A Fast Algorithm for Ranking Users by their Influence in Online Social Platforms

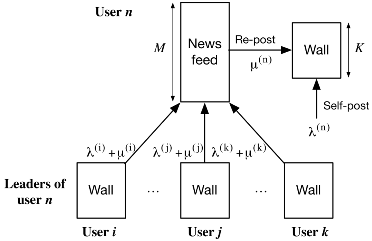
Nouamane Arhachoui   Esteban Bautista
Maximilien Danisch   Anastsios Giovanidis

NPA & ComplexNetworks teams
LIP6 - Sorbonne Université

July 4th, 2022

Outline

Social Platform Model



Social Graph:

- $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ where $(i, j) \in \mathcal{E}$ iff user i follows user j (i.e. j is a leader of i). $|\mathcal{N}| = N$, $|\mathcal{E}| = M$.
- Each node has a **Wall** (FIFO queue with his posts and re-posts) and a **Newsfeed** (FIFO queue with posts and re-posts of his/her leaders)

Activity rates for a user n

- $\lambda^{(n)}$: **posting rate** for user n, number of posts per unit of time created by n
- $\mu^{(n)}$: **re-posting rate** for user n, frequency with which n re-posts a random entry from his/her Newsfeed

Focus on posts of origin i

Presence on Newsfeeds

$\mathbf{p}_i = (p_i^{(1)} \; p_i^{(2)} \; \cdots \; p_i^{(N)})^T$

$\forall n \in \mathcal{N}$, $p_i^{(n)}$ is the expected percentage of posts originating from user i on the news-feed of user n

Presence on Walls

$\mathbf{q}_i = (q_i^{(1)} \; q_i^{(2)} \; \cdots \; q_i^{(N)})^T$

$\forall n \in \mathcal{N}$, $q_i^{(n)}$ is the expected percentage of posts originating from user i on the wall of user n (**influence of i on n**)

Focus on posts of origin i

Presence on Newsfeeds

$\mathbf{p}_i = (p_i^{(1)} \; p_i^{(2)} \; \cdots \; p_i^{(N)})^T$

$\forall n \in \mathcal{N}$, $p_i^{(n)}$ is the expected percentage of posts originating from user i on the news-feed of user n

Presence on Walls

$\mathbf{q}_i = (q_i^{(1)} \; q_i^{(2)} \; \cdots \; q_i^{(N)})^T$

$\forall n \in \mathcal{N}$, $q_i^{(n)}$ is the expected percentage of posts originating from user i on the wall of user n (**influence of** i **on** n)

The ψ-score

The influence of a user i over the entire network is:

$$\psi_i = \frac{1}{N}\sum_{n=1}^{N} q_i^{(n)} = \frac{1}{N}\mathbf{1}^T\mathbf{q}_i$$

Outline

Relation with PageRank

SCIENCES SORBONNE UNIVERSITÉ

Theorem

When all the users have the same activity, i.e. $\forall n \in [\![1, N]\!]\ \lambda^{(n)} = \lambda$ and $\mu^{(n)} = \mu$ and if $\frac{\mu}{\lambda+\mu} = \alpha \in [0, 1]$, then ψ-score = PageRank with a damping factor α

But the goal is not to compute PageRank with a new algorithm:

- ψ-score uses additional information useful for measuring the influence
- In social networks, users have heterogeneous behaviors (i.e. different λ and μ)
- A user with a high in-degree is popular but not necessarily influential

Outline

Compute the ψ-score

Theorem

The conservation law of posts on Newsfeeds implies the following linear systems:

$$\mathbf{p}_i = \mathbf{A}\mathbf{p}_i + \mathbf{b}_i$$
$$\mathbf{q}_i = \mathbf{C}\mathbf{p}_i + \mathbf{d}_i$$

where,

$$\mathbf{A} \in \mathbb{R}^{N\times N} : a_{ji} = \frac{\mu^{(i)}}{\sum_{\ell\in L^{(j)}} (\lambda^{(\ell)}+\mu^{(\ell)})} \mathbb{1}_{\{i\in L^{(j)}\}},$$

$$\mathbf{b}_i \in \mathbb{R}^{N} : b_{ji} = \frac{\lambda^{(i)}}{\sum_{\ell\in L^{(j)}} (\lambda^{(\ell)}+\mu^{(\ell)})} \mathbb{1}_{\{i\in L^{(j)}\}},$$

$$\mathbf{C} \in \mathbb{R}^{N\times N} : c_{ji} = \frac{\mu^{(j)}}{\lambda^{(j)}+\mu^{(j)}} \mathbb{1}_{\{j=i\}},$$

$$\mathbf{d}_i \in \mathbb{R}^{N} : d_{ji} = \frac{\lambda^{(i)}}{\lambda^{(i)}+\mu^{(i)}} \mathbb{1}_{\{j=i\}}$$

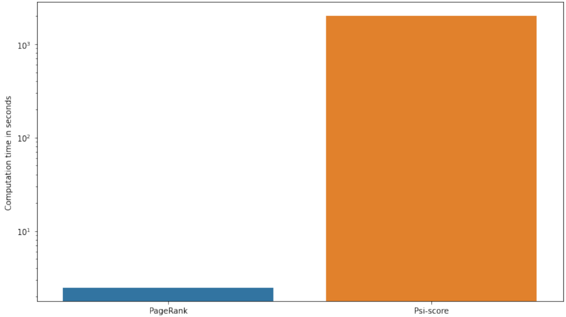
Outline

Problem statement

SCIENCES SORBONNE UNIVERSITÉ

Problem

- The current computation of the ψ-score vector is too slow (compared e.g. to PageRank)
- There are a linear system for each user in the network
- Solving N systems of N equations is required to get the ψ-score vector



Problem Statement

Given a social graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ where the nodes have a posting and sharing activity, we aim for an algorithm that computes the ψ-score for all nodes as fast as PageRank.

Outline

First step: Rewrite the system

SCIENCES SORBONNE UNIVERSITÉ

Instead of solving these N systems of N equations:

$$\mathbf{p}_i = \mathbf{A}\mathbf{p}_i + \mathbf{b}_i$$
$$\mathbf{q}_i = \mathbf{C}\mathbf{p}_i + \mathbf{d}_i$$

we can rewrite everything as follows:

$$\mathbf{P} = \mathbf{A}\mathbf{P} + \mathbf{B}$$
$$\mathbf{Q} = \mathbf{C}\mathbf{P} + \mathbf{D}$$

with,

$$\mathbf{P} = \begin{pmatrix}\mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_N\end{pmatrix}$$
$$\mathbf{Q} = \begin{pmatrix}\mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_N\end{pmatrix}$$
$$\mathbf{B} = \begin{pmatrix}\mathbf{b}_1 & \mathbf{b}_2 & \cdots & \mathbf{b}_N\end{pmatrix}$$
$$\mathbf{D} = \begin{pmatrix}\mathbf{d}_1 & \mathbf{d}_2 & \cdots & \mathbf{d}_N\end{pmatrix}$$

all square matrices.

Theorem

$\mathbf{A}$ is sub-stochastic $\Rightarrow \rho(\mathbf{A}) < 1$

$$\Rightarrow \mathbf{P} = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} = \sum_{t=0}^{\infty} \mathbf{A}^t\mathbf{B}$$

Second step: Get directly the ψ-score vector

In the same fashion, computing:

$$\forall i, \quad \psi_i = \frac{1}{N}\mathbf{1}^T\mathbf{q}_i$$

becomes:

$$\boldsymbol{\psi}^T = \frac{1}{N}\mathbf{1}^T\mathbf{Q}$$

where,

$$\boldsymbol{\psi} = \begin{pmatrix}\psi_1 & \psi_2 & \cdots & \psi_N\end{pmatrix}^T$$

is the ψ-score vector

Power-ψ

With all these changes, we obtain:

$$\boldsymbol{\psi}^T = \frac{1}{N}\left(\mathbf{s}^T\mathbf{B} + \mathbf{1}^T\mathbf{D}\right)$$

where,

$$\mathbf{s} = \sum_{t=0}^{\infty} \mathbf{1}^T\mathbf{C}\mathbf{A}^t$$

$\mathbf{C}$ and $\mathbf{D}$ are diagonal matrices. No need to compute their product with $\mathbf{1}$.

Algorithm

Algorithm 1: `Power-`ψ: Power iteration based algorithm for the ψ-score vector.

input : N number of users, $N \times N$ matrices $\mathbf{A}$ and $\mathbf{B}$, two vectors $\mathbf{c}$ and $\mathbf{d}$, $\mathbf{s}$-tolerance ε

output: vector $\boldsymbol{\psi}$ with the ψ-score of all users

$\mathbf{s} \leftarrow \mathbf{c}$;

$B\_norm \leftarrow \|\mathbf{B}\|$;

$t \leftarrow 0$;

$gap \leftarrow 1$;

while $(gap > \varepsilon)$ **do**

- $\mathbf{s}_{old} \leftarrow \mathbf{s}$;
- $\mathbf{s}^T \leftarrow \mathbf{s}_{old}^T \mathbf{A} + \mathbf{c}$;
- $gap \leftarrow B\_norm \, \|\mathbf{s}_{old} - \mathbf{s}\|$;
- $t \leftarrow t + 1$;

end

$\boldsymbol{\psi}^T \leftarrow \frac{1}{N} \left(\mathbf{s}^T \mathbf{B} + \mathbf{d}^T \right)$;

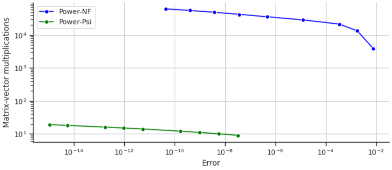
return $\boldsymbol{\psi}$;

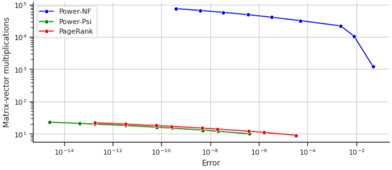
Numerical Analysis

SCIENCES SORBONNE UNIVERSITÉ

Settings

- Dataset: DBLP, citation network
- $N = 12\ 591$; $M = 49\ 743$





Conclusion and Future Work

SCIENCES SORBONNE UNIVERSITÉ

The proposed method

- is nearly as fast as PageRank
- outperforms the state-of-the-art alternative
- enables scalability for real-world datasets

Future work:

- Develop an algorithm that allows to have all the $\mathbf{p}_i$ and $\mathbf{q}_i$ vectors to have the influence on a specific user's newsfeed or wall

Appendix A: Obtaining the Power-ψ equation

SCIENCES SORBONNE UNIVERSITÉ

$$
\begin{aligned}
\psi^T &= \frac{1}{N}\mathbf{1}^T\mathbf{Q} \\
&= \frac{1}{N}\mathbf{1}^T(\mathbf{CP}+\mathbf{D}) \\
&= \frac{1}{N}\mathbf{1}^T\left[\mathbf{C}(\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}+\mathbf{D}\right] \\
&= \frac{1}{N}\mathbf{1}^T\left[\mathbf{C}\left(\sum_{t=0}^{\infty}\mathbf{A}^t\right)\mathbf{B}+\mathbf{D}\right] \\
&= \frac{1}{N}\left[\mathbf{1}^T\mathbf{C}\left(\sum_{t=0}^{\infty}\mathbf{A}^t\right)\mathbf{B}+\mathbf{1}^T\mathbf{D}\right] \\
&= \frac{1}{N}\left[\left(\sum_{t=0}^{\infty}\mathbf{1}^T\mathbf{C}\mathbf{A}^t\right)\mathbf{B}+\mathbf{1}^T\mathbf{D}\right]
\end{aligned}
$$