

Insights into Social Dynamics among US Congress Members through Twitter Interactions

Data to Paper

December 31, 2023

Abstract

Understanding social interactions among politicians is crucial for comprehending political landscapes. However, the analysis of Twitter interactions within the US Congress, particularly in relation to party affiliations and represented states, remains understudied. We address this research gap by presenting a comprehensive analysis of Twitter interactions among members of the 117th US Congress from both the House and the Senate. By employing a directed graph representation, we investigate the structure of the adjacency matrix, graph metrics, and the likelihood of interactions across party affiliations and represented states. Our findings reveal significant distinctions in Twitter interactions among different political parties, highlighting the influence of party affiliation on online interactions. Furthermore, we unveil the influential role of represented states in Twitter interactions, emphasizing the significance of geographical context in political discourse. We demonstrate that party affiliation fundamentally shapes the social network of Congress members and explore the impact of represented states. By contributing to a deeper understanding of the dynamics of online political communication within the US Congress, our research sheds light on the interplay between social media, party affiliations, and represented states. Limitations include the exclusion of members with fewer than 100 tweets, which may introduce certain biases. Overall, our results have important implications for understanding the multidimensional impacts of political and geographical factors on online interactions among politicians.

Results

We present the findings of our analysis of Twitter interactions among members of the 117th U.S. Congress, which includes a total of 475 Congress

members. The analysis aimed to uncover the influence of party affiliation and geographical context on these interactions.

We first investigated the party-wise interactions on Twitter. Our goal was to determine the likelihood of interactions between members of different political parties. To reach our findings, we employed a chi-square test, which tests whether two categorical variables are independent. In our case, the two variables were the party affiliations of the interacting congress members. The chi-square statistic was found to be $6.59 \cdot 10^3$, with a p-value of less than 10^{-6} , indicating a highly significant association between the party affiliations of interacting Congress members (Table 1). This suggests that party affiliation plays a crucial role in shaping the interactions among the Congress members on Twitter.

Table 1: Chi-square test of party-wise retweet interaction.

	Chi-Square Statistic	p-value	Degrees of Freedom
Party Interaction Analysis	$6.59 \cdot 10^3$	$<10^{-6}$	4

p-value: Probability of observing a value as extreme as or more extreme than the observed value under the null hypothesis

Degrees of Freedom: Number of independent pieces of information that go into the calculation of a statistic

Subsequently, we turned our attention to the state-wise engagements among Congress members on the same platform. The objective was to explore whether geographical representation affects the interaction patterns on Twitter. Applying the chi-square test again, we found a significant association between the represented states of the interacting Congress members ($\chi^2 = 9.47 \cdot 10^3$, $p < 10^{-6}$, $df = 2809$) (Table 2). These results imply that geographical context significantly impacts the Twitter interactions among Congress members.

Table 2: Chi-square test of state-wise retweet interaction.

	Chi-Square Statistic	p-value	Degrees of Freedom
State Interaction Analysis	$9.47 \cdot 10^3$	$<10^{-6}$	2809

p-value: Probability of observing a value as extreme as or more extreme than the observed value under the null hypothesis

Degrees of Freedom: Number of independent pieces of information that go into the calculation of a statistic

In addition to our primary results, we present some related statistics

to encapsulate the scale of our dataset. Our graph of Twitter interactions included 475 nodes, each representing a unique Congress member. The graph was directed, with a total of 13,289 edges, symbolising at least one interaction between the respective Congress members over the period of data collection.

Summarising, our analysis indicates that both party affiliations and represented states considerably influence the Twitter interactions among members of the 117th U.S. Congress. These results lay the groundwork for further investigations into the multidimensional impacts of the political and geographic, shaping online political communication within the U.S. Congress.

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
5 # Load congress members data from csv file
6 df_congress_members = pd.read_csv('congress_members.csv')
7
8 # Load congress edges data
9 G = nx.read_edgelist('congress_edges.dat',
10 ↪ create_using=nx.DiGraph())
11
12 # Open output file
13 with open("data_exploration.txt", "w") as file:
14
15     # Data size
16     file.write("# Data Size\n")
17     file.write(f"Number of Congress members:
18 ↪ {df_congress_members.shape[0]}\n")
19     file.write(f"Number of attributes per member:
20 ↪ {df_congress_members.shape[1]}\n")
21     file.write(f"Number of nodes in the graph:
22 ↪ {G.number_of_nodes()}\n")
23     file.write(f"Number of edges in the graph:
24 ↪ {G.number_of_edges()}\n\n")
25
26     # Summary statistics
27     file.write("# Summary Statistics\n")
28     # As there are no continuous variables in this dataset, we
29     ↪ will skip this section
30     file.write("N/A for this dataset\n\n")
31
32     # Categorical variables
33     file.write("# Categorical Variables\n")
34     categorical_cols = ['State', 'Party', 'Chamber']
35     for col in categorical_cols:
```

```

30     # Count of members per categorical variable
31     file.write(f"Count of members per {col}:
    ↪     \n{df_congress_members[col].value_counts()}\n")
32     # Most common category for each categorical variable
33     top_5 =
    ↪     df_congress_members[col].value_counts().idxmax()
34     file.write(f"Most common {col}: {top_5}\n\n")
35
36     # Missing values
37     file.write("# Missing Values\n")
38     file.write(f"Missing values in each
    ↪     column:\n{df_congress_members.isna().sum()}\n")
39
40 file.close()
41

```

B.2 Code Description

The code performs a data exploration analysis on a dataset that represents Twitter interactions among members of the 117th US Congress. The analysis is conducted in order to gain insights into the structure and characteristics of the network formed by these interactions.

First, the code loads the congress members' data from a CSV file, which contains information such as Twitter handles, represented states, party affiliations, and chambers of the members. It also loads the congress edges data, which represents the directed interactions between members on Twitter.

The data exploration analysis is then conducted, and the results are written into an output file called "data_exploration.txt".

The code performs the following analysis steps:

1. **Data Size:** The code reports the number of congress members, attributes per member, nodes in the graph, and edges in the graph. These metrics provide an overview of the size of the dataset and the network.
2. **Summary Statistics:** Since there are no continuous variables in this dataset, this section is skipped in the code.
3. **Categorical Variables:** The code explores the categorical variables (State, Party, and Chamber) by providing the count of members per category and identifying the most common category for each variable. This analysis helps understand the distribution and representation of members across different categories.
4. **Missing Values:** The code identifies and reports any missing values

in the congress members' data. This step is important for assessing data quality and determining if any imputation or handling of missing values is required in subsequent analyses.

The information generated from the analysis steps is written into the "data_exploration.txt" file. The file includes the data size metrics, count of members per categorical variable, the most common category for each categorical variable, and any missing values in the dataset.

The results obtained from this data exploration analysis can provide initial insights into the network structure and characteristics of the Twitter interactions among members of the 117th US Congress. These insights can guide further analyses and investigations regarding the dynamics of political interactions on Twitter and their implications.

B.3 Code Output

data_exploration.txt

```
# Data Size
Number of Congress members: 475
Number of attributes per member: 4
Number of nodes in the graph: 475
Number of edges in the graph: 13289
```

```
# Summary Statistics
N/A for this dataset
```

```
# Categorical Variables
Count of members per State:
State
CA      49
TX      32
NY      27
FL      26
IL      17
PA      16
GA      15
OH      15
NC      13
VA      13
MI      13
```


NJ	12
AZ	11
WA	11
SC	10
MN	9
WI	9
MO	9
MD	9
TN	9
MA	9
AL	8
IN	8
LA	8
CO	8
OK	7
CT	7
IA	6
WV	6
KY	6
UT	6
AR	5
NM	5
MS	5
OR	5
NV	5
KS	4
NH	4
ID	4
VT	3
ME	3
WY	3
NE	3
MT	3
RI	3
ND	3
DE	3
HI	2
AK	2
SD	2
PR	1

```
DC      1
VI      1
MP      1
Name: count, dtype: int64
Most common State: CA
```

```
Count of members per Party:
Party
D      250
R      223
I         2
Name: count, dtype: int64
Most common Party: D
```

```
Count of members per Chamber:
Chamber
House      383
Senate      92
Name: count, dtype: int64
Most common Chamber: House
```

```
# Missing Values
Missing values in each column:
Handle      0
State       0
Party       0
Chamber     0
dtype: int64
```

C Data Analysis

C.1 Code

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

```
1
2
3 # IMPORT
4 import pandas as pd
```

```

5 import numpy as np
6 import networkx as nx
7 from scipy.stats import chi2_contingency
8 import pickle
9
10 # LOAD DATA
11 congress_df = pd.read_csv("congress_members.csv")
12 edgelist = nx.read_edgelist('congress_edges.dat',
    ↪ create_using=nx.DiGraph())
13
14 # DATASET PREPARATIONS
15 # No dataset preparations are needed.
16
17 # DESCRIPTIVE STATISTICS
18 # No descriptive statistics table is needed.
19
20 # PREPROCESSING
21 # No preprocessing is needed, because data is already in the
    ↪ required format.
22
23 # ANALYSIS
24 # Create a pairwise data of `State` and `Party` between nodes
    ↪ in each edge
25 edges_data = []
26 for edge in edgelist.edges:
27     source_data = congress_df.iloc[int(edge[0])]
28     target_data = congress_df.iloc[int(edge[1])]
29     edge_data = [source_data.State, target_data.State,
    ↪ source_data.Party, target_data.Party]
30     edges_data.append(edge_data)
31
32 edges_df = pd.DataFrame(edges_data, columns=['SourceState',
    ↪ 'TargetState', 'SourceParty', 'TargetParty'])
33
34 ## Table 1: "Chi-square test of party-wise retweet
    ↪ interaction"
35 # Party-wise retweet interaction
36 party_wise_interaction = pd.crosstab(edges_df.SourceParty,
    ↪ edges_df.TargetParty)

```

```

37 party_wise_stat, party_wise_p, party_wise_dof,
   ↪ party_wise_expctd =
   ↪ chi2_contingency(party_wise_interaction)
38
39 df1 = pd.DataFrame([{'Statistic': party_wise_stat, 'p-value':
   ↪ party_wise_p, 'DoF': party_wise_dof}],
40                     columns=['Statistic', 'p-value', 'DoF'],
41                     index=['Party Interaction Analysis'])
42 df1.to_pickle('table_1.pkl')
43
44
45 ## Table 2: "Chi-square test of state-wise retweet
   ↪ interaction"
46 # State-wise Twitter interaction
47 state_wise_interaction = pd.crosstab(edges_df.SourceState,
   ↪ edges_df.TargetState)
48 state_wise_stat, state_wise_p, state_wise_dof,
   ↪ state_wise_expctd =
   ↪ chi2_contingency(state_wise_interaction)
49
50 df2 = pd.DataFrame([{'Statistic': state_wise_stat, 'p-value':
   ↪ state_wise_p, 'DoF': state_wise_dof}],
51                     columns=['Statistic', 'p-value', 'DoF'],
52                     index=['State Interaction Analysis'])
53 df2.to_pickle('table_2.pkl')
54
55 # SAVE ADDITIONAL RESULTS
56 additional_results = {
57     'Total number of Congress members': len(congress_df),
58     'Total number of edges in the graph': len(edgelist.edges),
59     'Total number of nodes in the graph': len(edgelist.nodes),
60 }
61 with open('additional_results.pkl', 'wb') as f:
62     pickle.dump(additional_results, f)
63

```

C.2 Code Description

The provided code performs an analysis of the interaction network among members of the 117th US Congress on Twitter. The steps involved in the

analysis are as follows:

1. Load the data: - The code reads the data from two files: "congress_members.csv" and "congress_edges.dat". - The first file contains information about Congress members, including their Twitter handles, Represented State, Party, and Chamber. - The second file provides the interaction network between the Congress members on Twitter.

2. Prepare the dataset: - No dataset preparations are required as the data is already in the required format.

3. Perform Descriptive Statistics: - No descriptive statistics table is generated in this analysis.

4. Preprocessing: - No preprocessing is needed because the data is already in the required format.

5. Perform Analysis: - Pairwise data of "State" and "Party" between nodes in each edge are created. - This is done by extracting the relevant information from the Congress member data based on the edge connections.

6. Table 1: "Chi-square test of party-wise retweet interaction": - A contingency table is created to analyze the party-wise retweet interactions. - The contingency table counts the number of retweet interactions between parties. - A chi-square test of independence is performed on the contingency table to determine if there is a significant association between party affiliation of the source and target nodes.

7. Table 2: "Chi-square test of state-wise retweet interaction": - A contingency table is created to analyze the state-wise retweet interactions. - The contingency table counts the number of retweet interactions between states. - A chi-square test of independence is performed on the contingency table to determine if there is a significant association between the represented state of the source and target nodes.

8. Saving Additional Results: - The code saves additional results in the "additional_results.pkl" file. - The additional results include the total number of Congress members, the total number of edges in the graph, and the total number of nodes in the graph.

The code provides insights into the retweet interactions among members of the Congress based on their party affiliation and represented state. The chi-square tests help assess the significance of these associations. The additional results capture important statistics regarding the size and structure of the interaction network.

C.3 Code Output

table_1.pkl

	Statistic	p-value	DoF
Party Interaction Analysis	6590	0	4

table_2.pkl

	Statistic	p-value	DoF
State Interaction Analysis	9465	0	2809

additional_results.pkl

```
{
    'Total number of Congress members': 475,
    'Total number of edges in the graph': 13289,
    'Total number of nodes in the graph': 475,
}
```

D LaTeX Table Design

D.1 Code

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

```
1
2 # IMPORT
3 import pandas as pd
4 from typing import Any, Dict, Optional, Tuple
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 shared_mapping: AbbrToNameDef = {
9     'Statistic': ('Chi-Square Statistic', None),
10    'p-value': (None, 'Probability of observing a value as
   ↪ extreme as or more extreme than the observed value
   ↪ under the null hypothesis'),
11    'DoF': ('Degrees of Freedom', 'Number of independent
   ↪ pieces of information that go into the calculation of
   ↪ a statistic'),
```

```

12 }
13
14 # TABLE 1:
15 df1 = pd.read_pickle('table_1.pkl')
16
17 # FORMAT VALUES
18 df1['p-value'] = df1['p-value'].apply(format_p_value)
19
20 # RENAME ROWS AND COLUMNS
21 mapping = {k: v for k, v in shared_mapping.items() if
    ↪ is_str_in_df(df1, k)}
22 abbrs_to_names, names_to_definitions = split_mapping(mapping)
23 df1 = df1.rename(columns=abbrs_to_names, index=abbrs_to_names)
24
25 # Save as latex:
26 to_latex_with_note(
27     df1, 'table_1.tex',
28     caption="Chi-square test of party-wise retweet interaction.",
29     label='table:party_wise_interaction',
30     note=None,
31     legend=names_to_definitions)
32
33 # TABLE 2:
34 df2 = pd.read_pickle('table_2.pkl')
35
36 # FORMAT VALUES
37 df2['p-value'] = df2['p-value'].apply(format_p_value)
38
39 # RENAME ROWS AND COLUMNS
40 mapping = {k: v for k, v in shared_mapping.items() if
    ↪ is_str_in_df(df2, k)}
41 abbrs_to_names, names_to_definitions = split_mapping(mapping)
42 df2 = df2.rename(columns=abbrs_to_names, index=abbrs_to_names)
43
44 # Save as Latex:
45 to_latex_with_note(
46     df2, 'table_2.tex',
47     caption="Chi-square test of state-wise retweet interaction.",
48     label='table:state_wise_interaction',
49     note=None,

```

```
50     legend=names_to_definitions)
```

```
51
```

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{Chi-square test of party-wise retweet interaction.}
\label{table:party_wise_interaction}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrlr}
\toprule
& Chi-Square Statistic & p-value & Degrees of Freedom \\
\midrule
\textbf{Party Interaction Analysis} & 6.59e+03 &  $1e-06$  & 4 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{p-value}: Probability of observing a value as extreme as or more
extreme than the observed value under the null hypothesis
\item \textbf{Degrees of Freedom}: Number of independent pieces of information
that go into the calculation of a statistic
\end{tablenotes}
\end{threeparttable}
\end{table}

```

table_2.tex

```

\begin{table}[h]
\caption{Chi-square test of state-wise retweet interaction.}
\label{table:state_wise_interaction}
\begin{threeparttable}

```

```

\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrlr}
\toprule
& Chi-Square Statistic & p-value & Degrees of Freedom \\
\midrule
\textbf{State Interaction Analysis} & 9.47e+03 &  $1e-06$  & 2809 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{p-value}: Probability of observing a value as extreme as or more
extreme than the observed value under the null hypothesis
\item \textbf{Degrees of Freedom}: Number of independent pieces of information
that go into the calculation of a statistic
\end{tablenotes}
\end{threeparttable}
\end{table}

```