Xuyang Ji

COMMUNITY AND SUBGROUPS DETECTION

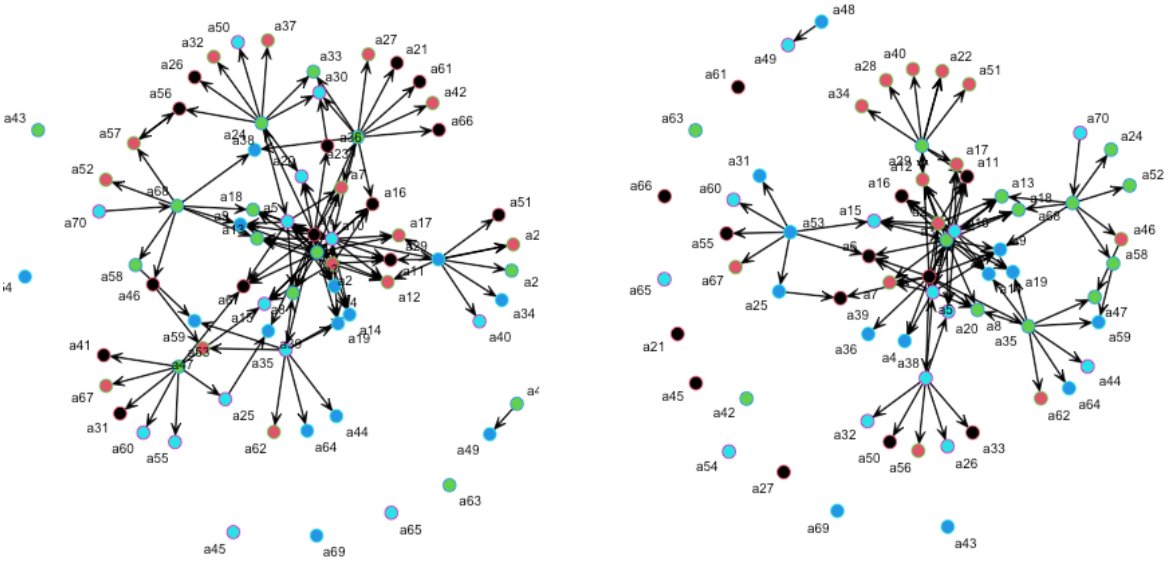
SUMMARY

In networks, it is common to find clusters of individuals who interact significantly and form distinct subgroups. For example, in a professional environment, it may manifest as a group of colleagues who share a common interest. In networks involving organizations, it can represent a collection of interacting organizations that operate as a unified entity, in which these clusters are commonly referred to as "communities" or "subgroups". The used data is extracted from the directed advice network for the midterm sunglass company project. For the simplicity of the plot, self-loops are omitted as it's more meaningful to focus on the direction of information flow and community structure. Several methods for identifying subgroups in the advice networks including, ***cutpoints and bridges, cliques, k-cores, modularity, and community detection algorithms.***

CUTPOINTS AND BRIDGES

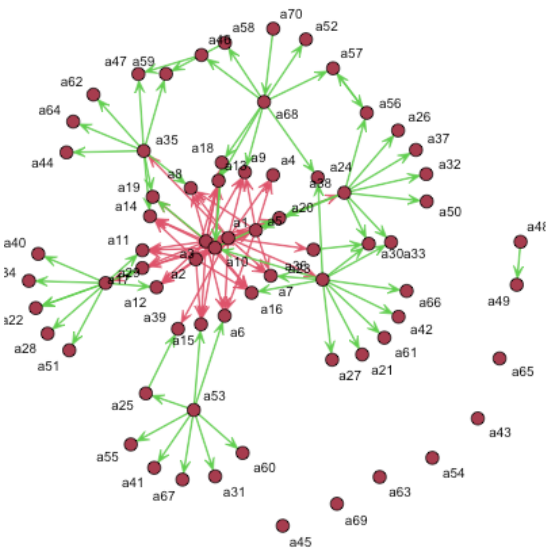
Cutpoints in a graph refer to vertices whose removal would increase the number of components in the network. They are significant when analyzing flow networks as their removal affects the connectivity properties of the graph. Cutpoints often occupy crucial positions, connecting different parts of the network. In the case of the directed advice network, a weak component rule is applied to identify cutpoints, given the network's low density.

By using the cpnet function, central nodes within the network are identified as cutpoints. Specifically, vertexes 23, 29, 35, 38, 53, and 68 are recognized as central nodes. The figures below illustrate the network before and after the removal of these central nodes. If these central nodes were dropped from the network, the result would be an increase in the number of disconnected subsets of actors. Consequently, the network would become sparser, and more components would emerge. This is shown in the figure on the right. The identification of cutpoints and understanding their impact on the network structure is crucial for determining critical positions in flow networks. By recognizing these central nodes and assessing their removal's consequences, one can gain insights into the network's connectivity and potential disruptions.



Bridges in a network are the equivalent of cutpoints for edges. An edge is considered a bridge if its removal would result in the division of one component into two. The bridge function analyzes each tie in

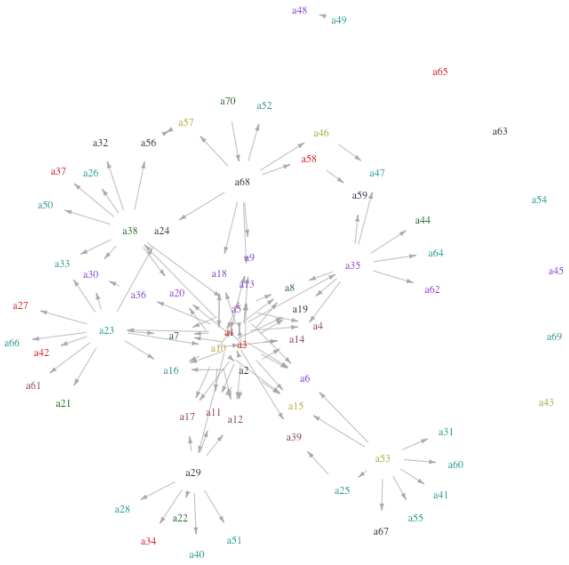
a directed network to determine if removing it changes the component count. It returns a logical vector of the same length as the number of ties, indicating which ties are bridges.



In the figure on the right, the edges highlighted in red represent the bridges within the network. These specific edges play a critical role in maintaining the connectivity of the network. If any of these bridges were to be removed, it would result in the separation of one component into two distinct components. Identifying bridges is valuable for understanding the structural integrity of the network and identifying vulnerable or critical edges. Bridges serve as key connections that maintain the flow of information or resources between different parts of the network.

CLIQUES

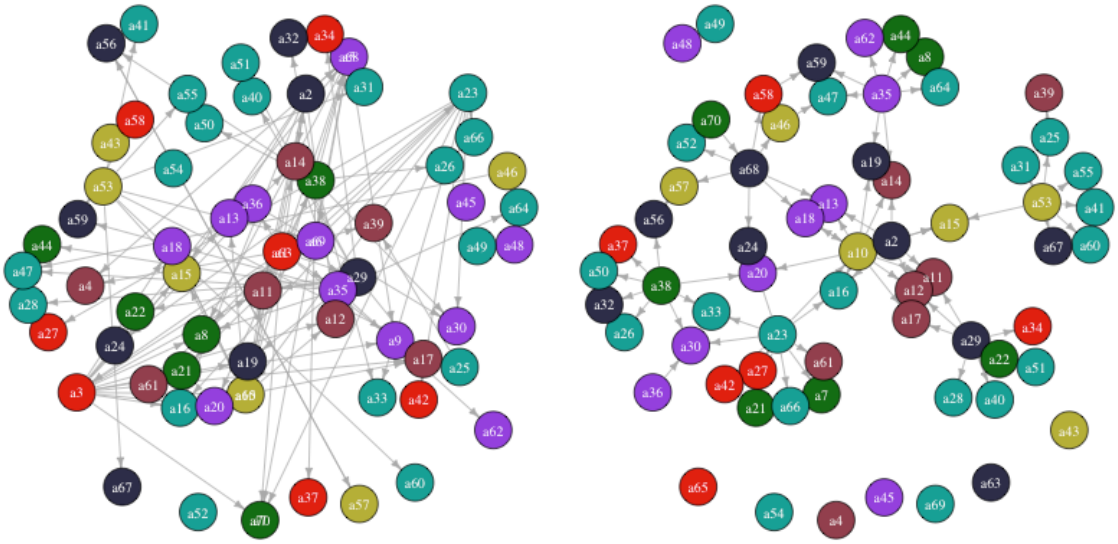
A clique in a network is a maximally complete subgraph where every node is directly connected to every other node within the subgraph. It represents a subset of nodes that have all possible ties among them. In this network analysis, the vertices represent employees, and in order to visually distinguish them based on their hobbies, the network is plotted with vertex labels colored according to their respective hobbies.



To assign the hobbies to the network nodes, a subset of rows is selected from the complete attributes dataset, specifically those rows matching the IDs of the vertices in the network. This generates a dataset called 'eAttr', which includes attributes relevant to the network nodes. The 'Hobby' attribute of each network node is assigned the corresponding values from 'eAttr'. Examining the cliques in the network, it is observed that there are a large number of cliques consisting of 70 vertices. To determine the number and characteristics of these cliques, the 'maximal.cliques' function is used. It identifies 91 cliques in the advising network, with a constraint of a minimum size of 3 nodes per clique. Analyzing the output of 'maximal.cliques', it is found that the largest clique in the network contains 4 vertices. Additionally, the six largest cliques are as follows:

Vertices 1: a9, a1, a3, a5, Vertices 2: a6, a1, a3, a5, Vertices 3: a8, a1, a3, a5, Vertices 4: a7, a1, a3, a5, Vertices 5: a5, a1, a10, a2, Vertices 6: a5, a1, a10, a3

After removing all the central vertexes, a much sparser network is shown below. Despite their usefulness in certain contexts, cliques have two major disadvantages that limit their utility in real-world social network analysis. Firstly, cliques represent a very conservative definition of a cohesive subgroup, as they require every member to be directly connected to every other member. This definition may exclude more loosely connected but still significant subgroups. Secondly, cliques are not very common in larger social networks. They are fragile and sensitive to even a single missing or added connection, making them less prevalent in real-world networks where connections may be sparser or varied. Therefore, while cliques can provide insights into specific subsets of highly connected nodes, they may not capture the full complexity and diversity of larger social networks.



K-CORES

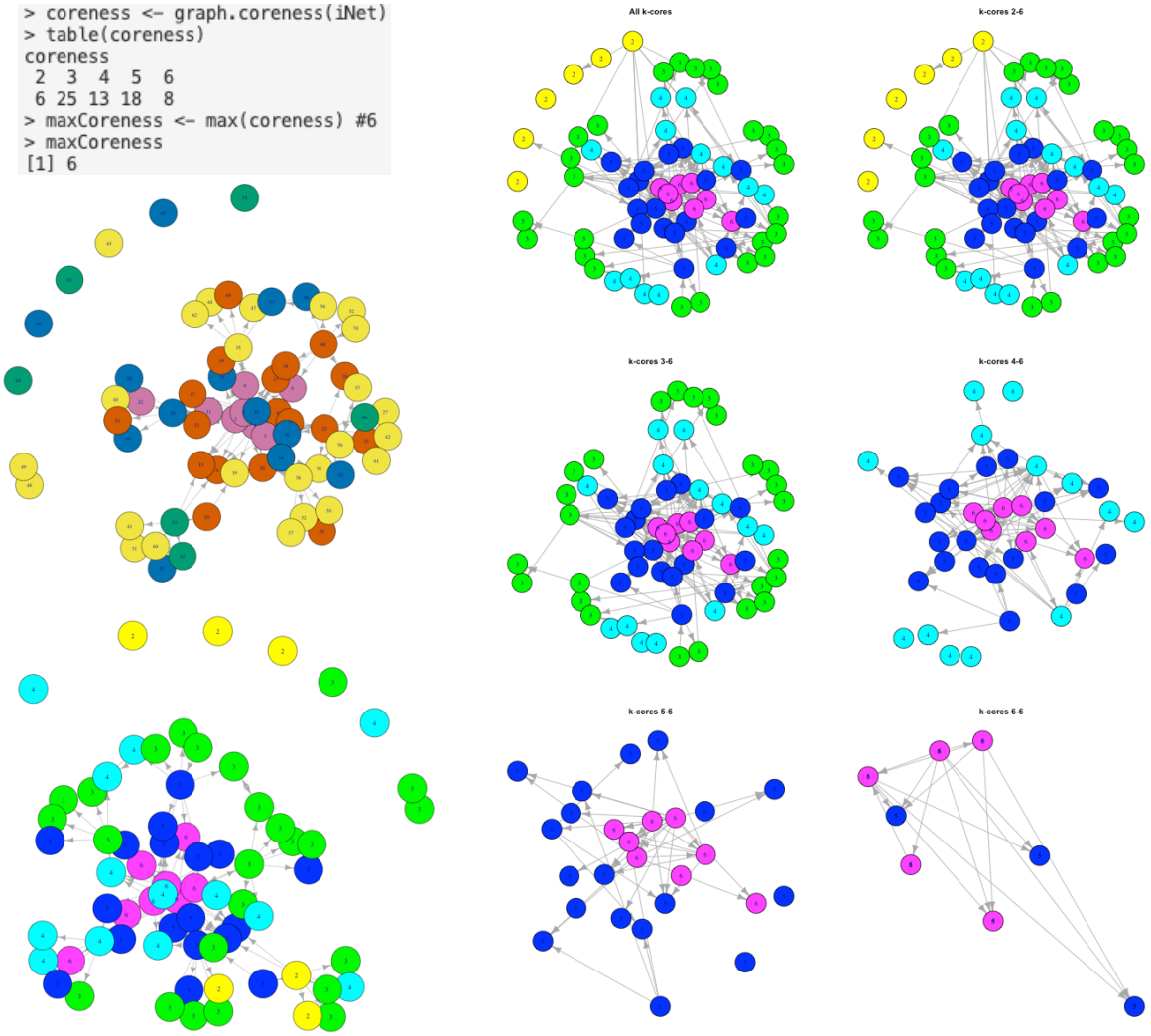
In network analysis, the k-core is a variation that addresses the rarity of cliques in observed social networks. A k-core is a maximal subgraph where each vertex is connected to at least k other vertices within the subgraph. Unlike cliques, k-cores offer several advantages: they are nested (each member of a higher k-core is also a member of a lower k-core), they do not overlap, and they are easy to identify.

To understand the k-core structure in the network, the graph's density is calculated, representing the ratio of the actual number of edges to the largest possible number of edges in the graph, assuming no multi-edges are present. This density value of 0.039 provides an indication of how interconnected the network is. The graph.coreness function is then utilized to identify the k-core structure in the network. It returns a vector that lists the highest core each vertex belongs to in the network. By examining these results, it becomes apparent that the k-cores in the network range from 2 to 6.

To gain a visual understanding of the k-core structure, the network is plotted using the k-core membership information. The nodes in the plot are labeled with their respective k-core membership values in the left below, allowing for a visual representation of the network's k-core structure. Analyzing the plot, it becomes evident that the center of the network is predominantly composed of vertices belonging to the highest k-core. This observation suggests that the core of the network is densely interconnected.

To further investigate subgroup patterns within the network, the induced.subgraph function is employed. This function enables the examination of progressively "peeling away" each lower k-core to reveal the structure and connectivity of the remaining higher k-cores.

```
> coreness <- graph.coreness(iNet)
> table(coreness)
coreness
 2  3  4  5  6
 6 25 13 18  8
> maxCoreness <- max(coreness) #6
> maxCoreness
[1] 6
```



MODURLARITY

Modularity is an essential characteristic of networks that plays a significant role in many community detection algorithms. It measures the structural properties of a network, specifically focusing on the degree of clustering observed within groups of nodes and the density of connections between these groups. Modularity provides a chance-corrected statistic that helps quantify the extent to which nodes form clusters that are denser internally and sparser externally.

In the case of the advising data, the influence of hobbies on subgroup structures becomes apparent. The node attribute in this analysis identifies the social group to which each employee belongs, such as poker, cooking, hiking, etc. The modularity score, calculated as 0.287 in this scenario, indicates that the advice social networks exhibit moderately low clustering with respect to the given hobbies grouping. This means

that there is a degree of clustering within the social groups defined by hobbies, but the density of connections between these groups is relatively higher than expected by chance.

The modularity score provides insight into the presence of community structure within the network. A higher modularity score suggests a stronger division into distinct communities, with tighter connections within the communities and sparser connections between them. Conversely, a lower modularity score indicates a more interconnected network, with less pronounced separation between communities. This knowledge can be valuable for further analysis, such as exploring the dynamics of interactions within and between social groups or identifying influential individuals bridging different communities.

COMMUNITY DETECTION ALGORITHM

For the directed network, edge-betweenness and infoMAP algorithms are used to detect underlying communities. Based on the output, the modularity score for the clustering algorithm using edge betweenness is 0.334. This indicates a relatively high level of clustering within the network, with dense connections within communities and sparse connections between them. Recall that the modularity score obtained by hobbies is 0.287, indicating a moderate level of clustering based on the actual hobby-based groups. When comparing the clustering results to the ground truth (Hobbies), the adjusted Rand index is 0.018, suggesting a moderate level of agreement between the clustering and the actual hobby-based groups.

The modularity score for the infomap clustering algorithm is 0.049. This score indicates a lower level of clustering compared to the edge betweenness algorithm. When comparing the infomap and edge betweenness clustering algorithms using the adjusted Rand index, the value obtained is 0.059. This indicates a moderate level of similarity between the two clustering results. Overall, the analysis suggests that the edge betweenness algorithm performs better than the infomap algorithm in terms of modularity and agreement with the actual hobby-based groups. However, there is still room for improvement in capturing the exact clustering structure of the network based on hobbies, as indicated by the moderate modularity scores.

```
> # Community detection algorithm for directed network
> ceb <-cluster_edge_betweenness(simplify(iNet))
> modularity(ceb)
[1] 0.3336058
> co <- cluster_infomap(simplify(iNet))
> modularity(co)
[1] 0.04994055
> compare(as.numeric(factor(V(iNet)$Hobbies)), co, method = "adjusted.rand")
[1] -0.006567825
> compare(as.numeric(factor(V(iNet)$Hobbies)), ceb, method = "adjusted.rand")
[1] 0.01751672
> compare(co,ceb, method="adjusted.rand")
[1] 0.05940629
```