Geographical and Chamber Influence on Twitter Interactions among US Congress Members

Data to Paper

December 26, 2023

Abstract

Twitter has become a prominent platform for political communication, but little is known about the dynamics of Twitter interactions among US Congress members. To bridge this research gap, we analyze a comprehensive dataset of Twitter interactions among members of the 117th US Congress. We construct a directed graph capturing these interactions and investigate the relationship between Congress members' attributes (Represented State, Political Party, and Chamber) and the occurrence of Twitter interactions. Our findings indicate that geographical location, represented by the state, and the members' chamber do not significantly influence the likelihood of Twitter interactions. These results provide insights into the social dynamics of US Congress members on Twitter and highlight the platform's role in facilitating engagement among politicians. However, our study is limited to a specific dataset and time period, necessitating further research to validate these findings and explore their implications for political discourse and decision-making processes.

Results

Understanding political communication dynamics on Twitter among US Congress members is crucial for gaining insights into regional political dynamics and the role of geographical location in online political discourse. To investigate if Twitter interactions vary depending on the represented state, we conducted a Chi-Squared Test of Independence between Represented State and Occurrence of Twitter Interactions (Table 1). The analysis aimed to determine if there is a significant association between the represented state and the likelihood of Twitter interactions. The results suggest that the represented state does not play a significant role in influencing Twitter

interactions among US Congress members, as the Chi-Squared Statistic was 25175 with a p-value of 0.405. This finding implies that geographical location may not be a primary factor driving interactions among US Congress members on Twitter.

Table 1: Results of Chi-Square Independence Test for Represented State and Twitter Interactions

	Chi-Squared Statistic	p-value
Chi-Squared Test: State and Interactions	25175	0.405

p-value: A measure of statistical significance indicating the strength of the evidence against the null hypothesis

Examining the relationship between the member's chamber and the occurrence of Twitter interactions provides insights into communication patterns among members of the House and Senate. We conducted a Chi-Squared Test of Independence between the Member's Chamber and Occurrence of Twitter Interactions (Table 2) to understand the impact of chamber affiliation on Twitter interactions. The analysis revealed that the member's chamber alone does not appear to have a significant impact on the occurrence of Twitter interactions among US Congress members, with a Chi-Squared Statistic of 475 and a p-value of 0.478. These results reflect that the patterns of engagement on Twitter are independent of whether a member serves in the House or the Senate.

Table 2: Results of Chi-Square Independence Test for Chamber and Twitter Interactions

	Chi-Squared Statistic	p-value
Chi-Squared Test: Chamber and Interactions	475	0.478

p-value: A measure of statistical significance indicating the strength of the evidence against the null hypothesis

In summary, the analysis of the dataset highlights the limited influence of the represented state and the member's chamber on the likelihood of Twitter interactions among members of the 117th US Congress. The findings suggest that Twitter interactions among US Congress members are not strongly determined by geographical location or chamber affiliation alone. These insights contribute to our understanding of the social dynamics of political communication, highlighting the role of Twitter as a platform for engagement among US Congress members.

Taken together, these results provide valuable context for further investigations into the implications of these interactions on political discourse and decision-making processes in the digital era.

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 numpy as np
5 import networkx as nx
```

```
import pickle
from sklearn.preprocessing import LabelEncoder
from scipy.stats import chi2_contingency

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

# DATASET PREPARATIONS
# No dataset preparations are needed.

# DESCRIPTIVE STATISTICS
# No descriptive statistics table is needed.

# PREPROCESSING
# Encoding categorical variables
state_encoder = LabelEncoder()
df_members['State_code'] = state_encoder.fit_transform(df_members['State'])
party_encoder = LabelEncoder()
df_members['Party_code'] = party_encoder.fit_transform(df_members['Party'])
chamber_encoder = LabelEncoder()
df_members['Chamber_code'] = chamber_encoder.fit_transform(df_members['Chamber'])

# ANALYSIS
## Table 1: "Chi-Squared Test of Independence between Represented State and Occurrence of Twitter Interactions"

# Create a contingency table for Twitter Interactions and Represented State
contingency_state_interactions = pd.crosstab(df_members['State_code'], df_edges)

# Perform the Chi-squared test of independence
chi2_stat, p_val, dof, ex = chi2_contingency(contingency_state_interactions)

```

```python
# Create a dataframe for the table with meaningful row labels
df1 = pd.DataFrame({"Chi-Squared Statistic": [chi2_stat], "p-value": [p_val]},
                   index=["Chi-Squared Test: State and Interactions"])
df1.to_pickle('table_1.pkl')

## Table 2: "Chi-Squared Test of Independence between the Member's Chamber and Occurrence of Twitter Interactions"

# Create a contingency table for Twitter Interactions and Chambers
contingency_chamber_interactions = pd.crosstab(df_members['Chamber_code'], df_edges)

# Perform the Chi-squared test of independence
chi2_stat, p_val, dof, ex = chi2_contingency(contingency_chamber_interactions)

# Create a dataframe for the table with meaningful row labels
df2 = pd.DataFrame({"Chi-Squared Statistic": [chi2_stat], "p-value": [p_val]},
                   index=["Chi-Squared Test: Chamber and Interactions"])
df2.to_pickle('table_2.pkl')

# SAVE ADDITIONAL RESULTS
additional_results = {
 'Total Number of Members': df_members.shape[0],
 'Total Number of Twitter Interactions': df_edges.size()
}
with open('additional_results.pkl', 'wb') as f:
    pickle.dump(additional_results, f)
```

C.2 Code Description

The code performs data analysis on the dataset of US Congress's Twitter interactions.

First, the required libraries are imported, including pandas for data ma-

nipulation, numpy for numerical operations, networkx for network analysis, and pickle for saving additional results.

The code then loads the dataset consisting of members of the 117th Congress and their Twitter interactions. The members' data is stored in a DataFrame called "df_members" and the interactions network is stored in a directed graph called "df_edges".

Next, the code encodes categorical variables in the members' DataFrame. The variables "State", "Party", and "Chamber" are encoded using LabelEncoder from the scikit-learn library. The encoded values are stored in new columns "State_code", "Party_code", and "Chamber_code" in the members' DataFrame.

The analysis section of the code includes the calculation of two Chi-Squared tests of independence:

1. Table 1: "Chi-Squared Test of Independence between Represented State and Occurrence of Twitter Interactions" - A contingency table is created to analyze the relationship between state and Twitter interactions. The table cross-tabulates the encoded state values and the occurrence of Twitter interactions. - The code then performs the Chi-Squared test of independence on the contingency table. - The results of the test, including the Chi-Squared statistic and p-value, are stored in a DataFrame called "df1" with a meaningful row label. The DataFrame is saved as a pickle file called "table_1.pkl".

2. Table 2: "Chi-Squared Test of Independence between the Member's Chamber and Occurrence of Twitter Interactions" - A contingency table is created to analyze the relationship between the member's chamber (House or Senate) and Twitter interactions. The table cross-tabulates the encoded chamber values and the occurrence of Twitter interactions. - The code then performs the Chi-Squared test of independence on the contingency table. - The results of the test, including the Chi-Squared statistic and p-value, are stored in a DataFrame called "df2" with a meaningful row label. The DataFrame is saved as a pickle file called "table_2.pkl".

Finally, the code saves additional results in a pickle file called "additional_results.pkl". The additional results include the total number of members in the dataset and the total number of Twitter interactions in the network.

Overall, the code performs data analysis by encoding categorical variables, conducting Chi-Squared tests of independence, and saving the results in pickle files for further analysis and reporting.

C.3 Code Output

table_1.pkl

```
                                                Chi-Squared Statistic p-value
Chi-Squared Test: State and Interactions                       25175.0  0.4055
```

table_2.pkl

```
                                                  Chi-Squared Statistic p-value
Chi-Squared Test: Chamber and Interactions                        475.0  0.4784
```

additional_results.pkl

```
{
    'Total Number of Members': 475,
    'Total Number of Twitter 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 my_utils import to_latex_with_note, format_p_value,
↪  is_str_in_df, split_mapping, AbbrToNameDef
from typing import Dict, Tuple, Any, Optional

# PREPARATION FOR ALL TABLES
mapping: AbbrToNameDef = {
    'Chi-Squared Statistic': ('Chi-Squared Statistic', None),
    'p-value': ('p-value', 'A measure of statistical
    ↪  significance indicating the strength of the evidence
    ↪  against the null hypothesis'),
}

# TABLE 1
# Load table
```

```python
df1 = pd.read_pickle('table_1.pkl')

# Apply value formatting
df1['p-value'] = df1['p-value'].apply(format_p_value)

# Apply renaming
mapping_table1 = {k: v for k, v in mapping.items() if
↪  is_str_in_df(df1, k)}
abbrs_to_names, legend = split_mapping(mapping_table1)
df1 = df1.rename(columns=abbrs_to_names, index=abbrs_to_names)

# Save as latex
to_latex_with_note(
 df1, 'table_1.tex',
 caption="Results of Chi-Square Independence Test for
 ↪  Represented State and Twitter Interactions",
 label='table:chi_squared_test_state_interactions',
 note=None,
 legend=legend
)

# TABLE 2
# Load table
df2 = pd.read_pickle('table_2.pkl')

# Apply value formatting
df2['p-value'] = df2['p-value'].apply(format_p_value)

# Apply renaming
mapping_table2 = {k: v for k, v in mapping.items() if
↪  is_str_in_df(df2, k)}
abbrs_to_names, legend = split_mapping(mapping_table2)

df2 = df2.rename(columns=abbrs_to_names, index=abbrs_to_names)

# Save as latex
to_latex_with_note(
 df2, 'table_2.tex',
 caption="Results of Chi-Square Independence Test for Chamber
 ↪  and Twitter Interactions",
```

```
 label='table:chi_squared_test_chamber_interactions',
 note=None,
 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]]]

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

```
\begin{table}[h]
\caption{Results of Chi-Square Independence Test for Represented State and
    Twitter Interactions}
\label{table:chi_squared_test_state_interactions}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrl}
\toprule
 & Chi-Squared Statistic & p-value \\
\midrule
\textbf{Chi-Squared Test: State and Interactions} & 25175 & 0.405 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{p-value}: A measure of statistical significance indicating the
    strength of the evidence against the null hypothesis
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_2.tex

```
\begin{table}[h]
\caption{Results of Chi-Square Independence Test for Chamber and Twitter
```

```
    Interactions}
\label{table:chi_squared_test_chamber_interactions}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrl}
\toprule
 & Chi-Squared Statistic & p-value \\
\midrule
\textbf{Chi-Squared Test: Chamber and Interactions} & 475 & 0.478 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{p-value}: A measure of statistical significance indicating the
    strength of the evidence against the null hypothesis
\end{tablenotes}
\end{threeparttable}
\end{table}
```