Vote Prediction Models for Signed Social Networks

Ananth Mahadevan

Aalto University

May 21, 2020

Overview

Voting and Signed Networks

Voting in Communities

- Communities need to take collective action
- Voting is a popular method
- Members of the community vote on the agenda
- E.g.
  - Politicians voting for bills in the parliament
  - Wikipedia users voting for promoting administrators
- Understanding voting behaviour is beneficial
- Can propose agendas items which will be be successful
- Identify ideological fault lines amongst members

Votes as Signed Graphs

- Votes are usually for or against an agenda
- Intuitively maps to positive and negative edges in signed graphs
- More tools to analyse voting patterns, e.g.,
  - Correlation clustering [Brito et al., 2020, Arinik et al., 2017]
  - Balance and Status [Levorato and Frota, 2016, Derr and Tang, 2018]
- Two main prediction tasks exist with regard to voting
  1. Predicting the Result
  2. Predicting an individual vote
- We focus on predicting votes

Predicting Votes

Predicting a vote can be split into two phases

1. **Who** will vote next
   - Same as link prediction task
   - Is trivial when voting order is known, e.g., parliament roll calls
   - Combinatorial if no known underlying process
2. **How** they will vote
   - Same as sign prediction task
   - Triad features encode balance and status theory
   - Train a supervised ML model using network and triad features [Leskovec et al., 2010a, Leskovec et al., 2010b]

We propose an *unsupervised iterative model* to predict the sign of a vote using balance and status theory.

Local Signed Network

Terminology

- Current voting session is a signed graph $S = (V_S, E_S, w_S)$
- It contains current voter v, candidate c and prior set of voters U
- We also have a signed *Relationship Graph*, $R = (V_R, E_R, w_R)$
- It is created from the history of voting sessions H
- The *Local Signed Network* is the intersection of these two graphs $LSN = S \cap R$
- Essentially the subgraph of the voter's neighbours in R who have already voted in S
- Can use balance and status theory in the LSN to predict votes

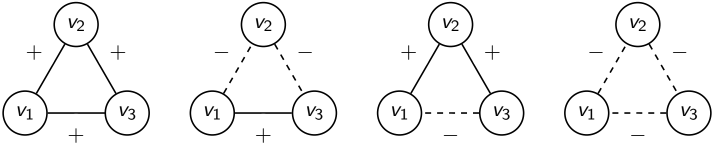
Wikipedia Elections

- RfAs are an election-esque process to gain admin rights
- Anyone can nominate a **candidate** or self-nominate
- 7 day long period of voting and discussion
- Any registered editor can comment and vote :
  - Support
  - Oppose
  - Neutral
- Highly intense process with Q&A with the candidate about Wiki policy etc
- Final decision is made by a bureaucrat
- Decision is based on if a **consensus** has been reached
- Successful RfAs usually have 75% support

Balance Theory

Balance Theory

- Balance theory works for undirected signed graphs
- A cycle is balanced if it has even number of negative links
- A graph is balanced if every cycle is balanced
- Triads B_1, B_2 are balanced
- Traids B_3 and B_4 are unbalanced



B_1 B_2 B_3 B_4

Agreement Graph

- Define a symmetric relationship based on **agreement**
- Subtract 0.5 from agreement ratio to get a signed measure
- Create an *Agreement Graph*, $A = (V_A, E_A, w_A)$

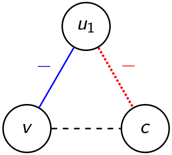
Signed Symmetric Measure

$$w_A((u, v)) = \frac{\text{Number of times } u \text{ and } v \text{ have voted similarly}}{\text{Number of common RfAs for } u \text{ and } v} - 0.5$$

How to measure balance?

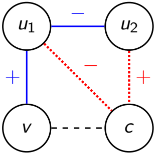
- Triads can be capture local balance of a network
- Need capture information from longer cycles [Chiang et al., 2011]
- Smallest eigenvalue λ_1 of the signed Laplacian $\overline{L}$ is a measure of the imbalance of the network [Hou, 2005]
- We state that a voter chooses to maintain the balance of the LSN
- Therefore, can use λ_1 as basis for predictions in model

Iterative Balance Model



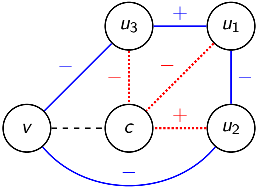
$\lambda_1^+ = 0, \lambda_1^- = 1$

(a) $i = 1$



$\lambda_1^+ = 0.58, \lambda_1^- = 0$

(b) $i = 2$



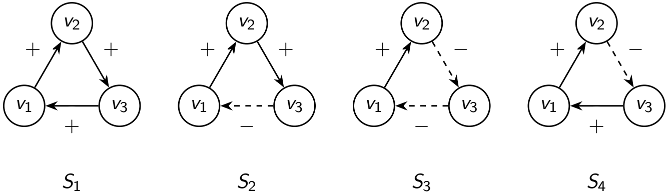
$\lambda_1^+ = 0.55, \lambda_1^- = 0.55$

(c) $i = 3$

Status Theory

Status Theory

- Status theory is based on relative merit of nodes
- Edge $u \xrightarrow{+} v$: v has higher status than u
- Edge $u \xrightarrow{-} v$: v has lower status than u
- Status and Balance theories can contradict each other when used to predict edges
- Triads S_2, S_3 and S_4 adhere to status theory
- Only triads S_1 and S_3 are balanced



Follows Graph

- Define a asymmetric relationship based on **following**
- Create an *Follow Graph*, $F = (V_F, E_F, w_F)$
- An edge $u \rightarrow v$ indicates u votes after v
- Weight is 0.5 subtracted from the *followe ratio*

Signed Asymmetric Measure

$$w_F((u, v)) = \frac{\text{Number of times } u \text{ voted after and agreed with } v}{\text{Number of times } u \text{ voted after } v} - 0.5$$

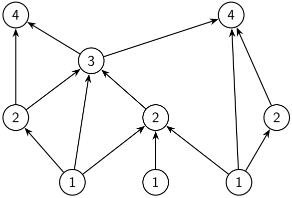
How to measure status?

- Directed triads and other graph motifs can be a proxy for measuring status [Liu et al., 2019]
- Heuristic for status like $\sigma(v) = d^{+}_{\text{in}}(v) + d^{-}_{\text{out}}(v) - d^{+}_{\text{out}}(v) - d^{-}_{\text{in}}(v)$ [Leskovec et al., 2010a]
- These still only capture local effect of status
- Cannot measure violations of status in larger cycles
- The concept of a nodes possessing an implicit status is very similar to a hierarchy

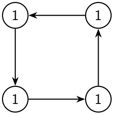
Agony

- Assume a ranking for nodes, $r : V \to \mathbb{N}$
- An edge $u \to v$ causes **agony** if $r(u) < r(v)$
- Quantified as $\max(r(u) - r(v) + 1, 0)$ for an edge
- Agony of a graph G wrt r is
  $A(G, r) = \sum_{(u,v) \in E} \max(r(u) - r(v) + 1, 0)$
- Agony of a graph is lowest possible agony over all rankings
  $A(G) = \min_{r \in \text{Rankings}} A(G, r)$
- Lower agony means better hierarchy $h(G) = 1 - \frac{1}{|E|} A(G)$

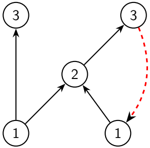
Examples



(a) DAG has perfect hierarchy, $h(G) = 1$ and agony of each edge is 0



(b) A cycle has no hierarchy, $h(G) = 0$ and each edge has agony of 1

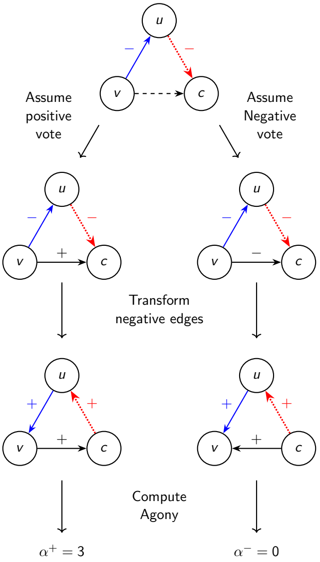


(c) Graph with some hierarchy $h(G) = 2/5$. Red dashed edge has agony of 3 and solid black edges have 0 agony.

Agony and Status

- [Gupte et al., 2011] and [Tatti, 2017] provide algorithms to compute agony of directed graphs
- Consider rank function r as status function σ
- For edg $u \rightarrow v$, $\sigma(u) < \sigma(v)$ is a status violation
- Therefore, agony is a measure of status violation of an edge
- The agony of a network is the **overall status compliance** of G
- What to do with signed edges?
- Flip edges $u \xrightarrow{-} v$ to $u \xleftarrow{+} v$
- Now, we say a voter chooses to reduce the agony of the LSN

Iterative Status Model



Iterative Prediction

Iterative Predictions and its Advantages

- Predictions with balance and status predictions are unsupervised
- Can be bootstrapped by starting with an empty graph R
- After predicting votes in a session, it updates R
- Iteratively gathers more information in R increasing performance
- Therefore, model can be trained on the entire dataset

Algorithm

Input: Candidate c, Relationship graph $R = (V_R, E_R, w_R)$, Order of voters in current session O and true votes w^*

Result: Predictions for current session

$k \leftarrow |O|$

$u \leftarrow O[1]$ // First voter

$V_S \leftarrow \{c, u\}$ // candidate and first voter

$E_S \leftarrow \{(u, c)\}$ // first vote

$w_S((u, c)) \leftarrow w^*((u, c))$ // Assign true vote

Initialize session graph $S = \{V_S, E_S, w_S\}$

$predictions \leftarrow \emptyset$

for $i \leftarrow 2$ **to** k **do**

  $v \leftarrow O[i]$

  $V_S \leftarrow V_S \cup \{v\}$

  $LSN \leftarrow S \cap R$

  $p \leftarrow Predict(v, c, LSN)$

  $predictions \leftarrow predictions \cup p$

  $E_S \leftarrow E_S \cup \{(v, c)\}$

  $w_S((v, c)) \leftarrow w^*((v, c))$ // Assign true vote

end

$Update(R, S)$ // Update Relationship graph

return $predictions$

Results

Tabular Results

Table: Information of relationship graphs of iterative models using entire WIKI-RFA dataset

Relationship Graph	$\|V\|$	$\|E\|$	density	largest component size
Agreement Graph	11924	2451028	0.0345	11908
Follow Graph	11924	3136303	0.0220	11563

Table: Results of iterative models on the complete WIKI-RFA dataset

Model	AUC-ROC	AUC-PR$_{pos}$	AUC-PR$_{neg}$
Baseline	0.5	0.784	0.216
Iterative Balance	0.835	0.935	0.635
Iterative Status	0.784	0.917	0.502

Plots

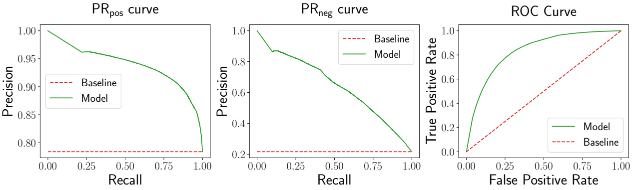


Figure: Plots for the Iterative Balance Model on the complete WIKI-RFA dataset

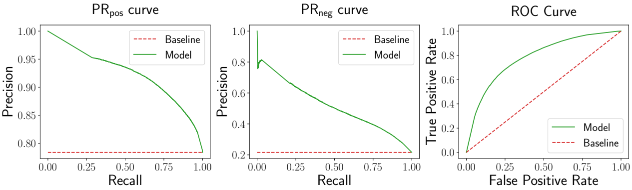


Figure: Plots for the Iterative Status Model on the complete WIKI-RFA dataset

Questions or Comments

Arinik, N., Figueiredo, R., and Labatut, V. (2017).
Signed graph analysis for the interpretation of voting behavior.

Brito, A. C. M., Silva, F. N., and Amancio, D. R. (2020).
A complex network approach to political analysis: Application to the brazilian chamber of deputies.
PLOS ONE, 15(3).

Chiang, K.-Y., Natarajan, N., Tewari, A., and Dhillon, I. S. (2011).
Exploiting longer cycles for link prediction in signed networks.
In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1157–1162.

Derr, T. and Tang, J. (2018).
Congressional vote analysis using signed networks.
In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1501–1502. IEEE.

Gupte, M., Shankar, P., Li, J., Muthukrishnan, S., and Iftode, L. (2011).
Finding hierarchy in directed online social networks.

In *Proceedings of the 20th international conference on World wide web*, pages 557–566.

Hou, Y. P. (2005).
Bounds for the least laplacian eigenvalue of a signed graph.
Acta Mathematica Sinica, 21(4):955–960.

Leskovec, J., Huttenlocher, D., and Kleinberg, J. (2010a).
Predicting positive and negative links in online social networks.
In *Proceedings of the 19th international conference on World wide web*, pages 641–650.

Leskovec, J., Huttenlocher, D., and Kleinberg, J. (2010b).
Signed networks in social media.
In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1361–1370.

Levorato, M. and Frota, Y. (2016).
Brazilian congress structural balance analysis.
Journal of Interdisciplinary Methodologies and Issues in Sciences.

Liu, S., Xiao, J., and Xu, X. (2019).

Link prediction in signed social networks: from status theory to motif families.
IEEE Transactions on Network Science and Engineering, pages 1–1.

Tatti, N. (2017).
Tiers for peers: a practical algorithm for discovering hierarchy in weighted networks.
Data Mining and Knowledge Discovery, 31(3):702–738.