Distinct Collaboration and Communication Patterns in Twitter Interactions among US Congress Members

Data to Paper

December 31, 2023

Abstract

Understanding online interactions among politicians is crucial for studying political communication and networked governance. However, limited research has explored the specific patterns of Twitter interactions among members of the US Congress. This study fills this gap by mapping the social connections and interactions among members of the 117th US Congress using a 4-month dataset. By analyzing attributes such as represented state, political party, and chamber, we uncover the distribution of interactions between members of the House of Representatives and the Senate. Our findings demonstrate differentiated collaboration and communication patterns between the chambers, supported by a significant relationship identified through a chi-square test of independence. While the dataset comprises XX Congress members, the exclusion of those with fewer than 100 tweets is a limitation that warrants further investigation. Overall, this research contributes to our understanding of online political interactions, shedding light on the evolving nature of democratic processes and their impact on political communication and networked governance.

Results

Our analysis began with an exploration of the distribution of Twitter interactions among members of the 117th US Congress. The motivation for this examination was to better comprehend the collaboration and communication patterns among the chambers due to their importance in political communication and networked governance. Our dataset consisted of 13,289 directed edges between the congress members, demonstrating their Twitter interaction patterns over a period of four months.

We first focused on the interactions between the House of Representatives and the Senate. The results illustrated in Table 1 showed 10,115 interactions initiated by the House of Representatives members, out of which a minor fraction, precisely 472, was directed towards the Senate. Meanwhile, the Senate members started 1,139 interactions, among which a comparably larger fraction, namely 1,563, was directed towards the House of Representatives. Such an evident disparity in the numbers suggested a clear difference in interaction patterns across the chambers.

Table 1: Distribution of interactions among House of Representatives and Senate Members

Chamber_Target Chamber_Source	House of Representatives	Senate
House of Representatives	10115	472
Senate	1139	1563

House of Representatives: Members of US Congress representing House
Senate: Members of US Congress representing Senate

To further quantify the observed interaction patterns, we conducted a chi-square test of independence, which measured if the interaction distribution across the chambers was purely by chance. As presented in Table 2, the chi-square statistic was significantly high, recording at 4727 with a p-value less than 10^{-6}. This clearly demonstrated that the distribution of interactions among the two chambers was not independent, and the chamber to which the congress members belonged did influence the way they interacted on Twitter.

Table 2: Chi-square Test of Independence Result

	Chi-square statistic	P-value
Chi-square Test	$4.73\ 10^3$	$<10^{-6}$

Chi-square statistic: Chi-square test statistic value indicating level of independence between variables
P-value: Statistical significance value of the Chi-square test statistic

To sum up, the analysis revealed distinct collaboration and communication patterns between the House of Representatives and the Senate. From the interaction frequency to the extent of influence the chamber had, our analysis indeed enriched our understanding of the networked dynamics of

online political interactions among members of the 117th US Congress and offered insights into the evolving democratic processes in networked governance and political communication.

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:

```python

import pandas as pd
import networkx as nx

# Load datasets
members = pd.read_csv("congress_members.csv")
edges = nx.read_edgelist("congress_edges.dat", create_using=nx.DiGraph())

summary_state = members.State.describe()
summary_party = members.Party.describe()
summary_chamber = members.Chamber.describe()
summary_missing = members.isnull().sum()

# Prepare report
data_size = f"Number of rows: {members.shape[0]}\nNumber of columns: {members.shape[1]}"
summary_statistics = f"State:\n{summary_state}\nParty:\n{summary_party}\nChamber:\n{summary_chamber}"
categorical_variables = f"Categorical columns: State, Party, Chamber\nMost common State: {summary_state.top}\nMost common Party: {summary_party.top}\nMost common Chamber: {summary_chamber.top}"
missing_values = f"Missing values:\n{summary_missing}"

report = f"# Data Size\n{data_size}\n\n# Summary Statistics\n{summary_statistics}\n\n# Categorical Variables\n{categorical_variables}\n\n# Missing Values\n{missing_values}"

with open("data_exploration.txt", 'w') as f:
    f.write(report)

# Network data
num_nodes = len(edges.nodes)
```

```python
num_edges = len(edges.edges)
network_data = f"\n\n# Network Data\nNumber of nodes:
↪  {num_nodes}\nNumber of edges: {num_edges}"
with open("data_exploration.txt", 'a') as f:
    f.write(network_data)

```

B.2 Code Description

The code performs data exploration on the dataset of US Congress Twitter interactions. The main steps of the analysis are as follows:

1. Loading the Datasets: The code loads two datasets - "congress_members.csv" and "congress_edges.dat". The first dataset contains information about the members of the US Congress, including their Twitter handles, represented state, party affiliation, and chamber. The second dataset represents the interaction network between members on Twitter.

2. Summary Statistics: The code calculates summary statistics for the categorical variables in the "congress_members.csv" dataset. It provides information such as the count, unique values, and the most common value for each variable - State, Party, and Chamber. These statistics help understand the distribution and representation of members across different categories.

3. Missing Values: The code checks for missing values in the "congress_members.csv" dataset and calculates the count of missing values for each variable. This information helps identify if any data is missing and may require further data cleaning or imputation steps.

4. Report Generation: The code generates a report summarizing the data size, summary statistics, categorical variables, and missing values. The report is written into a file named "data_exploration.txt". The report provides insights into the overall dataset and its characteristics, helping researchers understand the data before further analysis.

5. Network Data: The code calculates the number of nodes (members) and edges (interactions) in the interaction network represented by the "congress_edges.dat" dataset. This information gives an overview of the network's size and complexity.

The "data_exploration.txt" file contains the generated report, which includes: - Data Size: Number of rows and columns in the "congress_members.csv" dataset. - Summary Statistics: Summary statistics for categorical variables - State, Party, and Chamber. - Categorical Variables: Most common values for State, Party, and Chamber. - Missing Values: Count of missing values for each variable. - Network Data: Number of nodes and edges in the interaction network dataset.

The "data_exploration.txt" file serves as a reference for researchers to understand the dataset's characteristics, summary statistics, and network data before performing further analysis.

B.3 Code Output

data_exploration.txt

```
# Data Size
Number of rows: 475
Number of columns: 4

# Summary Statistics
State:
count     475
unique     54
top        CA
freq       49
Name: State, dtype: object
Party:
count     475
unique      3
top         D
freq      250
Name: Party, dtype: object
Chamber:
count       475
unique        2
top       House
freq        383
Name: Chamber, dtype: object

# Categorical Variables
Categorical columns: State, Party, Chamber
Most common State: CA
Most common Party: D
Most common Chamber: House

# Missing Values
```

```
Missing values:
Handle     0
State      0
Party      0
Chamber    0
dtype: int64

# Network Data
Number of nodes: 475
Number of edges: 13289
```

C Data Analysis

C.1 Code

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

```python

# IMPORT
import pandas as pd
import networkx as nx
import pickle
from scipy.stats import chi2_contingency

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

# DATASET PREPARATIONS
# Convert the edges to pandas DataFrame denoting source and
↪ target nodes
df_edges = pd.DataFrame(list(G.edges()),
↪ columns=['Source_Node', 'Target_Node'])

# Create new series with int type for merging operation
df_edges['Source_Node_int'] =
↪ df_edges['Source_Node'].astype(int)
df_edges['Target_Node_int'] =
↪ df_edges['Target_Node'].astype(int)
```

```

# Merge df_edges with df_members to get additional details for
↪  Source_Node and Target_Node chambers
df_edges = df_edges.merge(df_members[['Chamber']],
↪  left_on='Source_Node_int', right_index=True)
df_edges.rename(columns={'Chamber': 'Chamber_Source'},
↪  inplace=True)
df_edges = df_edges.merge(df_members[['Chamber']],
↪  left_on='Target_Node_int', right_index=True)
df_edges.rename(columns={'Chamber': 'Chamber_Target'},
↪  inplace=True)

# DESCRIPTIVE STATISTICS
# No descriptive statistics table is needed

# PREPROCESSING
# No preprocessing is needed, as the data are already suitable
↪  for our analysis

# ANALYSIS
## Table 1: "Distribution of interactions among House and
↪  Senate Members"
# Create a dataframe for Table 1
df1 = pd.crosstab(df_edges['Chamber_Source'],
↪  df_edges['Chamber_Target'])
df1.to_pickle('table_1.pkl')

# Calculate chi-square test of independence
chi2, p, dof, expected = chi2_contingency(df1)

## Table 2: "Chi-Square Test of Independence Result"
# Create a dataframe for Table 2
indexData = ['Chi-square Test']
df2_data = [{'Chi-square statistic': chi2, 'p-value': p}]
df2 = pd.DataFrame(df2_data, index=indexData)
df2.to_pickle('table_2.pkl')

# SAVE ADDITIONAL RESULTS
additional_results = {
 'Total number of observations': len(df_edges),
 'Chi-square Test Results': {'Chi-square statistic': chi2,
  ↪  'p-value': p}
}
with open('additional_results.pkl', 'wb') as f:
 pickle.dump(additional_results, f)

```

C.2 Code Description

The provided code conducts an analysis on a dataset representing Twitter interactions among members of the 117th US Congress. The analysis is performed in several steps:

1. Load the dataset: The code reads the 'congress_members.csv' file, which contains information about the Congress members, and creates a directed graph object 'G' from the 'congress_edges.dat' file, which represents the Twitter interactions between the members.

2. Dataset Preparations: The code converts the graph edges into a pandas DataFrame, allowing for easier manipulation and analysis. It also merges the DataFrame with the 'df_members' DataFrame to get additional details about the chambers of the source and target nodes.

3. Descriptive Statistics: No descriptive statistics table is created in this analysis, so this step is skipped.

4. Preprocessing: No preprocessing is performed on the data since it is already in a suitable format for analysis.

5. Analysis: a. Table 1: "Distribution of interactions among House and Senate Members": The code creates a contingency table using the 'crosstab' function to show the distribution of interactions between members of the House and the Senate. The resulting table is saved as 'table_1.pkl'.

b. Chi-Square Test of Independence: The code performs a chi-square test of independence on the contingency table created in the previous step using the 'chi2_contingency' function. It calculates the chi-square statistic, p-value, degrees of freedom, and expected frequencies. These results are saved as 'table_2.pkl'.

6. Saving Additional Results: The code saves the total number of observations and the results of the chi-square test of independence as a dictionary in the 'additional_results.pkl' file. This file can be used to further analyze and interpret the results.

Overall, the code performs an analysis on the Twitter interactions among members of the 117th US Congress, examining the distribution of interactions between the House and the Senate and conducting a chi-square test to determine if the interactions are independent of the chamber. The results of the analysis are saved as pickle files for further investigation.

C.3 Code Output

table_1.pkl

```
Chamber_Target  House  Senate
Chamber_Source
House           10115     472
Senate           1139    1563
```

table_2.pkl

```
                 Chi-square statistic p-value
Chi-square Test                  4727       0
```

additional_results.pkl

```
{
    'Total number of observations': 13289,
    'Chi-square Test Results': {'Chi-square statistic': 4727             ,
    'p-value': 0},
}
```

D LaTeX Table Design

D.1 Code

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

```python

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

# PREPARATION FOR ALL TABLES
# No shared mapping is required in this case as the tables
↪ labels are unique.

```

```python
# TABLE 1:
df = pd.read_pickle('table_1.pkl')

# RENAME ROWS AND COLUMNS
# Define labels to scientifically-suitable names
mapping: AbbrToNameDef = {
 'House': ('House of Representatives', 'Members of US Congress
  ↪  representing House'),
 'Senate': ('Senate', 'Members of US Congress representing
  ↪  Senate'),
}

abbrs_to_names, legend = split_mapping(mapping)
df = df.rename(columns=abbrs_to_names, index=abbrs_to_names)

# Save as latex:
to_latex_with_note(
 df, 'table_1.tex',
 caption="Distribution of interactions among House of
  ↪  Representatives and Senate Members",
 label='table:distribution_of_interactions',
 legend=legend)


# TABLE 2:
df = pd.read_pickle('table_2.pkl')

# FORMAT VALUES
# Applying `format_p_value` to p-value column
df['p-value'] = df['p-value'].apply(format_p_value)

# RENAME ROWS AND COLUMNS
# Define labels to scientifically-suitable names
mapping: AbbrToNameDef = {
 'Chi-square statistic': ('Chi-square statistic', 'Chi-square
  ↪  test statistic value indicating level of independence
  ↪  between variables'),
 'p-value': ('P-value', 'Statistical significance value of the
  ↪  Chi-square test statistic')
}
```

```

abbrs_to_names, legend = split_mapping(mapping)
df = df.rename(columns=abbrs_to_names, index=abbrs_to_names)

# Save as latex:
to_latex_with_note(
 df, 'table_2.tex',
 caption="Chi-square Test of Independence Result",
 label='table:chi_square_result',
 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):
```

```python
    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]]]

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_1.tex

```latex
\begin{table}[h]
\caption{Distribution of interactions among House of Representatives and Senate
    Members}
\label{table:distribution_of_interactions}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrr}
\toprule
Chamber\_Target & House of Representatives & Senate \\
Chamber\_Source &  &  \\
\midrule
\textbf{House of Representatives} & 10115 & 472 \\
\textbf{Senate} & 1139 & 1563 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
```

```
\item \textbf{House of Representatives}: Members of US Congress representing
    House
\item \textbf{Senate}: Members of US Congress representing Senate
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_2.tex

```
\begin{table}[h]
\caption{Chi-square Test of Independence Result}
\label{table:chi_square_result}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrl}
\toprule
 & Chi-square statistic & P-value \\
\midrule
\textbf{Chi-square Test} & 4.73e+03 & $<$1e-06 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{Chi-square statistic}: Chi-square test statistic value indicating
    level of independence between variables
\item \textbf{P-value}: Statistical significance value of the Chi-square test
    statistic
\end{tablenotes}
\end{threeparttable}
\end{table}
```