Analyzing Twitter Interactions of the 117th United States Congress

Ivy Li, Tingting Liang, Sam Ream

June 7, 2024

1 Introduction

The `congress-Twitter` dataset [1][2] is a directed network of 475 nodes, each representing the Twitter (rebranded to X in 2023) account of a single member of the 117th United States Congress, and 13,289 edges representing interactions (i.e., retweets, quote tweets, replies, or mentions of another member's tweet) of these accounts between February 9, 2022 and June 9, 2022. Additionally, each edge contains an associated weight, representing the empirical probability of the interaction occurring (based upon the total number of times the interaction occurred). To enhance our analysis of the dataset, additional vertex attributes were added separately. These included the name of the congressperson, their respective chamber, party affiliation, state, and Census Bureau-designated region of the state.

In this analysis, we will examine the nature of the 117th US Congress' Twitter interactions: in particular, we will investigate how a member's chamber, party affiliation, and the geographical location of their district impacts which members they interact with on Twitter. Moreover, we will look for the existence of community structures in the Twitter interaction network, and whether these structures—if they exist—are correlated with chamber, party affiliation, or geographical location.

2 Methodology

The bulk majority of our analysis is performed with R, using the `igraph`, `Matrix`, `tidyverse`, and `rjson` packages.

To input the `congress-Twitter` data into R, the `fromJSON()` function from the `rjson` file was used to read the data in from a `json` file. The input data was then transformed into a dataframe, from which a list of edges was constructed. This list of edges was then converted into a directed `igraph` object using the `graph.edgelist()` function from the `igraph` package. Edge weights and appropriate node attributes were then added to this graph from a separately constructed dataframe. Finally, the graph was converted into an undirected and unweighted graph using the `as.undirected()` function. For detailed code of this process, see Listing 1 and Listing 2. The visualization and analysis of the network was then performed primarily using `igraph` functions.

3 Analysis

3.1 Subgraphs

To visualize and detect any interesting features of the network, we produced 4 induced sub-graphs of 75 randomly sampled nodes. (We settled on 75 nodes over a larger number of sampled nodes due to the limited size of our network, which had 475 nodes overall, and to improve the clarity of visualizations.) For each subgraph, 3 visualizations were produced: one using the `layout_with_kk()` layout showing inter-party interactions and node degrees (see Figure 2), one using the `layout_on_grid()` layout showing inter-chamber interactions (see Figure 3), and one using the `layout_in_circle()` layout showing interactions between regions (see Figure 4).

All three visualizations for the first induced subgraph are shown in Figure 1. In the leftmost plot of Figure 1, we see that edges between parties (colored purple) tend to be less common overall compared to edges within each party: this trend continues in Figure 2 and indicates that congressmembers tend to interact with members of their own party over members of the opposing party on Twitter. In the central plot of Figure 3, we see that edges between the two congressional chambers (colored green) are less common than edges within each chamber. This is also true of the other plots in Figure 3 and indicates that congressmembers may have some preference for interacting with other members in their chamber over members of the opposing chamber on Twitter. In the rightmost plot of Figure 1, we see see that edges between regions (colored blue) are much more common than edges that originate and terminate in the same region. The same can be said of the remaining plots of Figure 4, indicating that congressmembers do not show a particular preference for interacting with other members from the same region over members from other regions.

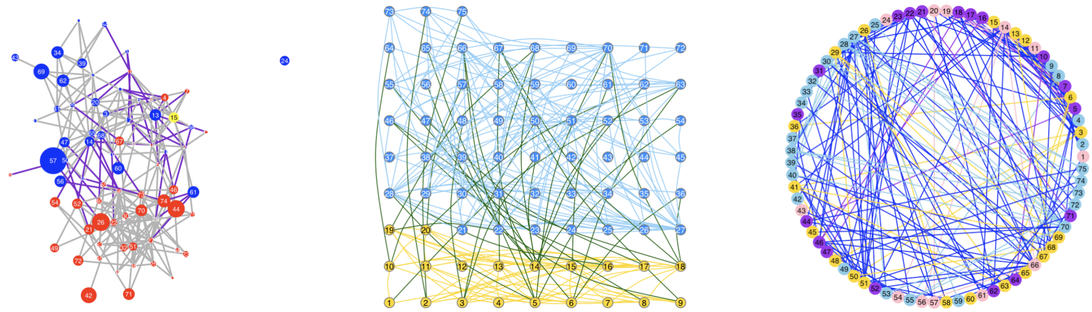

Figure 1: Visualizations of the first induced subgraph. The leftmost figure shows nodes sized by their respective degree and colored by party affiliation, along with purple edges indicating edges between parties. The center figure shows nodes colored based on chamber. Edges that originate and terminate within the same chamber are colored the same as the nodes, while edges that terminate in the opposing chamber are colored green. The rightmost figure shows nodes colored by region. Edges that originate and terminate in the same region are colored the same as their respective nodes, while edges that terminate in a region different from their original region are colored blue.

3.2 Network Metrics

Our analysis of network metrics for the `congress-Twitter` dataset was performed on both the directed and undirected versions of the network. The results of this analysis is summarized in Table 1. The directed edge density is 0.05901843, while the undirected edge density is 0.09080169: this indicates a fairly well connected network. The reciprocity of the graph is 0.461469, which means less than half of the interactions aren't reciprocated: that is, if a congressmember interacts with another member's tweets, only about half the time does the other member also interact with that member's tweets. This is rather surprising, as we were expecting more mutual interactions between congressmembers. The transitivity of the network is 0.269535, which indicates a moderate amount of connectivity, i.e. the graph is not very densely clustered. The directed mean distance between nodes is 2.357161 and the undirected mean distance between nodes is 2.063886. That is, on average, any pair of nodes is separated by 2 edges, indicating that the network is well connected. The diameter of the directed network is 6, in line with the commonly seen "6 degrees of separation" in many networks. The farther nodes are two outliers, Claudia Tenney (A Repubican House member from New York) and Gregoro Sablan (a Democratic delegate to the House of Representatives from the Northern Mariana Islands). The undirected network is weakly connected, which means a path exists between any pair of nodes we get that our network, but not necessarily in both directions. There are no articulation points in the network. Kevin McCarthy (Republican representative for California) was the congressperson with the highest degree of 284 (127 in and 157 out) and Gregoro Sablan (Democratic representative from the Northern Mariana Islands) was the congress person with the lowest degree of 2 (1 in and 1 out).

Network Metric	Directed	Undirected
Edge Density	0.05901843	0.09080169
Transitivity	0.269535	0.269535
Reciprocity	0.461469	———
Average Path Length	2.357161	2.063886
Diameter	6	4
Connected	Weakly	Yes
Articulation Points	0	0
Highest Degree	284	214
Lowest Degree	2	2

Table 1: Various network metrics for the directed and undirected version of the `congress-Twitter` network.

Table 2 shows network metrics for eight different induced subgraphs. The subgraphs consist of all

Democrats in the network, all Republicans in the network, all senators in the network, and all representatives in the network, with a directed and undirected version of each. The edge density and transitivity for all eight subgraphs is higher than the edge density for the overall network, indicating a higher degree of connectivity within each group relative to the overall network. Similarly, the reciprocity of the four directed subgraphs is higher than the reciprocity of the overall network, indicating that congressmembers prefer to mutually interact with members from their own party and chamber. Finally, the average path length and diameter for each subgraph is lower than their respective counterparts for the overall network, again indicating a higher degree of connnectivity within each group relative to the overall network.

Network Metric	Democrats	Republicans	Senate	House
Edge Density (D)	0.09484285	0.1091324	0.1866937	0.06913592
Edge Density (U)	0.1447859	0.1648402	0.2747253	0.1062704
Transitivity	0.3058064	0.3418851	0.3995948	0.2975693
Reciprocity	0.4734122	0.4895397	0.5284709	0.4628769
Average Path Length (D)	2.179667	2.129141	1.970414	2.312977
Average Path Length (U)	1.934927	1.891458	1.741997	2.034011
Diameter (D)	5	5	4	6
Diameter (U)	4	3	3	4

Table 2: Network metrics for various induced subgraphs of the `congress-Twitter` network generated based on various vertex attributes. Metrics evaluated on directed subgraphs are indicated by a (D) and metrics evaluated on undirected subgraphs are indicated by a (U).

3.3 Community Detection

We ran a total of 8 different community detection algorithms on the `congress-Twitter` dataset, the results of which are displayed in Table 3. We see that the majority of the algorithms return somewhere between 2 and 5 groups, with the exception of the `cluster_edge_betweenness()` algorithm, which returned a total of 13 groups. Upon closer inspection however, all but 2 of these groups were of size 1, and when we consider only groups of size larger than 2, we find that the algorithms all return between 2 and 4 groups, with the majority finding 3 groups.

Community Detection Algorithm	No. of Groups	No. of Groups of Size > 2	Modularity
`cluster_fast_greedy()`	3	3	0.4003141
`cluster_edge_betweenness()`	13	2	0.3441079
`cluster_infomap()`	3	3	0.4099317
`cluster_label_prop()`	2	2	0.3409326
`cluster_leading_eigen()`	4	4	0.3847427
`cluster_louvain()`	4	3	0.4131818
`cluster_spinglass()`	5	3	0.4132002
`cluster_walktrap()`	3	3	0.4113495

Table 3: Results of various community detection algorithms from the `igraph` package when applied to the `congress-Twitter` network.

The modularity of these communities found by the community detection algorithms all range from 0.34 to 0.41, which means that there is some community structure present. To examine whether any of the network attributes (chamber, party affiliation, region, etc.) is the basis for the formation of these groups, visualization of the communities returned by each algorithm were made. Figure 5 shows a plot of the communities returned by the `cluster_spinglass()` algorithm. From the plot, we can see the three main groups the network is divided into: these appear to be the House Democrats (circular nodes with a green center and blue border), the House Republicans (circular nodes with a turquoise center and red border), and the Senate (square nodes with a yellow center and blue or red border). So it appears that the community structure in the `congress-Twitter` network is based on the party and chamber that a particular node belongs to.

The `cluster_leading_eigen()` algorithm was the only algorithm that returned 4 groups of size larger than 2. If we visualize the results of the algorithm, as in Figure 6, we see that the four communities appear to be: (1) the House Democrats (circular nodes with a green center and blue border), the House Republicans (circular nodes with a turquoise center and red border), the Senate Democrats (square nodes with a purple center and blue border), and the Senate Republicans (square nodes with a yellow center and red border). Just as with the `cluster_spinglass()` algorithm, it seems that party and chamber are the basis of communities in this network.

The `igraph` package also has functionality that permits the manual creation of communities using the `make_clusters()` function. Using this function, we created communities based on particular node attributes and evaluated the modularity of the resulting groups, the results of which are shown in Table 4. For instance, dividing the nodes based on their respective states yields 54 groups (there are more than 50 groups here, as representatives from several US territories were included in the dataset) with a modularity of 0.0860. Dividing the nodes based on their respective Census Bureau-designated regions yields 5 groups with a modularity of 0.107. Both of these are low modularity scores, indicating that neither state nor region are a large factor in the community structure of the network. On the other hand, dividing the nodes into two groups by chamber and then by party (the three Independents in the dataset, Kyrsten Sinema, Bernie Sanders, and Angus King all caucus with the Democrats and were included with the Democrats for this analysis) yields modularity scores of 0.163 and 0.339 respectively, indicating that chamber and party are more significant factors in the community structure of the network than region, with party being the primary factor (having the largest modularity score). When chamber and party are taken into consideration together to yield 4 groups, the resulting modularity score is 0.384. This is comparable to the performance of the community detection algorithms in Table 3, and conclusively indicates that the community structure in the network is based upon the party and chamber that a node belongs to.

Attribute(s)	Number of Groups	Modularity
State	54	0.08597953
Region	5	0.1067211
Chamber	2	0.1630579
Party	2	0.3393424
Chamber and Party	4	0.3837733

Table 4: Modularity scores for manually created communities based on various vertex attributes of the `congress-Twitter` network using the `make_clusters()` function.

3.4 Adjacency Matrix

Figure 7 displays the adjacency matrix for the `congress-Twitter` network with nodes ordered by the chamber they belong to. In upper left corner of the visualization, we see a dark square corresponding to interactions between Senate members, while interactions between House members are represented in the larger and slightly lighter square in the bottom right corner. Both of these regions have a higher density of edges than in the rectangular regions outside at the top right and bottom left, indicating that members of Congress favor intra-chamber interactions over inter-chamber interactions (i.e., members of the Senate interact more with other senators, and the same for members of the House).

The adjacency matrix for the network can also be organized by the state each node belongs to, as shown in Figure 8. We can make out a diagonal running from the upper left corner to the bottom right corner of the plot: this represents edges between congressmembers from the same state. The diagonal is darker than the surrounding regions, which indicates members of Congress tend to favor intra-state interactions over inter-state interactions (i.e. a member from Iowa is more likely to interact with other Iowan congressmembers).

The adjacency matrix can also be ordered by the region that each node belongs to, as shown in Figure 9. In this plot, we see four very faint squares along the main diagonal, representing interactions between states. These squares are only slightly darker than the surrounding regions of the plot, indicating that there is a slight preference for intra-region interactions amongst US congressmembers.

Figure 10 shows the adjacency matrix for the network organized by party, with organization within each party by chamber. In the visualization, we can see four smaller squares at the upper left corner of each quadrant: these represent interactions between members of the Senate, with the two along the main diagonal from the upper left corner to the bottom right corner representing intra-party interactions within the Senate, and the remaining two representing inter-party interactions within the Senate. The two larger squares on the main diagonal represent intra-party interactions within the House. We see that the squares along the main diagonal are darker than the surrounding regions, indicating that intra-party interactions within each chamber are favored over over intra-party interactions across chambers. Additionally, the two smaller squares in the upper left corners in the upper right and lower left quadrants are also slightly darker than the surrounding regions, indicating that senators prefer to interact more with other senators, even when they are not from the same party. Overall, this visualization further backs up the existence of community structures in the network based on party and chamber.

4 Results

Based on our network visualizations, we see that intra-party interactions tend to be favored over inter-party interactions (Figure 2), intra-chamber interactions are favored over inter-chamber interactions (Figure 3), and there is no preference for intra-region interactions over inter-region interactions (Figure 4). Calculating network metrics for subgraphs based on these attributes confirms these conclusions: in Table 2, we see that the edge density and transitivity of the subgraphs is higher than the overall edge density and transitivity in Table 1, while the average path length and diameter tend to be lower. Performing community detection on the network further bolsters this conclusion: in Figures 5 and 6, we see distinct communities returned by the `cluster_spinglass()` and `cluster_leading_eigen()` algorithms based on the chamber and party of each node. Similarly, the adjacency matrices (Figures 7, 8, 9, and 10) highlight congressmembers preference for intra-party, intra-chamber, intra-state, and intra-region interactions.

We have seen that congress members are more likely to interact with other congress members on Twitter if they are both in the same chamber and/or party. Conversely, members are more likely to tweet other members between regions rather than within. In terms of network metrics, for being a larger social network, the edge density of about 0.0590 is relatively high. Having a lower transitivity of about 0.2695 is not surprising considering that congressmen have their own political objectives but will occasionally team up to accomplish certain goals. The reciprocity of about 0.4615 is surprisingly low, as we'd expect more congressmen to reply to each other and have political discourse on Twitter. The average path length of about 2.3572 is not surprising considering the size of our network and how intertwined the congressmen are. Additionally, having a diameter of 6 is not surprising considering we have a couple outliers (Claudia Tenney and Gregoro Sablan). There are also no articulation points present, which aligns with the egalitarian structure of congress.

5 Conclusion

Overall, we have found community structures in our network based on the chamber and party attributes. Congressmembers tend to prefer interacting with members from their own party, chamber, and state, with a slight preference for members from their own region. One surprising aspect of the network was the reciprocity, which we found to be close to 0.5: we expected congressmembers to mutually interact with each other with a greater frequency. However, other aspects of the network were more in line with expectations: big names like Kevin McCarthy and Nancy Pelosi were among the highest degree nodes in the network.

In this analysis, we did not take into account the edge weights of the network. A future analysis could incorporate these weights—which represent the empirical probability of interactions between nodes—to determine what further insights may be gleaned from their inclusion. Another direction for future analysis could incorporate ideology and political climate to investigate if certain events impact the interaction of congressmembers on Twitter.

6 References

[1] C.G. Fink, K. Fullin, G. Gutierrez, N. Omodt, S. Zinnecker, G. Sprint, and S. McCulloch: A centrality measure for quantifying spread on weighted, directed networks. Physica A, 2023.

[2] C.G. Fink, N. Omodt, S. Zinnecker, and G. Sprint: A Congressional Twitter network dataset quantifying pairwise probability of influence. Data in Brief, 2023.

7 Appendix

7.1 Figures

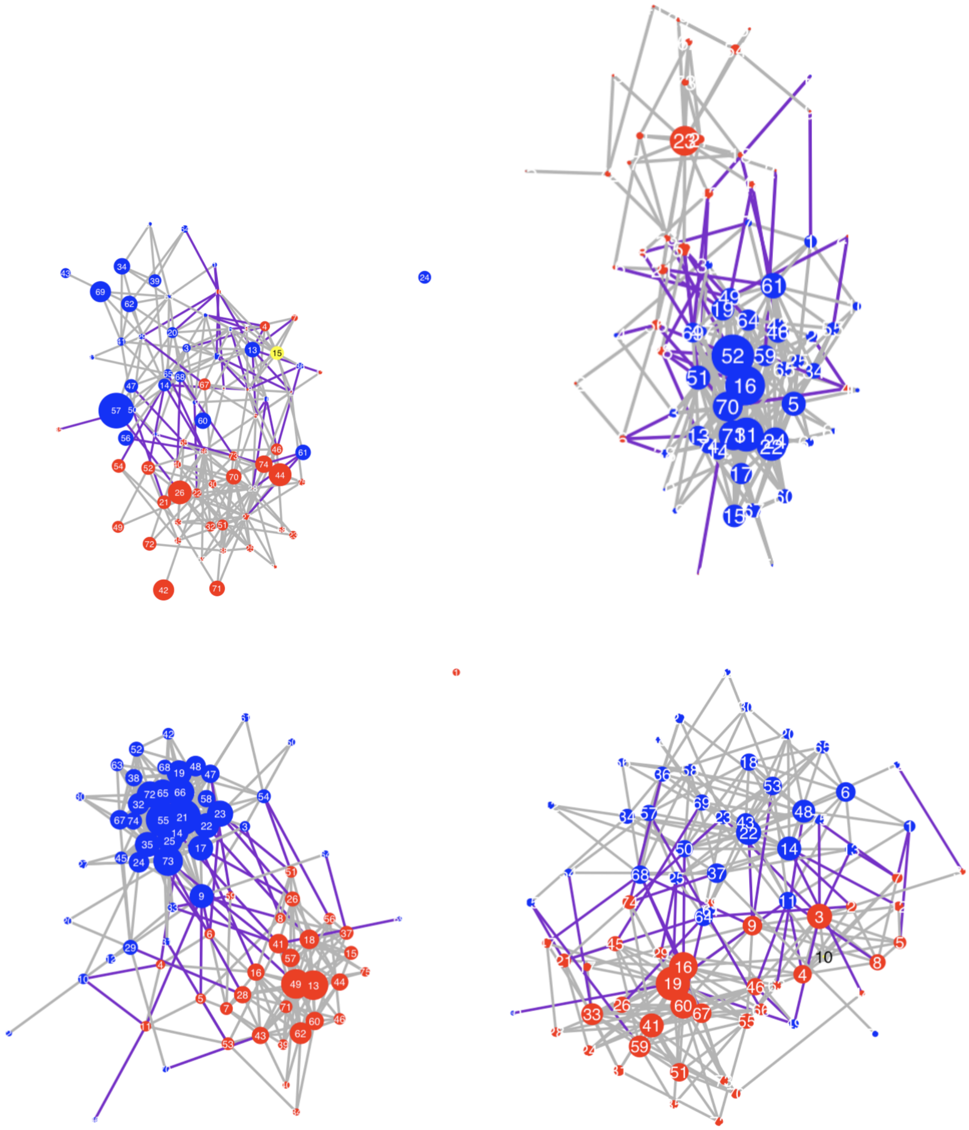

Figure 2: The images above shows the plots of 4 induced sub-graphs using `layout_with_kk()` colored by party affiliations. The plots shows the inter-party interactions. The nodes are colored in red representing Republicans, blue representing Democrats and yellow representing Independents. Edges between nodes in the same party are in gray, and edges in purple represents connection across parties. The plots shows 2 clusters split by Democrates and Republican across samples which indicates there is strong connections within parties. Some node also interacts outside of their party. Node size are determine by degree. We observed an even distribution of interactions across parties. This suggests that members from both parties engage in similar levels of interaction within the network.

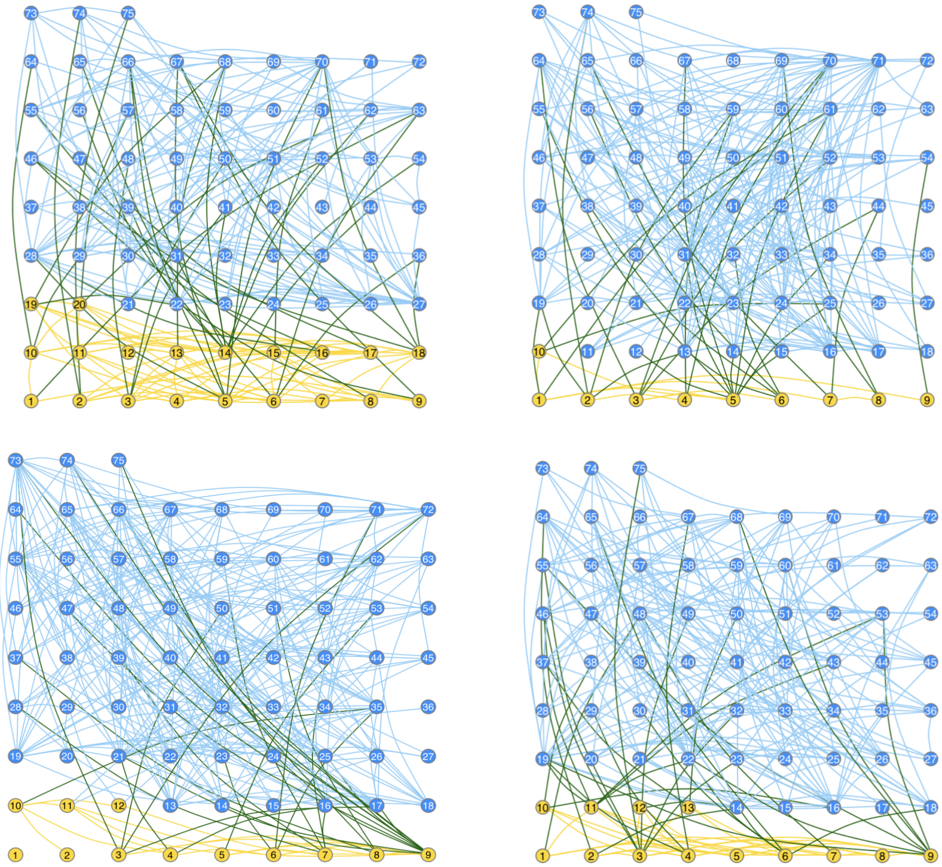

Figure 3: The images above shows the plots of 4 induced sub-graphs using `layout_on_grid()` showing interactions between chambers. The nodes are colored by chamber, where blue nodes represents the the House of Representatives and gold nodes represents the Senate. Edges between nodes in the same chamber are colored according to their node colors respectively. Edges between nodes of different chambers are represented in dark green. The plot shows 2 clusters split by chambers meaning there are more interactions between members of the same chamber than across. However, interaction across chambers also exists.

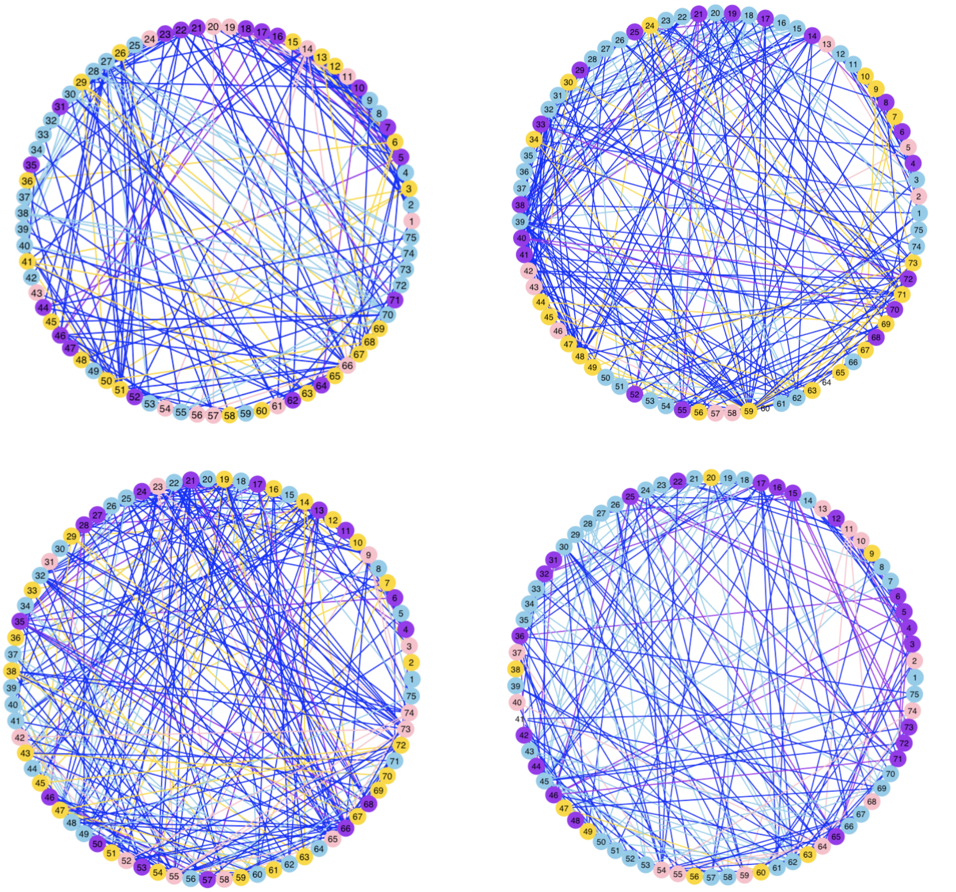

Figure 4: The images above shows the plots of 4 induced sub-graphs using `layout_in_cicle()` showing interactions between regions. Nodes are colored by regions(Northeast, Midwest, West, and South). Edges between nodes of the same region is colored by their node color. The Blue edges between nodes represents the interactions between regions. The plots across different samples shows that there's more interactions between regions than within.

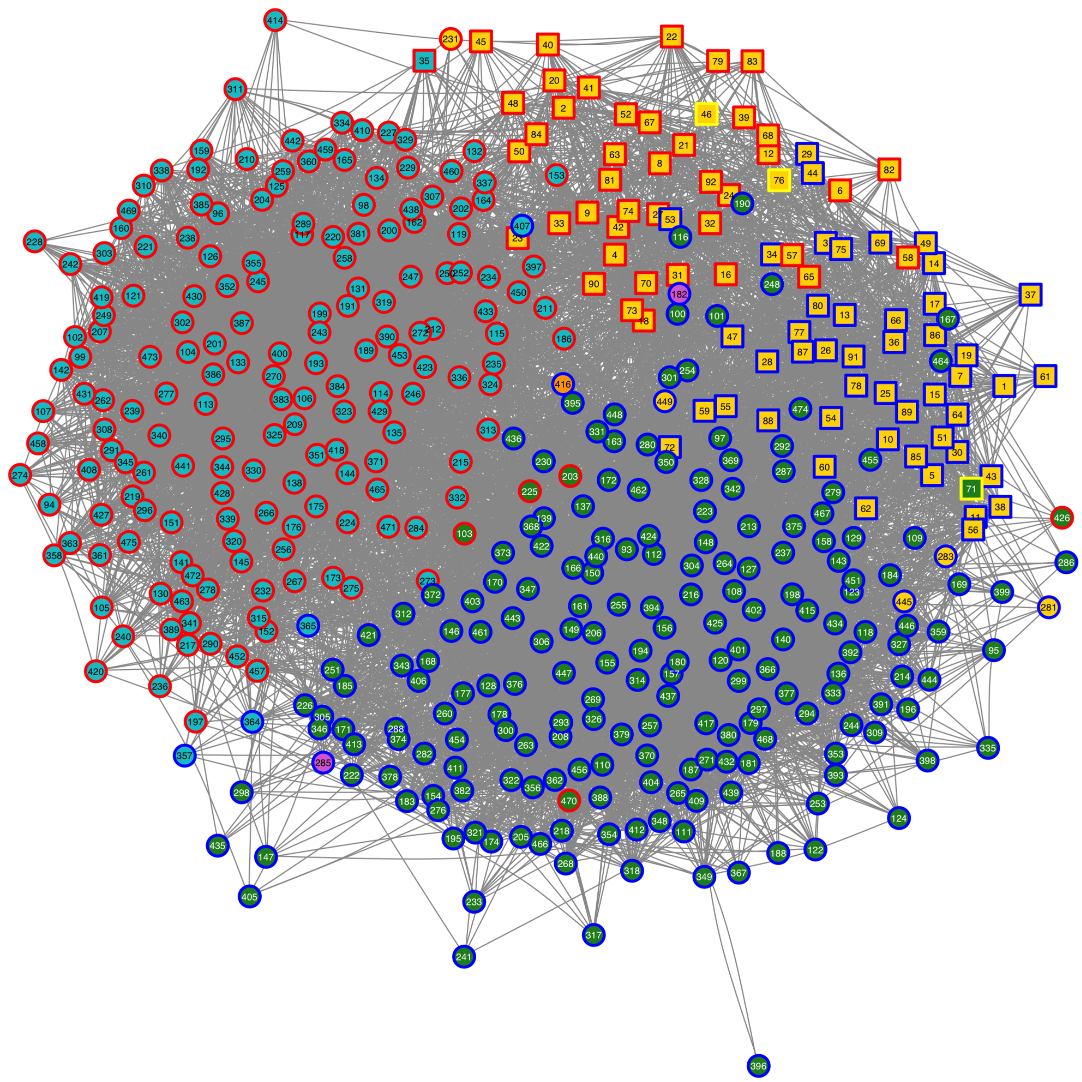

Figure 5: Plot of the `congress-Twitter` network with communities found using `cluster_spinglass()`. The community a node belongs to is denoted by the central color of the node: the frame color denotes the party affiliation for each node (blue for Democrats, yellow for Independents, and red for Republicans), and the shape denotes the chamber for each node (square for the Senate and circular for the House). Overall, three distinct communities were found: one corresponding to all parties in the Senate, and two corresponding to each of the major parties in the House.

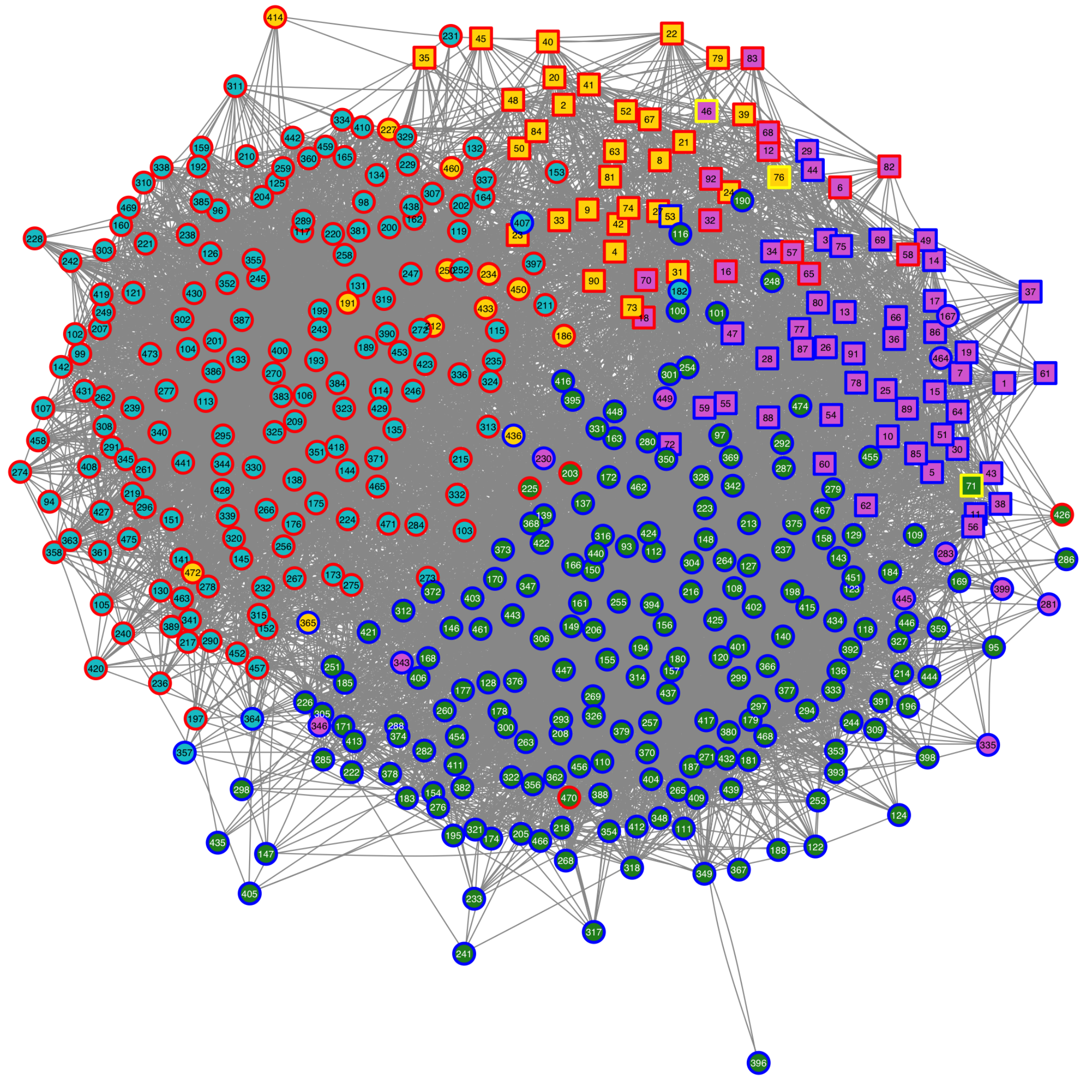

Figure 6: Plot of the `congress-Twitter` network with communities found using `cluster_leading_eigen()`. The community a node belongs to is denoted by the central color of the node: the frame color denotes the party affiliation for each node (blue for Democrats, yellow for Independents, and red for Republicans), and the shape denotes the chamber for each node (square for the Senate and circular for the House). Overall, four distinct communities were found, primarily divided by party affiliation and chamber.

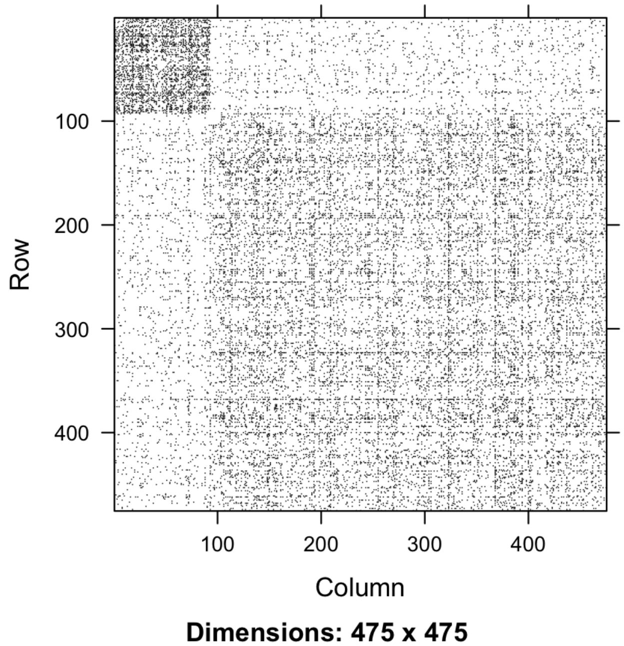

Figure 7: Adjacency matrix for the congress-Twitter network with nodes ordered based on chamber.

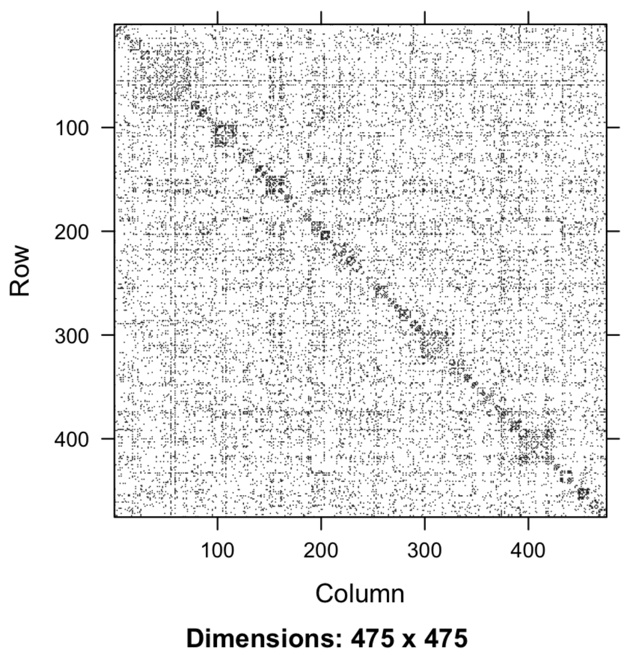

Figure 8: Adjacency matrix for the congress-Twitter network with nodes ordered alphabetically based on state.

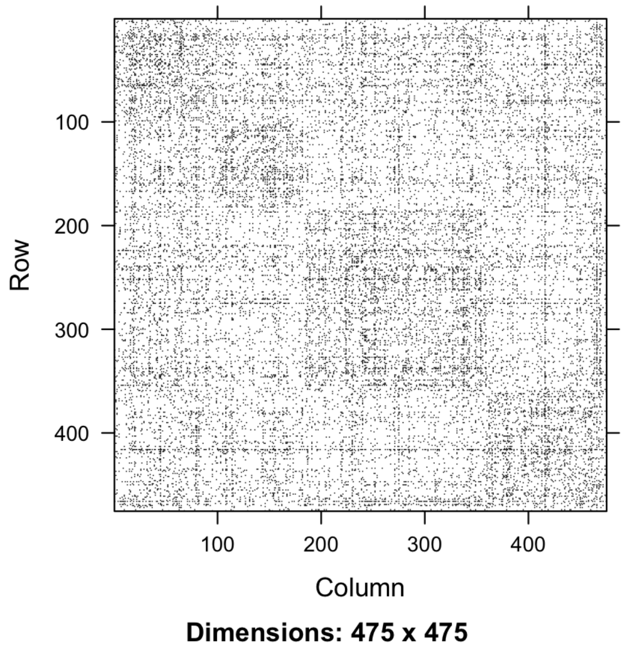

Figure 9: Adjacency matrix for the `congress-Twitter` network with nodes ordered by their Census-Bureau designated region (Midwest, Northeast, South, and West: US territories were grouped together in a separate category).

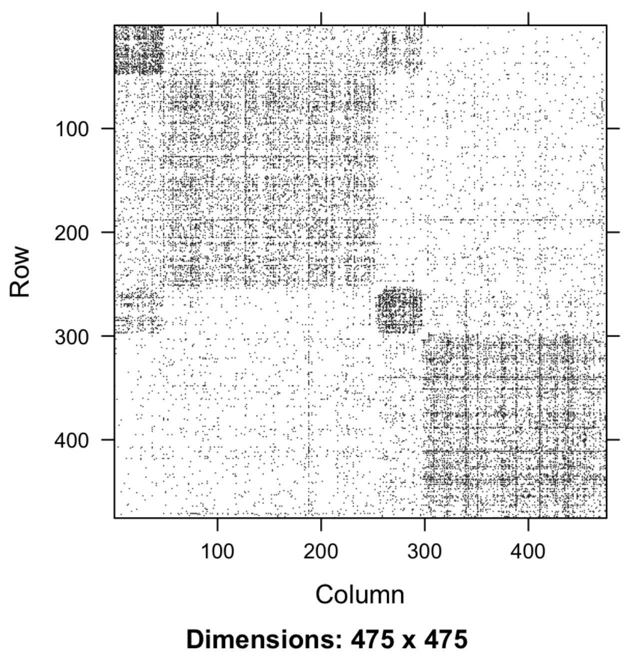

Figure 10: Adjacency matrix for the `congress-Twitter` network with nodes ordered alphabetically based on party affiliation (Democrats, Independents, then Republicans) and chamber.

7.2 Code

```r
library(tidyverse)
library(rjson)

# load data from JSON file
congress_twit <- fromJSON(file = <FILE>)

# prepare dataframe for directed edges (columns 1 and 2) and weights
ctwit_df <- data.frame(from = c(), to = c(), weight = c())

# loop through all nodes in the dataset
for (i in 1:475) {

  # extract nodes that have an edge to node i
  from <- congress_twit[[1]]$inList[[i]]

  # reindex node numbers to start from 1
  if (length(from) != 0) {
    from <- from + 1
  }

  # add directed edges and weights to dataframe
  to <- rep(i, length(from))
  weight <- congress_twit[[1]]$inWeight[[i]]
  ctwit_df <- bind_rows(ctwit_df, data.frame(from = from, to = to, weight = weight))
}
```

Listing 1: Sample code to load the `congress-Twitter` data into R from a JSON file and convert the edges and weights into a dataframe.

```r
# select edges from dataframe (first two columns)
EL <- subset(ctwit_df, select = c(1, 2))

# convert dataframe of edges into an edgelist
ctwit.edgelist <- matrix(unlist(EL), ncol=2)

library(igraph)

# convert edgelist into a directed igraph graph
ctwit.igraph <- graph.edgelist(ctwit.edgelist)

# add weight and node attributes to graph
E(ctwit.igraph)$weight <- ctwit_df$weight
V(ctwit.igraph)$username <- congress_attributes$Username
V(ctwit.igraph)$congressperson <- congress_attributes$Congressperson
V(ctwit.igraph)$chamber <- congress_attributes$Chamber
V(ctwit.igraph)$party <- congress_attributes$Party
V(ctwit.igraph)$state <- congress_attributes$State
V(ctwit.igraph)$region <- congress_attributes$Region

# convert graph into an undirected graph
ctwit.igraph_und <- as.undirected(
  ctwit.igraph,
  mode = "collapse",
  edge.attr.comb = "ignore" # removes edge weights
)
```

Listing 2: Sample code to convert the `congress-Twitter` data into an undirected and unweighted graph with node attributes for each member's Twitter username, name, chamber, party affiliation, state, and region.