Identifying Local Community Influencers in Undirected Social Networks by Clustering and Centrality

By MENOR Edrik Jan Valdez (20696958)

Group 7

Introduction

The advent of social media has revolutionized people-to-people connections, in terms of frequency and connectivity. A person may find it easier to increase its fame via user interaction. On the other hand, certain algorithms make a person harder, in a social network, to reach out to its peers outside its community; in a “filter bubble” phenomenon. At the same time, localized targeting is increasingly an ideal business strategy, as evidenced in the rise of TikTok (Hern, 2022). These reasons state the importance of targeting users that the locally influential in a community.

Community detection in graphs is itself a classical graph-level machine learning task, in which graphs are partitioned into subgraphs or components. There exist algorithms performing such task where computational time and output are varying. In other words, each clustering algorithm divides one social network into communities each in a different manner.

In each community, there exist a handful of users in which influence is exerted over most users in that community. For such users to be influential, they may be required to enjoy high degree of connectivity with other users, but the definition of such connectivity is yet to be discovered. Different centrality measures can be used to gauge the connectivity of every node in a community, where the outcomes are differing for each method.

This project, which is to identify local community influencers, is segregated into two parts: examining how social networks can be divided into non-overlapping communities using different unsupervised clustering methods such that each community has dense connections, and examining how the most influential/connected user in a community can be found using different centrality methods. **Figure 1** specifies how the task of identifying these influencers is conducted.

The quality of clustering and its computational performance is also conducted, in order to determine the best clustering algorithm and centrality measures.

Related concepts:

Concepts related to this project are **Topological Graph Clustering** and **Node Centrality.**

Topological Graph Clustering

A graph can be divided into different clusters based on topology, Particularly, each cluster is assumed to have more edges inside the cluster than that connecting outside the cluster. A range of algorithms, including Markov clustering (Dongen, 2000), Spectral clustering, coupled with k-means(Sharma, 2022) and k-medoid(Park and Jun, 2009) clustering methods are being studied since these algorithms are likely to be biased towards subgraphs with dense edges.

Node centrality

For each graph, there exists a node that has the most central position relative to other nodes. Then the positions of other nodes are ranked relative to the most central node, by employing certain centrality measures. The definition of centrality differs, by the highest degree, the shortest paths, and others. Thus, some important centrality measures, which are degree centrality, closeness centrality and eigenvector(Bonacich, 1987) centrality are being studied in this project. However, betweenness centrality will not be studied since high betweenness is unlikely to imply popularity or influence.

Problem definition

Undirected social networks

For the simplicity of this project, it is assumed that relationships in social networks are bilateral and ambiguous to relationship types. A social network is represented as an undirected, unweighted and featureless graph . Each user is denoted by a node , and a bilateral relationship between the two users exists iff for two nodes , there exists edge .

Community

A social network can be divided into non-overlapping communities. A community is represented as a complete subgraph , where and . A community should have a sensible size, preferably at least 5 members. So, the constraint is being added. Each node represents a user/entity in a community. For two nodes , there exists an edge if users are connected. **Figure 1** shows that graph has communities.

Local Influencers

Each community has one and only node with the highest centrality, which represents that the user exerts the highest degree of influence through connections and power. These users are defined to be “local influencers”. For the degree of influence , the local influencer is represented by node such that . Here, can be defined by various centrality measures including degree centrality and eigenvector centrality.

Methodology

This project intends to construct a method to identify local community influencers by two methods in sequence: (1): Partition a social network into communities. (2): Find the most influential user in each community.

1. Graph clustering

To partition social networks into communities, the following algorithms are being used:

1: Markov clustering (Dongen, 2000)

Markov clustering is a clustering method based on unbiased random walks on a graph. Such random walks would mainly be more likely to be inside clusters than across clusters. That is, edges that are across clusters are much less likely to be travelled on.

.

1. Compute column stochastic matrix where for ,
2. While is not converged do
   1. Construct matrix where for ,
   2. Delete edges where and
3. Obtain clusters

Algorithm 1: Markov clustering method

Given a graph , there is a normalized column stochastic matrix for representing random walks, where for each column vector representing node , each non-zero element is valued as the probability of walking from node to node .

The expansion factor is an integer that represents the number of steps taken on a random walk. The matrix is multiplied times, to calculate the probability of travelling from node to node within walks of length , .

The inflation factor is a real number used to augment the probability of travelling from node to node . By principle, that is to increase the difference of the probability of edges walked inside communities with that of edges walked across communities using the exponential power of . To do so, is being multiplied to the power of , then normalized with other edges incident with node , in a similar manner as the softmax function.

Then the edges that are extremely unlikely to be travelled on, say for any edge , have a probability less than an extremely small number stated in Algorithm 1. These edges would be deleted to make the connections between clusters sparse.

According to Algorithm 1, the steps of expanding, inflating and deleting are repeated until the column stochastic matrix reaches its convergent state, meaning that no more random walks can be done to discover communities.

Finally, for each column vector that represents node , if there is a non-zero element representing node , then node and belong to same cluster.

The tuple needs to be selected so that the clustering algorithm give the best results, which can be measured with the modularity index. However, the modularity index will be studied in the Evaluation section.

2: Spectral clustering (from class)

Spectral clustering uses some eigenvectors of the Laplacian matrix of a graph to identify clusters.

such that clustering labels are well distributed

1. Extract degree and adjacency matrices and
2. Calculate Laplacian
3. Calculate the first eigenvalues and corresponding eigenvectors of matrix in ascending order of eigenvalues
4. Perform -means/medoid clustering on the nd to th eigenvectors in ascending order of eigenvalues
5. Cluster nodes by clustering labels

Algorithm 2: Spectral clustering algorithm implemented

First, a matrix embedding of the representation of the graph needs to be determined, so the graph Laplacian

is calculated, given the degree matrix and adjacency matrix . The adjacency matrix alone is insufficient for initiating this clustering method.

Next, a spectral embedding of the graph needs to be obtained. This can be done by calculating the smallest eigenvalues of matrix and its corresponding eigenvectors. Note that the smallest eigenvalue is always 0, which has no practical meaning in this algorithm other than representing the complete graph.

Instead, the spectral embedding required is where its -th column is the eigenvector associated with the th smallest eigenvalue, and the -th row represents the -th node.

However, certain classical clustering algorithms, including k-means and k-medoid, are still required to map these nodes to corresponding clusters. Here, the number is the number of clusters needed. The best selection for requires the clustering labels from these k-means/k-medoid methods to be well-distributed, say having an appropriate standard deviation, etc, which will be discussed in the Evaluation section as well.

2.1: K-means clustering (Sharma, 2022)

K-means clustering is a classical clustering method based on distance between centroids and nodes. In principle, graphs will need to be embedded through a representation learning method, such as the spectral embedding discussed earlier.

Input:

1. Select nodes (centroids) from nodes randomly
2. For nodes to , excluding the nodes (centroids) chosen earlier,
   1. Choose a node from the nodes (centroids) such that , where can be any distance measure is minimum
   2. Label the node with the node
3. If average is not minimum, i.e. node not the closest to each other, repeat steps 1 and 2
4. End

Algorithm 2.1: Rough procedure of k-means clustering

First, the number of clusters required needs to be determined. Then in most cases, nodes will be randomly selected from nodes.

Next, for the rest of the nodes, the distance between such node, say , and the previously selected centroid is being calculated. The distance measure can range from Euclidean distance to Manhattan distance. The centroid should also be chosen for the distance to be minimum. After that, the node should be labelled to associate with node .

Then the average distance between centroid and the nodes that are labelled with node is estimated. If the average distance is not the minimum, this implies that the clusters induced from the previous attempt is not dense enough. The clustering procedure will need to restarted.

Otherwise, the clustering procedure is complete.

2.2: K-medoid clustering (Park and Jun, 2009)

The k-medoid clustering method is a variant of the k-means algorithm in that it biases on associating outliers with the closest cluster instead of excluding or made as its own cluster.

Input:

1. For each node pair calculate distance where is any distance measure

2. For node , calculate as

1. Pick the nodes with the smallest values of , as attractors
2. Perform step 2 in Algorithm 2.1
3. Update the attractors with the closest neighbouring nodes of the previous attractors
4. Perform step 2 in Algorithm 2.1
5. If the criteria stated in Step 3 of Algorithm 2.1 is not satisfied, repeat steps 6 and 7
6. End

Algorithm 2.2: Rough procedure of k-medoid clustering

The similarity between k-means and k-medoid clustering is the labelling of nodes with “centres” determined by mutual distance. But the difference is that k-medoids, unlike k-means, associates outliers with the closest dense cluster, as demonstrated in Step 2 of 2.2 instead of singular centres.

This reduces the risk of excluding distant nodes and cluster granularization.

3. Louvain algorithm (Blondel et al., 2008)

This is a hierarchical clustering algorithm aimed at maximizing the modularity of a graph. The measure of modularity will be discussed as the evaluation section.

Input:

* 1. Assign a community label to each node
  2. For each node , consider nodes where
     1. Label node , with the neighbouring node such that is locally maximized and only positive

Here is defined as

Where is the number of edges from node to nodes in community

* + 1. If no positive , node retains its original label
  1. Let communities found be new nodes. Let the value of new edges be the number of edges between two communities.
  2. If modularity is maximum then STOP
  3. Else repeat steps a and b
  4. Return list of communities

Algorithm 3: Louvain algorithm

The idea of this algorithm is to aggregate communities from a singleton of node to subset of nodes such that the modularity is maximized.

During the process, virtual nodes representing communities are formed.

1. Node centrality measures

To search for the most influential user in a community, i.e. find the node with the highest centrality, the following centrality measures are being employed:

1: Degree centrality

This centrality method stipulates that the most central node in a graph has the highest degree among all nodes. The centrality equation is defined as

2: Closeness centrality

This centrality method stipulates that the most central node in a graph has the shortest length of distances, i.e. shortest paths to all other nodes. The centrality equation is defined as

where is the length of the shortest path from node to node .

3: Eigenvector centrality (Bonacich, 1987)

This centrality method stipulates that the most central node in a graph explicitly has the highest degree of influence. The centrality is calculated as follows:

1. Given the adjacency matrix of network
2. Find the largest eigenvalue and corresponding eigenvector such that
3. Let centrality , where is the element is vector corresponding to node .

Each influencer found will be placed in a list specified in **Figure 1.**

Datasets

The datasets that are tested on are representations of small social networks, of which members are human or animals that are undirected, unweighted and without node features.

(Zachary, 1977) (Lusseau et al., 2003)

(Firth & Sheldon, 2015) (Newman, 2006) (Leskovec et al., 2010)

|  |  |  |  |
| --- | --- | --- | --- |
| **Dataset no.** | **Dataset name** | **Number of nodes** | **Number of edges** |
| 1 | Zachary’s karate club | 34 | 78 |
| 2 | Dolphins | 62 | 159 |
| 3 | Aves-wildbird-network-3 | 169 | 1615 |
| 4 | Aves-wildbird-network | 202 | 11780 |
| 5 | Ca-netscience | 379 | 914 |
| 6 | Wiki-vote | 889 | 2914 |

Experiment parameters and metrics

Depending on the topology of the datasets listed above, and the parameters required for clustering method, the parameters are set as follows:

Markov clustering (Dongen, 2000):

Inflation factor : 1.1, 1.2, 1.3, …, 5.8, 5.9 Expansion factor : 2, 3, 4, 5, 6

During the clustering process of the graph, the inflation factor and the expansion factor giving the highest modularity will be selected as the final parameters for outputting the list of influencers.

Spectral clustering:

Smaller datasets: Bigger datasets = 5 up to 70

During the clustering process of the graph, the statistical standard deviation of clustering is also calculated, so that a high relates to granular clusters, while a low relates to chunky clusters.

Louvain clustering [10]:

There are virtually no parameters required since the algorithm is aimed at maximized modularity, but there is a resolution parameter that changes the cluster size. Normally, indicates larger clusters, and this project favours slightly larger communities, so set

To evaluate the computational performance, the time required to complete a round of clustering and centrality measuring is being recorded. The clustering methods are being tested on the following specifications:

CPU: AMD Ryzen 5 5600H

RAM: 16GB

GPU: AMD Radeon iGPU

OS: Windows 11 Pro

To evaluate the consistency and accuracy of the clustering and centrality methods, the Jaccard index , where

Is being calculated, where are the lists of influencers retrieved by two of the tested clustering and centrality methods. A higher indicates a better score.

Evaluation

Preliminary metric: Modularity (Malliaros & Vazirgiannis, 2013)

It is common to study the clustering quality of graph by using a modularity index , ranged from . Closer to indicates excellent quality, while closer to indicates the opposite. Here, m can be defined as

where for , adjacency matrix , and

Topological analysis

During the Markov clustering process, the modularity is being calculated for each expansion factor and inflation factor . This gives a view on the topological properties of each network tested.

Also, in the spectral clustering process, the statistical standard deviation of clustering labels is also calculated with the number . This gives an overview of the ideal number of clusters for each network. However, spectral clustering cannot be done on dataset 5 and 6, which will be explain in the Analysis part.

Dataset 1

From **Figure 2a**, the highest modularity, which is between and , occurs when . This means that a community inside this network has a maximal diameter of 2. The communities in this network may be dense. Also, according to **Figure 3a**, the number of communities is estimated to be 2 or 3.

Dataset 2

From **Figure 2b,** the highest modularity, which is around , occurs when . This means that a community inside this network has a maximal diameter of 6. The communities in this network may be sparse. Also, according to **Figure 3b,** the number of communities is estimated to be 4 or 5.

Dataset 3

From **Figure 2c,** the highest modularity, which is around 0.6, occurs when . This means that a community inside this network has a maximal diameter of 2. The communities in this network may be dense. Also, a modularity difference of around between and shows that communities are distant from each other. Also, according to **Figure 3c**, the number of communities is estimated to be between 5 to 40.

Dataset 4

From **Figure 2d,** the highest modularity, which is more than , occurs when , showing that a community as a diameter of . When , the modularity converges at around . This indicates that nodes are densely clustered into their own communities. Also, according to **Figure 3d,** the number of communities is estimated to be between 5 to 20.

Dataset 5

From **Figure 2e,** the highest modularity, between and , occurs when . There is no definite maximal diameter of a community, so the network is likely to have a lack of communities.

Dataset 6

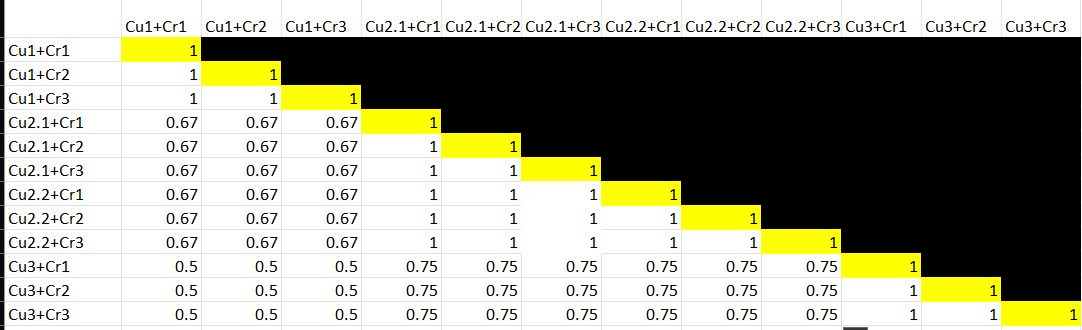
From **Figure 2f,** the highest modularity is around when . This means the network has several dense communities of diameter 2.

Experimental results

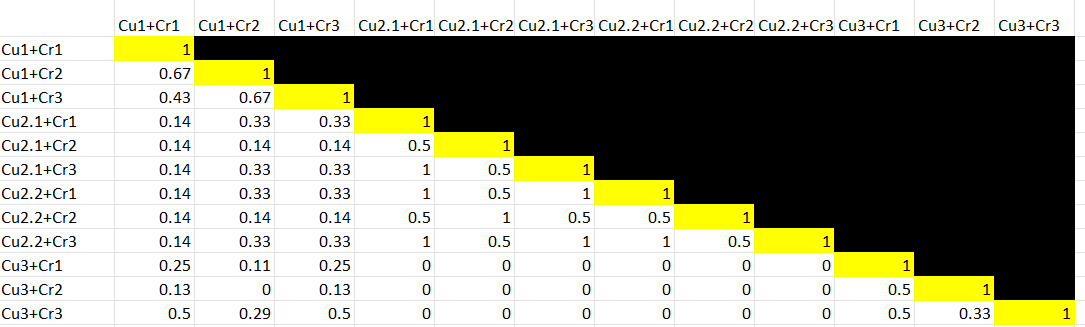
Denote Cu1 as Markov clustering, Cu2.1 as spectral clustering with k-means, Cu 2.2 as spectral clustering with k-medoid, Cu3 as Louvain clustering.

Denote Cr1 as Degree centrality, Cr2 as closeness centrality and Cr3 as eigenvector centrality.

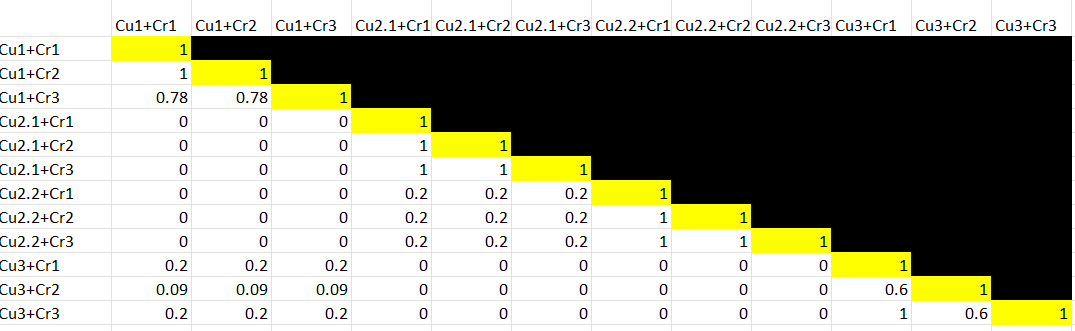
Dataset 1



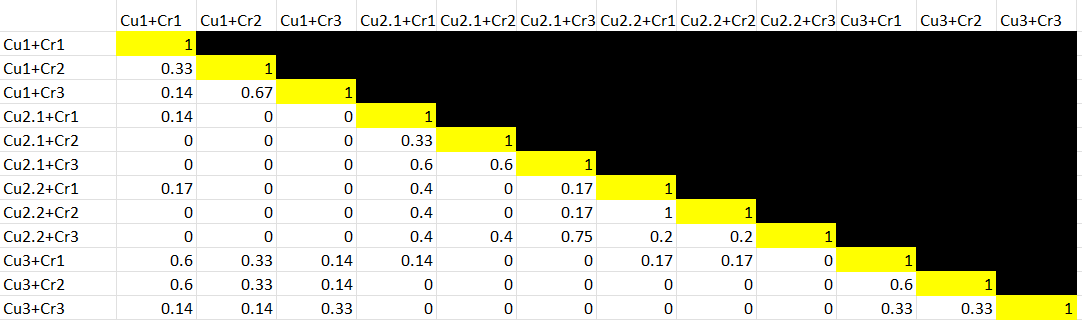
Dataset 2



Dataset 3



Dataset 4



Dataset 5

Shape

Description automatically generated

Dataset 6

Shape

Description automatically generated

Time computed:

Graphical user interface, table

Description automatically generated

Note that the red parts indicate that the algorithm cannot be proceeded.

Analysis

On clustering

The clustering methods divide graphs each in a different manner. These methods occupy most of the computational process, thus contributing to the computational time required. Based on the computational time required but less on the Jaccard indices, we now analyse the following cluster algorithms:

1. Markov clustering

This clustering method is built upon the degree of every node in the network and random walks with a fair probability, like PageRank. Based on the high Jaccard indices based on the Markov clustering and any centrality measure, this clustering method generally gives the most consistent results. For each social network tested, the method divides the network into the same, or nearly the same communities. For example, the wiki-vote network (dataset 6) has the Jaccard index of between 0.9 and 1 between lists of influencers created by Markov clustering and any centrality measure.

However, this clustering method is also the slowest out of all clustering methods tested. For instance, for the same dataset 6, it took as long as almost 5 hours to produce a list of influencers. This is likely due to the matrix multiplication in Step 2a of Algorithm 1, which requires vast amounts of computational power, and the issue accentuates with the increasing expansion factor . Also, the requirement for the convergence of the matrix does not help in reducing computational time either.

Therefore, Markov clustering is a consistent but extremely slow clustering method.

1. Spectral clustering

This clustering methods is based on the spectral embedding which is derived from extracting the *k* smallest eigenvalues and eigenvectors of the graph Laplacian. By principle, this algorithm is reliant on the number , where . Performing -means and -medoid clustering on the spectral embedding is also required.

When considering the computational time required, the clustering process is fairly fast. This is because instead of performing matrix multiplication, it only requires eigen-decomposition which uses an identity matrix, an integer and subtraction. The largest dataset possibly tested on (dataset 4) only requires at most 2.9s for the entire clustering process to complete.

Unfortunately, this clustering method generally gives inconsistent lists of influencers, judging from the generally low Jaccard indices between lists from spectral clustering and lists from any clustering method tested in all datasets. This is possibly due to the random selection of nodes that become “centres” of communities, especially in k-means clustering, which may result in division into different communities every time the algorithm is run. However, that problem is less severe when run alongside k-medoids clustering, which according to Algorithm 2.2, determines the “centroids” based on the network topology beforehand, and considers outliers as part of clusters. Thus, k-medoids clustering can output lists of influencers with a generally slightly higher Jaccard index.

Another disadvantage is that spectral clustering cannot be ideally performed on larger networks (datasets 5 and 6). This is because attempting to perform eigen-decomposition on large networks would extremely likely to return eigenvalues on the complex space, which cannot be mapped into the real values.

Therefore, spectral clustering is relatively light and modular, but the results are inconsistent and worse, non-existent.

1. Louvain clustering

This clustering method is basically aggregating communities from singletons of single nodes to subgraphs consisting of several nodes. It does not require parameters, meaning that it is an unsupervised algorithm, although some parameters, such as resolution can be set to adjust the sizes of communities.

The major advantage of this clustering method is the fast computational time, as the majority of procedures require less than a second to compute. For every node , its neighbouring nodes are being considered, then these two nodes will be clustered as one community if the modularity is maximized. The point is that communities identified are being considered as nodes for every iteration, which drastically reduces the number of nodes being processed, thus contributing to the notably lower computational time.

The aggregation of nodes based on modularity is similar to neighbourhood sampling in GNNs, since the algorithm aggregates the neighbouring nodes around any node , where the distance between the two nodes is being summed and averaged as shown in Algorithm 3. The result is the node-level representations of communities, similar to how graph representation learning is done. In the end, for each network, the Louvain algorithm identifies the same communities consistently.

The only downside is that the resulting Jaccard indices is generally lower than that of Markov clustering, but that issue is most likely caused by differing centrality measures.

On centrality

Given a set of communities from the original social network, the decisive factor for forming lists of influencers are the centrality measures. In this analysis, the Jaccard indices for different clustering methods with the same clustering method is compared.

Table

Description automatically generated

1. Degree centrality

The degree centrality is based on the degree of a node in a community. This represents the number of relationships a member has, and implies that more relationships make way for greater influence. The advantage is that it is straightforward to calculate.

The Jaccard indices about the degree centrality shown on the table are fairly high, possibly because the most central nodes inside the networks tested usually have the highest degree among all nodes.

1. Closeness centrality

The closeness centrality is calculated on the shortest paths between nodes in a community. This implies that the user with least number of “brokers” can find it easier to influence all other users, which is also connected to community density.

The Jaccard indices about the closeness centrality shown on the table are the lowest, possibly because there is not only one node with the least average shortest path.

1. Eigen centrality

The eigen centrality is calculated by creating a one-dimensional embedding which is the eigenvector associated with the largest eigenvalue . Then the largest element of vector corresponds to the user with the most influence.

The Jaccard indices about the eigen centrality are very slightly higher than that of the degree centrality, possibly because the spectral embeddings are most likely to be unique.

Recommendations

The Markov clustering method can divide a social network into one set of communities in a highly consistent manner, although the computational complexity is extremely high.

On the other hand, the Louvain clustering method has a rather low computational complexity and detects communities in a way like GNN-based graph representation learning.

The calculation of the eigencentrality also produces a one-dimensional graph embedding via eigen decomposition, in which these embeddings have a high propensity to be unique for each community.

Therefore, disregarding the time constraints, identifying local community influencers in social networks is best done by Markov clustering method and using eigencentrality measure.

Considering time constraints however, identifying local community influencers in social networks is best done by Louvain clustering method and using eigencentrality measure.

Conclusion

In this project, many clustering methods and centrality measures are being studied, both of which differ fundamentally, as in computation methods. The process of graph clustering and measuring node centrality has a critically important application in social network analysis, in discovering how certain people hold more social influence over all others (e.g. Key Opinion Leaders) and exploiting it. It is believed that both in real life and in this experimental project, there are differing views on who holds such power.

On the other hand, certain clustering methods featured in this project each utilize a different principle, leading to different results and computational time; as is for centrality measures. After all, this project reveals that a few clustering methods demonstrate better performance and consistency than that of other clustering methods, and there may be one centrality measure that is more uniform than that of other centrality methods.

It is admittable that other clustering and centrality measure are developed in the future for further exploration.

References

Hern, A. (2022, October 24). How TikTok's algorithm made it a success: 'it pushes the boundaries'. The Guardian. Retrieved December 15, 2022, from <https://www.theguardian.com/technology/2022/oct/23/tiktok-rise-algorithm-popularity>

Dongen van S. M. (2000). Graph clustering by flow simulation (dissertation). Universiteit Utrecht.

Sharma, N. (2022, November 14). K-means clustering explained. neptune.ai. Retrieved December 15, 2022, from <https://neptune.ai/blog/k-means-clustering>

Park, H.-S., & Jun, C.-H. (2009). A simple and fast algorithm for K-medoids clustering. Expert Systems with Applications, 36(2), 3336–3341. <https://doi.org/10.1016/j.eswa.2008.01.039>

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 2008(10). <https://doi.org/10.1088/1742-5468/2008/10/p10008>

Bonacich, P. (1987). Power and centrality: A family of measures. American Journal of Sociology, 92(5), 1170–1182. <https://doi.org/10.1086/228631>

Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. Journal of Anthropological Research, 33(4), 452–473. <https://doi.org/10.1086/jar.33.4.3629752>

Lusseau, D., Schneider, K., Boisseau, O. J., Haase, P., Slooten, E., & Dawson, S. M. (2003). The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. Behavioral Ecology and Sociobiology, 54(4), 396–405. <https://doi.org/10.1007/s00265-003-0651-y>

Firth, J. A., & Sheldon, B. C. (2015). Experimental manipulation of avian social structure reveals segregation is carried over across contexts. Proceedings of the Royal Society B: Biological Sciences, 282(1802), 20142350. <https://doi.org/10.1098/rspb.2014.2350>

Newman, M. E. (2006). Finding community structure in networks using the eigenvectors of matrices. Physical Review E, 74(3). <https://doi.org/10.1103/physreve.74.036104>

Leskovec, J., Huttenlocher, D., &amp; Kleinberg, J. (2010). Signed networks in Social Media. Proceedings of the 28th International Conference on Human Factors in Computing Systems - CHI '10. <https://doi.org/10.1145/1753326.1753532>

[10] [louvain\_communities — NetworkX 2.8.8 documentation](https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.community.louvain.louvain_communities.html)

Malliaros, F. D., & Vazirgiannis, M. (2013). Clustering and community detection in directed networks: A survey. Physics Reports, 533(4), 95–142. <https://doi.org/10.1016/j.physrep.2013.08.002>

Appendix

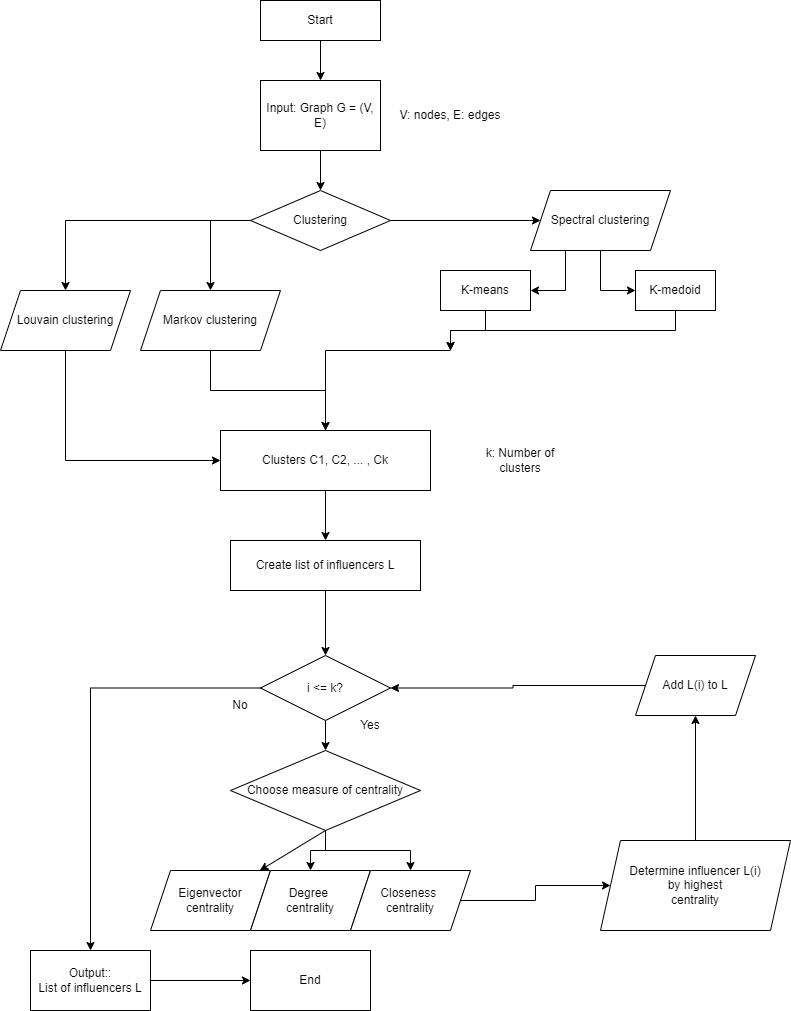
Figure 1: Flowchart of solution

Figure 2: Markov clustering analysis graphs

Figure 2a

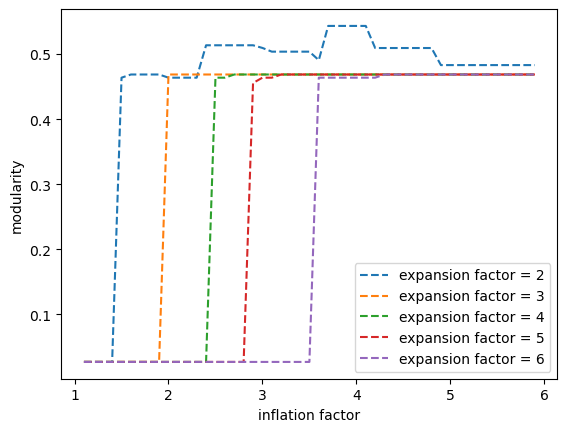


Figure 2b

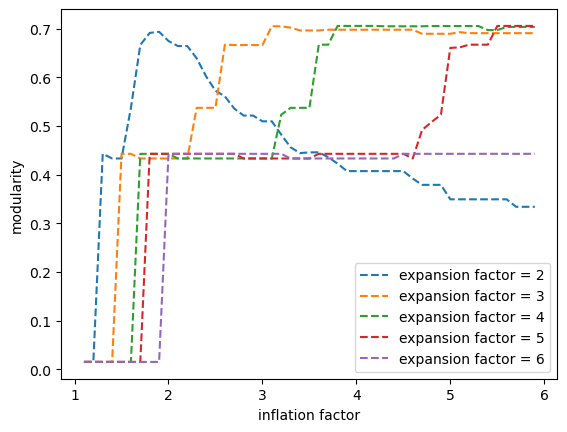


Figure 2c

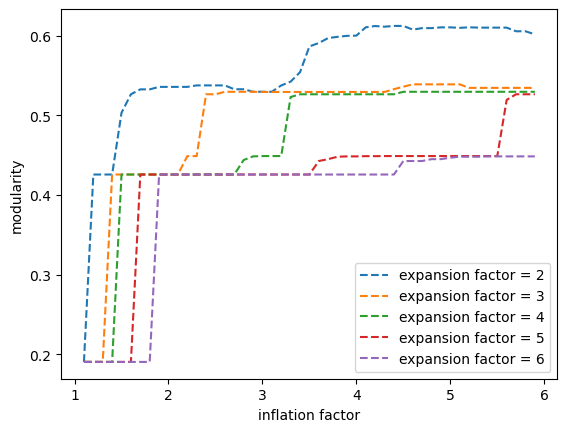


Figure 2d

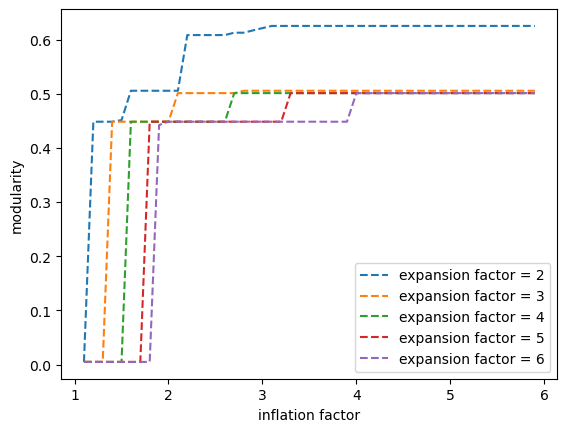


Figure 2e

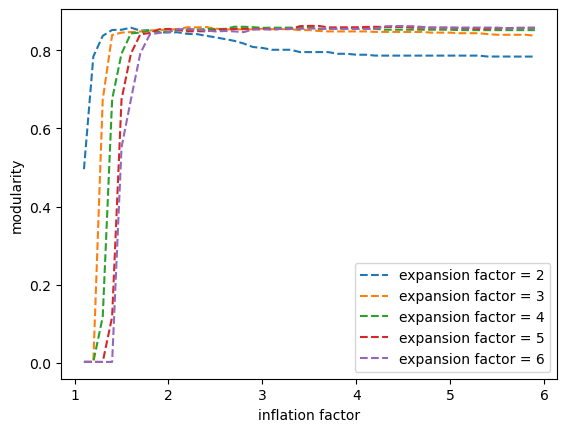


Figure 2f

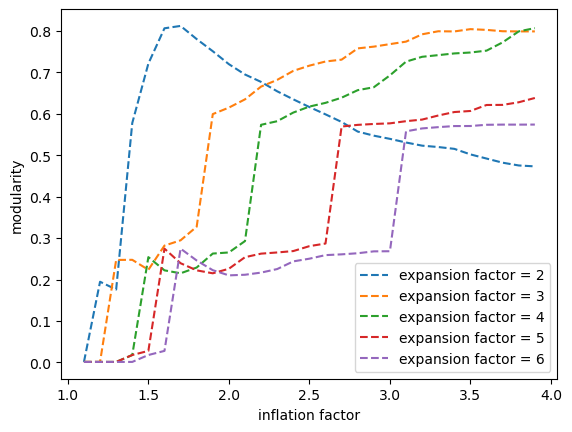


Figure 3: Spectral clustering graphs

Figure 3a

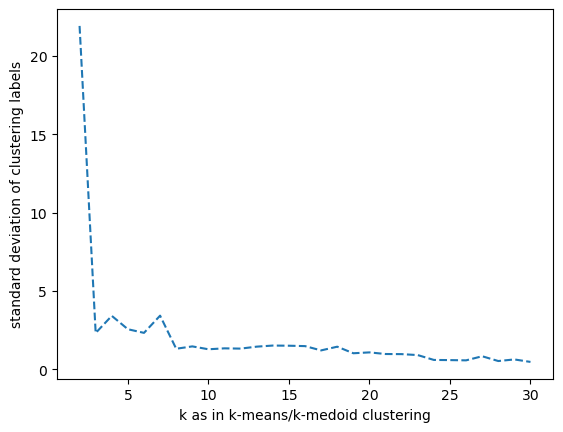


Figure 3b

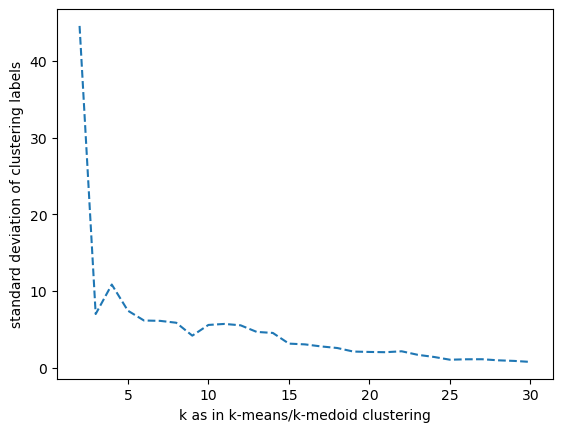


Figure 3c

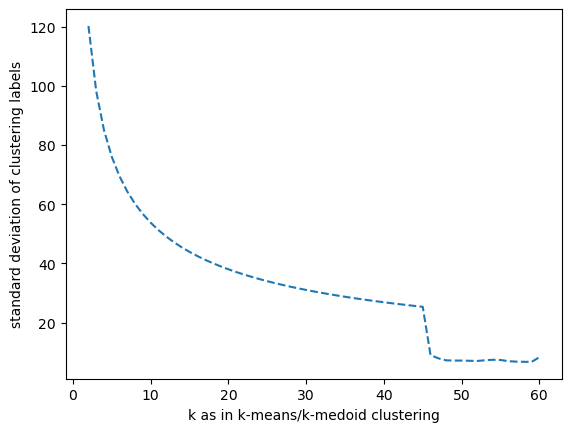
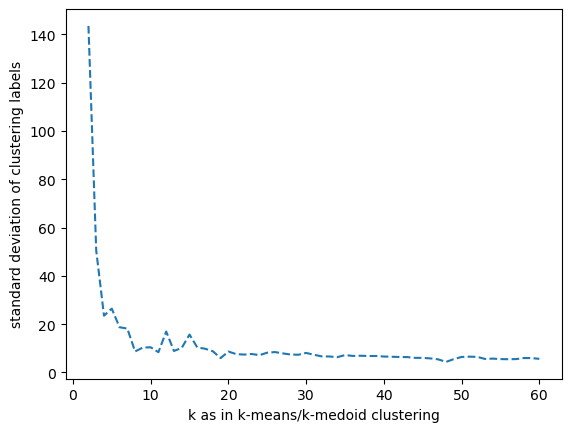


Figure 3d



.