Xuyang Ji

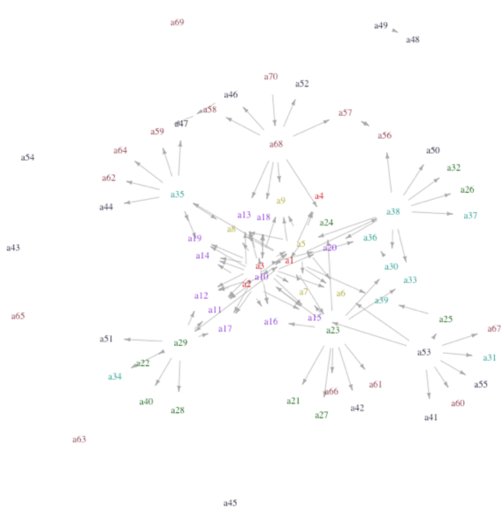
Positional Analysis in Network

Positional analysis aims to find actors or groups of actors who occupy similar positions or roles in a network. Unlike modularity-based community detection, where nodes are grouped based on their connections with each other, positional analysis groups nodes based on their similar patterns of relations. These positions can be seen as roles, as individuals in similar positions tend to perform similar functions within the network, although this is not always the case. The used data is extracted from the directed advice network for the midterm project. For the simplicity of the plot, self-loop are omitted as it's more meaningful to focus on the direction of information flow.

As positional analysis relies on identifying structurally equivalent nodes, here a more generalized notion of structural equivalence is taken, where nodes are considered to share a position or role when they have generally similar patterns of relations, rather than exact ones. Several strategies for identifying roles in the advice networks includes, Brokerage Typology, Structural Equivalence, Isomorphic Local Graphs, Block Modeling with concor, and Stochastic block modeling. A more cohesive and supportive advice network is generated with the name and position generator.

Baseline Network Plot – categorized by Department

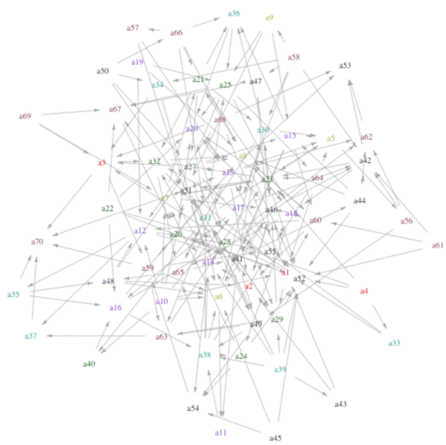
In order to visually distinguish the departments, the network is plotted with vertex labels colored according to their respective departments, as shown below. The attribute dataset is derived by selecting specific rows from the complete attributes.csv dataset, based on matching IDs with the names of vertices in the network. Furthermore, it is reordered to match the node order within the network. This process generates the 'eAttr' dataset, which exclusively includes attributes relevant to the network nodes. The 'Department' attribute of each network node is then assigned the corresponding values from 'eAttr'.



It is observed that the executive members occupy a central position within the network, indicating their influential role in the overall structure. This centrality suggests that executive members are likely to play a crucial role in facilitating information flow and decision-making processes within the organization. On the other hand, department managers tend to play the role of advice-givers, providing guidance and support to employees within their respective departments. This pattern suggests that department managers are more directly involved in fostering communication and sharing knowledge within their specific teams. These findings highlight the distinct roles and positions of executive members and department managers, shedding light on the hierarchical dynamics and communication patterns within the organization.

Random Network Generated by Name & Position Generator

Using randomly ordered name generators, the effect of name generators' relative position on the likelihood of respondents' satisficing in their response is tested. The goal is to increase the probability of edges in the random graph by modifying the probability vector generated by the *runif* function.



To enhance visual differentiation between departments, the network visualization incorporates vertex labels colored according to their respective departments. This color-coded representation provides a clear visual distinction and aids in identifying different organizational units. Finally, the 'Department' attribute of each network node is assigned the corresponding values extracted from 'eAttr', serving as the true class and enabling accurate association of departments with their respective nodes.

Algorithm 1. Brokerage

Unlike Burt's measure, the Gould-Fernandez measure below uses the department attribute to measure brokerage roles, sample output is shown below. Based on the output, node 'a1' has no incoming or outgoing ties and is not involved in any brokerage relationships. Therefore, it does not fill any of the mentioned roles (Coordinator, Itinerant Broker/Consultant, Representative, Gatekeeper, or Liaison). It serves as a standalone node within the network, in our case, the CEO of the company.

```
> ##Vignett 1. Brokerage roles&position
> bNet <- as.network(adviceE, loops =TRUE, multiple=TRUE, directed=TRUE)
> bNet %v% 'department' <- as.character(eAttr$Department)
> adviceBro<-  brokerage(bNet, 'department')
> head(adviceBro$raw.nli)
   w_I w_O b_IO b_OI b_O  t
a1   0   0    0    0   0  0
a2   1   0    5    0   0  6
a3   0   2    0    1   7 10
a4   0   0    0    0   0  0
a5   0   4    0    0   8 12
a6   0   0    0    0   0  0
```

Node 'a2' has one incoming tie weighted by advice and no outgoing ties, initiated five brokerage ties received by others, no brokerage ties received from others, and is involved in six brokerage in total. Therefore, node 'a2' fills the roles of Coordinator, Representative, and Liaison. Node 'a3' has no incoming ties, two outgoing ties weighted by advice, didn't initiate any brokerage ties received by others, received one brokerage tie from others, and is involved in ten brokerage ties in total. The total number of ties (both incoming and outgoing) is 10. Therefore, node 'a3' fills the roles of Itinerant Broker/Consultant, Gatekeeper, and Liaison.

Based on the output for the random generated network, the values of these brokerage measures for the first few nodes in the network. For example, node 'a1' has 5 outgoing ties to nodes in other departments (w_O = 5), and it serves as a liaison role between nodes from different departments (b_O = 18). Similarly, node 'a12' has 3 outgoing ties to nodes in other departments (w_O = 3) and 4 incoming ties from nodes in other departments (b_IO = 4), indicating its role as both an itinerant broker/consultant and a representative. The output provides insights into the brokerage roles and positions within the random network, shedding light on the patterns of a more connected interdepartmental connections and the influence of specific nodes in facilitating communication and collaboration between departments.

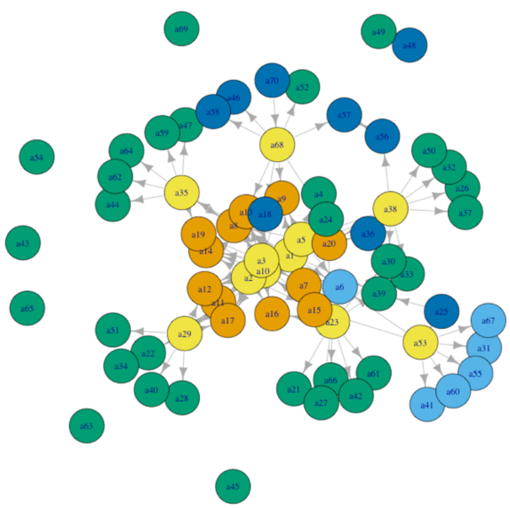
```
> ##Vignett 1. Random Network- Brokerage roles&position
> bNet <- as.network(random[,1:2], loops =TRUE, multiple=TRUE, directed=TRUE)
> bNet %v% 'department' <- as.character(eAttr$Department)
> adviceBro<-  brokerage(bNet, 'department')
> head(adviceBro$raw.nli)
    w_I w_O b_IO b_OI b_O  t
a1    0   5    0    0  18 23
a10   0   0    0    0   0  0
a11   0   0    0    0   2  2
a12   0   3    4    0  13 20
a13   0   5    0    0  29 34
a14   0   4    0   15  11 30
```

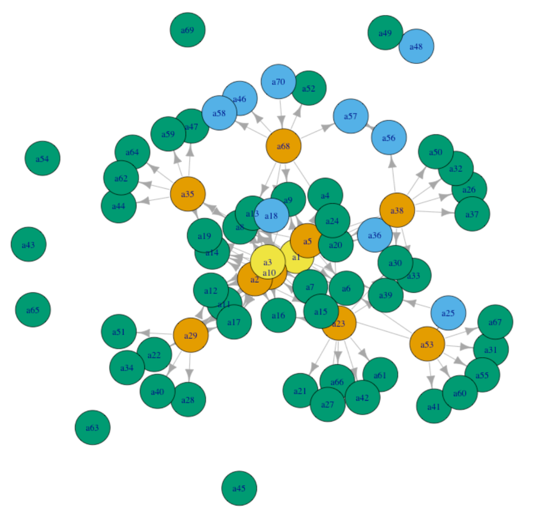
The explanations for other nodes can be derived in a similar manner by considering their respective values for `w_I`, `w_O`, `b_IO`, `b_OI`, `b_O`, and `t`. By incorporating the node explanations, we gain a clearer understanding of the specific roles and positions that each node fulfills within the network, allowing for a more nuanced analysis of their influence on information flow and network dynamics.

Algorithm 2. Structural Equivalence

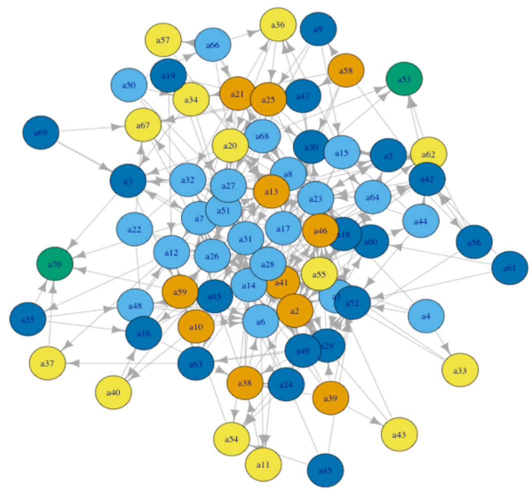
In the concept of structural equivalence, two nodes are considered structurally equivalent when they share identical relationships with all other nodes in the network. This can be assessed by comparing the sets of neighbors for each node or by computing the absolute difference between the row values of a matrix representing the relationships of the two nodes. By examining these criteria, we can determine whether two nodes exhibit structural equivalence in the network.

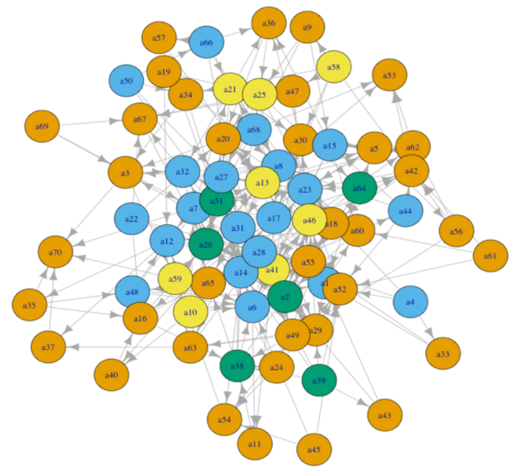
As an example, for nodes ‘a1’ and ‘a2’, the sum of their absolute differences is greater than 0, hence they are not precisely equivalent. By extrapolating every pair of nodes in the advice network then convert into a similarity matrix, sets of similar actors are identified using K-Means Clustering algorithm. The following plots show the actors who are proximately similar when set the number of clusters equal to 4 and 5.





Recall that a2 represents an executive member of the sunglass company, however, grouped together as occupying the same positions as the employees when set the number of cluster equals to 4. The reason behind it is the strict definition of structural equivalence - which relies on comparing the precise set of actors that each node is connected to. As a result, our method is confusing similarity with closeness, and as a result, it turns out, nodes that we deem structurally inequivalent can be regard as similar. On the other hand, the cluster on the left is more reasonable as it groups most executive and management members into the same clusters. The plots below are generated from the random network, as there are few isolated vertex, meaning all employees are getting advices from others.

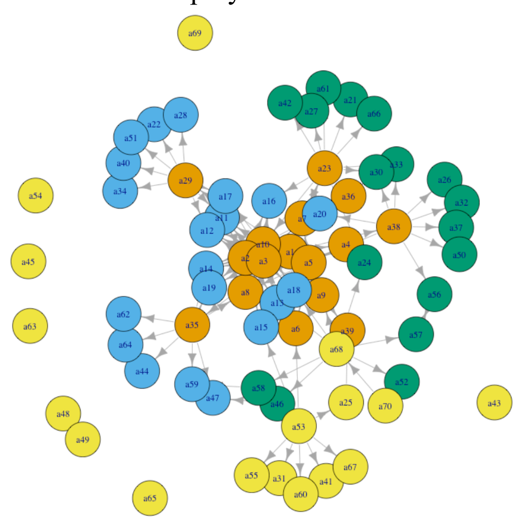




Algorithm 3. Block Modeling with CONCOR

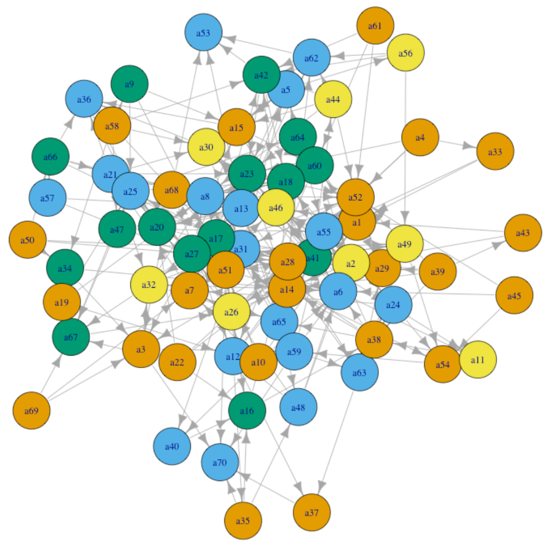
The CONCOR algorithm operates on a similar concept of equivalence as described in structural equivalent, which involves identifying structurally equivalent nodes based on their relationship patterns. However, CONCOR takes advantage of correlation and employs matrix stacking to generate a block model that incorporates multiple relationships simultaneously. The key insight of CONCOR is that, by repeatedly running correlation on the results of this initial correlation, the data will eventually converge to only -1s and 1s. The final blocks suggest 4 clusters as below, where nodes are colored by positions. Based on the output, there are some entry and medium-level employees are classified as the same as those in the executive groups.

```
> split_results_corred <- lapply(split_results, cor_many_times)
> groups_2 <- lapply(split_results_corred, function(x) x[, 1] > 0)
> split_results_again <- lapply(groups_2,
+     function(x) list(adj_mat[, names(x[x])], adj_mat[, names(x[!x])]))
> split_results_again <- unlist(split_results_again, recursive = F)
> final_blocks <- lapply(split_results_again, colnames)
> final_blocks
[[1]]
 [1] "a1"  "a10" "a5"  "a6"  "a7"  "a8"  "a9"  "a2"  "a3"  "a4"  "a23" "a29"
[13] "a35" "a36" "a38" "a39"

[[2]]
 [1] "a11" "a12" "a13" "a14" "a15" "a16" "a17" "a18" "a19" "a20" "a22" "a28"
[13] "a34" "a40" "a51" "a44" "a47" "a59" "a64" "a62"

[[3]]
 [1] "a21" "a24" "a61" "a66" "a27" "a33" "a30" "a42" "a26" "a32" "a37" "a50"
[13] "a56" "a46" "a52" "a57" "a58"

[[4]]
 [1] "a25" "a31" "a41" "a43" "a45" "a48" "a49" "a53" "a55" "a60" "a67" "a54"
[13] "a63" "a65" "a68" "a69" "a70"
```



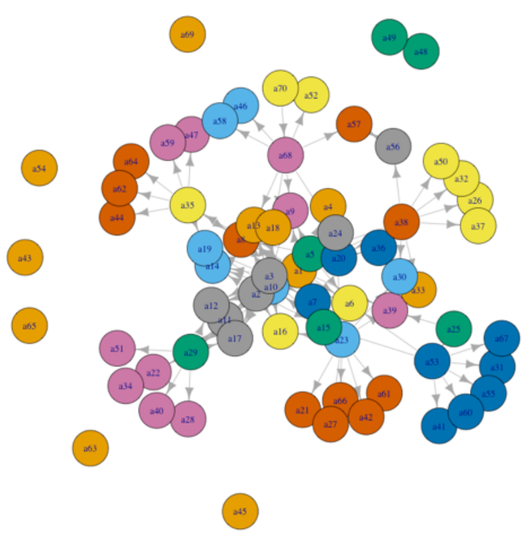
*The plots below shows the output for random network.

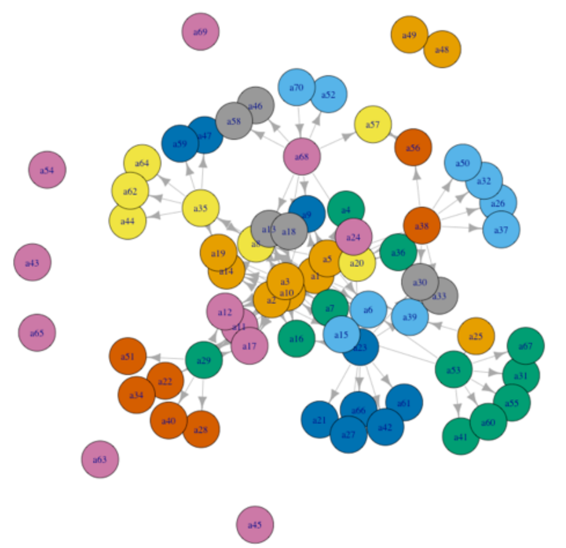
```
> groups_2 <- lapply(split_results_corred, function(x) x[, 1] > 0)
> split_results_again <- lapply(groups_2,
+     function(x) list(adj_mat[, names(x[x])], adj_mat[, names(x[!x])]))
> split_results_again <- unlist(split_results_again, recursive = F)
> final_blocks <- lapply(split_results_again, colnames)
> final_blocks
[[1]]
 [1] "a69" "a7"  "a3"  "a68" "a1"  "a14" "a37" "a29" "a28" "a61" "a52" "a58"
[13] "a51" "a54" "a50" "a38" "a45" "a15" "a43" "a39" "a35" "a33" "a22" "a19"
[25] "a10" "a4"

[[2]]
 [1] "a53" "a21" "a12" "a57" "a36" "a31" "a8"  "a65" "a48" "a55" "a5"  "a63"
[13] "a62" "a59" "a70" "a13" "a6"  "a40" "a24" "a25"

[[3]]
 [1] "a67" "a66" "a41" "a64" "a23" "a60" "a42" "a47" "a34" "a17" "a9"  "a18"
[13] "a27" "a20" "a16"

[[4]]
[1] "a26" "a2"  "a30" "a56" "a49" "a46" "a44" "a11" "a32"
```



Algorithm 4. Isomorphic Local Graphs

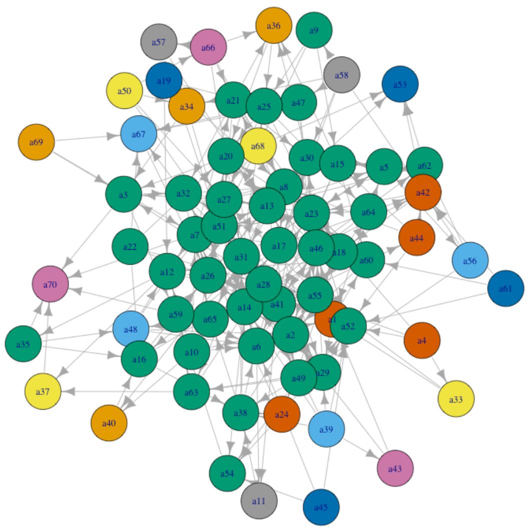
To identify actors who are precise structural equivalent, isomorphic is used by relaxing the condition that nodes be tied to precisely the same set of nodes by defining them as structurally equivalent as long as their local neighborhoods are automorphic. The underlying algorithm, bliss, essentially permutes the matrices of the two networks it is comparing to see if, under any of the different permutations, the two matrices are equivalent. In other words, Isomorphic local graph analysis focuses on identifying similar subgraphs within a larger network, where subgraphs are smaller sections of the network that exhibit identical structural characteristics.





By setting the neighborhood size to 2 and 3, and loop through the different local neighborhoods and evaluate whether they are automorphic, the output plot on the right shows nodes who share isomorphic neighborhoods using 3 clustering is much more reasonable.

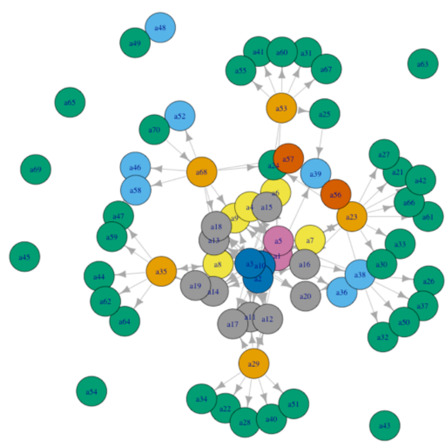
The following plot shows the output for the random network when the neighborhood size is set to 3. Where the algorithm groups most employees as the same cluster. Analyzing a random network within an isomorphic local graph context allows for the examination of structural similarities and patterns within the advice network. Examining the isomorphic local graph properties of a random network can provide insights into the underlying mechanisms that shape its structure. It can reveal if the network contains repeated patterns or clusters of nodes with similar connectivity, highlighting potential organizational or functional modules within the network.

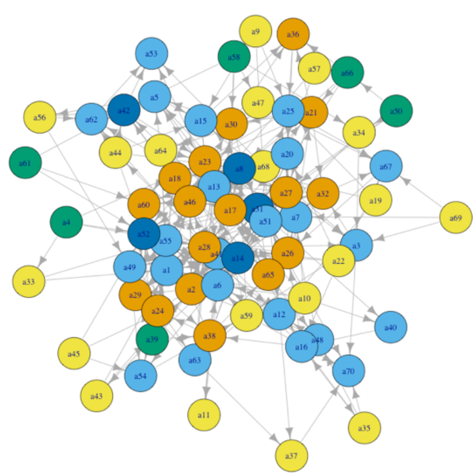


Comparing to the true plot, employees within each class/department tend to interact more frequently with others from the same class than with those from a different class. However, when we generate a random network, there is a possibility that by chance, individuals from different classes are connected more frequently than expected. This can lead to the formation of a subgroup that includes people from different departments, which is not reflective of the true social structure. The output highlights the importance of distinguishing between a random network and the true social network. Random networks can produce patterns that differ significantly from the actual social structure, emphasizing the need for careful analysis and interpretation when studying real-world networks.

Algorithm 5. Stochastic Block Models

In an SBM, the network is partitioned into a predefined number of blocks or communities. Each node is assigned to one of these blocks, and the goal is to uncover the underlying block structure and the connections between them. The connections within a block are typically denser compared to connections between blocks. Rather than sharing the same set of nodes, actors that share a role or position will have the same probability of being attached to all other alters in the network. Under this operationalization, equivalence is not absolute, but probabilistic. Based on the output of true scenarios on the left, it can be concluded that the community structure captured by the model accurately represents the underlying social dynamics in the advice network.





Xuyang Ji

The output on the right shows the SBMs output using random network, which serves as a benchmark for validation. By analyzing SBMs on random networks with known properties, we can assess the performance and effectiveness of the SBM inference methods. In this case, the SBM failed to accurately recovers the block structure and connection probabilities in the random network, it demonstrates the model's validity and the inefficacy of the inference technique.