PART II:
COMMUNITY ANALYSIS

The problem of finding and understanding the structure of groups in large and complex networks is called community detection. This technique is especially useful for social media algorithms or in machine learning to discover groups of similar interests and categorise them based on the connections they have. A popular application of community detection is in diffusion problems, like when assessing whether a product will be adopted or a *meme* will go viral. If the decision to adopt is largely dependent on the decision of others, then the probability of adoption can either be trapped by highly clustered ties or reinforced by them. People that have more friends are encouraged to adopt when more of their friends adopt, which is much easier when all of their friends are friends with each other. This marginal benefit diminishes, however, when communities become so clustered that no paths exist to reach outside communities. In such conditions, a *meme* introduced to a community is more likely to circulate the same users and quickly become redundant before it can ever hope to reach outside users. Additionally, uncovering communities in a network can help shape the type of interventions that have a higher chance of succeeding when introduced to a group of people. When attempting to reduce smoking habits of students in a high-school, for example, it could be more (cost) effective to strategically target densely connected communities of smokers by only focusing on the group "leaders" as opposed to the whole group.

Methods

Finding the optimal solution to detection problems can be somewhat challenging, especially since there can be any number of communities in a given network and each of varying sizes. One method that tries to uncover optimal communities is **modularity optimization**.

> ***What Is Modularity***
> Modularity is a measure of the number of intra-community ties or within-group ties (relative to the expected number of ties formed when running a random process). The modularity algorithm is optimized such that the nodes inside a community are densely connected within community and sparsely connected between communities. Modularity scores of +1 mean that all the edges in a community are connecting nodes within the community. A score of 0 would mean that the community has half its edges connecting nodes within the same community, and half connecting nodes outside the community. A score of -1 means that there are no edges connecting nodes within the community, and they instead all connect nodes outside the community. A modularity score of 0.7 and above is a generally accepted sign of a good partition.

Steps in this module are as follows:

1. **Modularity Optimisation** -> to choose the resolution parameter that finds optimal communities.
2. **Community Detection** -> to group members in communities according to their connectivity.
3. **Model Evaluation** -> to compare the partition outcomes of different detection algorithm.

1 OPTIMISATION

Maximizing modularity is computationally complex, thus various heuristic approaches are applied when trying to locate good "local" maxima of modularity. Even if these heuristics cannot make any guarantees about the true global optimum, it is still good practice to try let the data "tell" you where "good" or "robust"

values of gamma or the resolution parameter might be. A good heauristic for optimising the resolution parameter (gamma) is by plotting the number of communities at different values of gamma.

> ***What is a Resolution Parameter (Gamma)***
> Resolution is a parameter that affects the size of recovered clusters or controls the number of communities detected. Higher resolutions lead to more communities of smaller sized clusters, while lower resolutions lead to fewer communities of larger sized clusters.

When choosing the gamma, it is best to find the point that results in robust and resilient community detection. This is at the point on the graph where the number of communities detected seemingly plateaus (i.e. the number of communities detected is constant for atleast 2 or more consecutive gamma values). It is also good to note that the smaller the gamma value, the less penalty for weakly connected nodes to be grouped together in the same community.

2 DETECTION

Two popular modularity-based techniques for uncovering community structure are the **Louvain** and **Leiden** algorithms.

2.1 Community Louvain

> ***What is Louvain Method***
> The louvain method is a community structure based on the multilevel algorithm of Blondel et al (2008). This is a bottom-up algorithm that starts with a weighted network of N nodes. In the first phase, the algorithm assigns a different community to each node of the network. Then for each node, it considers the neighbours and evaluates the gain of modularity by removing the particular node from the current community and placing it in the neighbour's community. The node will be placed in the neighbour's community if the gain is positive and maximized. The node will remain in the same community if there is no positive gain. This process is applied repeatedly and for all nodes until no further improvement is found. The first phase of the louvain algorithm stops when a local maxima of modularity is obtained. In the second phase, the algorithm builds a new network considering communities found in the first phase as nodes. Once the second phase is completed, the algorithm will reapply the first phase to the resulting network. These steps are repeated until there are no changes in the network and maximum modularity is obtained.

2.2 Community Leiden

What is Leiden Method
The leiden method finds the community structure of the graph using the leiden algorithm of Traag, van Eck & Waltman (2019). It is an adaptation of the louvain method, designed to improve the tendency for the louvain community detection algorithm to discover communities that are internally disconnected (weakly connected communities). In addition to the phases used in louvain algorithm, leiden uses one more phase which tries to refine the discovered partitions. In the refinement phase, the algorithm tries to identify refined partitions from the partitions proposed by the first phase. Communities proposed by the first phase may further split into multiple partitions in the second phase. The refinement phase does not follow a greedy approach and may merge a node with a randomly chosen community which increases the quality function. This randomness allows discovering the partition space more broadly. Also in the first phase, leiden follows a different approach to the louvain. Instead of visiting all the nodes in the network after the first visit to all nodes has been completed, leiden only visits those nodes whose neighbourhood has changed.

In the analysis that follows, I use community leiden as my primary community detection method.

3 EVALUATION

There are various quantitative measures for comparing two partitions against each other. Some of the most popular are pair counting scores like “Rand” and “adjusted Rand”; and the various information-theoretic measures like “variation of information”, “normalized mutual information”, and “adjusted mutual information”.

What is Variation of Information (VI)
The variation of information or shared information distance is a measure of the distance between two clusterings (partitions of elements). As a standardised score, the variation of information value ranges between 0 (no correlation) and 1 (perfect correlation).

What is Adjusted Rand Index (ARI)
The adjusted rand index is introduced to determine whether two cluster results are similar to each other. It calculates the similarity between two cluster results by taking all points identified within the same cluster. In general, an ARI value lies between 0 and 1. The index value is equal to 1 only if a partition is completely identical to the intrinsic structure and close to 0 following a random partition.

What is Normalized Mutual Information (NMI)
Normalized mutual information is a measure used to evaluate network partitioning performed by community finding algorithms. Normalized mutual information is a normalisation of the mutual information (MI) score to scale the results between 0 (no mutual information) and 1 (perfect correlation).

4 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.

Blondel, V., Guillaume, J., Lambiotte, R. and Lefebvre, E., 2008. Fast unfolding of communities in large networks. IOPscience.

Newman, M. E. J., "Modularity and community structure in networks," PNAS, 2006, 103 (23), 8577–8582.

Newman and M. Girvan, "Finding and evaluating community structure in networks," Physical Review E, 2004, 69 (2).

Newman, M. E. J., "Fast algorithm for detecting community structure in networks," Physical review E, 2004, 69 (6).

Traag, V. A., Waltman, L. and van Eck N. J., "From Louvain to Leiden: guaranteeing well-connected communities," Scientific Reports, 2019, 9 (5233).

Mucha, Peter., "Community Detection," Social Networks and Health Workshop, 2019. Available at: https://sites.duke.edu/dnac/13-community-detection/

Moody, J., "Network Visualisation and Communities," Social Networks and Health Workshop, 2019. Available at: https://sites.duke.edu/dnac/15-network-visualization-and-communities/

Jayawickrama, Thamindu Dilshan., "Community Detection Algorithms," 2021. Available at: https://towardsdatascience.com/community-detection-algorithms-9bd8951e7dae

Noesis, "Community Detection Algorithms," Network-Oriented Exploration, Simulation, and Induction System. Available at: https://noesis.ikor.org/wiki/algorithms/community-detection