Ideology Detection for Twitter Users via Link Analysis

Yupeng Gu[1], Ting Chen[1], Yizhou Sun[1], Bingyu Wang[2]

[1]University of California, Los Angeles

[2]Northeastern University

July 7, 2017

Overview

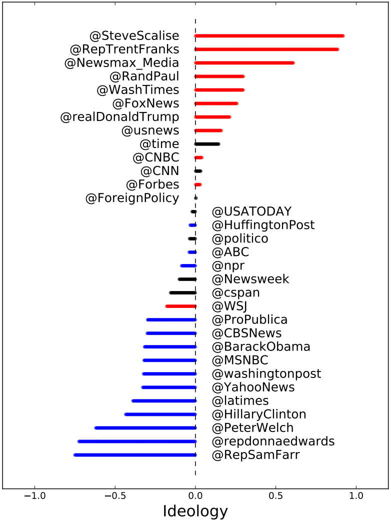
Background

Latent feature (embedding) detection for nodes in the network.
Input: a network of nodes and links (e.g. Twitter).

Figure 1 : Twitter network

Background

Output: node representation in a vector space $\mathbb{R}^K$ ($K = 1$ below).



Background

Applications:

Background

Applications:

- Understand people's tastes/opinions & Advertising

Background

Applications:

- Understand people's tastes/opinions & Advertising
- Clustering / Classification (lower dimensional vector representation)

Background

Applications:

- Understand people's tastes/opinions & Advertising
- Clustering / Classification (lower dimensional vector representation)
- Visualization (2D/3D vector representation)

Background

Applications:

- Understand people's tastes/opinions & Advertising
- Clustering / Classification (lower dimensional vector representation)
- Visualization (2D/3D vector representation)

How to estimate node representation in a network?

Simple and intuitive on *homogeneous* networks (i.e. single type of node and edge).

Figure 2 : *Friendship* between *people*

Intuition

Simple and intuitive on *homogeneous* networks (i.e. single type of node and edge).

- Homophily assumption: connected nodes (neighbors) should be close in vector space (e.g. [MSLC01, ME11])

Intuition

Simple and intuitive on *homogeneous* networks (i.e. single type of node and edge).

- Homophily assumption: connected nodes (neighbors) should be close in vector space (e.g. [MSLC01, ME11])
- Random walk-based approaches: propagation (e.g. [PARS14])

Challenge

How about *heterogeneous* networks (i.e. networks with multiple types of edges)?

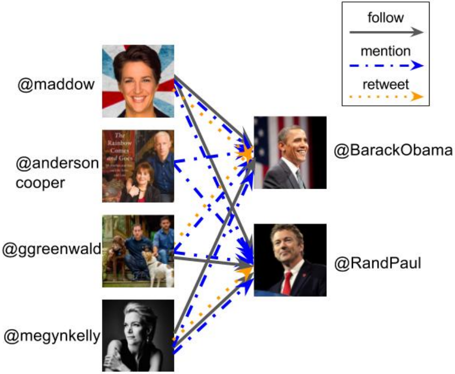


Figure 2 : Multiple types of edges: *follow*, *mention*, *retweet* on Twitter

Challenge

Possible solutions:

- Require domain knowledge or experts to assign weights to each type of link, e.g. $w_{retweet} = 2 \times w_{follow}$

Challenge

Possible solutions:

- Require domain knowledge or experts to assign weights to each type of link, e.g. $w_{retweet} = 2 \times w_{follow}$
  ... not realistic for most cases; thus not easily generalized

Challenge

Possible solutions:

- Require domain knowledge or experts to assign weights to each type of link, e.g. $w_{retweet} = 2 \times w_{follow}$
  ... not realistic for most cases; thus not easily generalized
- Task-specific (even if network is the same)

Challenge

Possible solutions:

- Require domain knowledge or experts to assign weights to each type of link, e.g. $w_{retweet} = 2 \times w_{follow}$
  ... not realistic for most cases; thus not easily generalized
- Task-specific (even if network is the same)
  ... makes it even trickier

Challenge

Possible solutions:

- Require domain knowledge or experts to assign weights to each type of link, e.g. $w_{retweet} = 2 \times w_{follow}$
  ... not realistic for most cases; thus not easily generalized
- Task-specific (even if network is the same)
  ... makes it even trickier
- Cross validation on weight assignments $(w_1, w_2, \cdots, w_T) \in \mathbb{R}^T$

Challenge

Possible solutions:

- Require domain knowledge or experts to assign weights to each type of link, e.g. $w_{retweet} = 2 \times w_{follow}$
  ... not realistic for most cases; thus not easily generalized
- Task-specific (even if network is the same)
  ... makes it even trickier
- Cross validation on weight assignments $(w_1, w_2, \cdots, w_T) \in \mathbb{R}^T$
  ... too expensive; unable to enumerate all possible configurations

Challenge

Our proposed method:

- Able to detect users' latent features in heterogeneous networks
- Able to automatically learn and interpret weights (strength) for each type of links
- Scalable to large networks

Networks with a **single** link type:

Model

Networks with a **single** link type:

- Similarity in the network: neighbors
- Similarity in the vector space $\mathbb{R}^K$: inner product

Model

Networks with a **single** link type:

- Similarity in the network: neighbors
- Similarity in the vector space $\mathbb{R}^K$: inner product

Probability model of link generation, while preserving similarity in two spaces.
A directed link $u_i \to u_j$ is the outcome of the interaction of u_i's representation $\mathbf{p}_i \in \mathbb{R}^K$ and u_j's representation $\mathbf{q}_j \in \mathbb{R}^K$.

Model

The binary status (presence/absence) of a social link from u_i to u_j is modeled as a Bernoulli event with parameter

$$p(e_{ij} = 1) = \sigma(\mathbf{p}_i \cdot \mathbf{q}_j + b_j) \tag{1}$$

where $\sigma(x) = 1/(1 + e^{-x})$ and b_j is a bias (popularity) term for u_b.

Model

The binary status (presence/absence) of a social link from u_i to u_j is modeled as a Bernoulli event with parameter

$$p(e_{ij} = 1) = \sigma(\mathbf{p}_i \cdot \mathbf{q}_j + b_j) \tag{1}$$

where $\sigma(x) = 1/(1 + e^{-x})$ and b_j is a bias (popularity) term for u_b.

Model parameters: $\{\mathbf{p}_i\}_{i=1}^N$, $\{\mathbf{q}_i\}_{i=1}^N \subset \mathbb{R}^K$, $\{b_i\}_{i=1}^N \subset \mathbb{R}$.

Model

The log-likelihood of observing the whole network G is then

$$\log p(G) = \sum_{(i,j):e_{ij}=1} \log p(e_{ij} = 1) + \sum_{(i,j):e_{ij}=0} \log \Big(1 - p(e_{ij} = 1)\Big) \qquad (2)$$

Model

The log-likelihood of observing the whole network G is then

$$\log p(G) = \sum_{(i,j):e_{ij}=1} \log p(e_{ij} = 1) + \sum_{(i,j):e_{ij}=0} \log \Big(1 - p(e_{ij} = 1)\Big) \quad (2)$$

Negative sampling strategy is used to speed up computation:

$$\log p(G) \approx \sum_{(i,j):e_{ij}=1} \log p(e_{ij} = 1) + \sum_{(i,j):e_{ij}\in S_-} \log \Big(1 - p(e_{ij} = 1)\Big) \quad (3)$$

where $|S_-| = |\{(i,j)|e_{ij} = 1\}|$.

Model

The log-likelihood of observing the whole network G is then

$$\log p(G) = \sum_{(i,j):e_{ij}=1} \log p(e_{ij}=1) + \sum_{(i,j):e_{ij}=0} \log\Big(1 - p(e_{ij}=1)\Big) \qquad (2)$$

Negative sampling strategy is used to speed up computation:

$$\log p(G) \approx \sum_{(i,j):e_{ij}=1} \log p(e_{ij}=1) + \sum_{(i,j):e_{ij}\in S_-} \log\Big(1 - p(e_{ij}=1)\Big) \qquad (3)$$

where $|S_-| = |\{(i,j)|e_{ij}=1\}|$.

Standard optimization techniques (e.g. stochastic gradient descent) can be applied on the objective function to infer model parameters.

Model

Networks with **multiple** link types ($r = 1, \cdots, R$):

Model

Networks with **multiple** link types ($r = 1, \cdots, R$):

$\mathbf{p}_i$ remains the same; while $\mathbf{q}_i^{(r)}$ and $b_i^{(r)}$ becomes relation-specific.

Model

Networks with **multiple** link types ($r = 1, \cdots, R$):

$\mathbf{p}_i$ remains the same; while $\mathbf{q}_i^{(r)}$ and $b_i^{(r)}$ becomes relation-specific. Accordingly, the probability for a link of type r is generalized to

$$p(e_{ij}^{(r)} = 1) = \sigma(\mathbf{p}_i \cdot \mathbf{q}_j^{(r)} + b_j^{(r)}) \tag{4}$$

Model

Networks with **multiple** link types ($r = 1, \cdots, R$):

$\mathbf{p}_i$ remains the same; while $\mathbf{q}_i^{(r)}$ and $b_i^{(r)}$ becomes relation-specific. Accordingly, the probability for a link of type r is generalized to

$$p(e_{ij}^{(r)} = 1) = \sigma(\mathbf{p}_i \cdot \mathbf{q}_j^{(r)} + b_j^{(r)}) \tag{4}$$

Objective function

$$J = \sum_{r=1}^{R} w_r \cdot \Big(\sum_{(i,j):e_{ij}^{(r)}=1} \log p(e_{ij} = 1) + \sum_{(i,j):e_{ij} \in S_{-}^{(r)}} \log \big(1 - p(e_{ij} = 1)\big) \Big) \tag{5}$$

s.t

$$\Big(\prod_{r=1}^{R} w_r \Big)^{1/R} = 1$$

Model parameter $w_r \in \mathbb{R}^+$ indicates the strength of each type of link.

Optimization

Optimization is done by updating $\{w\}$ and $\{P, Q, b\}$ iteratively (fixing each other).

Optimization

Optimization is done by updating $\{w\}$ and $\{P, Q, b\}$ iteratively (fixing each other).

Update relation weight w

closed-form solution using Lagrange multiplier

Optimization

Optimization is done by updating $\{w\}$ and $\{P, Q, b\}$ iteratively (fixing each other).

Update relation weight w

closed-form solution using Lagrange multiplier

Update vector representation P, Q, b

stochastic gradient ascent

Optimization

Optimization is done by updating $\{w\}$ and $\{P, Q, b\}$ iteratively (fixing each other).

Update relation weight w

closed-form solution using Lagrange multiplier

Update vector representation P, Q, b

stochastic gradient ascent

Time complexity: $O(\sum_{r=1}^{R} E_r)$ where E_r is the number of edges of type r (for each iteration). Usually requires a few iterations to converge.

Data description

Experiment

Data description

- We first identify all members of the 113^{th} U.S. congress (2013-2015) on Twitter.

Experiment

Data description

- We first identify all members of the 113^{th} U.S. congress (2013-2015) on Twitter.
- We then use Twitter's REST and streaming API to collect a subset of their followees and followers.

Experiment

Data description

- We first identify all members of the 113^{th} U.S. congress (2013-2015) on Twitter.
- We then use Twitter's REST and streaming API to collect a subset of their followees and followers.
- All users' recent tweets are collected to extract their mention and retweet behaviors.

Experiment

Data description

- We first identify all members of the 113^{th} U.S. congress (2013-2015) on Twitter.
- We then use Twitter's REST and streaming API to collect a subset of their followees and followers.
- All users' recent tweets are collected to extract their mention and retweet behaviors.
- A heterogeneous network is built with 3 relations.

Experiment

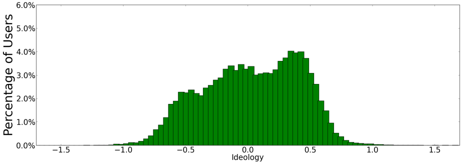
Data description

- We first identify all members of the 113^{th} U.S. congress (2013-2015) on Twitter.
- We then use Twitter's REST and streaming API to collect a subset of their followees and followers.
- All users' recent tweets are collected to extract their mention and retweet behaviors.
- A heterogeneous network is built with 3 relations.

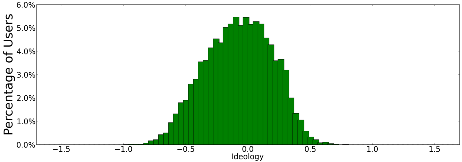
Relation	*follow*	*mention*	*retweet*
Number of users	46,477	34,775	30,990
Number of links (including multiplicity)	1,764,956	2,395,813	718,124

Table 1 : Statistics for Twitter Dataset

Evaluation



(a) Ideology distribution for core users (follow more than 20 politicians)



(b) Ideology distribution for peripheral users

Figure 3 : Political ideology distribution of Twitter users

Evaluation

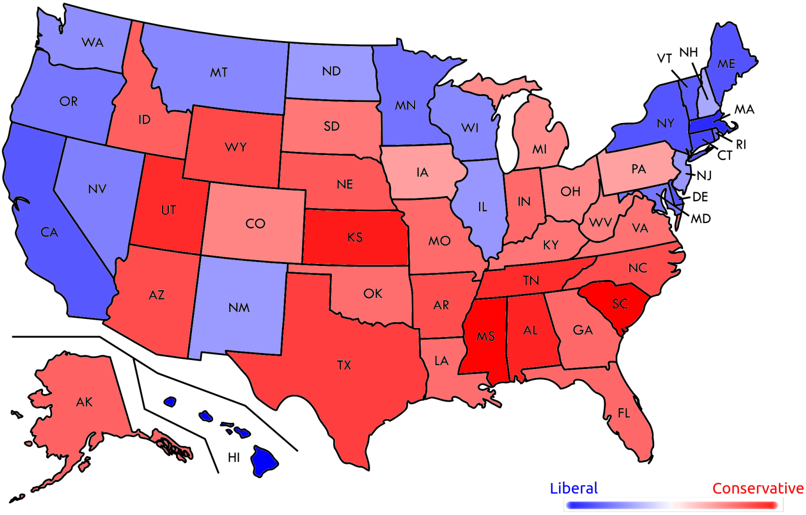


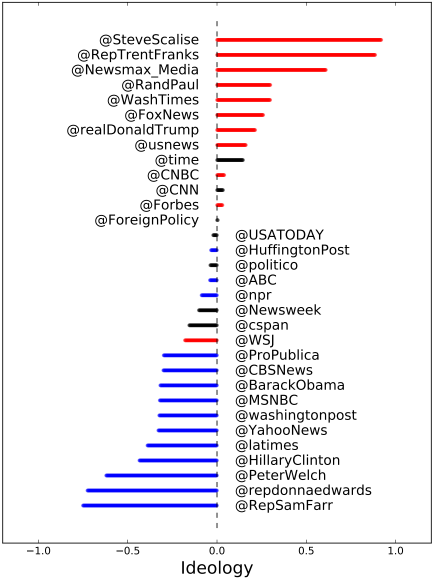
Figure 4 : Average ideology for Twitter users in each state. Darker red means more conservative, while darker blue means more liberal.

Evaluation

Relation r	follow	mention	retweet
Weight w_r	0.866	1.035	1.117

Table 2 : Weights of different link types

Case Studies



Conclusion

- A scalable approach on political ideology detection for Twitter users.
- Our method is easily generalized to other social networks and information networks.
- Future work: incorporate text information (if available) in order to leverage sentiment information.

Reference

Aditya Krishna Menon and Charles Elkan.
Link prediction via matrix factorization.
In *ECML/PKDD'11*, pages 437–452, 2011.

Miller McPherson, Lynn Smith-Lovin, and James M Cook.
Birds of a feather: Homophily in social networks.
Annual review of sociology, pages 415–444, 2001.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena.
Deepwalk: Online learning of social representations.
In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

Thanks! Q&A

Email: ypgu@cs.ucla.edu

Homepage: http://web.cs.ucla.edu/~ypgu/