# CS529 HW4

## Q1 – Les Miserables Experiments

### Q1A1 – Cluster Memberships

#### Girvan – Newman Clusters

Total clusters detected: **11**

*Table 1: Samples in Cluster 0*

| ID | Name |
|---|---|
| 0.0 | Myriel |
| 1.0 | Napoleon |
| 2.0 | MlleBaptistine |
| 3.0 | MmeMagloire |
| 4.0 | CountessDeLo |
| 5.0 | Geborand |
| 6.0 | Champtercier |
| 7.0 | Cravatte |
| 8.0 | Count |
| 9.0 | OldMan |

*Table 2: Samples in Cluster 1*

| ID | Name |
|---|---|
| 12.0 | Marguerite |
| 16.0 | Tholomyes |
| 17.0 | Listolier |
| 18.0 | Fameuil |
| 19.0 | Blacheville |
| 20.0 | Favourite |
| 21.0 | Dahlia |
| 22.0 | Zephine |
| 23.0 | Fantine |
| 30.0 | Perpetue |

*Table 3: Samples in Cluster 2*

| ID | NamBelow is the |
|---|---|

| | clusters detected by Girvan-Newman with the layout of the nodes editted for clarity.e |
|---|---|
| 46.0 | Jondrette |
| 47.0 | MmeBurgon |

*Table 4: Samples in Cluster 3*

| ID | Name |
|---|---|
| 48.0 | Gavroche |
| 55.0 | Marius |
| 57.0 | Mabeuf |
| 58.0 | Enjolras |
| 59.0 | Combeferre |
| 60.0 | Prouvaire |
| 61.0 | Feuilly |
| 62.0 | Courfeyrac |
| 63.0 | Bahorel |
| 64.0 | Bossuet |
| 65.0 | Joly |
| 66.0 | Grantaire |
| 76.0 | MmeHucheloup |

*Table 5: Samples in Cluster 4*

| ID | Name |
|---|---|
| 10.0 | Labarre |
| 11.0 | Valjean |
| 13.0 | MmeDeR |
| 14.0 | Isabeau |
| 15.0 | Gervais |
| 29.0 | Bamatabois |
| 31.0 | Simplice |
| 32.0 | Scaufflaire |
| 33.0 | Woman1 |
| 34.0 | Judge |
| 35.0 | Champmathieu |
| 36.0 | Brevet |
| 37.0 | Chenildieu |
| 38.0 | Cochepaille |

*Table 6: Samples in Cluster 5*

| ID | Name |
|---|---|
| 26.0 | Cosette |
| 43.0 | Woman2 |
| 49.0 | Gillenormand |
| 50.0 | Magnon |
| 51.0 | MlleGillenormand |
| 52.0 | MmePontmercy |
| 53.0 | MlleVaubois |
| 54.0 | LtGillenormand |
| 56.0 | BaronessT |
| 72.0 | Toussaint |

*Table 7: Samples in Clusters 6-9*

| ID | Name | Cluster ID |
|---|---|---|
| 28.0 | Fauchelevent | 6 |
| 44.0 | MotherInnocent | 6 |
| 45.0 | Gribier | 6 |
| 40.0 | Boulatruelle | 7 |
| 67.0 | MotherPlutarch | 8 |
| 73.0 | Child1 | 9 |
| 74.0 | Child2 | 9 |

*Table 8: Samples in Clusters 10*

| ID | Name |
|---|---|
| 24.0 | MmeThenardier |
| 25.0 | Thenardier |
| 27.0 | Javert |
| 39.0 | Pontmercy |
| 41.0 | Eponine |
| 42.0 | Anzelma |
| 68.0 | Gueulemer |
| 69.0 | Babet |
| 70.0 | Claquesous |
| 71.0 | Montparnasse |
| 75.0 | Brujon |

## Modularity Clusters

Total clusters detected: **6**

*Table 9: Samples in Clusters 0*

| ID | Name |
|---|---|
| 0.0 | Myriel |
| 1.0 | Napoleon |
| 2.0 | MlleBaptistine |
| 3.0 | MmeMagloire |
| 4.0 | CountessDeLo |
| 5.0 | Geborand |
| 6.0 | Champtercier |
| 7.0 | Cravatte |
| 8.0 | Count |
| 9.0 | OldMan |

*Table 10: Samples in Clusters 1*

| ID | Name |
|---|---|
| 12.0 | Marguerite |
| 16.0 | Tholomyes |
| 17.0 | Listolier |
| 18.0 | Fameuil |
| 19.0 | Blacheville |
| 20.0 | Favourite |
| 21.0 | Dahlia |
| 22.0 | Zephine |
| 23.0 | Fantine |

*Table 11: Samples in Clusters 2*

| ID | Name |
|---|---|
| 24.0 | MmeThenardier |
| 25.0 | Thenardier |
| 40.0 | Boulatruelle |
| 41.0 | Eponine |
| 42.0 | Anzelma |
| 68.0 | Gueulemer |
| 69.0 | Babet |
| 70.0 | Claquesous |
| 71.0 | Montparnasse |

| | |
|---|---|
| 75.0 | Brujon |

*Table 12: Samples in Clusters 3*

| ID | Name |
|---|---|
| 29.0 | Bamatabois |
| 34.0 | Judge |
| 35.0 | Champmathieu |
| 36.0 | Brevet |
| 37.0 | Chenildieu |
| 38.0 | Cochepaille |

*Table 13: Samples in Clusters 4*

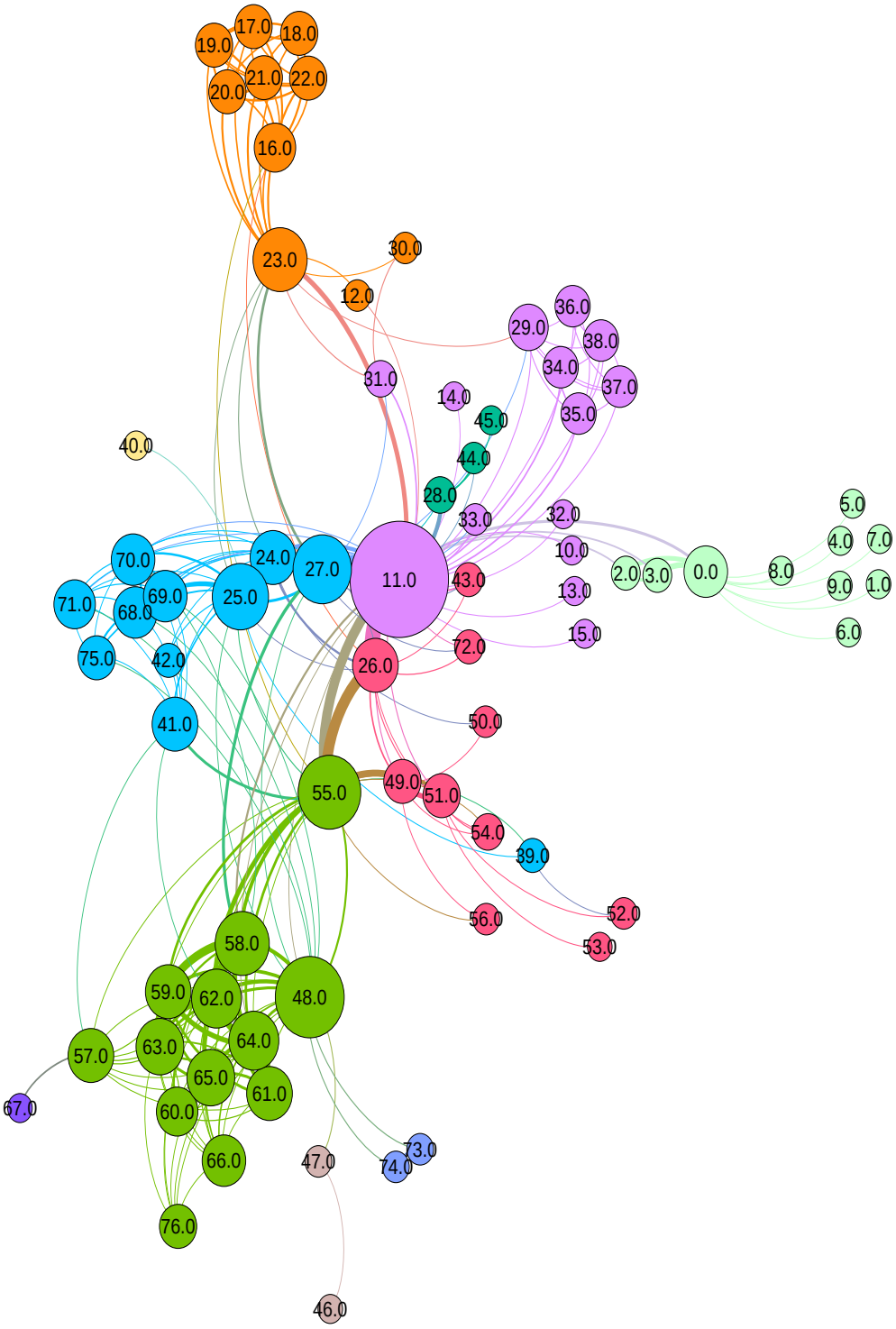
| ID | Name |
|---|---|
| 10.0 | Labarre |
| 11.0 | Valjean |
| 13.0 | MmeDeR |
| 14.0 | Isabeau |
| 15.0 | Gervais |
| 26.0 | Cosette |
| 27.0 | Javert |
| 28.0 | Fauchelevent |
| 30.0 | Perpetue |
| 31.0 | Simplice |
| 32.0 | Scaufflaire |
| 33.0 | Woman1 |
| 39.0 | Pontmercy |
| 43.0 | Woman2 |
| 44.0 | MotherInnocent |
| 45.0 | Gribier |
| 49.0 | Gillenormand |
| 50.0 | Magnon |
| 51.0 | MlleGillenormand |
| 52.0 | MmePontmercy |
| 53.0 | MlleVaubois |
| 54.0 | LtGillenormand |
| 55.0 | Marius |
| 56.0 | BaronessT |
| 72.0 | Toussaint |

*Table 14: Samples in Clusters 5*

| ID | Name |
|---|---|
| 46.0 | Jondrette |
| 47.0 | MmeBurgon |
| 48.0 | Gavroche |
| 57.0 | Mabeuf |
| 58.0 | Enjolras |
| 59.0 | Combeferre |
| 60.0 | Prouvaire |
| 61.0 | Feuilly |
| 62.0 | Courfeyrac |
| 63.0 | Bahorel |
| 64.0 | Bossuet |
| 65.0 | Joly |
| 66.0 | Grantaire |
| 67.0 | MotherPlutarch |
| 73.0 | Child1 |
| 74.0 | Child2 |
| 76.0 | MmeHucheloup |

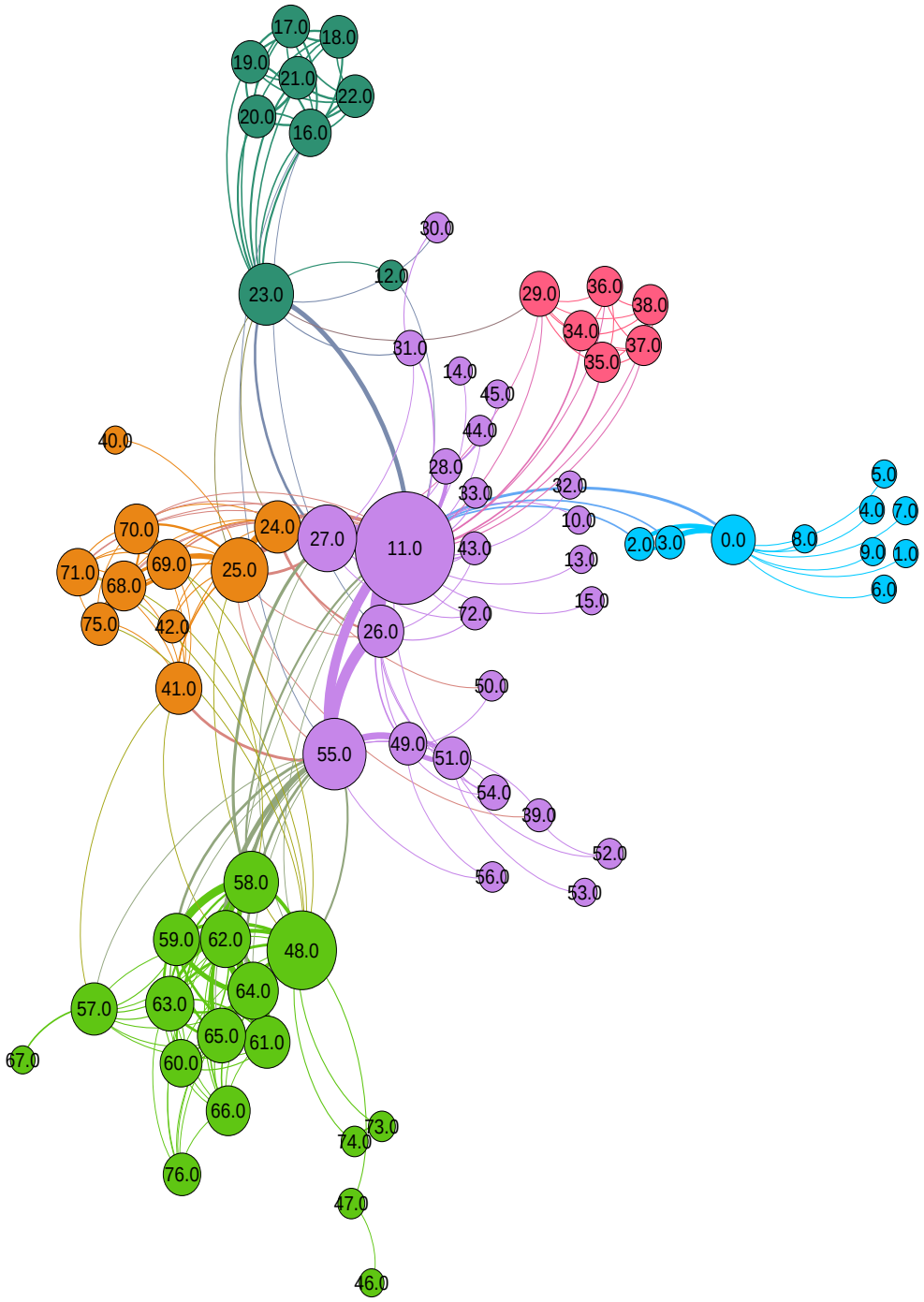
## Q1B – Girvan – Newman Visualization

Below is the clusters detected by Girvan-Newman with the layout of the nodes editted for clarity.



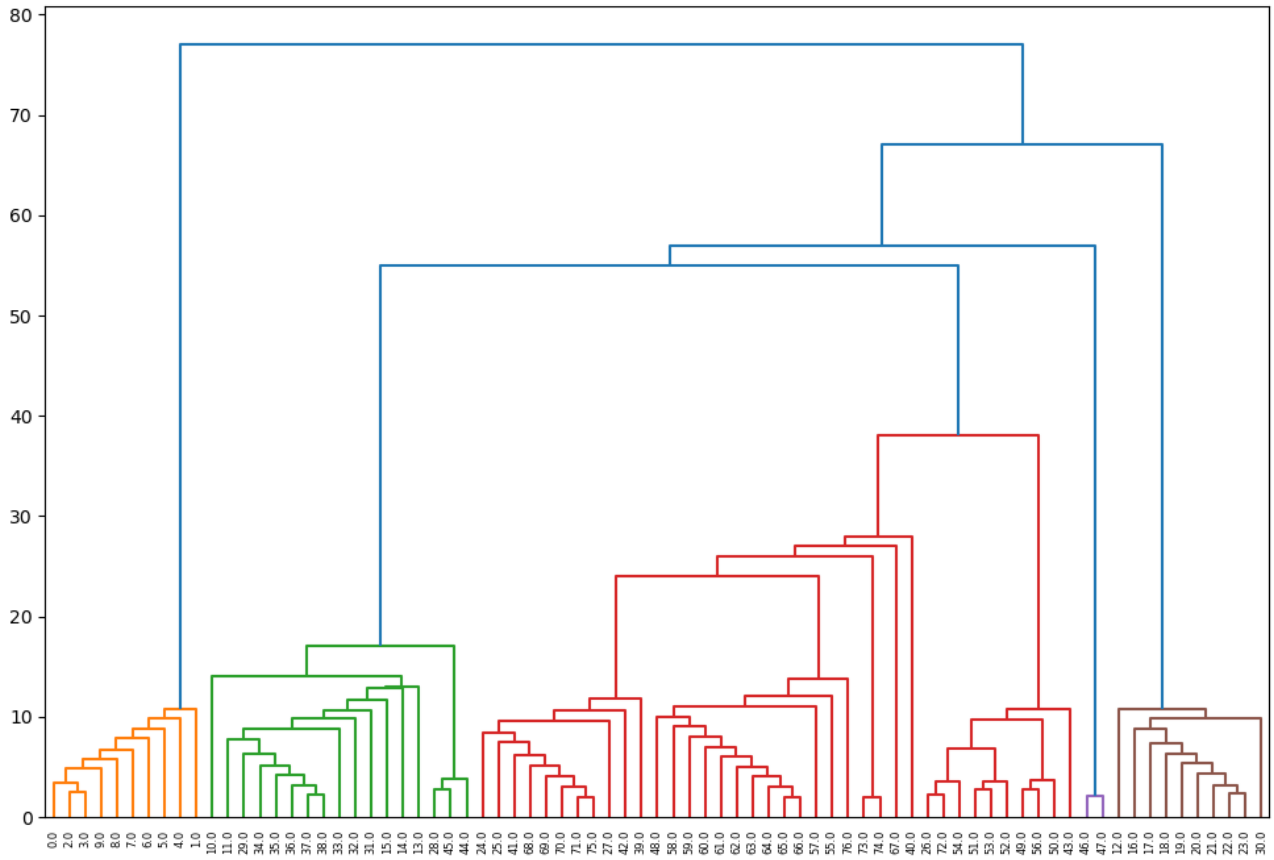
## Q1C – Modularity Visualization

Below is the clusters deteted by Modularity-based clustering with the layout of the nodes editted for clarity.



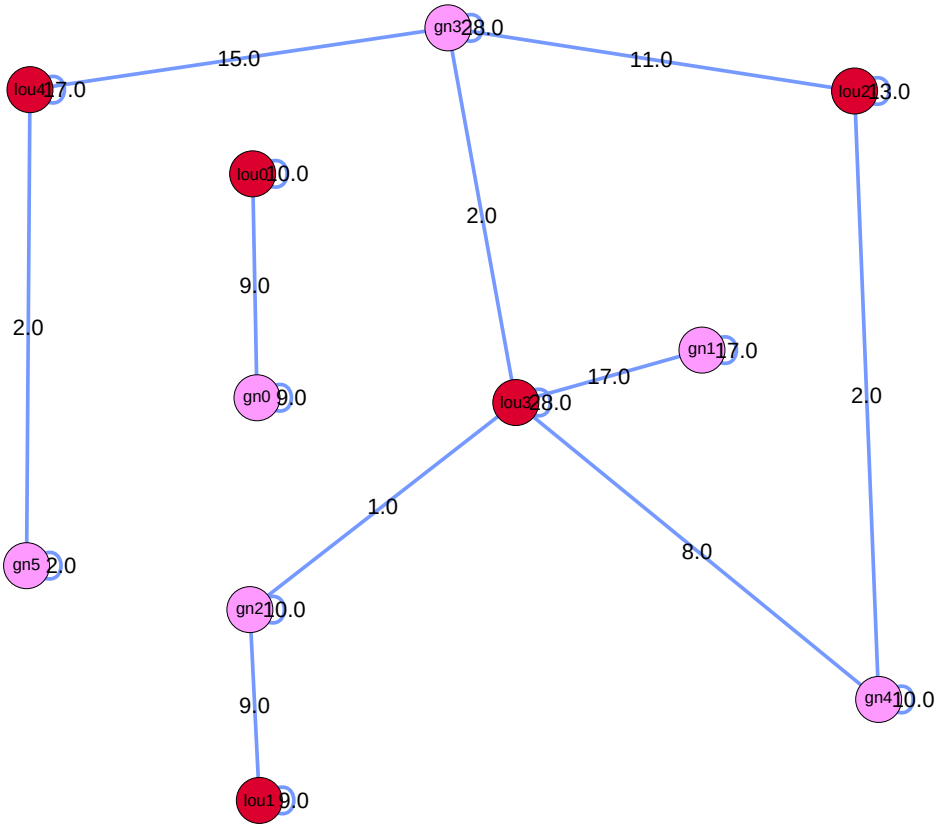
## Q1D – Dendogram Visualization

The following dendogram is generated based on Girvan-Newman implementation by Python NetworkX package. Since this packages behaves differently than Gephi plugin for the same algorithm, 5 clusters are detected instead of 11, as shown below. In the figure, the vertical axis (y) indicates cluster distances where horizontal axis (x) shows node ids.



## Q1E – Cluster Relations

The figure below shows the relationship between the clusters detected by Girvan-Newman and Louvain clustering algorithms, obtained by one-mode mapping of node – detected clusters bipartite network over the detected clusters.




In this figure, clusters detected by Python NetworkX package are employed. Red nodes are the clusters detected by the Louvain algorithm (named *'louX'*, for an integer *X* in [0,4]), the pink nodes are the clusters the figure in Section *Q1D – Dendogram Visualization* is based on, obtained by Girvan-Newman algorithm (named *'gnY'*, for an integer *Y* in [0,5]). In total, there are 11 individual clusters. Weights over the edges indicate, by one-mode collapse, the number of shared people between two clusters. If no shared people exist, there's no edge between the pair. As before, self-loops show the number of people assigned to each cluster in the bipartite network. Based on this figure, we argue that if the edgeweight between two nodes is high, these clusters are similar since this observation translates to how many unique individuals exist in both clusters. We can exclude self-loops from this generalization since source and destination clusters in case of a self-loop are identical.

# Q2 – Open-Ended Questions

## Q2A – Structural Equivalence vs Cohesive Groups

Two nodes are *structurally equivalent* if they are exactly substitutable[1]. This suggests that the nodes in question must have identical properties with respect to some defined criteria such as having exactly the same relations to a set of vertices, same centrality metrics, etc. Actors that are structurally equivalent occupy identical roles in a given graph and are subjected to the same limitations in regards to real-life interpretations. However, due to the strength of this relation, complete structural equivalence is rare in real-life scenarios, thus, approximations are usually preferred in network analysis.

Similarly, based on node-wise definition of structural equivalence, we can argue that path and graph-based structural equivalence suggests that a pair of paths or graphs are structurally equivalent if the vertices and vertex relationships visible within the path or graph are identical. In this sense, two companies hiring employees with the exact expertise, excelling in the same field, working with the same clients and manufacturing the same products, etc. would be structurally equivalent in a commerce graph. Just like before, this type of structural equivalence is rare.

On the other hand, *cohesive groups* (or *cohesive subgroups)*, while lacking a unified definition, defines a set of vertices in a tightly-connected cluster. In general, there are four main concepts that characterizes cohesive groups[2]: 1) the mutuality of ties, 2) closeness or reachability of subgroup members, 3) frequency of ties among group members, 4) relative frequency of intra-group connections compared to outside connections. Based on these criterion, a cohesive group is similar to the small-world phenomenon, or a dense cluster: In both cases, we know that vertices lie close to each other and have many connections to other members; while ties to outside nodes are sparse. This suggests that members of cohesive groups share similar properties and are generally isolated from the outside world.

These definitions of structural equivalence (SE) and cohesive groups (CG) both suggest that detected units, be a vertex or a graph, show similar properties. However, one advantage of SE over CG is that SE does not require a clustering to be present in the graph in question. Therefore, SE is still useful even when the graph no clustering or intertwining cluster boundaries, cases where it is difficult to clearly detect and separate cluster boundaries. Fully connected or dense graphs are examples to such scenarios.

In addition, structural equivalence can be more specific in regards to the interpretation of the nature of the relationship between to actors or graphs. Consider a graph consisting of familial relationships. While clusters in such a graph would likely result in extended families, structural equivalence of two nodes would suggest they have the same role in family hierarchy such as cousins, grandparents, etc. Based on this, we can argue that clusters generalize the characteristics of their members while structural equivalence can focus on individual traits.

1 Network positions and social roles: The idea of equivalence.
https://faculty.ucr.edu/~hanneman/nettext/C12_Equivalence.html

2 Social Network Analysis, Methods and Applications.
https://www.cambridge.org/core/books/social-network-analysis/cohesive-subgroups/B3AA2C2D10B5A4D29B8D9EA8E46314CE

## Q2B – Structural Equivalence & Cohesive Groups In TwiBot-20

In the context of my term project on political bot detection, cohesive groups or clusters in general have many use cases based on a few hypotheses. We first make the distinction between automated accounts referred to as bots and other users, accounts affiliated with both individuals and corporate entities, that we refer to as real accounts. Based on these definitions, some of our hypotheses are as follows:

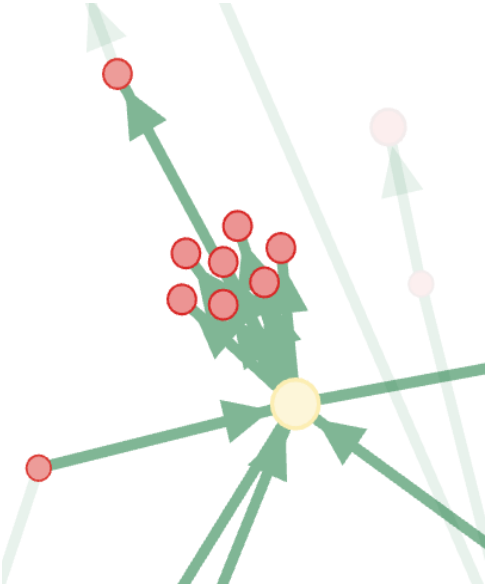


*Figure 1: Examplary cluster of bots following the same hub*

1. Bot accounts are created with for a specific set of purposes. Some of these goals include: spreading misinformation, news broadcasting, political campaigns, etc. Therefore, from the perspective of the account creator or its admin, the measure of a bot's impact increases proportional to the number of real accounts it can reach and/or influence. This means that, in the context of political bot detection, bots are more likely to follow many real accounts compared to other automated users in order to boost their exposure. For example, consider the figure shown to the left. Inside, we can see an example cluster of users that are expected to be bots (red) following the same hub which is expected to be a real account (yellow). By performing hierarchical clustering, we can capture such occurrences and verify the validity of our analysis based on ground truths.

2. Real accounts, while they may have a specific political agenda or a function to fulfill, generally are not interested in increasing their exposure since they already hold a certain political prominence. Individuals that are party leaders, publicly elected officials, law enforcement officials, journalists, etc. are examples of such people. Therefore, these individuals are likely follow a mixture of real accounts based on their political affiliations and bots which is likely to be inevitable. Combined with the previous hypothesis that suggests bots tend to center around the hubs, we can detect real accounts based on the characteristics of their neighborhoods. Computing approximate structural equivalence for hubs is a candidate method for achieving real account detection.

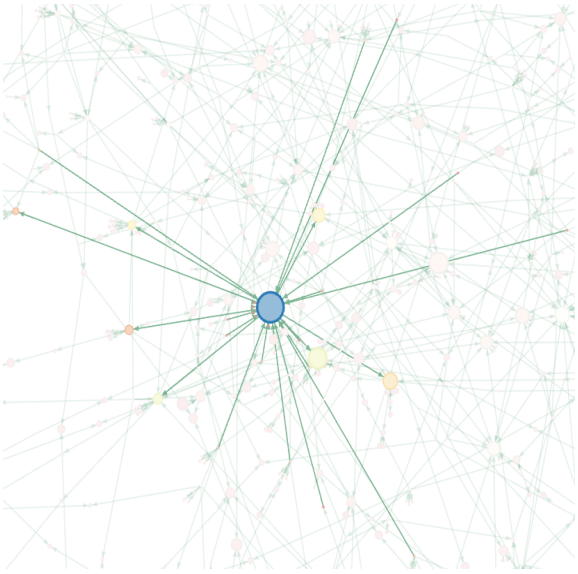


*Figure 2: Example hub suspected to be a real account.*