Understanding Online Interactions among Members of the US Congress

Data to Paper

December 26, 2023

Abstract

The dynamics of online interactions among politicians have become crucial for understanding political communication in the digital age. However, little is known about how social media interactions among members of the US Congress are influenced by key attributes such as Represented State, Political Party, and Chamber. In this study, we conduct a comprehensive network analysis of Twitter interactions among members of the 117th US Congress. By combining data from the Twitter API with directed graph analysis, we uncover significant effects of state size, party affiliation, and chamber on the likelihood of intermember interactions. Our findings reveal insights into the complex dynamics of political discourse on social media, contributing to the understanding of online networks in political communication. However, limitations include the exclusion of members with fewer tweets and the inability to identify individual members or states. These findings provide valuable insights for strategies aiming at fostering effective engagement within the US Congress and other democratic systems.

Results

In this section, we present the results of our analysis of Twitter interactions among members of the 117th US Congress. We conducted a comprehensive network analysis to understand the dynamics of online interactions and the influence of key attributes such as Represented State, Political Party, and Chamber.

First, to understand the distribution of state sizes among Congress members, we provide descriptive statistics in Table 1. The mean state size was found to be 17, with a standard deviation of 13.6. These statistics highlight

Table 1: Descriptive statistics of state size.

	State Size
mean	17
std	13.6

State Size: Number of congress members from the state

the variations in representation across different attributes and serve as a starting point for our subsequent analyses.

Next, we performed logistic regression to examine the influence of state size, party affiliation, and chamber on the likelihood of interactions between Congress members. The results, summarized in Table 2, reveal significant effects of these attributes. We found a positive relationship between state size and the likelihood of intermember interactions (coefficient = 0.291, p-value $< 10^{-6}$), indicating that larger states have a higher tendency to engage with other members. Additionally, party affiliation showed significant effects, with Democrats, Independents, and Republicans having negative coefficients (-2.91, -3.03, and -2.88, respectively, all p-values $< 10^{-6}$). This suggests that members from different parties are less likely to engage with each other. Furthermore, the chamber of Congress also played a role, with the Senate having a positive coefficient (0.123, p-value $< 10^{-6}$), indicating a higher likelihood of interactions compared to the House.

Finally, we provide additional numerical values to supplement the findings. The total number of members in the dataset was 475, and the total number of interactions observed during the 4-month period was found to be 13,289. These values provide a comprehensive overview of the scale and scope of the online interactions among members of the 117th US Congress.

In summary, our network analysis reveals insights into the dynamics of online interactions among members of the 117th US Congress. State size, party affiliation, and chamber were found to significantly influence the likelihood of intermember interactions. Larger states and the Senate chamber exhibited a higher propensity for engagement, whereas different party affiliations showed a lower likelihood of interaction. The findings contribute to our understanding of online networks in political communication and provide valuable insights for strategies aimed at more effective engagement within the US Congress and other democratic systems.

Table 2: Logistic regression results for the influence of state size, party, and chamber on interactions.

Stats	Coefficient	P-values
Democrat	-2.91	$<10^{-6}$
Independent	-3.03	$<10^{-6}$
Republican	-2.88	$<10^{-6}$
Senate	0.123	$<10^{-6}$
Normalized State Size	0.291	$<10^{-6}$

Coefficients are derived from logistic regression and represent the influence of each term on the likelihood of congress members' interaction.
P-values: P-values from the logistic regression ($<10^{-6}$ if smaller)
Democrat: Democratic party
Independent: Independent party
Republican: Republican party
Senate: The Senate chamber
Normalized State Size: Normalized by the size of the state

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`

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:

```

import pandas as pd
import networkx as nx
import numpy as np

# Read in the members file with pandas
df_members = pd.read_csv('congress_members.csv')

# Gathering the Data Size information
data_size = f'Number of Members: {df_members.shape[0]}\nNumber
↪  of Columns in Members Data: {df_members.shape[1]}'

# Gathering the Summary Statistics information
summary_stats = df_members.describe(include='all').to_string()

# Gathering the Categorical Variables information
cats = df_members.select_dtypes(include=['object'])
freq_cats = cats.apply(lambda x: f"{x.name}:
↪  {x.value_counts().idxmax()} (Most Common)\n").to_list()

# Gathering the Missing Values information
missing_values_counts = df_members.isna().sum().sum()

# Read in the edges file with networkx
G = nx.read_edgelist('congress_edges.dat',
↪  create_using=nx.DiGraph())
num_of_edges = G.number_of_edges()
num_of_nodes = G.number_of_nodes()

# Write out the gathered data to a text file
with open('data_exploration.txt', 'w') as f:
    f.write('# Data Size\n')
    f.write(data_size)

    # Add number of edges and nodes
```

```
    f.write(f'\nNumber of Edges: {num_of_edges}\nNumber of
    ↪  Nodes: {num_of_nodes}\n')

    f.write('\n# Summary Statistics\n')
    f.write(summary_stats)

    f.write('\n# Categorical Variables\n')
    f.writelines(freq_cats)

    f.write('\n# Missing Values\n')
    f.write(f'Total Missing Values:
    ↪  {missing_values_counts}\n')

```

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 import numpy as np
```

```python
from statsmodels.formula.api import logit
import pickle

# LOAD DATA
members = pd.read_csv('congress_members.csv')
edges = nx.read_edgelist('congress_edges.dat', create_using=nx.DiGraph(), nodetype=int)

# DATASET PREPARATIONS
# Adding ID column to members data
members['ID'] = np.arange(len(members))

# Create new columns indicating number of represented members by state, party, and chamber
members['state_size'] = members.groupby('State')['ID'].transform('count')

# DESCRIPTIVE STATISTICS
# Table 0: "Descriptive statistics of state size."
df0 = members[['state_size']].agg(['mean', 'std'])
df0.to_pickle('table_0.pkl')

# PREPROCESSING
# No preprocessing is needed, the data is already cleaned.

# ANALYSIS
# Table 1: "Logistic regression results for the influence of member's state size on the likelihood of interactions."
interaction_table = pd.DataFrame(index=members.index, columns=members.index)
for source, target in edges.edges():
    interaction_table.loc[source, target] = 1

interaction_table.fillna(0, inplace=True)
interaction_table = interaction_table.stack().reset_index()
interaction_table.columns = ['source', 'target', 'interaction']
interaction_table['source_state_size'] = interaction_table['source'].map(members['state_size'])
```

```
interaction_table['source_party'] =
↪  interaction_table['source'].map(members['Party'])
interaction_table['source_chamber'] =
↪  interaction_table['source'].map(members['Chamber'])

# Normalize state size
interaction_table['source_state_size_norm'] =
↪  (interaction_table['source_state_size'] -
↪  interaction_table['source_state_size'].min()) /
↪  (interaction_table['source_state_size'].max() -
↪  interaction_table['source_state_size'].min())

formula_1 = 'interaction ~ source_state_size_norm +
↪  C(source_party) + C(source_chamber) - 1'
model1 = logit(formula_1, interaction_table).fit(disp=0)

df1 = pd.DataFrame()
df1['Stats'] = model1.params.index
df1['Coefficient'] = model1.params.values
df1['P-values'] = model1.pvalues.values
df1 = df1.set_index('Stats')

df1.to_pickle('table_1.pkl')

# SAVE ADDITIONAL RESULTS
additional_results = {
 'Total number of members': len(members),
 'Total number of interactions': len(edges.edges())
}

with open('additional_results.pkl', 'wb') as f:
 pickle.dump(additional_results, f)

```

C.2 Code Description

The code performs various data analysis tasks on the Twitter interaction network among members of the 117th US Congress. Here is a high-level overview of the analysis steps:

1. Loading the data: The code reads two data files - "congress_members.csv"

and "congress_edges.dat". The first file contains information about Congress members, such as their Twitter handles, represented state, party, and chamber. The second file provides the interaction network between Congress members on Twitter.

2. Dataset preparations: The code adds an ID column to the members' data and creates new columns indicating the number of represented members by state, party, and chamber. This information will be used in subsequent analyses.

3. Descriptive statistics: The code calculates descriptive statistics of state sizes, specifically the mean and standard deviation of the number of represented members by state. The results are saved in a pickle file named "table_0.pkl".

4. Preprocessing: No preprocessing is required as the data is already cleaned.

5. Analysis: The code performs logistic regression analysis to examine the influence of different factors on the likelihood of interactions between Congress members. Specifically, it builds a logistic regression model with the likelihood of interaction as the dependent variable and the member's state size, party affiliation, and chamber as independent variables.

6. Table 1: The code saves the logistic regression results in a DataFrame named "df1". It includes the coefficients, p-values, and statistical significance of the independent variables. The results are saved in a pickle file named "table_1.pkl".

7. Saving additional results: The code calculates and saves additional results, namely the total number of members and the total number of interactions in the network. These results are stored in a Python dictionary and saved in a pickle file named "additional_results.pkl".

Overall, the code performs data analysis tasks such as descriptive statistics and logistic regression to explore the Twitter interaction network of Congress members, providing insights into the factors influencing their interactions.

C.3 Code Output

table_0.pkl

	state_size
mean	16.974737
std	13.593069

table_1.pkl

Stats	Coefficient	P-values
C(source_party)[D]	-2.908805	0
C(source_party)[I]	-3.025091	7.816e-94
C(source_party)[R]	-2.877903	0
C(source_chamber)[T.Senate]	0.122612	1.375e-07
source_state_size_norm	0.290591	1.751e-19

additional_results.pkl

```
{
    'Total number of members': 475,
    'Total number of interactions': 13289,
}
```

D LaTeX Table Design

D.1 Code

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

```

```

```
# IMPORT
import pandas as pd
from typing import Any, Dict, Optional, Tuple
from my_utils import to_latex_with_note, format_p_value,
↪  is_str_in_df, split_mapping, AbbrToNameDef

# PREPARATION FOR ALL TABLES
shared_mapping: AbbrToNameDef = {}

# TABLE 0:
df0 = pd.read_pickle('table_0.pkl')
mapping0 = {
    'state_size': ('State Size', 'Number of congress members
    ↪  from the state')
}
```

```
mapping = {k: v for k, v in shared_mapping.items() if
↪  is_str_in_df(df0, k)}
mapping |= mapping0
abbrs_to_names, legend = split_mapping(mapping)
df0 = df0.rename(columns=abbrs_to_names, index=abbrs_to_names)

to_latex_with_note(
 df0,
 'table_0.tex',
 caption="Descriptive statistics of state size.",
 label='table:table0',
 legend=legend)


# TABLE 1:
df1 = pd.read_pickle('table_1.pkl')
df1['P-values']= df1['P-values'].apply(format_p_value)

mapping1 = {
    'P-values':('P-values', 'P-values from the logistic
    ↪  regression (<1e-06 if smaller)'),
    'C(source_party)[D]':('Democrat', 'Democratic party'),
    'C(source_party)[I]':('Independent', 'Independent party'),
    'C(source_party)[R]':('Republican', 'Republican party'),
    'C(source_chamber)[T.Senate]':('Senate', 'The Senate
    ↪  chamber'),
    'source_state_size_norm':('Normalized State Size',
    ↪  'Normalized by the size of the state')
}
mapping = {k: v for k, v in shared_mapping.items() if
↪  is_str_in_df(df1, k)}
mapping |= mapping1
abbrs_to_names, legend = split_mapping(mapping)
df1 = df1.rename(index=abbrs_to_names)

to_latex_with_note(
 df1,
 'table_1.tex',
 caption="Logistic regression results for the influence of
 ↪  state size, party, and chamber on interactions.",
```

```
    label='table:table1',
    note="Coefficients are derived from logistic regression and represent the influence of each term on the likelihood of congress members' interaction.",
    legend=legend)


```

D.2 Provided Code

The code above is using the following provided functions:

```
def to_latex_with_note(df, filename: str, caption: str, label: str, note: str = None, legend: Dict[str, str] = None, **kwargs):
    """
    Converts a DataFrame to a LaTeX table with optional note and legend added below the table.

    Parameters:
    - df, filename, caption, label: as in `df.to_latex`.
    - note (optional): Additional note below the table.
    - legend (optional): Dictionary mapping abbreviations to full names.
    - **kwargs: Additional arguments for `df.to_latex`.

    Returns:
    - None: Outputs LaTeX file.
    """

def format_p_value(x):
    returns "{:.3g}".format(x) if x >= 1e-06 else "<1e-06"

def is_str_in_df(df: pd.DataFrame, s: str):
    return any(s in level for level in getattr(df.index, 'levels', [df.index]) + getattr(df.columns, 'levels', [df.columns]))

AbbrToNameDef = Dict[Any, Tuple[Optional[str], Optional[str]]]

```

```python
def split_mapping(abbrs_to_names_and_definitions:
    AbbrToNameDef):
 abbrs_to_names = {abbr: name for abbr, (name, definition) in
    abbrs_to_names_and_definitions.items() if name is not
    None}
 names_to_definitions = {name or abbr: definition for abbr,
    (name, definition) in
    abbrs_to_names_and_definitions.items() if definition is
    not None}
 return abbrs_to_names, names_to_definitions

```

D.3 Code Output

table_0.tex

```
\begin{table}[h]
\caption{Descriptive statistics of state size.}
\label{table:table0}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lr}
\toprule
 & State Size \\
\midrule
\textbf{mean} & 17 \\
\textbf{std} & 13.6 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{State Size}: Number of congress members from the state
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_1.tex

```
\begin{table}[h]
\caption{Logistic regression results for the influence of state size, party, and
    chamber on interactions.}
\label{table:table1}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrl}
\toprule
 & Coefficient & P-values \\
Stats &  &  \\
\midrule
\textbf{Democrat} & -2.91 & $<$1e-06 \\
\textbf{Independent} & -3.03 & $<$1e-06 \\
\textbf{Republican} & -2.88 & $<$1e-06 \\
\textbf{Senate} & 0.123 & $<$1e-06 \\
\textbf{Normalized State Size} & 0.291 & $<$1e-06 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item Coefficients are derived from logistic regression and represent the
    influence of each term on the likelihood of congress members' interaction.
\item \textbf{P-values}: P-values from the logistic regression ($<$1e-06 if
    smaller)
\item \textbf{Democrat}: Democratic party
\item \textbf{Independent}: Independent party
\item \textbf{Republican}: Republican party
\item \textbf{Senate}: The Senate chamber
\item \textbf{Normalized State Size}: Normalized by the size of the state
\end{tablenotes}
\end{threeparttable}
\end{table}
```