Wikipedia Elections

Studying social networks with graphs

Table of Contents

Project Overview

Wikipedia is a free encyclopedia written collaboratively by users from all over the world. However, a small part of the contributors are administrators (users with certain permissions for maintenance)
In order for a user to become an administrator, a Request for adminship (RfA) is issued and the Wikipedia community via a public discussion or a vote decides who to promote to adminship.

This project aims to study the complexity of social interactions by utilizing weighted networks and graph scoring.

Inspiration:

- initially driven by technology (not data)
- agnostic data (e.g., user 3 voted for user 25 as neutral (0)) that allow to withhold any biases and assumption

Dataset I - Initial Format & Assumptions

Wikipedia elections (http://cs.stanford.edu/people/jure/pubs/triads-chi10.pdf).
2,800 elections with around 100,000 total votes and about 7,000 users participating in the elections.

Initial Data format:

E: is election successful (1) or not (0)
T: time election was closed
U: user id (and username) of editor that is being considered for promotion
N: user id (and username) of the nominator
V: <vote(1:support, 0:neutral, -1:oppose)> <user_id> <time> <username>

Assumptions:

1. Balanced data (complete data)
2. Equally weighted and registered data

```
E   1
T   2004-09-21 01:15:53
U   30  cjcurrie
N   32  andyl
V   1   3   2004-09-14 16:26:00 ludraman
V   -1  25  2004-09-14 16:53:00 blankfaze
V   1   4   2004-09-14 17:08:00 gzornenplat
V   1   5   2004-09-14 17:37:00 orthogonal
V   1   6   2004-09-14 19:28:00 andrevan
V   1   7   2004-09-14 19:37:00 texture
V   1   8   2004-09-14 21:04:00 lst27
V   1   9   2004-09-14 21:30:00 mirv
V   1   10  2004-09-14 22:13:00 anv°rion
V   0   26  2004-09-14 22:18:00 grunt
V   0   27  2004-09-15 03:19:00 slowking
V   0   28  2004-09-15 03:20:00 neutrality
V   1   11  2004-09-15 04:28:00 merovingian
V   1   12  2004-09-15 06:56:00 wile
V   1   13  2004-09-15 09:19:00 sjc
V   1   14  2004-09-15 12:20:00 172
V   0   29  2004-09-16 00:58:00 ugen64
V   1   15  2004-09-16 14:50:00 danny
V   1   16  2004-09-16 15:31:00 simonp
V   1   17  2004-09-17 13:49:00 jwrosenzwei
V   1   18  2004-09-17 20:57:00 adam
V   1   19  2004-09-17 22:11:00 ffirehorse
V   1   20  2004-09-18 00:02:00 michael
V   1   21  2004-09-18 01:06:00 rhymeless
V   1   22  2004-09-20 05:36:00 bearcat
V   1   23  2004-09-20 14:28:00 cryptoderk
V   1   24  2004-09-20 22:20:00 jayjg
```

Dataset II - Mutation

Initial Data Format
(multi-dimensional)

→

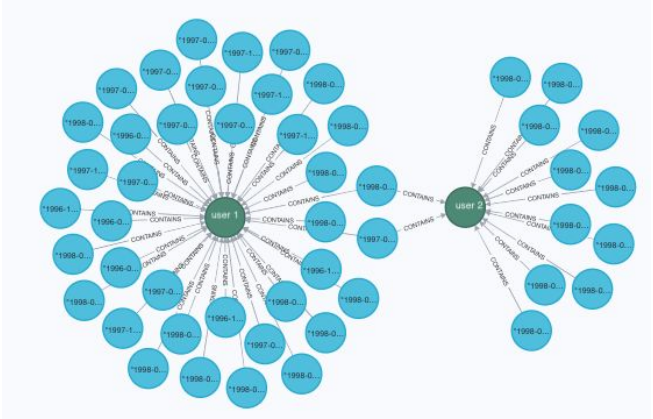
Entity-Centric Data
(Election Data vs Voter Data vs Candidate Data)

→

Graph Data
(Weighted and Directional Data)

```
E   1
T   2004-09-21 01:15:53
U   30  cjcurrie
N   32  andyl
V   1   3   2004-09-14 16:26:00 ludraman
```

→



Project Implementation Methodology

1. Data Mutation in 2 steps:
    a. Flatten data in 2 directions
    b. Construct directed and weighted relationship between data entities
2. Build ML model for identifying clustering of connected groups (socal clicks)
3. Based on ML groups identification introduce 3rd affecting variable for study of dynamics
4. Analysis & Findings: Identify voting patterns

Tools, Techniques, and Theory

- Python and Pandas for Data Manipulation
- Neon4j for Graph Data
- K-Means algorithm to identify groups and clusters
- Dijkstra Algorithm Implementation (in C++ because I re-purposed my prior work) for testing results
- “Signed Networks in Social Media” by Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg
- “Predicting Positive and Negative Links in Online Social Networks” by Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg

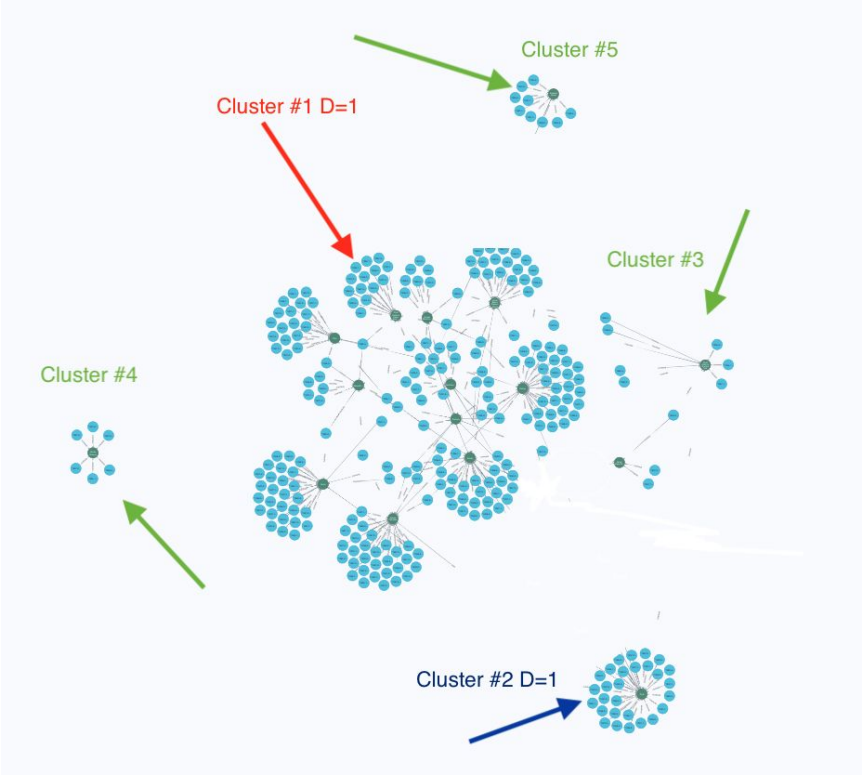
Results & Findings - Clusters

5 Main clusters Identified:

- Cluster #1 influenced 75% of all election results
- Cluster #2 influenced 11% of all election results
- Clusters #3, #4, and #5 influenced the remaining 14% of all election results

Findings (based on assumption that data is complete and equally weighted):

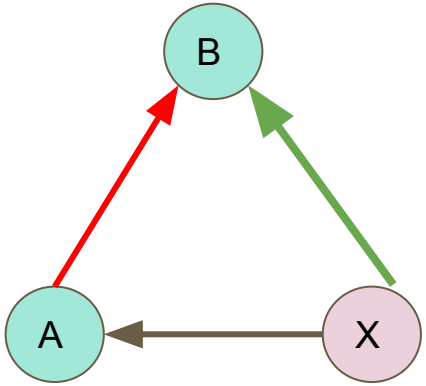
1. Users tend to vote for users that they know
2. Reciprocal behavior could lead to bias OR sign of acknowledgement
3. Closed community could lead to limited information on a subject



Results & Findings - Theories of signed networks

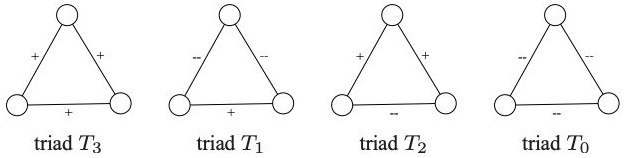
"You scratch my back and I'll scratch yours" OR Theories of signed networks (balance)

Nodes - 7,118
Edges - 103,747
+ edges - 72.3%
− edges - 27.7%
Triads - 48,567



		Wikipedia			
T_3	+ + +	555,300	0.702	0.489	379.6
T_1	+ − −	163,328	0.207	0.106	289.1
T_2	+ + −	63,425	0.080	0.395	-572.6
T_0	− − −	8,479	0.011	0.010	10.8

Data from "Signed Networks in Social Media" by Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg



Findings (based on assumption that data is complete and equally weighted

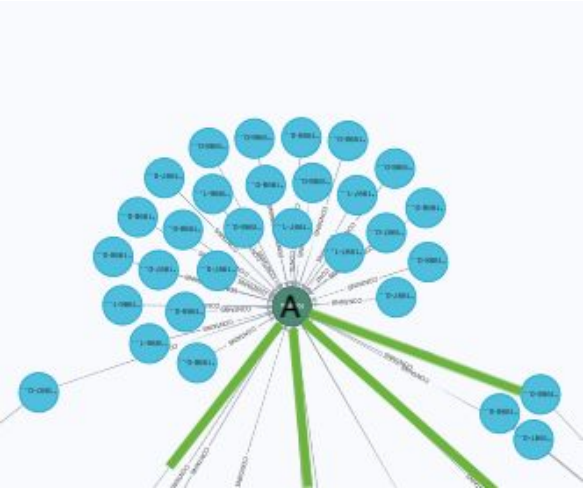
More in depth analysis of various triads is necessary to draw a concrete conclusion, but based on the data from + edges and cluster sizes (see previous slide) we can conclude that Wikipedia voting database confirms Theory of signed network (balance)

Results & Findings - Theory of Positive Nodes

Polarized data (all positive and all negative nodes)

Theory assumption: if node A gets only positive votes, it will vote only positive as well.

Nodes - 7,118
Edges - 103,747
+ nodes - 1.8%
− nodes - 5.6%
Triads +++ 3,312
Triads --- 7,124



Reflection/Limitation of the project

Critique of the project:

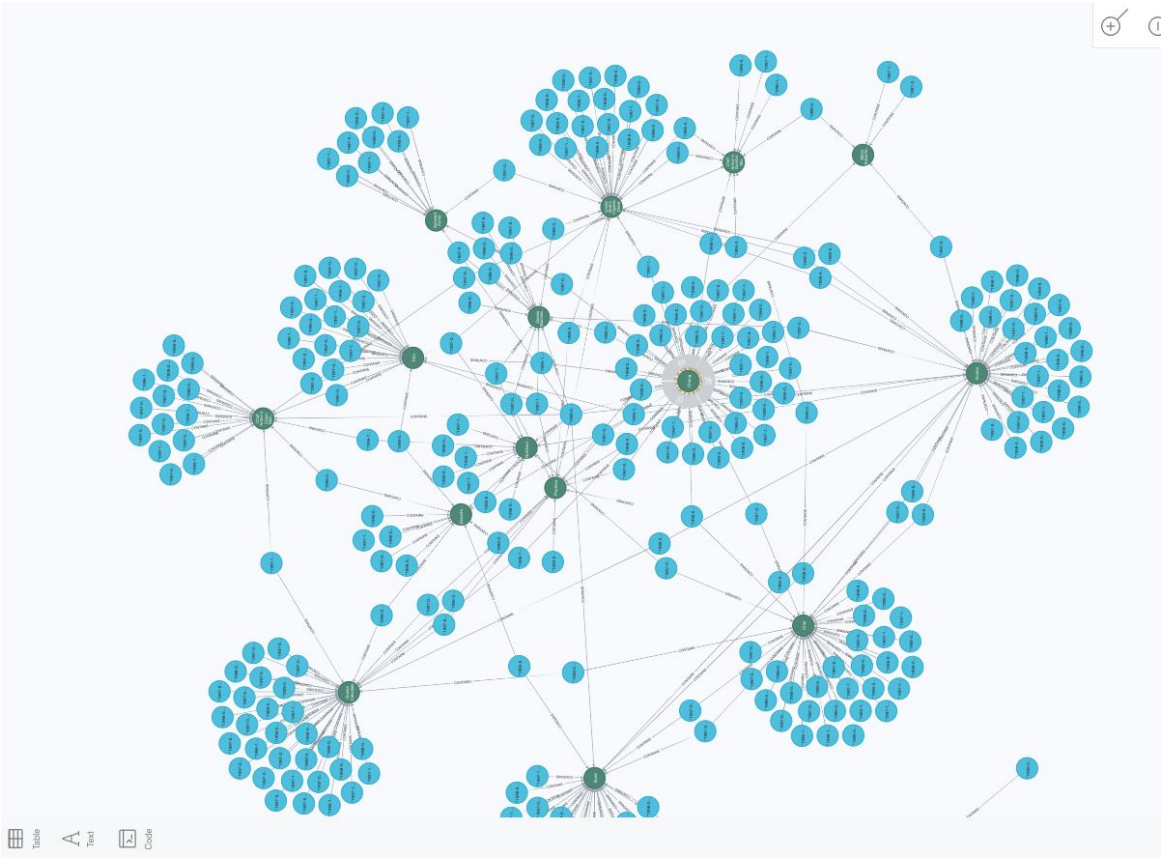
1. Subject /Data Selection
2. Not sure if K-means was the right algorithm for cluster identification
3. Although data is weighted, it's weight is not fully explored in the project

Main Takeaways:

1. Graph data for large and weighted dataset is NOT user-friendly
2. Graph analysis requires post-validation
3. Graph data is extremely interactive and excellent for testing specific hypothesis
4. Graph Data is excellent for defining the strength and depth of relationship
5. Neo4j - very user-friendly and easy tool to use
6. Graph databases is a tool for testing hypothesis for SME (subject matter experts)

Challenges

Trying to draw and test ANY hypothesis from THIS :)



Wishlist

1. Blend Graph Data with powerful clustering ML algorithm
2. Leverage weight of relationship into analysis
3. Perform a controlled performance comparison between hard-coded Dijkstra Algorithm vs Neo4j and Graph API
4. Develop full-stack application with Graph Data

THANK YOU!