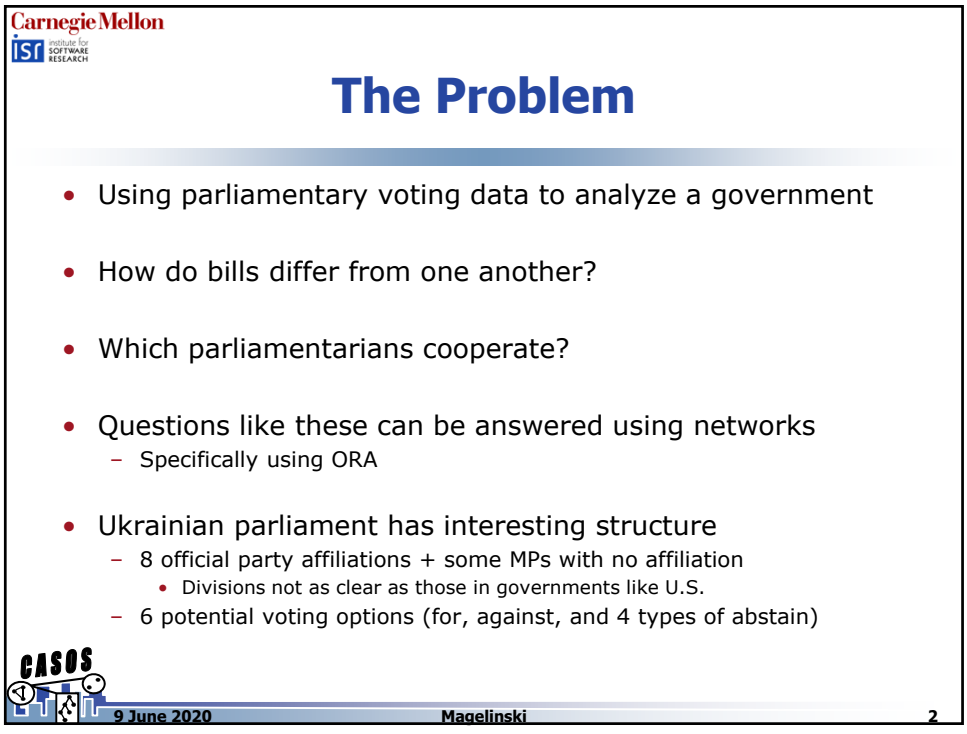
<Your Name>


CASOS
Case Study:
Finding Factions from Ukrainian Legislative Data
Tom Magelinski
tmagelin@andrew.cmu.edu
institute for SOFTWARE RESEARCH
Carnegie Mellon
Center for Computational Analysis of Social and Organizational Systems
http://www.casos.cs.cmu.edu/


Carnegie Mellon
institute for SOFTWARE RESEARCH
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)
CASOS
9 June 2020
Magelinski
2

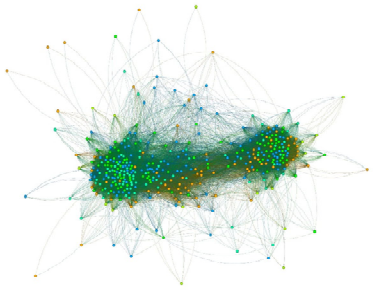

CASOS

<Your Name>

Carnegie Mellon
ISR institute for SOFTWARE RESEARCH

# 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





Carnegie Mellon
ISR institute for SOFTWARE RESEARCH

# 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



CASOS

<Your Name>

Carnegie Mellon
ISR institute for SOFTWARE RESEARCH

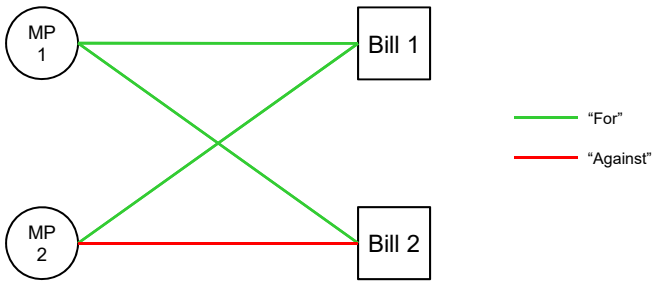
# Bipartite Networks & Symbolic Links

- Bipartite: network connecting one nodeset to another, with no connections between
  - MP's (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



Carnegie Mellon
ISR institute for SOFTWARE RESEARCH

# Bipartite Networks & Symbolic Links

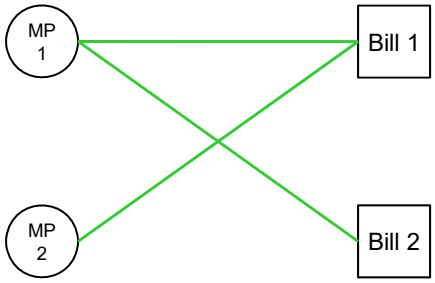



- No connections between MPs, or between Bills
  - Good only for MP-Bill Analysis (popularity)
    - Bill 1 is more popular here

<Your Name>

Carnegie Mellon
isr institute for SOFTWARE RESEARCH

# Bipartite Networks & Symbolic Links




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



Carnegie Mellon
isr institute for SOFTWARE RESEARCH

# 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

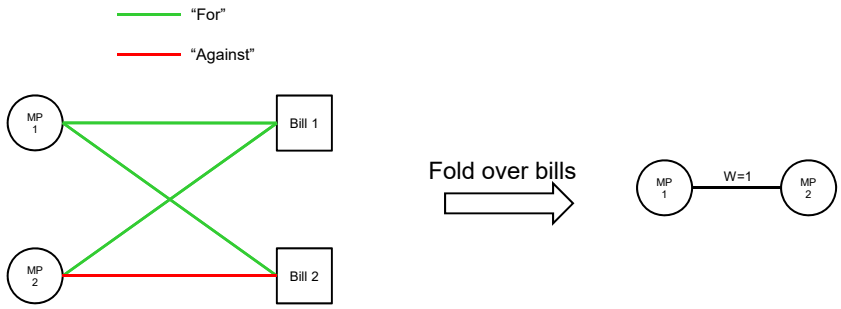
$$A_{MP\,x\,MP} = A_{MP\,x\,Bill} * A^{T}_{MP\,x\,Bill} = A_{MP\,x\,Bill} * A_{Bill\,x\,MP}$$

  - $A_{MP\,x\,MP}$ is the adjacency matrix for the MP to MP network, where links are *weighted* by number of bills they agreed upon

<Your Name>

Carnegie Mellon
ISR institute for SOFTWARE RESEARCH

# Unipartite Analysis (Folding)




- Now we can compare MPs to each other



Carnegie Mellon
ISR institute for SOFTWARE RESEARCH

# 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)

<Your Name>

Carnegie Mellon
institute for SOFTWARE RESEARCH

# Look at the Data

- Open in Excel
- 'ukrainian_sample_votes.csv' :

| | Source Node | Target Node | Link Type |
|---|---|---|---|
| 1152 | Masorina Olena Sergeevna | 2540? [10] | 3 |
| 2329 | Masorina Olena Sergeevna | 3371-1 [1] | 3 |

- 'ukrainian_sample_MPs.csv' :

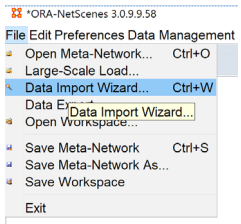
| | Name | faction | gender |
|---|---|---|---|
| 0 | Abdullin Alexander Raftakatovich | All-Ukrainian Association (Fatherland) | male |
| 1 | Agafonova Natalia Volodymyrivna | Peter Porchenko Bloc | female |




Carnegie Mellon
institute for SOFTWARE RESEARCH

# Import Data into ORA

- Open data import wizard:




- "Import excel or text delimited files"
- "Table of network links"
- "Next" in bottom right corner




<Your Name>

Carnegie Mellon
institute for SOFTWARE RESEARCH

# 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\tmagelin\Desktop\si_case_study\ukrainian_sample_votes.csv Browse



Carnegie Mellon
institute for SOFTWARE RESEARCH

# Import Data into ORA

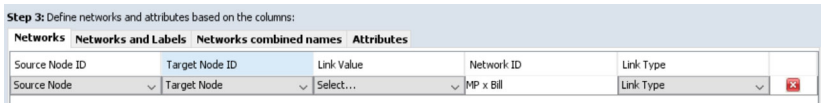
- Step 2
- Under SOURCE NODE:
  - Node Names
  - Nodeset Class: Agent
  - Nodeset Name: MP
- Under TARGET NODE:
  - Node Names
  - Nodeset Class: Belief
  - Nodeset Name: Bill

<Your Name>

Carnegie Mellon
isr institute for SOFTWARE RESEARCH

# Import Data into ORA

- Step 3
- Hit "New"
- Match dropdowns like below:




- Hit "Next" then "Finish"



Carnegie Mellon
isr institute for SOFTWARE RESEARCH

# 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'



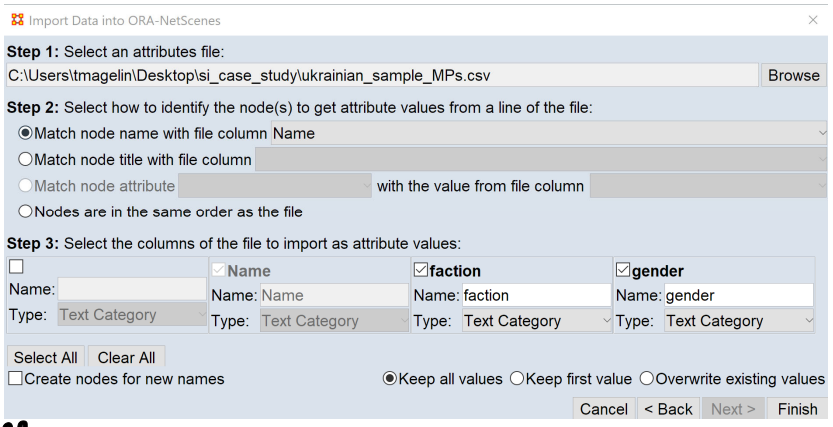
CASOS

<Your Name>

Carnegie Mellon
institute for SOFTWARE RESEARCH

# Import Data into ORA

- Match the values below:






Carnegie Mellon
institute for SOFTWARE RESEARCH

# METHOD 1: BIPARTITE ANALYSIS (AGENT-BILL NETWORK)

<Your Name>

Carnegie Mellon
isr institute for SOFTWARE RESEARCH

# Clean Data

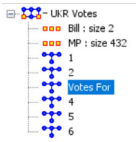
- A look at the readme.txt shows that there are 6 voting options
- For this study, we only care about votes "for" or linkweight=3
- Goal: create 2 binary networks
  - Agent-Bill connected with "for" votes
  - Agent-Bill connected with "non-for" votes
- Method: Use network unions



Carnegie Mellon
isr institute for SOFTWARE RESEARCH

# Clean Data: Rename Votes For

- Our "3" network encodes links from "for" votes
- Simply rename this as "Votes For"

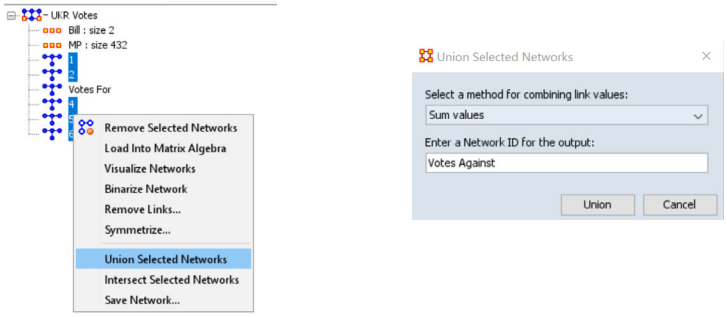


<Your Name>

Carnegie Mellon
institute for SOFTWARE RESEARCH

# Clean Data: Votes Against

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

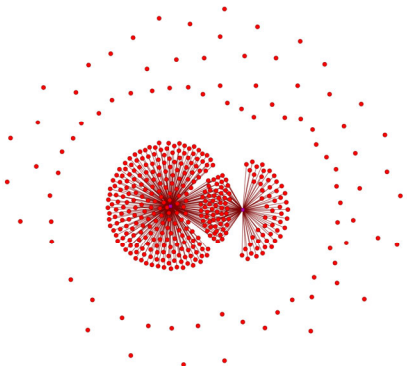




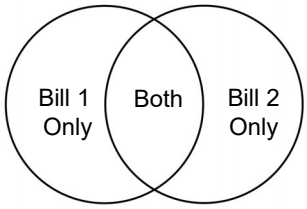

Carnegie Mellon
institute for SOFTWARE RESEARCH

# Visualize the Agent-Bill Network

- "Visualize only this network" on the votes-for network



Think of it as a vote "for" Venn diagram:

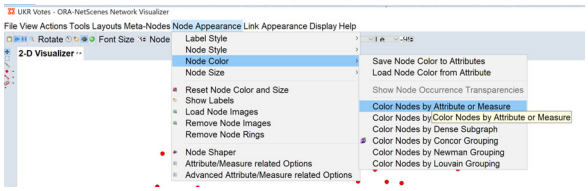



Neither

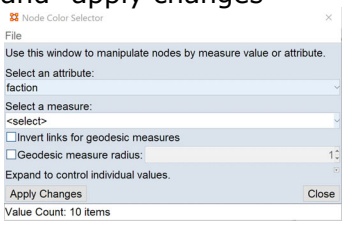
<Your Name>

# Color by Attribute

- "Color Nodes by Attribute"



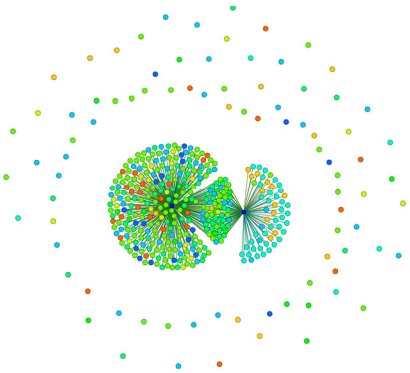

- Select "faction" and "apply changes"






# Color by Attribute

- "Color Nodes by Attribute"



- Focus on the ratios, and what colors are *not* present

<Your Name>

Carnegie Mellon
institute for SOFTWARE RESEARCH

# 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



Carnegie Mellon
institute for SOFTWARE RESEARCH

# METHOD 2: UNIPARTITE ANALYSIS (AGENT-AGENT NETWORK)

<Your Name>

Carnegie Mellon
institute for SOFTWARE RESEARCH

# Constructing the Agent-Agent Network

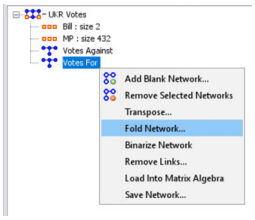
- MP-Bill network might not be the best
- Some aspects counter intuitive
  - "isolates" actually linked to single vote "for" MPs
- Visualization less useful with more than 3 bills
- Use MP-MP network instead
  - Link weight is the number of times two MPs agreed on a bill
  - Need to add instances of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



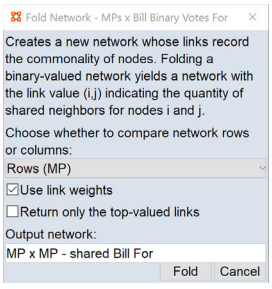
Carnegie Mellon
institute for SOFTWARE RESEARCH

# Constructing the Agent-Agent Network

- Fold vote "for" network:



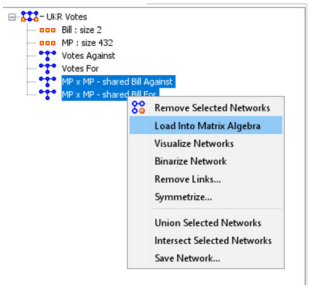

- Rename output and press "fold"
- Repeat with "against" network



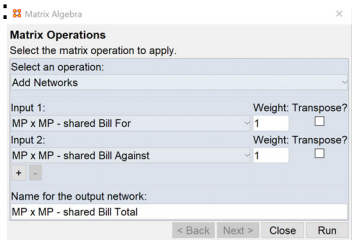
<Your Name>

# Constructing the Agent-Agent Network

- Load networks into matrix algebra:



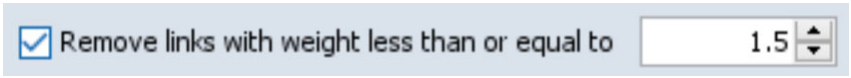

- Add Networks:






# 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)



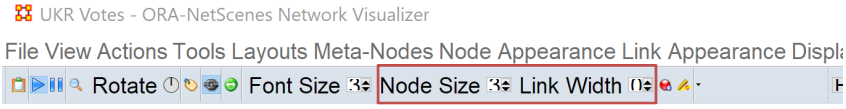

- Make sure the box is checked!!

<Your Name>

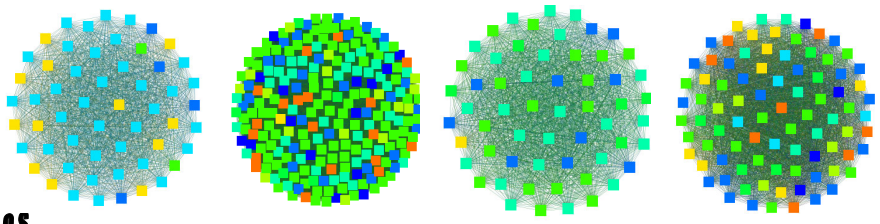
Carnegie Mellon
institute for SOFTWARE RESEARCH

# Visualize the Agent-Agent Network

- Increase node size and decrease link weight using arrows




- Color by faction:





Carnegie Mellon
institute for SOFTWARE RESEARCH

# 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



CASOS

<Your Name>

Carnegie Mellon
institute for SOFTWARE RESEARCH

# Overall Conclusions

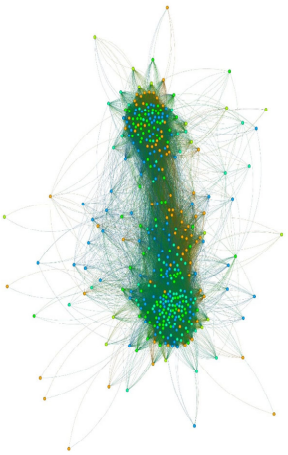
- Matrix algebra / link operations are extremely useful
  - Especially for symbolic links
  - Separate a network into multiple networks (for/against)
- Must be careful visualizing bipartite data
  - Especially with symbolic weighting
- Folding a network can be used to answer different research questions
- Network visualization is quick and powerful
  - Especially for network attributes



Carnegie Mellon
institute for SOFTWARE RESEARCH

# 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"

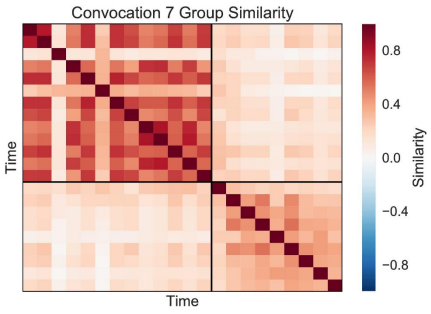


<Your Name>

Carnegie Mellon
isr institute for SOFTWARE RESEARCH

# 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






CASOS