

Insights from Twitter Interactions among US Congress Members

Data to Paper

December 26, 2023

Abstract

Understanding the social dynamics and interactions among members of the US Congress is crucial for unraveling the complex political relationships within the legislative body. However, the structural properties and factors influencing these interactions remain understudied. To address this gap, we conduct a comprehensive analysis of a Twitter interaction dataset among members of the 117th US Congress, spanning a 4-month period. By leveraging a directed graph framework and employing graph metrics and statistical analysis, we uncover intriguing insights into this dynamic network. Our findings demonstrate that the chamber affiliation does not significantly influence the density of out-degree interactions, suggesting similar engagement patterns across the House and Senate. Additionally, we observe variations in the degree centrality among political parties, indicating distinct levels of Twitter engagement among Democrats, Republicans, and Independents. These results contribute to our understanding of the social dynamics within Congress and highlight the potential of directed graph analysis for studying political networks. It is important to note that our analysis focuses solely on Twitter interactions and does not capture the full spectrum of congressional communication. Nonetheless, this study enhances our knowledge of the role of social media in political discourse and decision-making processes.

Results

In this study, we present the results of a comprehensive analysis conducted on a directed graph mapping the Twitter interactions among members of the 117th US Congress. Consisting of 475 nodes and 13289 edges, and exhibiting a density of 0.05902, the network generated offers deep insights into the structural properties and factors driving these interactions.

Our first step involved investigating the connection between a member’s house affiliation (House or Senate) and their out-degree interaction density. To generate a thorough understanding of the patterns in interaction density across different houses, we adopted an analysis of variance (ANOVA) approach, the details and results of which are presented in Table 1. The ANOVA results revealed a significant lack of difference in the density of out-degree interactions exhibited by members of the House or the Senate (F statistic = 0.657, p-value > 0.05). These findings suggest that the engagement patterns on Twitter among the congressional members remains unaffected by their house affiliations. In other words, the extent of connectivity or interaction between any given member and all other members does not vary significantly across the two houses.

Table 1: ANOVA test of association between chamber affiliation and out-degree centrality

	F statistic	p-value
Chamber	0.657	0.418

Out-degree centrality is a measure of the proportion of all other congress members a given member interacts with on Twitter.

p-value: value of the statistical significance test

Expanding on these findings, our analysis then shifted focus onto studying the degree of centrality within each political party. The aim was to unearth underlying variations in Twitter engagement patterns among Democrats, Republicans, and Independents. As displayed in Table 2, our computations revealed an average degree centrality of 0.117 for Democrats, 0.0717 for Independents, and 0.119 for Republicans. These figures highlight slightly higher levels of engagement on Twitter by Democrats and Republicans compared to Independents. As degree centrality reflects the fraction of the graph that a node is connected to, these results represent a more extensive network of interactions pertaining to Democrats and Republicans.

Overall, this analytical study based on the Twitter interactions among the members of the 117th US Congress has brought forth intriguing insights into the interconnected political network that forms part of the legislative body. Although there was no discernible difference in interaction density driven by house affiliations, evident variations in degree centrality among the different political parties were detected. Being solely reliant on Twitter interactions for the analysis might present a limitation in terms of capturing the full spectrum of congressional communication, but the insights garnered

Table 2: Degree centrality by political party

Average Degree Centrality	
Party	
Democrat	0.117
Independent	0.0717
Republican	0.119

Average Degree Centrality: Average centrality measure indicating the proportion of out-degree interaction within the Twitter network

pave the way for future, expansive investigations within the politico-social dynamic sphere.

A Data Description

Here is the data description, as provided by the user:

*** Rationale:**

The dataset maps US Congress's Twitter interactions into a directed graph with social interactions (edges) among Congress members (nodes). Each member (node) is further characterized by three attributes: Represented State, Political Party, and Chamber, allowing analysis of the adjacency matrix structure, graph metrics and likelihood of interactions across these attributes.

*** Data Collection and Network Construction:**

Twitter data of members of the 117th US Congress, from both the House and the Senate, were harvested for a 4-month period, February 9 to June 9, 2022 (using the Twitter API). Members with fewer than 100 tweets were excluded from the network.

- ``Nodes``. Nodes represent Congress members. Each node is designated an integer node ID (0, 1, 2, ...) which corresponds to a row in ``congress_members.csv``, providing the member's Represented State, Political Party, and Chamber.
- ``Edges``. A directed edge from node *i* to node *j* indicates that member *i* engaged with member *j* on Twitter at least once during the 4-month data-collection period. An engagement is defined as a tweet by member *i* that mentions member *j*'s handle, or as retweets, quote tweets, or replies of *i* to a tweet by member *j*.

*** Data analysis guidelines:**

- Your analysis code should NOT create tables that include names of Congress members, or their Twitter handles.
- Your analysis code should NOT create tables that include names of States, or their two-letter abbreviations. The code may of course do statistical analysis of **properties** related to States, but should not single out specific states.

2 data files:

File #1: "congress_members.csv"

A csv file of members of the 117th Congress, including their Twitter handles, Represented State, Party, and Chamber.

Data source: ``https://pressgallery.house.gov/member-data/members-official-`

twitter-handles`.

Rows are ordered according to the node ID, starting at 0.

Fields:

`Handle`: Twitter handle (without `@`)
`State`: Categorical; Two-letter state abbreviation; including also: "DC", "PR",
"VI", "AS", "GU", "MP".
`Party`: Categorical; Party affiliation ("D", "R", or "I")
`Chamber`: Categorical; The member's chamber ("House", "Senate")

Here are the first few lines of the file:

```
```output
Handle,State,Party,Chamber
SenatorBaldwin,WI,D,Senate
SenJohnBarrasso,WY,R,Senate
SenatorBennet,CO,D,Senate
...

```

File #2: "congress\_edges.dat"

This file provides the interaction network between members of the 115th US Congress on Twitter.

Download and adapted from: [https://snap.stanford.edu/data/congress-twitter`](https://snap.stanford.edu/data/congress-twitter)

Each line contains two integers (i, j), indicating a directed edge from node ID i to node ID j, compatible with `nx.read_edgelist('congress_edges.dat', create_using=nx.DiGraph())`. An i->j edge indicates that Congress member i had at least one tweet engaging with Congress member j during the 4-month collection period.

## B Data Exploration

### B.1 Code

The Data Exploration was carried out using the following custom code:

```
1
2 import pandas as pd
3 import networkx as nx
4 import numpy as np
5
6 # Read in the members file with pandas
7 df_members = pd.read_csv('congress_members.csv')
8
9 # Gathering the Data Size information
10 data_size = f'Number of Members: {df_members.shape[0]}\nNumber
 ↳ of Columns in Members Data: {df_members.shape[1]}'
11
12 # Gathering the Summary Statistics information
13 summary_stats = df_members.describe(include='all').to_string()
14
15 # Gathering the Categorical Variables information
16 cats = df_members.select_dtypes(include=['object'])
17 freq_cats = cats.apply(lambda x: f"{x.name}:
 ↳ {x.value_counts().idxmax()} (Most Common)\n").to_list()
18
19 # Gathering the Missing Values information
20 missing_values_counts = df_members.isna().sum().sum()
21
22 # Read in the edges file with networkx
23 G = nx.read_edgelist('congress_edges.dat',
 ↳ create_using=nx.DiGraph())
24 num_of_edges = G.number_of_edges()
25 num_of_nodes = G.number_of_nodes()
26
27 # Write out the gathered data to a text file
28 with open('data_exploration.txt', 'w') as f:
29 f.write('# Data Size\n')
30 f.write(data_size)
31
32 # Add number of edges and nodes
```

```

33 f.write(f'\nNumber of Edges: {num_of_edges}\nNumber of
 ↪ Nodes: {num_of_nodes}\n')
34
35 f.write('\n# Summary Statistics\n')
36 f.write(summary_stats)
37
38 f.write('\n# Categorical Variables\n')
39 f.writelines(freq_cats)
40
41 f.write('\n# Missing Values\n')
42 f.write(f'Total Missing Values:
 ↪ {missing_values_counts}\n')
43

```

## B.2 Code Description

The provided code performs data exploration on the given dataset of US Congress Twitter interactions. The code aims to analyze and summarize the dataset by extracting various information and writing it to the "data\_exploration.txt" file.

First, the code reads the "congress\_members.csv" file using pandas, which contains information about Congress members such as their Twitter handles, represented states, party affiliations, and chambers. The code then gathers data size information, including the number of members and the number of columns in the members' data.

Next, the code calculates and adds information about the interaction network by reading the "congress\_edges.dat" file using networkx. It analyzes the number of edges and nodes in the network.

Moving on, the code collects summary statistics of the categorical variables in the members' data and writes them to the output file. It provides information like the most common value for each categorical variable (e.g., the most common represented state, party affiliation, and chamber).

Furthermore, the code calculates the total number of missing values in the members' data and includes it in the output file. This provides an overview of the data quality and the extent of missing information.

Finally, all the gathered information is written to the "data\_exploration.txt" file. The file includes sections such as data size, summary statistics, categorical variables, and missing values. This allows for a comprehensive understanding and analysis of the US Congress Twitter dataset.

The output file serves as a summary and reference for the exploratory

analysis of the dataset. It provides key details and statistical information that can aid in further research, data preprocessing, and decision-making processes.

### B.3 Code Output

#### data\_exploration.txt

```
Data Size
Number of Members: 475
Number of Columns in Members Data: 4
Number of Edges: 13289
Number of Nodes: 475

Summary Statistics

```

	Handle	State	Party	Chamber
count	475	475	475	475
unique	475	54	3	2
top	SenatorBaldwin	CA	D	House
freq	1	49	250	383

```
Categorical Variables
Handle: SenatorBaldwin (Most Common)
State: CA (Most Common)
Party: D (Most Common)
Chamber: House (Most Common)

Missing Values
Total Missing Values: 0
```

## C Data Analysis

### C.1 Code

The Data Analysis was carried out using the following custom code:

```
1
2 # IMPORT
3 import pandas as pd
4 import networkx as nx
5 from scipy import stats
```



```

6 import pickle
7
8 # LOAD DATA
9 members = pd.read_csv('congress_members.csv')
10 edges = nx.read_edgelist('congress_edges.dat',
 ↪ create_using=nx.DiGraph())
11
12 # DATASET PREPARATIONS
13 # No dataset preparations are needed.
14
15 # DESCRIPTIVE STATISTICS
16 # No descriptive statistics table is needed.
17
18 # PREPROCESSING
19 # Creating numeric IDs for members; adding this column to
 ↪ members dataframe.
20 members['ID'] = range(len(members))
21
22 # ANALYSIS
23 ## Table 1: "Test of Association between Chamber and Density
 ↪ of Out-Edges"
24 # Create an out-degree centrality measure for each node
 ↪ (member of congress)
25 out_degrees =
 ↪ pd.DataFrame.from_dict(nx.out_degree_centrality(edges),
 ↪ orient='index', columns=['Out-degree Centrality'])
26
27 # Label node IDs and merge with members dataframe
28 out_degrees['ID'] = out_degrees.index.map(int)
29 result_table_1 = members.merge(out_degrees, on='ID')
30
31 house_out_degrees = result_table_1[result_table_1['Chamber']
 ↪ == 'House']['Out-degree Centrality']
32 senate_out_degrees = result_table_1[result_table_1['Chamber']
 ↪ == 'Senate']['Out-degree Centrality']
33
34 f_stat, p_val = stats.f_oneway(house_out_degrees,
 ↪ senate_out_degrees)
35 df1 = pd.DataFrame({'F Stat': [f_stat], 'p-value': [p_val]},
 ↪ index=['Chamber'])

```

```

36 df1.to_pickle('table_1.pkl')
37
38 ## Table 2: "Degree Centrality by Party"
39 # State level centrality measure
40 degree_centrality =
 ↪ pd.DataFrame.from_dict(nx.degree centrality(edges),
 ↪ orient='index', columns=['Degree Centrality'])
41 degree_centrality['ID'] = degree_centrality.index.map(int)
42 result_table_2 = members.merge(degree_centrality, on='ID')
43
44 by_party = result_table_2.groupby('Party')['Degree
 ↪ Centrality'].mean()
45 df2 = pd.DataFrame(by_party).rename(columns={'Degree
 ↪ Centrality': 'Average Degree Centrality'})
46 df2.to_pickle('table_2.pkl')
47
48 # SAVE ADDITIONAL RESULTS
49 additional_results = {
50 'Number of Nodes': edges.number_of_nodes(),
51 'Number of Edges': edges.number_of_edges(),
52 'Density of Network': nx.density(edges)
53 }
54
55 with open('additional_results.pkl', 'wb') as file:
56 pickle.dump(additional_results, file)
57

```

## C.2 Code Description

The code performs an analysis of the Twitter interaction network among members of the 117th US Congress. The data consists of two files: "congress\_members.csv" which provides information about the Congress members, and "congress\_edges.dat" which represents the network of interactions between the members on Twitter.

The analysis code performs the following steps:

1. Load the data: The code reads the "congress\_members.csv" file using the pandas library to create a DataFrame containing information about the members, and reads the "congress\_edges.dat" file using the networkx library to create a directed graph representing the Twitter interactions.
2. Dataset Preparations: No specific dataset preparations are performed

in this code.

3. Descriptive Statistics: No descriptive statistics table is generated in this code.

4. Preprocessing: The code assigns numeric IDs to the members and adds this column to the members DataFrame.

5. Analysis: a. Table 1: "Test of Association between Chamber and Density of Out-Edges": - Compute the out-degree centrality measure for each node (member) in the graph, which represents the proportion of outgoing edges from a node. - Merge the out-degree centrality values with the members DataFrame based on the node ID. - Separate the out-degree centrality values for members of the House and Senate into separate series. - Perform a one-way analysis of variance (ANOVA) test to determine the association between the Chamber (House or Senate) and the density of out-edges. - Save the results of the test, including the F-statistic and p-value, as a DataFrame and store it as "table\_1.pkl".

b. Table 2: "Degree Centrality by Party": - Compute the degree centrality measure for each node (member) in the graph, which represents the proportion of edges connected to a node. - Merge the degree centrality values with the members DataFrame based on the node ID. - Group the members by their political party and calculate the average degree centrality for each party. - Save the results of the average degree centrality by party as a DataFrame and store it as "table\_2.pkl".

6. Additional Results: - Compute and store additional results about the network, including the number of nodes, number of edges, and density of the network, as a dictionary. - Save the dictionary as a pickled file named "additional\_results.pkl".

The code enables analysis of interactions between Congress members on Twitter, examining factors such as the association between the chamber and the density of out-edges, as well as the average degree centrality by party. Additionally, it provides summary statistics about the network structure through the additional\_results.pkl file.

### C.3 Code Output

**table\_1.pkl**

	F Stat	p-value
Chamber	0.65654	0.4182

table\_2.pkl

Average Degree Centrality	
Party	
D	0.117359
I	0.071730
R	0.119231

additional\_results.pkl

```
{
 'Number of Nodes': 475,
 'Number of Edges': 13289,
 'Density of Network': 0.05902,
}
```

## D LaTeX Table Design

### D.1 Code

The LaTeX Table Design was carried out using the following custom code:

```
1
2
3 # IMPORT
4 import pandas as pd
5 from my_utils import to_latex_with_note, format_p_value,
 ↪ is_str_in_df, split_mapping, AbbrToNameDef
6
7 # PREPARATION FOR ALL TABLES
8
9 # Shared mapping
10 shared_mapping: AbbrToNameDef = {
11 'F Stat': ('F statistic', None),
12 'p-value': ('p-value', 'value of the statistical
 ↪ significance test'),
13 'Party': ('Political Party', 'D: Democrat, R: Republican,
 ↪ I: Independent'),
```

```

14 'Average Degree Centrality': ('Average Degree Centrality',
 ↪ 'Average centrality measure indicating the proportion
 ↪ of out-degree interaction within the Twitter
 ↪ network'),
15 'D': ('Democrat', None),
16 'I': ('Independent', None),
17 'R': ('Republican', None)
18 }
19
20 # TABLE 1
21 df1 = pd.read_pickle('table_1.pkl')
22
23 # Format statistical significance values
24 df1['p-value'] = df1['p-value'].apply(format_p_value)
25
26 # Rename rows and columns
27 mapping1 = {k: v for k, v in shared_mapping.items() if
 ↪ is_str_in_df(df1, k)}
28 abbrs_to_names1, legend1 = split_mapping(mapping1)
29 df1 = df1.rename(columns=abbrs_to_names1,
 ↪ index=abbrs_to_names1)
30
31 # Save as latex
32 to_latex_with_note(
33 df1,
34 'table_1.tex',
35 caption="ANOVA test of association between chamber
 ↪ affiliation and out-degree centrality",
36 label='table:anova_1',
37 note="Out-degree centrality is a measure of the proportion of
 ↪ all other congress members a given member interacts with
 ↪ on Twitter.",
38 legend=legend1
39)
40
41 # TABLE 2
42 df2 = pd.read_pickle('table_2.pkl')
43
44 # Rename rows and columns

```

```

45 mapping2 = {k: v for k, v in shared_mapping.items() if
 ↪ is_str_in_df(df2, k)}
46 abbrs_to_names2, legend2 = split_mapping(mapping2)
47 df2 = df2.rename(index=abbrs_to_names2)
48
49 # Save as latex
50 to_latex_with_note(
51 df2,
52 'table_2.tex',
53 caption="Degree centrality by political party",
54 label='table:degree_centrality_party',
55 note=None,
56 legend=legend2
57)
58

```

## D.2 Provided Code

The code above is using the following provided functions:

```

1 def to_latex_with_note(df, filename: str, caption: str, label:
 ↪ str, note: str = None, legend: Dict[str, str] = None,
 ↪ **kwargs):
2 """
3 Converts a DataFrame to a LaTeX table with optional note and
 ↪ legend added below the table.
4
5 Parameters:
6 - df, filename, caption, label: as in `df.to_latex`.
7 - note (optional): Additional note below the table.
8 - legend (optional): Dictionary mapping abbreviations to full
 ↪ names.
9 - **kwargs: Additional arguments for `df.to_latex`.
10
11 Returns:
12 - None: Outputs LaTeX file.
13 """
14
15 def format_p_value(x):
16 returns "{:.3g}".format(x) if x >= 1e-06 else "<1e-06"

```

```

17
18 def is_str_in_df(df: pd.DataFrame, s: str):
19 return any(s in level for level in getattr(df.index,
 ↪ 'levels', [df.index]) + getattr(df.columns, 'levels',
 ↪ [df.columns]))
20
21 AbbrToNameDef = Dict[Any, Tuple[Optional[str], Optional[str]]]
22
23 def split_mapping(abbrs_to_names_and_definitions:
 ↪ AbbrToNameDef):
24 abbrs_to_names = {abbr: name for abbr, (name, definition) in
 ↪ abbrs_to_names_and_definitions.items() if name is not
 ↪ None}
25 names_to_definitions = {name or abbr: definition for abbr,
 ↪ (name, definition) in
 ↪ abbrs_to_names_and_definitions.items() if definition is
 ↪ not None}
26 return abbrs_to_names, names_to_definitions
27

```

### D.3 Code Output

table\_1.tex

```

\begin{table}[h]
\caption{ANOVA test of association between chamber affiliation and out-degree
centrality}
\label{table:anova_1}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrl}
\toprule
& F statistic & p-value \\
\midrule
\textbf{Chamber} & 0.657 & 0.418 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize

```

```

\item Out-degree centrality is a measure of the proportion of all other congress
 members a given member interacts with on Twitter.
\item \textbf{p-value}: value of the statistical significance test
\end{tablenotes}
\end{threeparttable}
\end{table}

```

#### table\_2.tex

```

\begin{table}[h]
\caption{Degree centrality by political party}
\label{table:degree_centrality_party}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lr}
\toprule
& Average Degree Centrality \\\
Party & \\\
\midrule
\textbf{Democrat} & 0.117 \\\
\textbf{Independent} & 0.0717 \\\
\textbf{Republican} & 0.119 \\\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{Average Degree Centrality}: Average centrality measure indicating
 the proportion of out-degree interaction within the Twitter network
\end{tablenotes}
\end{threeparttable}
\end{table}

```