

One method to visually assess how well nodes fit their assigned cluster is to plot the pairwise relationship of each node within a cluster. These pairwise plots display a matrix of plots where each node assigned to a cluster is mapped against each feature included in the model.

> ***What are Pairwise Relationship Plots***
> A pairwise relationship plot is an effective exploratory data tool that helps to simultaneously visualise the distribution of multiple variables and the relationships between each pair of variables. Each panel in the off-diagonal matrix contains a scatter plot that displays the correlation between two variables/features. Plots on the diagonal are density plots that map the distribution of each category for each continuous variable/feature.

5 REFERENCES

Goyal, Sanjeev., Connections: An Introduction to the Economics of Networks, Princeton, NJ: Princeton University Press, 2007.

Jackson, Matthew O., Social and Economic Networks, Princeton, NJ: Princeton University Press, 2008.

Wasserman, Stanley and Katherine Faust, Social Network Analysis, New York, NY: Cambridge University Press, 2007.

Watts, Duncan J., Small Worlds: the dynamics of networks between order and randomness, Princeton, NJ: Princeton University Press, 2005.

Moody, J., "Positional Analysis for Social Networks and Health," Social Networks and Health Workshop, 2021. Available at: https://sites.duke.edu/dnac/files/2021/05/PositionalAnalysis.pdf

Hoffman, Mark, "Positional analysis in networks," Bookdown. Available at: https://bookdown.org/markhoff/social_network_analysis/positional-analysis-in-networks.html

Arvai, Kevin., "K-Means Clustering in Python: A Practical Guide," Real Python. Available at: https://realpython.com/k-means-clustering-python/

Durukan, Emre., "K-Means Clustering in Python," Medium, 2021. Available at: https://medium.com/swlh/k-means-clustering-in-python-6c2d7ea01af1