



Case Study: Ukrainian Voting Data

Tom Magelinski

tmagelin@andrew.cmu.edu



Carnegie Mellon

Center for Computational Analysis of
Social and Organizational Systems
<http://www.casos.cs.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
 - Not as partisan as governments like U.S.
 - 6 potential voting options (for, against, and 4 types of abstain)



13 June 2019

Magelinski

2



Skills Used

- Analyze bipartite network data with symbolic weights
- Clean data with ORA
- Matrix algebra
 - Copy networks
 - Transform networks
- Fold networks
 - Turning bipartite networks to unipartite networks
- Visual network insights
 - Analyze networks and their attributes
 - Partial visualizations of data for better insights



13 June 2019

Magelinski

3

Look at the Data

- Open in excel
- 'ukrainian_sample_votes.csv' :

	Source Node	Target Node	Link Value
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



13 June 2019

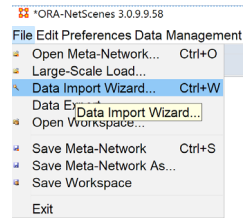
Magelinski

4



Import Data into ORA

- Open data import wizard:



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



13 June 2019

Magelinski

5

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\magelin\Desktop\si_case_study\ukrainian_sample_votes.csv

Browse



13 June 2019

Magelinski

6



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



13 June 2019

Magelinski

7

Import Data into ORA

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

Step 3: Define networks and attributes based on the columns:

Networks Networks and Labels Networks combined names Attributes

Source Node	Target Node	Link Value	Network	Network Column
Source Node	Target Node	Link Value	MP x Bill	Select

New Clear

- Hit "Next" then "Finish"



13 June 2019

Magelinski

8



Import Data into ORA

- Reopen Data Import Wizard
- "Table of Node Attributes"
- "Add to your existing metanetwork"
 - MPs only
- Hit "browse" and find 'ukrainian_sample_MPs.csv'



13 June 2019

Magelinski

9

Import Data into ORA

- Match the values below:

Import Data into ORA-NetScenes

Step 1: Select an attributes file:
 C:\Users\lrmagelin\Desktop\si_case_study\ukrainian_sample_MPs.csv Browse

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:

	<input checked="" type="checkbox"/> Name	<input checked="" type="checkbox"/> faction	<input checked="" type="checkbox"/> gender
Name:	Name: Name	Name: faction	Name: gender
Type: Text Category	Type: Text Category	Type: Text Category	Type: Text Category

Select All Clear All

☐ Create nodes for new names ☒ Keep all values ☐ Keep first value ☐ Overwrite existing values

Cancel < Back Next > Finish



13 June 2019

Magelinski

10



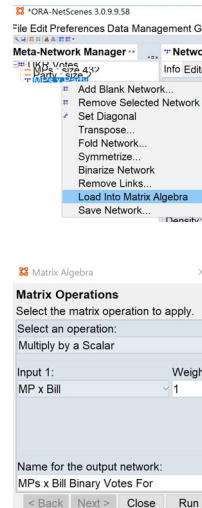
METHOD 1: BIPARTITE ANALYSIS (AGENT-BILL NETWORK)

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 link operations

Clean Data: Copy Network

- Load network into matrix algebra:
- Multiply network by 1, and run with new name:
- Run 2 times with different names, "MPs x Bill Binary Votes For" and "MPs x Bill Binary Votes Against"



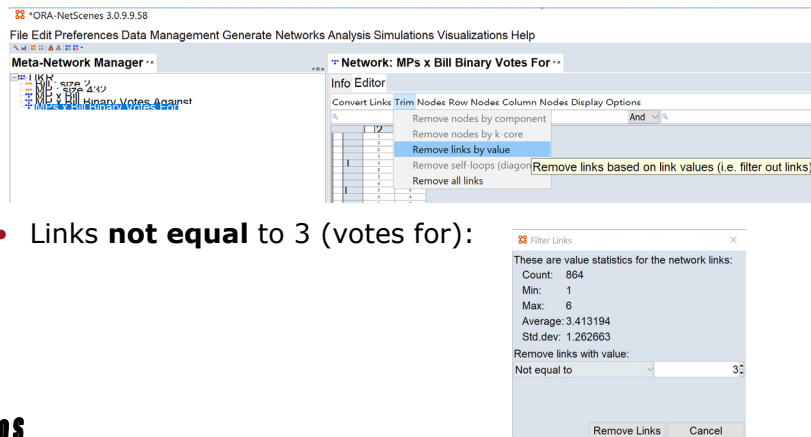
13 June 2019

Magelinski

13

Clean Data: Trim Links

- Remove links by value:
- Links **not equal** to 3 (votes for):



13 June 2019

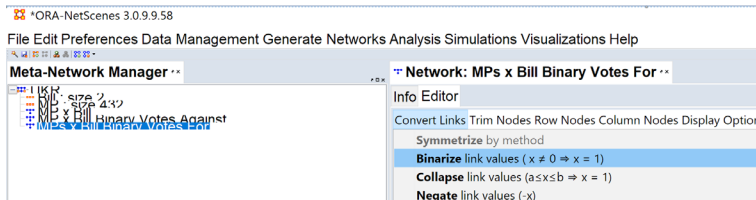
Magelinski

14



Clean Data: Binarize Network

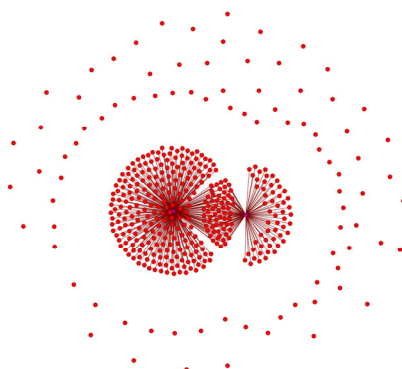
- Binarize network:



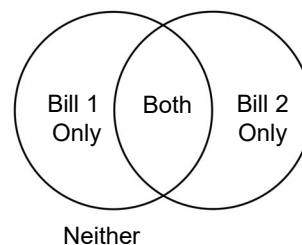
- Done!
- Repeat the last 2 steps for the vote *against* network, this time selecting "equal to" instead of "not equal to"

Visualize the Agent-Bill Network

- "Visualize only this network"

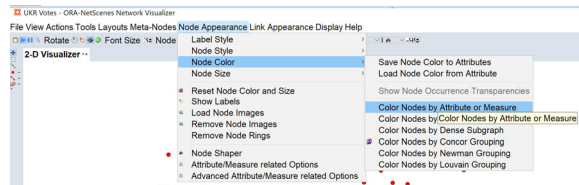


Think of it as a vote
"for" Venn diagram:

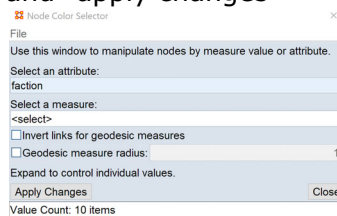


Color by Attribute

- “Color Nodes by Attribute”

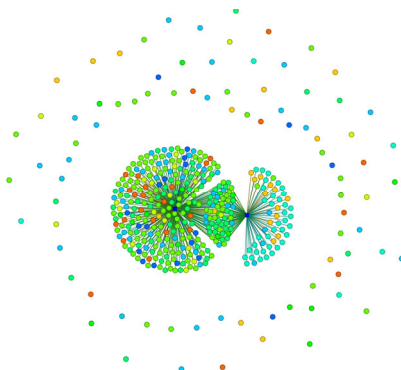


- Select “faction” and “apply changes”



Color by Attribute

- “Color Nodes by Attribute”



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

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



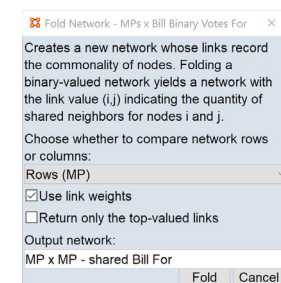
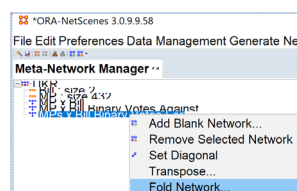
13 June 2019

Magelinski

21

Constructing the Agent-Agent Network

- Fold vote "for" network:
- Rename output and press "fold"
- Repeat with "against" network



13 June 2019

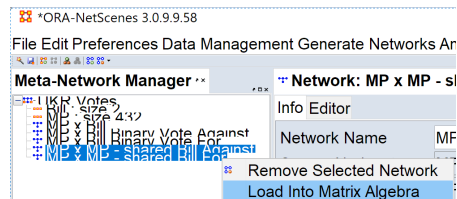
Magelinski

22

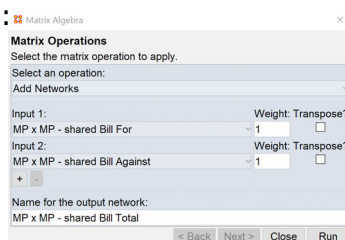


Constructing the Agent-Agent Network

- 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)

☒ Remove links with weight less than or equal to

- Make sure the box is checked!!

This says 1.5



Visualize the Agent-Agent Network

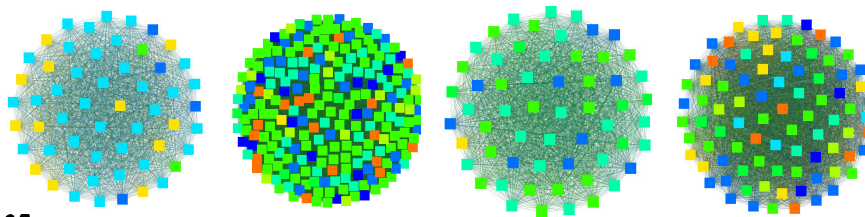
- Increase node size and decrease link weight using arrows

UKR Votes - ORA-NetScenes Network Visualizer

File View Actions Tools Layouts Meta-Nodes Node Appearance Link Appearance Display

Rotate Font Size Node Size Link Width

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

