Ranking Online Social Users by their Influence (a.k.a. *the* Ψ*-score*)

Anastasios Giovanidis

Sorbonne University CNRS-LIP6

Avignon University – Journée GdR COSMOS
21 October 2022

Problem

- **Context:** Generic Online Social Platform.
- **Contributions:**
  (1) Develop an analytical model to sufficiently describe the platform.
  (2) Derive a new measure to rank users based on their influence.
  (3) Propose efficient ranking algorithms, and relate to PageRank.

Problem

- **Context:** Generic Online Social Platform.
- **Contributions:**
  (1) Develop an analytical model to sufficiently describe the platform.
  (2) Derive a new measure to rank users based on their influence.
  (3) Propose efficient ranking algorithms, and relate to PageRank.

☞: Models for social platforms (not networks!) are missing:

- either data analysis from logs (no structural understanding)
- or, models for epidemics and/or opinion dynamics (graph-based only).

Motivation

Platforms and User Ranking

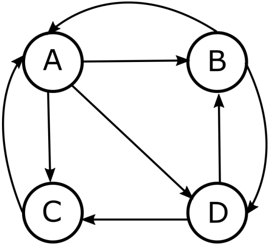
In Twitter, Facebook, Weibo, Instagram, etc.

The million follower falacy:
Audience Size Doesn't Prove Influence on Twitter.
[Cha et al 2010 from MPI-SWS]

☞ Important (missing) ingredients in the models!

- Friendship Graph (Leaders/Followers)
- What (content), when (frequency) and how (prefs) do users post?
- What does the platform show them on their Newsfeed?

Directed Social Graph

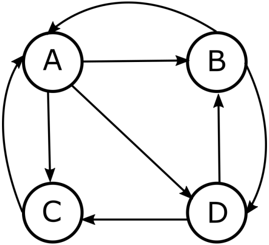


$$\begin{array}{cc} \begin{matrix} A & B & C & D \end{matrix} & \\ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} & \begin{matrix} A \\ B \\ C \\ D \end{matrix} \end{array}$$

- Constant number N of users.
- Leader-graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, Leader matrix.
- $(i, j) \in \mathcal{E}$ directed edge $i \to j$, when i follows j (point to leader).
- For each user i: set of Followers $\mathcal{F}^{(i)}$, set of Leaders $\mathcal{L}^{(i)}$,

e.g. $\mathcal{F}^{(A)} = \{B,\ C\}$, $\mathcal{L}^{(A)} = \{B,\ C,\ D\}$

Who's the influencer?

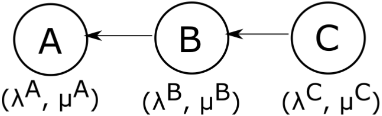


$$\begin{array}{cc} \begin{array}{cccc} A & B & C & D \end{array} & \\ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} & \begin{array}{c} A \\ B \\ C \\ D \end{array} \end{array}$$

Centrality measures for directed graphs:

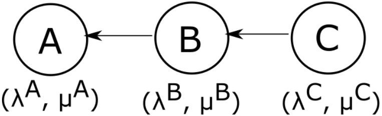
- #Followers (arg max in-degree: $A,\ B,\ C,\ D$) – not much to say...
- Closeness, (arg max: A) Betweenness (arg max: A)
- PageRank $\pi = \frac{1}{9}(3,\ 2,\ 2,\ 2)$ with damping $\beta = 1$ (arg max: A)

(Q1) User Activity?



- Post frequency per node: $\lambda^n \geq 0$ [post/time], $\mu^n \geq 0$ [re-post/time].

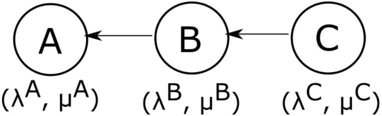
(Q1) User Activity?



- Post frequency per node: $\lambda^n \geq 0$ [post/time], $\mu^n \geq 0$ [re-post/time].

If $\lambda^A > 0$, $\lambda^B > 0$, and $\mu^B > 0$, $\mu^C > 0 \Rightarrow$
posts from both A and B will reach user C.

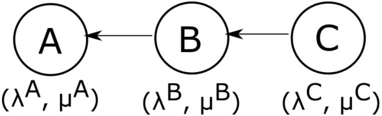
(Q1) User Activity?



- Post frequency per node: $\lambda^n \geq 0$ [post/time], $\mu^n \geq 0$ [re-post/time].

If $\lambda^B = 0$ but $\mu^B > 0 \rightarrow$ only posts from A will reach C.

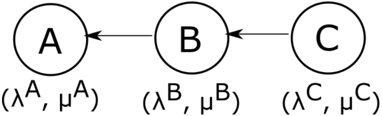
(Q1) User Activity?



- Post frequency per node: $\lambda^n \geq 0$ [post/time], $\mu^n \geq 0$ [re-post/time].

If $\mu^B = 0$ but $\lambda^B > 0 \rightarrow$ only posts from B will reach C.

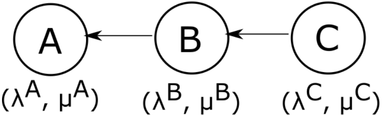
(Q1) User Activity?



- Post frequency per node: $\lambda^n \geq 0$ [post/time], $\mu^n \geq 0$ [re-post/time].

☞ Also! User may select what to re-post based on preferences!

(Q1) User Activity?

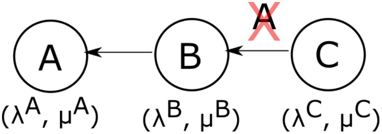


- Post frequency per node: $\lambda^n \geq 0$ [post/time], $\mu^n \geq 0$ [re-post/time].

☞ Also! User may select what to re-post based on preferences!

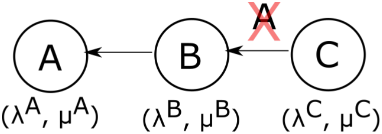
If $\lambda^A > 0$, $\lambda^B > 0$, and $\mu^B > 0$, $\mu^C > 0 \Rightarrow$
user C may chose to repost only posts from A and not from B.

(Q2) Newsfeed algorithms?



☞ If the platform does not show to C content from user A, then such content will never reach C, even if B shares it on their Wall!

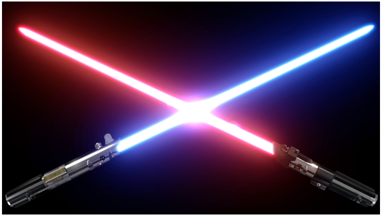
(Q2) Newsfeed algorithms?



☞ If the platform does not show to C content from user A, then such content will never reach C, even if B shares it on their Wall!

Existing models are not sufficient...

A new hope (model & metric)



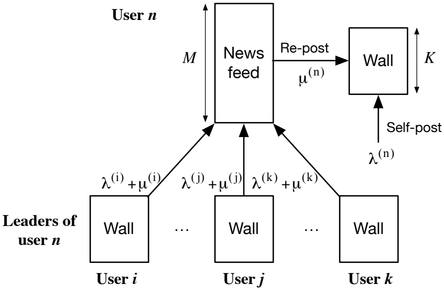
Build a mathematical model that combines:

- Graph structure.
- Post/Share user activity.
- Platform mechanisms and user decisions.

☞ and rank user by their post-related influence.

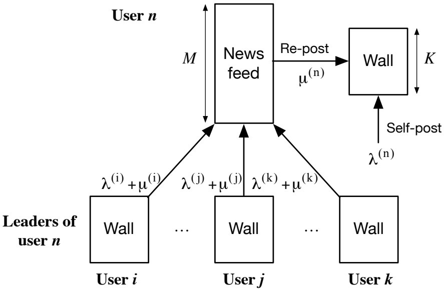
The Model

Social Platform Model



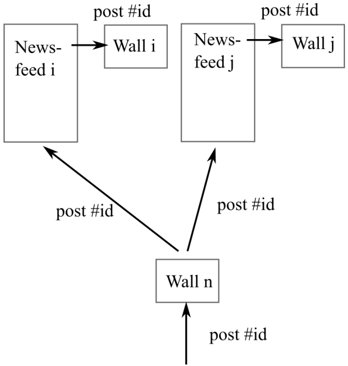
- Number of users N form a directed social graph.
- Each user n has a set of followers $\mathcal{F}^{(n)}$ and a set of leaders $\mathcal{L}^{(n)}$.
- Each user has 2 lists: a Wall of size K and a Newsfeed of size M.
- Each user has a self-post rate $\lambda^{(n)}$ and a re-post (share) rate $\mu^{(n)}$.

Social Platform Model



- Rate of post arrival on Wall n: $\lambda^{(n)} + \mu^{(n)} > 0$.
- Assumption: Content posted and re-posted on a user's Wall instantaneously appears on the Newsfeeds of all his followers.
- Rate of post arrival on Newsfeed n: $\sum_{i \in \mathcal{L}^{(n)}} (\lambda^{(i)} + \mu^{(i)})$.

Posts with multiple clones



☞ The model allows multiple clones of a single post during the diffusion.

☞ It allows in each Newsfeed for competition among posts of various origins to get user attention (re-post).

Simplified Modelling Assumptions

1. Poisson process for posts and re-posts.
2. User behaviour: Random selection (from Newsfeed).
3. Platform mechanism: Random eviction (both Wall and Newsfeed)[1]

The resulting model is Markovian!

[1] Result can be shown valid for other selection/eviction policies (e.g. FIFO)

States

A post generated by user "i" takes as label the author's index "i".

State of user n

$x_i^{(n)}(t) =$ number of posts with origin i on Newsfeed n at time t.
$y_i^{(n)}(t) =$ number of posts with origin i on Wall n at time t.

Newsfeed	$(X_1^{(n)}, \ldots, X_N^{(n)})$	$\xrightarrow{\textit{steady state}}$	$M \cdot (p_1^{(n)}, \ldots, p_N^{(n)})$
Wall	$(Y_1^{(n)}, \ldots, Y_N^{(n)})$	$\xrightarrow{\textit{steady state}}$	$K \cdot (q_1^{(n)}, \ldots, q_N^{(n)})$

Difficulties: The number of states is enormous. A user's Newsfeed and Wall states are coupled with the state of other users.

☞ Two methods: (I) approximate (Infocom'19) and (II) exact (ToN'21)

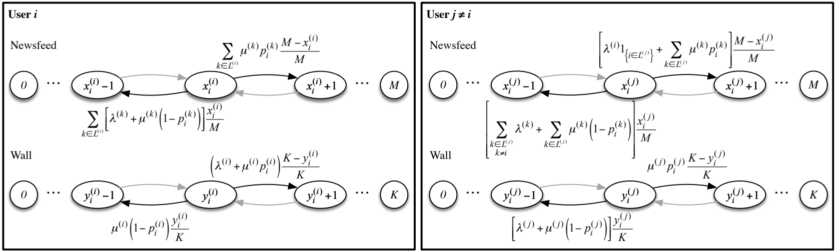
Method I: mean-field approximation

Steps of the solution method:

1. State aggregation: focus on posts from origin user i and aggregate all other posts as "-i".
2. "Mean field approximation": states of leaders evolve as if in steady-state.

$\Rightarrow$ The problem decouples to $2N$ **1-D** Markov Chains.

Aggregate-state transitions



For user n's Newsfeed the steady-state probabilities $p_i^{(n)}$ can be derived:

$$p_i^{(n)} = \sum_{x_i^{(n)}=0}^{M} \pi(x_i^{(n)}) \frac{x_i^{(n)}}{M}.$$

☞ This is the aggregate fixed-point for posts from i. Similarly for Wall.

Method II: exact

Post i Conservation Law (on user n Newsfeed)

$$X_i^{(n)}(0) + N_{in,i}((0,T]) \quad = \quad N_{out,i}((0,T]) + X_i^{(n)}(T).$$

Method II: exact

Post i Conservation Law (on user n Newsfeed)

$$X_i^{(n)}(0) + N_{in,i}((0,T]) \quad = \quad N_{out,i}((0,T]) + X_i^{(n)}(T).$$

- Expected incoming posts of origin i (Poisson process + filtering)

$$\mathbb{E}\left[N_{in,i}((0,T])\right] = \lambda^{(i)} T \mathbf{1}_{i \in \mathcal{L}^{(n)}} + \sum_{k=1}^{L} \mathbb{E}\left[\int_0^T \frac{X_i^{(k)}(t)}{M} \mu^{(k)} dt\right]$$

- Expected outgoing posts of origin i

$$\mathbb{E}\left[N_{out,i}((0,T])\right] = \mathbb{E}\left[\int_0^T \frac{X_i^{(n)}(t)}{M} dt\right] \sum_{k=1}^{L} \left(\lambda^{(k)} + \mu^{(k)}\right)$$

The balance equations

Solution

Newsfeed

$$
\begin{aligned}
(\textit{Newsfeed } i): \quad p_i^{(i)} \sum_{k \in \mathcal{L}^{(i)}} \left(\lambda^{(k)} + \mu^{(k)}\right) &= \sum_{k \in \mathcal{L}^{(i)}} \mu^{(k)} p_i^{(k)}, \\
(\textit{Newsfeed } j): \quad p_i^{(j)} \sum_{k \in \mathcal{L}^{(j)}} \left(\lambda^{(k)} + \mu^{(k)}\right) &= \lambda^{(i)} \mathbf{1}_{\{i \in \mathcal{L}^{(j)}\}} + \sum_{k \in \mathcal{L}^{(j)}} \mu^{(k)} p_i^{(k)},
\end{aligned}
$$

Wall (determined by Newsfeed)

$$
\begin{aligned}
(\textit{Wall } i): \quad (\lambda^{(i)} + \mu^{(i)}) q_i^{(i)} &= \lambda^{(i)} + \mu^{(i)} p_i^{(i)}, \\
(\textit{Wall } j): \quad (\lambda^{(j)} + \mu^{(j)}) q_i^{(j)} &= \mu^{(j)} p_i^{(j)}.
\end{aligned}
$$

Corollaries

- Wall steady-state is completely determined by Newsfeed

$$
\begin{aligned}
q_i^{(i)} &= \frac{\lambda^{(i)}}{\lambda^{(i)} + \mu^{(i)}} + \frac{\mu^{(i)}}{\lambda^{(i)} + \mu^{(i)}} \cdot p_i^{(i)}, \\
q_i^{(j)} &= \frac{\mu^{(j)}}{\lambda^{(j)} + \mu^{(j)}} \cdot p_i^{(j)}.
\end{aligned}
$$

- Insensitivity in list size. The steady-state probabilities depend neither on size M of the Newsfeed, nor on size K of the Wall.

General validity

Holds for several alternative selection / eviction policies.

- Random Selection / FIFO eviction:
  new post enters at top of the list to evict the oldest.
- Random Selection / TTL eviction:
  each post has a fixed Time-To-Live in the list before eviction.
- Newest Selection / Random eviction:
  a user always selects the most fresh entry.

Matrix-form linear system

Unknown vectors $\mathbf{p}_i := (p_i^{(1)}, \ldots, p_i^{(N)})^T$ and $\mathbf{q}_i := (q_i^{(1)}, \ldots, q_i^{(N)})^T$

$$\begin{aligned} \mathbf{p}_i &= \mathbf{A} \cdot \mathbf{p}_i + \mathbf{b}_i \\ \mathbf{q}_i &= \mathbf{C} \cdot \mathbf{p}_i + \mathbf{d}_i. \end{aligned}$$

$\mathbf{A}$	$a_{j,k} := \frac{\mu^{(k)}}{\sum_{\ell \in \mathcal{L}(j)} (\lambda^{(\ell)} + \mu^{(\ell)})} \mathbf{1}_{\{k \in \mathcal{L}(j)\}}$	$\mathbf{b}_i$	$b_{j,i} := \frac{\lambda^{(i)}}{\sum_{\ell \in \mathcal{L}(j)} (\lambda^{(\ell)} + \mu^{(\ell)})} \mathbf{1}_{\{i \in \mathcal{L}(j)\}}$
$\mathbf{C}$	$c_{j,k} := \frac{\mu^{(j)}}{\lambda^{(j)} + \mu^{(j)}} \mathbf{1}_{\{j=k\}}$	$\mathbf{d}_i$	$d_{j,i} := \frac{\lambda^{(i)}}{\lambda^{(i)} + \mu^{(i)}} \mathbf{1}_{\{j=i\}}$

Propagation matrix **A**

The matrix **A** is:

- non-negative.
- row sub-stochastic: the sum of all its rows is less or equal to 1.
- a weighted version of the adjacency (Follower) matrix

Propagation matrix **A**

The matrix **A** is:

- non-negative.
- row sub-stochastic: the sum of all its rows is less or equal to 1.
- a weighted version of the adjacency (Follower) matrix

☞ Weights: users that never re-post from their Newsfeed alter the possibilities of post propagation in the graph.

Propagation matrix **A**

The matrix **A** is:

- non-negative.
- row sub-stochastic: the sum of all its rows is less or equal to 1.
- a weighted version of the adjacency (Follower) matrix

☞ Weights: users that never re-post from their Newsfeed alter the possibilities of post propagation in the graph.

For the spectral radius, if $\lambda^{(n)} > 0$ for all users,

$$\rho(\mathbf{A}) < 1$$

Closed-form solution

Theorem

If $\lambda_n > 0$, $\forall n$ the solution of the linear system is unique,

$$\begin{aligned}\mathbf{p}_i &= (\mathbf{I}_N - \mathbf{A})^{-1}\mathbf{b}_i \\ \mathbf{q}_i &= \mathbf{C}\,(\mathbf{I}_N - \mathbf{A})^{-1}\mathbf{b}_i + \mathbf{d}_i.\end{aligned}$$

Iterative solution

Fixed-point algorithm (Power Iteration)

For any initialization vector $\mathbf{p}_i(0)$, the discrete-time linear system converges towards the previous fixed-point solution when $t \to \infty$.

$$\mathbf{p}_i(t) = \mathbf{A} \cdot \mathbf{p}_i(t-1) + \mathbf{b}_i$$

The rate of convergence is the spectral radius

$$\min_{j=1\ldots N} r(j) \leq \rho(\mathbf{A}) \leq \max_{j=1\ldots N} r(j)$$

where $r(j)$ is the sum of row-j of $\mathbf{A}$.

Influence score Ψ

The average influence of user i on the network is

$$\Psi_i \quad := \quad \frac{1}{N} \sum_{n \in \mathcal{N}} q_i^{(n)}.$$

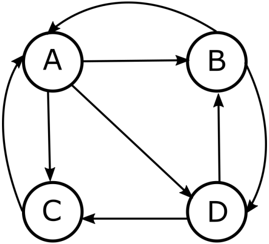
- $q_i^{(n)}$: Expected percentage of posts from i on the Wall of n.
- We can omit self-influence.
- Other metrics based on $q_i^{(n)}$ and $p_i^{(n)}$ can be suggested.

Relation with PageRank

Theorem

In the homogeneous case of user activity $\lambda^{(n)} = \lambda$, $\mu^{(n)} = \mu$,
the score Ψ is equal to PageRank with damping factor $\beta = \frac{\mu}{\lambda+\mu}$.

Who's the influencer?



$$\begin{array}{cc} \begin{array}{cccc} A & B & C & D \end{array} & \\ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} & \begin{array}{c} A \\ B \\ C \\ D \end{array} \end{array}$$

- $\pi = (0.331,\ 0.223,\ 0.223,\ 0.223)$, PageRank with $\beta = 0.95$.
- $\Psi = (0.331,\ 0.223,\ 0.223,\ 0.223)$, Ψ-score with $\lambda = 0.105,\ \mu = 2$.

- ☞ For $\lambda^C = 3\lambda$, $\lambda^{A,B,D} = \lambda$: $\Psi = (0.234,\ 0.156,\ 0.451,\ 0.159)$.
- ☞ For $\mu^C = 0$, $\mu^{A,B,D} = \mu$: $\Psi = (0.122,\ 0.231,\ 0.468,\ 0.179)$.

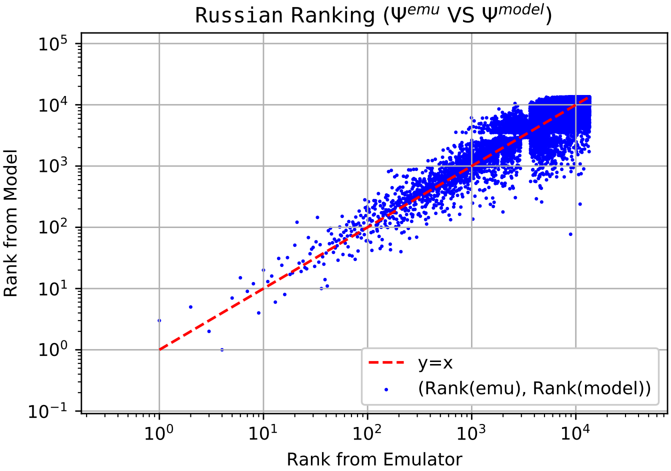
Data Logs

Trace Russian[2]

Time window	57 days
# users	181 621
# tweets	674 292
# retweets	1 271 073
Mean #tweet/user	3.71
Mean #retweets/user	7.00
Max #tweet	4 834
Max #retweet	2 811
% users with #tweet > 0	54.45
% users with #retweet > 0	63.60

[2] https://www.kaggle.com/borisch/russian-election-2018-twitter

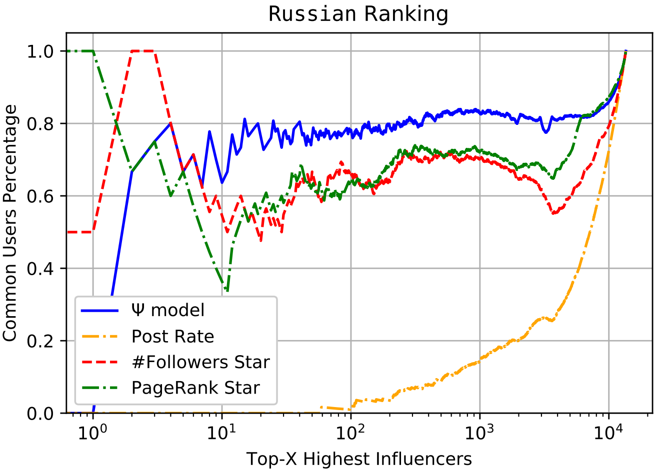
Rus: Model VS Trace I



Rus: Top-10

User	Rank^{emul}	Rank^{model}	Out-degree	$\lambda[sec^{-1}](10^{-7})$
# 20905367	1	3	6 676	42.91
# 82299300	2	5	7 833	96.03
# 494076761	3	2	6 963	439.30
# 615422017	4	1	5 474	639.54
# 174953869	5	7	1 742	118.51
# 711363811	6	15	1 309	8.17
# 36309919	7	9	4 516	118.51
# 1867848452	8	12	1 235	6.13
# 34200559	9	4	3 571	982.81
# 50597428	10	20	1 156	8.17

Rus: Model VS Trace II

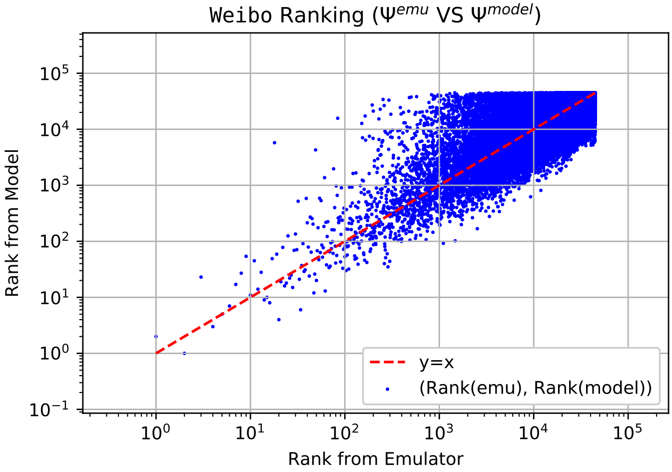


Trace Weibo[3]

Time window	1 216 days
# users	1 340 816
# tweets	232 978
# retweets	33 307 189
Mean #tweet/user	0.17
Mean #retweets/user	24.84
Max #tweet	3 718
Max #retweet	1 032
% users with #tweet > 0	03.55
% users with #retweet > 0	99.54

[3] From paper [Zha et al 2013] https://aminer.org/influencelocality

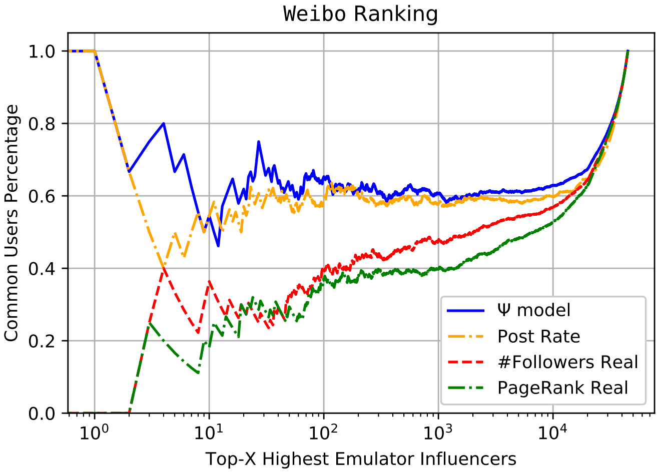
Weibo: Model VS Trace I



Weibo: Top-10

User ID #	$\Psi^{\mathbf{emu}}$ 10^{-3}	$\Psi^{\mathbf{model}}$ 10^{-3}	Rank *emu*	Rank *model*	Follow *Real*	$\lambda[s^{-1}]$ 10^{-6}
519514	37.08	63.88	1	2	459	31.7
490872	24.68	93.99	2	1	595	35.4
1004172	13.30	7.05	3	23	520	2.0
482551	11.53	47.56	4	3	1247	11.0
110361	7.07	13.19	5	5	288	10.3
244531	7.05	12.33	6	7	312	10.4
296675	6.77	8.30	7	17	347	8.7
980392	6.70	5.94	8	27	230	6.2
153610	6.22	2.93	9	54	81	3.6
821785	6.21	11.32	10	11	1084	2.7

Weibo: Model VS Trace II



Fast Ψ-Scoring[4]

[4] Proc. of the 2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) 2022

Computational issues

- Ψ-score generalizes PageRank for non-homogeneous node activity.
- Drawback: it scales poorly to large datasets.
- For a network of N users, it requires to solve N linear systems of equations of size N (or S systems for opinions, each of size N).

We need to make Ψ-score scale at least as fast as PageRank!

Transformation

Novel transformation of the balance equations permits to solve **a single system of size** N for all Ψ-scores instead of N systems.

$$\psi^T = \frac{1}{N}\left[\left(\sum_{t=0}^{\infty}\mathbf{c}^T\mathbf{A}^t\right)\mathbf{B}+\mathbf{d}^T\right].$$

- $\mathbf{A}$ is the propagation matrix.
- $\mathbf{B}$ is the column-concatenation of $\mathbf{b}_n$.
- $\mathbf{c}^T := \mathbf{1}^T\mathbf{C}$.
- $\mathbf{d}^T := \mathbf{1}^T\mathbf{D}$, where $\mathbf{D}$ is the column-concatenation of $\mathbf{d}_n$.

Fast algorithm

Define the series

$$\mathbf{s}_t^T = \sum_{\tau=0}^{t} \mathbf{c}^T \mathbf{A}^\tau, \qquad t = 1, \ldots$$

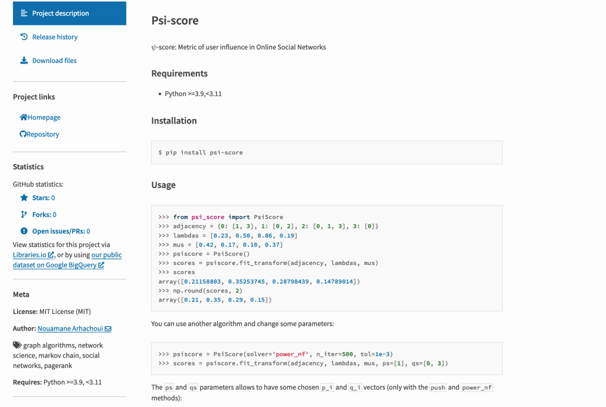
Then the following recursion holds

$$\mathbf{s}_t^T = \mathbf{s}_{t-1}^T \mathbf{A} + \mathbf{c}^T, \qquad t = 1, \ldots$$

Then we iterate until the desired performance gap is achieved

$$\varepsilon_t = \left\| \mathbf{s}_t^T - \mathbf{s}_{t-1}^T \right\| \leq \varepsilon \quad \Rightarrow \quad \delta_t = \left\| \boldsymbol{\psi}_t^T - \boldsymbol{\psi}_{t-1}^T \right\| \leq \frac{\varepsilon \left\| \mathbf{B} \right\|}{N}.$$

The PSI-score Python package


Project description
Release history
Download files

Project links
Homepage
Repository

Statistics
GitHub statistics:
Stars: 0
Forks: 0
Open issues/PRs: 0
View statistics for this project via Libraries.io, or by using our public dataset on Google BigQuery

Meta
License: MIT License (MIT)
Author: Nouamane Arhachoui
graph algorithms, network science, markov chain, social networks, pagerank
Requires: Python >=3.9, <3.11

Psi-score

ψ-score: Metric of user influence in Online Social Networks

Requirements

- Python >=3.9,<3.11

Installation

```
$ pip install psi-score
```

Usage

```
>>> from psi_score import PsiScore
>>> adjacency = {0: [1, 3], 1: [0, 2], 2: [0, 1, 3], 3: [0]}
>>> lambdas = [0.23, 0.50, 0.86, 0.19]
>>> mus = [0.42, 0.17, 0.10, 0.37]
>>> psiscore = PsiScore()
>>> scores = psiscore.fit_transform(adjacency, lambdas, mus)
>>> scores
array([0.21158803, 0.35253745, 0.28798439, 0.14789014])
>>> np.round(scores, 2)
array([0.21, 0.35, 0.29, 0.15])
```

You can use another algorithm and change some parameters:

```
>>> psiscore = PsiScore(solver='power_nf', n_iter=500, tol=1e-3)
>>> scores = psiscore.fit_transform(adjacency, lambdas, mus, ps=[1], qs=[0, 3])
```

The `ps` and `qs` parameters allows to have some chosen `p_i` and `q_i` vectors (only with the `push` and `power_nf` methods):

Figure: https://pypi.org/project/psi-score/

Numerical Analysis

Dataset name	Type	#Nodes	#Edges
DBLP	Citation Network	12 591	49 743
Twitter	Social Network	465 017	834 797
Facebook	Social Network	63 731	817 035
HepPh arXiv	Citation Network	34 546	421 578

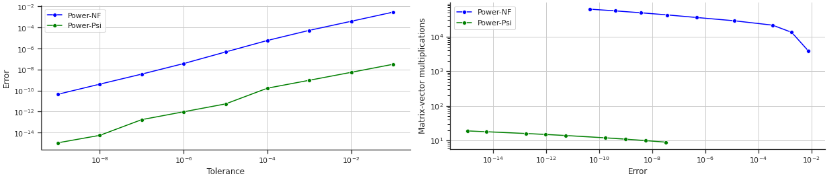
Three types of experiments:

- Precision assessment for a given tolerance criterion
- Performance assessment for a measured error
- Speed assessment for a given tolerance

Two scenarios:

- heterogeneous activity scenario
- homogeneous activity scenario

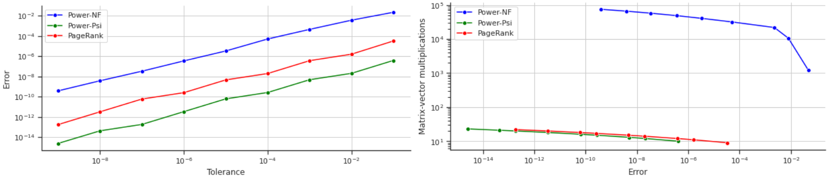
(i) Heterogeneous scenario



Dataset	Power-NF	Power-ψ
DBLP	17.805 sec	0.029 sec
Facebook	1764.226 sec	0.307 sec
Twitter	14526.039 sec	0.634 sec
HepPh	272.358 sec	0.622 sec

with $\varepsilon = 10^{-9}$

(ii) Homogeneous scenario



Dataset	PageRank	Power-NF	Power-ψ
DBLP	0.023 sec	20.775 sec	0.034 sec
Facebook	0.308 sec	2253.302 sec	0.454 sec
Twitter	0.584 sec	17411.146 sec	0.806 sec
HepPh	0.361 sec	360.769 sec	0.908 sec

with $\varepsilon = 10^{-9}$

Ψ-score and opinions[5]

[5] Proc. of Complex Networks and their Applications (CNA) 2022

Opinion diffusion

- A set of $S > 1$ opinions indexed by $s = 1, \ldots, S$.
- Instead of marking posts by author n we can mark by opinion s.
- The re-post rate of each user n is again $\mu^{(n)}$.
- The post rate of each user n is composed of

$$\lambda^{(n)} = \sum_{s=1}^{S} \lambda_s^{(n)}.$$

☞ We are looking for the percentage of s-opinions on the newsfeeds of any user $p_s^{(n)}$.

Balance of opinions

Balance of opinions on newsfeeds

For any opinion s, the average proportions $p_s^{(1)}, \ldots, p_s^{(N)}$ of s-opinions on each newsfeed at equilibrium are the solution of the following system

$$p_s^{(n)} \sum_{k \in \mathcal{L}^{(n)}} (\lambda^{(k)} + \mu^{(k)}) = \sum_{k \in \mathcal{L}^{(n)}} (\lambda_s^{(k)} + \mu^{(k)} p_s^{(k)})$$

Trace #Elysée2017fr[6]

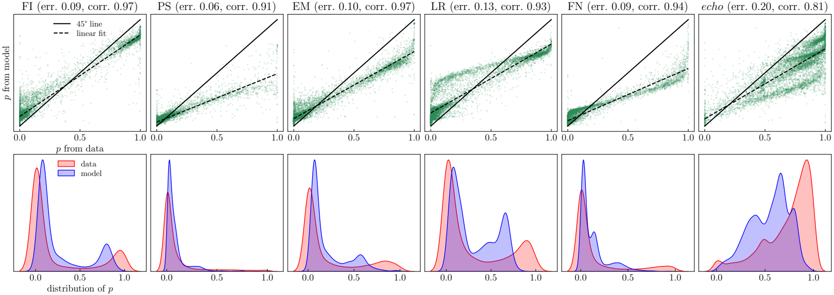
Time window	Nov. 2016 - May 2017
# users	22 853
# tweets	24M
# retweets	7.7M
# opinions	5

5 opinions correspond to known political affiliations FI, PS, EM, LR, FN.

Followers graph: 8,277 users and 975,168 edges

[6]Fraisier, O., Cabanac, G., Pitarch, Y., Besancon, R., Boughanem, M.: #Elysée2017fr: The 2017 French Presidential Campaign on Twitter. In: Proceedings of the 12th International AAAI Conference on Web and Social Media (2018).

Opinion validation on a Twitter dataset



Method to increase opinion diversity

Goal: maximise average diversity of content on the newsfeeds.

Method: insert posts into the newsfeeds (recommendation).

- $x_s^{(n)}$: rate at which we insert posts from party s into n's newsfeed
- B budget: no more than a proportion B of recommended content on newsfeeds

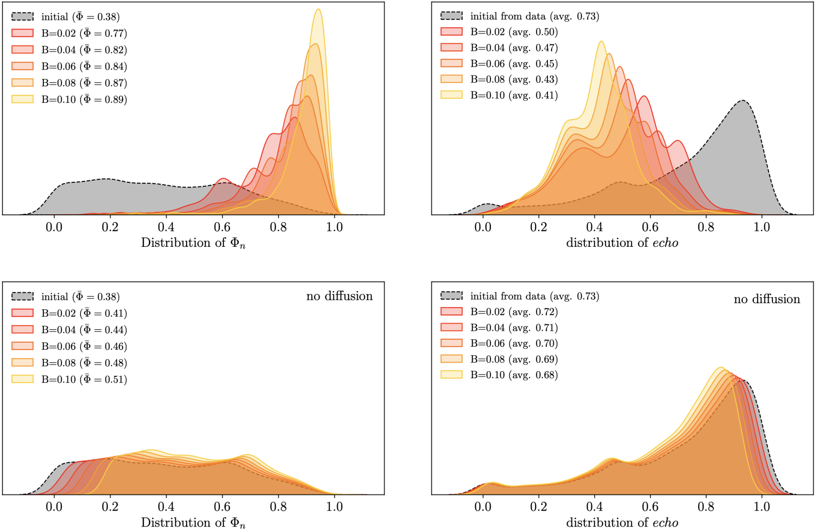
Objective: find $x_s^{(n)}$ for all n, s to maximise average diversity under budget B.

Optimisation problem formulation

$$
\begin{aligned}
\underset{x,p}{\operatorname{argmax}} \quad & \frac{1}{N}\sum_{n}\left(\frac{S}{S-1}\sum_{s=1}^{S} p_s^{(n)}(1-p_s^{(n)})\right) \\
\text{s.t.} \quad & \text{for all } n,s: \\
& \underbrace{\frac{p_s^{(n)}}{1-B}\sum_{k\in\mathcal{L}^{(n)}}(\lambda^{(k)}+\mu^{(k)}) = x_s^{(n)} + \sum_{k\in\mathcal{L}^{(n)}}(\lambda_s^{(k)}+\mu^{(k)}p_s^{(k)}),}_{\textit{model equation}} \\
& \underbrace{\sum_{s} x_s^{(n)} = \frac{B}{1-B}\sum_{k\in\mathcal{L}^{(n)}}(\lambda^{(k)}+\mu^{(k)}),}_{\textit{budget constraint}} \\
& x_s^{(n)}, p_s^{(n)} \geq 0.
\end{aligned}
$$

Optimisation problem

- quadratic objective with linear constraints
- 83K variables
- 50K constraints
- Gurobi solver (barrier algorithm)
- runtime $\sim$10min


initial ($\bar{\Phi} = 0.38$)
B=0.02 ($\bar{\Phi} = 0.77$)
B=0.04 ($\bar{\Phi} = 0.82$)
B=0.06 ($\bar{\Phi} = 0.84$)
B=0.08 ($\bar{\Phi} = 0.87$)
B=0.10 ($\bar{\Phi} = 0.89$)
0.0 0.2 0.4 0.6 0.8 1.0
Distribution of Φ_n
initial from data (avg. 0.73)
B=0.02 (avg. 0.50)
B=0.04 (avg. 0.47)
B=0.06 (avg. 0.45)
B=0.08 (avg. 0.43)
B=0.10 (avg. 0.41)
0.0 0.2 0.4 0.6 0.8 1.0
distribution of *echo*
initial ($\bar{\Phi} = 0.38$)
B=0.02 ($\bar{\Phi} = 0.41$)
B=0.04 ($\bar{\Phi} = 0.44$)
B=0.06 ($\bar{\Phi} = 0.46$)
B=0.08 ($\bar{\Phi} = 0.48$)
B=0.10 ($\bar{\Phi} = 0.51$)
no diffusion
0.0 0.2 0.4 0.6 0.8 1.0
Distribution of Φ_n
initial from data (avg. 0.73)
B=0.02 (avg. 0.72)
B=0.04 (avg. 0.71)
B=0.06 (avg. 0.70)
B=0.08 (avg. 0.69)
B=0.10 (avg. 0.68)
no diffusion
0.0 0.2 0.4 0.6 0.8 1.0
distribution of *echo*

Outro

Ψ-score Contributors

- Bruno Baynat (Sorbonne University, LIP6)
- Cléménce Magnien (CNRS, Sorbonne University, LIP6)
- Antoine Vendeville (PhD candidate UCL London, UK)
- Effrosyni Papanastasiou (PhD candidate, Sorbonne University)
- Benjamin Guedj (INRIA & UCL London, UK)
- Esteban Bautista (Post-doc, Sorbonne University, LIP6)
- Maximilien Danisch (Sorbonne University, LIP6)
- Nouamane Arhachoui (PhD candidate, Sorbonne University)
- Ricardo Lopez-Dawn (PhD candidate, Sorbonne University)

Ψ-score Publications

Relevant Publications

- *Ranking Online Social Users by their Influence*, (A. Giovanidis, B. Baynat, C. Magnien, A. Vendeville), IEEE/ACM Trans. on Networking 2021. doi: 10.1109/TNET.2021.3085201
- *Performance Analysis of Online Social Platforms*, (A. Giovanidis, B. Baynat, A. Vendeville), INFOCOM 2019. doi: 10.1109/INFOCOM.2019.8737539
- *Social Influencer Selection by Budgeted Portfolio Optimization.*, (R. Lopez-Dawn, A. Giovanidis), WiOpt 2021
- *A Fast Algorithm for Ranking Users by their Influence in Online Social Platforms*, (N. Arhachoui, E. Bautista, M. Danisch, A. Giovanidis), The 2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2022.
- *Opening up echo chambers via optimal content recommendation*, (A. Vendeville, A. Giovanidis, E. Papanastasiou, B.Guedj), 11th International Conference on Complex Networks and their Applications, CNA 2022

Ψ-score Software

☞ Check the code on github to play around!

- Pypi:
  https://pypi.org/project/psi-score/
  Source code: https://github.com/NouamaneA/psi-score
- Power-Iteration for Ψ-score, simulators, etc.
  https://github.com/yokaiAG/social-platform-model
- The code for the echo-chamber problem is available at
  https://github.com/AntoineVendeville/Opening-up-echo-chambers

Funding

ANR JCJC: FairEngine (2020-2024)

Contact

anastasios.giovanidis@lip6.fr