Case Study: Finding Factions from Ukrainian Legislative Data

Tom Magelinski
tmagelin@andrew.cmu.edu

The Problem

- Using parliamentary voting data to analyze a government
- How do bills differ from one another?
- Which parliamentarians cooperate?
- Questions like these can be answered using networks
  - Specifically using ORA
- Ukrainian parliament has interesting structure
  - 8 official party affiliations + some MPs with no affiliation
    - Divisions not as clear as those in governments like U.S.
  - 6 potential voting options (for, against, and 4 types of abstain)
Ukrainian Factions

- Ongoing research in CASOS to look at all bills to understand factions and how they change

- We’ll looked at 2 bills here
  - makes things easy to interpret / visualize

Skills Used

- Analyze bipartite network data with symbolic weights

- Clean data with ORA

- Using Link Types
  - Network Unions

- Fold networks
  - Turning bipartite networks to unipartite networks

- Visual network insights
  - Analyze networks and their attributes
  - Partial visualizations of data for better insights
Bipartite Networks & Symbolic Links

- Bipartite: network connecting one nodeset to another, with no connections between
  - MPs (nodeset 1) are connected to Bills (nodeset 2) based on their vote

- Weights often represent strength or distance, but not always

- Symbolic weights are also useful
  - Symbolic weights can represent the type of connection (for, against a bill, for example)

- Symbolic weights must be treated differently
  - We’ll show how to manipulate and compare them

No connections between MPs, or between Bills
- Good only for MP-Bill Analysis (popularity)
  - Bill 1 is more popular here
Bipartite Networks & Symbolic Links

- For symbolic weights, visualization per link type is usually most interpretable

Unipartite Analysis (Folding)

- For conclusions within a nodeset, we need a unipartite graph
  - MP x MP or Bill x Bill

- This is done through folding
  - Matrix multiplication of the adjacency matrix with its transpose

\[ A^T_{MP\times MP} = A^T_{MP\times Bill} \times A_{MP\times Bill} = A_{MP\times Bill} \times A_{Bill\times MP} \]

- \( A^T_{MP\times MP} \) is the adjacency matrix for the MP to MP network, where links are weighted by number of bills they agreed upon
Unipartite Analysis (Folding)

• Now we can compare MPs to each other

Folding with Symbolic Weights

• Folding assumes weights are not symbolic

• ORA: use symbolic weights to construct separate networks
  - MP x Bill (Only votes for)
  - MP x Bill (Only votes against)
  - Etc

• Fold these separately
  - MP x MP (weights = #bills both voted “for”)
  - MP x MP (weights = #bills both voted “against”)

• Add them
  - MP x MP (weights = #bills with same vote of any kind)
Look at the Data

- Open in Excel
- `ukrainian_sample_votes.csv`:

<table>
<thead>
<tr>
<th>Source Node</th>
<th>Target Node</th>
<th>Link Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1152</td>
<td>25407 [10]</td>
<td>3</td>
</tr>
<tr>
<td>2329</td>
<td>3371-1 [1]</td>
<td>3</td>
</tr>
</tbody>
</table>

- `ukrainian_sample_MPs.csv`:

<table>
<thead>
<tr>
<th>Name</th>
<th>faction</th>
<th>gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abdullin Alexander Raftakovych</td>
<td>All-Ukrainian Association (Fatherland)</td>
<td>male</td>
</tr>
<tr>
<td>Agafonova Natalia Volodymyrivna</td>
<td>Peter Porchenko Bloc</td>
<td>female</td>
</tr>
</tbody>
</table>

Import Data into ORA

- Open data import wizard:

  - “Import excel or text delimited files”
  - “Table of network links”
  - “Next” in bottom right corner
Import Data into ORA

- Give your network a name:
  - Create a new meta-network with name: UKR Votes
- “Next” in bottom right corner
- Select your file path:
  - Step 1: Select a file containing table data with column headers:
  - C:/Users/imagelini/Desktop/ai_case_study/ukrainian_sample_votes.csv

Import Data into ORA

- Step 2
- Under SOURCE NODE:
  - Node Names
  - Nodeset Class: Agent
  - Nodeset Name: MP
- Under TARGET NODE:
  - Node Names
  - Nodeset Class: Belief
  - Nodeset Name: Bill
Import Data into ORA

• Step 3

• Hit “New”

• Match dropdowns like below:

• Hit “Next” then “Finish”

Import Data into ORA

• Reopen Data Import Wizard

• “Table of Node Attributes”

• “Add to your existing meta-network”
  – MPs only

• Hit “browse” and find ‘ukrainian_sample_MPs.csv’
Import Data into ORA

- Match the values below:

   ![Import Data into ORA-NetScenes](image)

   **Step 1:** Select an attributes file:
   - C:\Users\magelink\Desktop\case_study\ukrainian_sample_MPs.csv
   
   **Step 2:** Select how to identify the node(s) to get attribute values from a line of the file:
   - □ Match node name with file column Name
   - □ Match node title with file column
   - □ Match node attribute with the value from file column
   - □ Nodes are in the same order as the file

   **Step 3:** Select the columns of the file to import as attribute values:
   - Name Type: Text Category
   - Name faction Type: Text Category
   - Name gender Type: Text Category

   ![Select columns](image)

- **METHOD 1:**
  **BIPARTITE ANALYSIS**
  **(AGENT-BILL NETWORK)**
Clean Data

- A look at the readme.txt shows that there are 6 voting options

- For this study, we only care about votes “for” or linkweight=3

- Goal: create 2 binary networks
  - Agent-Bill connected with “for” votes
  - Agent-Bill connected with “non-for” votes

- Method: Use network unions

Clean Data: Rename Votes For

- Our “3” network encodes links from “for” votes

- Simply rename this as “Votes For”
Clean Data: Votes Against

- Now, we want to combine all other networks into one
- Use a network union, summing the values

Visualize the Agent-Bill Network

- “Visualize only this network” on the votes-for network

Think of it as a vote “for” Venn diagram:

- Bill 1 Only
- Both
- Bill 2 Only
- Neither
Color by Attribute

- "Color Nodes by Attribute"

- Select “faction” and “apply changes”

- Focus on the ratios, and what colors are \textit{not} present
Conclusions About Bills

- **Bill 1**
  - More votes for
  - Favored by Presidential Party, Radical Party, UNION

- **Bill 2**
  - Less popular
  - Favored by Opposition bloc, Revival

- **Overall**
  - Seem like opposing bills (not much overlap, opposing parties)
  - Party bias noticeable but far from perfect

METHOD 2: UNIPARTITE ANALYSIS (AGENT-AGENT NETWORK)
Constructing the Agent-Agent Network

- MP-Bill network might not be the best
- Some aspects counter intuitive
  - “isolates” actually linked to single vote “for” MPs
- Visualization less useful with more than 3 bills
- Use MP-MP network instead
  - Link weight is the number of times two MPs agreed on a bill
  - Need to add instances of voting “for” together and voting “against” together
- Better to answer questions about MPs instead of questions about bills

Constructing the Agent-Agent Network

- Fold vote “for” network:
  - Rename output and press “fold”
- Repeat with “against” network
Constructing the Agent-Agent Network

- Load networks into matrix algebra:
- Add Networks:

Visualize the Agent-Agent Network

- “Visualize only this network”
- “Load normally”
- Agents can agree between 0 and 2 times
  - Want to only see strongest ties (weight = 2)

- Make sure the box is checked!!
Visualize the Agent-Agent Network

- Increase node size and decrease link weight using arrows

- Color by faction:

Conclusions about MPs

- MPs affiliated with the opposition block vote together, and rarely with others

- MPs not affiliated with a faction are spread over all the groups

- Presidential party members mostly in one group, but there are members in all the other groups

- Grouping not fully defined by parties
  - More interesting results from more data
Overall Conclusions

- Matrix algebra / link operations are extremely useful
  - Especially for symbolic links
  - Separate a network into multiple networks (for/against)

- Must be careful visualizing bipartite data
  - Especially with symbolic weighting

- Folding a network can be used to answer different research questions

- Network visualization is quick and powerful
  - Especially for network attributes

Research on Factions

- When all available bills are studied, networks get more complex

- Not all bills are equal, so we have developed weighting strategies to get the most meaningful connections

- Community detection algorithms are used to find “factions”
Research on Factions

- Faction dynamics are used to find change points
- Look at “snapshots” of a network and compare similarity
- Change-point seen here is the revolution