Performance Analysis of Online Social Platforms

Abstract—**We introduce an original mathematical model to analyze the diffusion of posts within a generic online social platform. Each user of such a platform has his own Wall and Newsfeed, as well as his own self-posting and re-posting activity. As a main result, using our developed model, we derive in closed form the probabilities that posts originating from a given user are found on the Wall and Newsfeed of any other. These probabilities are the solution of a linear system of equations. Conditions of existence of the solution are provided, and two ways of solving the system are proposed, one using matrix inversion and another using fixed-point iteration. Comparisons with simulations show the accuracy of our model and its robustness with respect to modeling assumptions related to user behavior. The novelty introduced by this article is that platform design and user activity (self- and re-posting) can be as important as the social graph structure when analyzing user influence in competing environments, and should always be taken into account.**

I. Introduction

Online Social Platforms (OSPs) play a major role in the way individuals communicate with each other, share news and get informed. Today such platforms host billions of user profiles. Although OSPs differ from one another, most of them share a common structure, which allows users to post messages on their Wall and read posts of others on a separate Newsfeed. Most OSPs also permit re-posting from Newsfeed to Wall, in order to facilitate information diffusion. With each re-post (or “share”, or “re-tweet”) the information becomes visible to a new audience, which may choose to adopt it or not, thus spreading further the post or halting its diffusion. In this way, posts originally generated by some user circulate inside the social network [1]. When the post is gradually adopted by a considerable proportion of the users, we see large cascades of information appear, and we refer to such posts as “viral” [2].

Understanding how information spreads through OSPs is very important as it affects the opinion of the population over several subjects of every-day social life. Companies want to determine the set of most influential users for better marketing of their products [3], and they would like to predict information cascades [4]. Such research is critical also because spreading of influence can have malevolent purposes instead [5], such as the spread of misinformation (“fake news”). To be able to develop defense mechanisms against such social attacks, a concrete mathematical analysis of post diffusion through OSPs is necessary.

Related literature on the topic has mainly focused on models for opinion dynamics given a social graph, but has not yet considered either the OSP structure or user activity. It would be useful to have an analytical model that explains what makes a post become viral within the platforms, how posts of different user origin compete for visibility, and what is the role of user activity. Instead, most available research on understanding cascades and post diffusion in OSPs is data-driven [1], [2], [4].

A. Related Literature

In most relevant research on opinion dynamics, individuals are seen as agents whose relation is described by a social graph. Each agent has a certain opinion and at each step this opinion is updated through interaction with his direct peers. Such models can be grouped into two general categories.

1) Dynamics with Binary opinions: There are only two possible opinions that agents can take. A large amount of work descends from the *voter model* [6], where opinion dynamics are based on imitation. A variation has been studied in [7], where agents with persisting opinions are included. For further extensions, see also [8]. Another group of work is related to epidemic spread. An agent is “susceptible” when his opinion is 0 and becomes “infected” when he adopts opinion 1, through social interaction. In [3] two such mechanisms for opinion updates have been proposed.

2) Dynamics with Continuous opinions: Several works in the literature have inherited and extended the original model of DeGroot [9]. In this, each agent updates his continuous opinion by forming at each step a weighted linear combination of the current opinions of his peers. Variations of this model consider the inclusion of persistent agents [10]. In [11] this update mechanism is used to formulate and solve an opinion manipulation problem. To account for more realistic social behavior, the authors in [12] consider opinion dynamics where agents interact in pairs when their opinions are already close.

Data, OSPs, and Cascades: Instead of modeling opinion dynamics, recent works rather use available data to investigate more practically how posts spread within OSPs. The authors in [1] describe diffusion patterns that arise in specific online domains. Data analysis of large Facebook cascades is performed in [2]. Interestingly, the authors in [4] propose ways to predict cascade growth using machine learning tools.

User activity: In [13] the authors identify user activity as an important control tool for influence maximisation. Making extensive use of datasets, they study the appropriate times for a user to post or re-post in an OSP in order to maximise the probability of audience response.

An interesting analytical effort to relate user activity with OSP design and post diffusion is made in [14]. The authors use temporal point processes to model posting and re-posting activity of a user. They highlight the importance of the Newsfeed in post propagation and map user activity to post visibility, building on the idea that a post can be adopted by a follower when it is visible on his Newsfeed and not pushed away by competing posts. Their model, however, treats only a single user Newsfeed and does not consider the dynamics of the entire social graph. Furthermore, the dynamics of the Newsfeed list are inaccurately mimicked by a FIFO queue.

B. Our Contributions

In this work we propose an analytical model for post diffusion in OSPs, which considers the entire social graph and allows users to generate new posts, or share on their Wall existing posts they find on their Newsfeed. The system is described in Section II. By incorporating the Wall and Newsfeed lists, we allow posts from different origin to compete for the attention of each user. Re-posting activity allows shared posts to further become visible to the users' followers. Re-posting thus plays the role of an information "valve". To the best of our knowledge such analytical model is completely original.

The model describing the generic OSP is presented in Section III. To cope with the enormous state-space of the initial model, we introduce an accurate approximation which decomposes the state description. We further simplify the model by focusing on posts from a single user while aggregating all the rest. In steady-state, the "aggregated" description results in a system of linear equations for the unknown influence probabilities. Its closed-form solution is given in Th. 3. Conditions for the existence of a solution, and an iterative method for the solution are provided in Section IV. [We plan to make the code freely available.] Extended numerical experiments (Section V) verify the validity of our model, and highlight the importance of user activity and OSP structure in better understanding the spread of posts in OSPs. Conclusions are drawn in Section VI.

II. System description

Let us first describe a generic social network platform, such as Facebook or Twitter. A set of users generate and share some content, denoted as *posts*, through the platform. Each user has a list of *followers* and a list of *leaders*. A user can simultaneously be follower and/or leader of others. As a follower, he (she) is interested in the content posted by his (her) leaders. With each user two lists of posts are associated, namely a *Newsfeed* and a *Wall*. A user's Newsfeed is constantly fed by the content that all of his leaders post on their Walls. A user's Wall is fed (i) by his self-generated posts that draw influence from the "outside world", and (ii) by posts that he shares from his Newsfeed. Hence, a user's Wall is a list of self-posts and re-posts. The generic social network platform is illustrated in Figure 1.

A. Assumptions on the system and notations

We consider a constant number N of active users, forming the set $\mathcal{N}$. Users are labelled by an index $n = 1, \dots, N$. We denote by $\mathcal{F}^{(n)}$ and $\mathcal{L}^{(n)}$ the list of followers and the list of leaders of user n. Without loss of generality, we use the directed Leader-graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$. Each pair of nodes $(i, j) \in \mathcal{E}$, corresponds to a directed edge from i to j, when i is a leader of j, i.e., $i \in \mathcal{L}^{(j)}$. We denote by $\mathbf{L}$ the $N \times N$ adjacency matrix of the Leader-graph, whose coefficients are given by: $\ell_{i,j} = \mathbf{1}_{\{i \in \mathcal{L}^{(j)}\}}$, where $\mathbf{1}_{\{.\}}$ is the indicator function. We

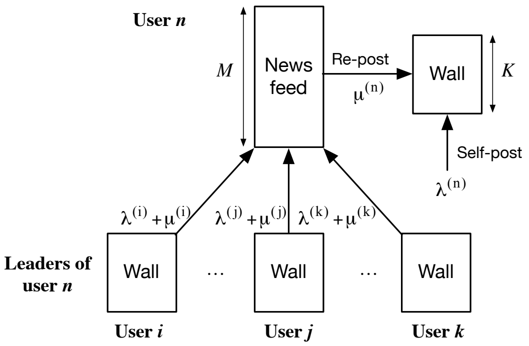

Fig. 1: The social platform from the point of view of user n.

assume that each user n has at least one leader, i.e., $\mathcal{L}^{(n)} \neq \emptyset$, $\forall n$. The Follower-matrix is by definition $\mathbf{F} := \mathbf{L}^T$.

The sizes of both Wall and Newsfeed are considered to be constant. We thus fix K the size of a Wall (total number of posts found on the Wall of each user) and M the size of a Newsfeed. This is reasonable if we assume that only a certain number of most recent posts is considered relevant, and users don't tend to scroll down to access older posting history.

We denote by $\lambda^{(n)}$ [posts/unit time] the rate with which user n generates new posts on his Wall, and by $\mu^{(n)}$ the rate with which user n visits his Newsfeed and selects one of the M entries to re-post on his Wall (note here that each visit implies re-posting). As a result, posts arrive on the n-th Wall with a total rate $\lambda^{(n)} + \mu^{(n)}$ [posts/unit time]. Additionally, we make the assumption that content posted on a users's Wall instantaneously appears on the Newsfeeds of his followers. As a result, the input rate of posts in the n-th Newsfeed, is $\sum_{j \in \mathcal{L}^{(n)}} (\lambda^{(j)} + \mu^{(j)})$. Given that the two lists associated per user have fixed size, then with each new entry one element has to be removed from the list and replaced by the new one. For the user activity we require $\lambda^{(n)} + \mu^{(n)} > 0$, $\forall n$.

Finally, any post originally generated by a given user n takes as label the author's index n, and will keep this label throughout its lifespan inside the network.

B. Influence metric of interest

The aim is to estimate the influence of a specific user, say user i, over the entire network. In order to define the metric of interest, we first define the *influence* of user i on user n, $q_i^{(n)}$, as the steady-state probability that a post found on the Wall of user n is of label i, i.e., has been originally created by user i. Note that these probabilities are performance parameters that will be the output of the developed models. They obviously satisfy $\sum_{i=1}^{N} q_i^{(n)} = 1$, $\forall n$. With the above, we propose the following metric of influence,

$$\Psi_i = \frac{1}{N-1} \sum_{n \neq i} q_i^{(n)}. \qquad (1)$$

It corresponds to the average probability that a post on the Wall of any user $n \neq i$, has origin i. Note here that the suggested metric averages over all users in the network, but excludes the original user i. Other metric definitions are also possible.

III. Markovian models

A. Modeling assumptions

For the analysis, we make the following assumptions:

- **Poisson arrivals**. For any user n the generation of new posts on his Wall follows a Poisson process with rate $\lambda^{(n)}$ and the re-posting activity from his Newsfeed follows a Poisson process with rate $\mu^{(n)}$.
- **Random selection**. When a user visits his own Newsfeed, we assume that he selects at random one of the M entries to re-post on his Wall.
- **Random eviction**. A novel entry on the Wall or Newsfeed list will push out an older entry of random position.

Thanks to these assumptions, the resulting models developed in the following are Markovian. Indeed, all inter-arrival times between posts and re-posts are exponential and all choices are probabilistic. The random selection is consistent with common practice in real life, because it is the actual content of posts rather than the order of appearance in the Newsfeed which plays major role in the decision to re-post. The random eviction from the Wall is less realistic, since new entries are normally placed at the top of the Wall list (which would correspond to an oldest eviction policy). The validity and robustness of all these assumptions will be evaluated through simulations. As will be seen in Section V-B they have a very limited impact on performance.

B. Detailed Model

The full state-description for this system is an N-tuple $(\mathbf{U}^{(1)}, \dots, \mathbf{U}^{(N)})$, where $\mathbf{U}^{(n)} = (\mathbf{x}^{(n)}, \mathbf{y}^{(n)})$ is the state of user n (at a given time t, omitted in notations for sake of clarity). $\mathbf{x}^{(n)}$ is the state of his Newsfeed and $\mathbf{y}^{(n)}$ the state of his Wall. The random eviction and random selection assumptions allow to describe the system-state evolution without using information over the order of posts in the lists. Then, $\mathbf{x}^{(n)} = (x_1^{(n)}, \dots, x_N^{(n)})$, where $x_i^{(n)}$ counts the number of posts with user-origin i found in the Newsfeed of user n. Similarly, $\mathbf{y}^{(n)} = (y_1^{(n)}, \dots, y_N^{(n)})$ where $y_i^{(n)}$ counts the number of posts with origin i found in the Wall of user n.

With all the assumptions described in Section III-A, it can be shown that the stochastic process resulting from the full state-description $(\mathbf{U}^{(1)}, \dots, \mathbf{U}^{(N)})$, is a continuous-time Markov chain model. As such, its steady-state can theoretically be solved. However, even for very small values of the system parameters the number of states will be enormous, whereas the state of a user's Newsfeed and Wall is coupled with the state of other users. As a result, any solution using, e.g., a numerical method, would be computationally intractable. For this reason we introduce in the next subsection a simple approximation that decouples the state-space and considerably reduces the solution complexity with negligible loss in precision.

Before presenting the decomposed model, it is important to understand where the coupling between states of different users appears. Consider user n and focus on label i posts. A leader k of user n will re-post from his own Newsfeed to his own Wall a post of label i with probability $x_i^{(k)}/M$, due to the random selection policy. This post will appear immediately in the Newsfeed of user n, thus changing its state $\mathbf{y}^{(n)}$. Hence, the evolution of the state of user n depends not only on his own current state and on his own activity, but also on the states and activity of all of his leaders.

C. Decomposed Model

We now develop an approximate Markovian model which will eventually lead to a closed-form solution for the system's steady-state. The main idea is the following: for a given user n, the state transitions of his Newsfeed and Wall will still be a function of his own current state and activity, as well as the activity of all of his leaders. But they will not depend anymore on the current states of the user's leaders (as shown above), rather on their *average probabilities in steady-state*, which at this point are unknown values. In this way, the original full state-description can be decomposed into $2N$ independent state-descriptions resulting in $2N$ decoupled Markov Chains, each one associated with the Newsfeed and the Wall of a given user. The coupling between the produced $2N$ Markov Chains exists only through the unknown values of the steady-state probabilities.

More precisely, let us consider user n. We denote by $p_i^{(n)}$ the steady-state probability for a post in his Newsfeed to be of label i, i.e., to originate from user i. Similarly, we have already defined in Section II-B $q_i^{(n)}$ as the steady-state probability for a post on the Wall of user n to be of label i. These quantities are the model *unknowns*. Note that these probabilities are actually the ergodic means of the related user states, i.e., $p_i^{(n)} = \mathbb{E}[\frac{x_i^{(n)}}{M}]$ and $q_i^{(n)} = \mathbb{E}[\frac{y_i^{(n)}}{K}]$.

The Newsfeed state of user n evolves as follows:

$$(x_1^{(n)}, x_i^{(n)}, x_j^{(n)}, x_N^{(n)}) \quad \overset{f_{i,j}^{(n)}}{\rightarrow} \quad (x_1^{(n)}, x_i^{(n)} + 1, x_j^{(n)} - 1, x_N^{(n)}),$$

where $f_{i,j}^{(n)}$ is the transition rate between the two represented states (when $x_j^{(n)} > 0$) and is given by:

$$f_{i,j}^{(n)} = \left(\lambda^{(i)} \mathbf{1}_{\{i \in \mathcal{L}^{(n)}\}} + \sum_{k \in \mathcal{L}^{(n)}} \mu^{(k)} p_i^{(k)} \right) \frac{x_j^{(n)}}{M}. \qquad (2)$$

Indeed, $\lambda^{(i)}$ is the rate with which user i generates a new own post on his Wall, post that instantaneously appears on the Newsfeed of user n, if $i \in \mathcal{L}(n)$. A post of label i can as well appear in the Newsfeed of n through re-posting. This occurs when one of the n's leaders visits his own Newsfeed and re-posts a label i post on his own Wall. For leader k such event occurs with rate $\mu^{(k)} \frac{x_i^{(k)}}{M}$, following the random selection policy. From the decoupling, $\mu^{(k)} p_i^{(k)}$ is an estimation of this rate and this is where the approximation lies. Finally, $\frac{x_j^{(n)}}{M}$ is the probability that an incoming post (with label i) replaces an old post of label j, by the principle of random eviction.

From the stationary probability π of this Markov chain, we can derive the steady-state probabilities $p_i^{(n)}$ for $i = 1, \dots, N$:

$$p_i^{(n)} = \sum_{(x_1^{(n)},\dots,x_N^{(n)}) \mid \sum_{j=1}^N x_j^{(n)} = M} \pi(x_1^{(n)}, \dots, x_N^{(n)}) \frac{x_i^{(n)}}{M}. \quad (3)$$

As can be seen, the unknown probabilities $p_i^{(n)}$ depend on the steady-state solution of a Markov chain (through relation (3)) whose transition rates depend in turn on the probabilities $p_i^{(\cdot)}$ (through relation (2)). As a result, the probabilities $p_i^{(n)}$ are solution of a fixed-point problem, that can be solved using an iterative procedure. Note that at each iteration of this procedure, one has to solve numerically a complicated N-dimensional Markov chain. We don't detail here the global algorithm, as we will show in the next section that we can drastically simplify the model and thus the analysis.

The Wall state of user n evolves as follows:

$$(y_1^{(n)}, y_i^{(n)}, y_j^{(n)}, y_N^{(n)}) \quad \overset{g_{i,j}^{(n)}}{\to} \quad (y_1^{(n)}, y_i^{(n)} + 1, y_j^{(n)} - 1, y_N^{(n)}),$$

where, using a similar approximation, the transition rate $g_{i,j}^{(n)}$ (when $y_j^{(n)} > 0$) is given by:

$$g_{i,j}^{(n)} = \left(\lambda^{(n)} \mathbf{1}_{\{i=n\}} + \mu^{(n)} p_i^{(n)}\right) \frac{y_j^{(n)}}{K}. \quad (4)$$

Indeed, as stated earlier, the state of the Wall of user n can only evolve with a rate $\lambda^{(n)} + \mu^{(n)}$. With a rate $\lambda^{(n)}$, user n generates a new post on his own Wall, and with a rate $\mu^{(n)} p_i^{(n)}$ user n recopies on his Wall from his own Newsfeed a post with label i. This is, again, an approximation resulting from the state decoupling. This incoming post will replace an old post of label j with a probability $\frac{y_j^{(n)}}{K}$ due to random eviction.

From the stationary probability of this Markov chain, we derive the steady-state probabilities $q_i^{(n)}$ for $i = 1, \dots, N$ as:

$$q_i^{(n)} = \sum_{(y_1^{(n)},\dots,y_N^{(n)}) \mid \sum_{j=1}^N y_j^{(n)} = K} \pi(y_1^{(n)}, \dots, y_N^{(n)}) \frac{y_i^{(n)}}{K}. \quad (5)$$

Note that from the relations (4) and (5) Wall probabilities $q_i^{(n)}$ do not result from a fixed-point solution, and they can directly be derived from the Newsfeed probabilities $p_i^{(n)}$.

D. Aggregated Model

The state-space of the decoupled Markov chains associated with both Newsfeed and Wall of a user can still be very large. Each chain is N-dimensional. For any feasible state of the n-th Newsfeed, i.e., combination of posts $\mathbf{x}^{(n)} = (x_1^{(n)}, \dots, x_N^{(n)})$, it holds that $\sum_{i=1}^N x_i^{(n)} = M$. Hence, the size of its state-space corresponds to the number of ways to put M undifferentiated objects in N distinct boxes, and is equal to $\binom{M+N-1}{M}$. Similarly, any state $\mathbf{y}^{(n)} = (y_1^{(n)}, \dots, y_N^{(n)})$ of the Wall of user n is such that $\sum_{i=1}^N y_i^{(n)} = K$. As a result, the size of the Wall state-space is equal to $\binom{K+N-1}{K}$. These state-spaces can still be huge for large values of N, M, or K.

We now present an aggregated Markov chain model that gets around the problem by considering a reduced state-space, but *without introducing any additional approximation*. As a result, the aggregated model has the same accuracy as the decomposed model presented in the previous section.

Our starting point here is the decomposed model. The idea is to focus on a particular user i and develop an aggregated model that will only be able to calculate the influence of user i on the entire network. Of course, one can successively apply the technique to all $i = 1, \dots, N$ in order to determine eventually the influence of all users.

We thus particularize a given user i and aggregate the state-space of the system as follows. We keep the system state as the n-tuple $(\mathbf{U}^{(1)}, \dots, \mathbf{U}^{(N)})$, where $\mathbf{U}^{(n)} = (\mathbf{x}^{(n)}, \mathbf{y}^{(n)})$ is the state of user n, $\mathbf{x}^{(n)}$ being the state of his Newsfeed and $\mathbf{y}^{(n)}$ the state of his Wall. But we now consider only the number of posts issued from the particularized user i and the number of posts issued from other users, indexed as $-i$. As a result, the state of the Newsfeed is reduced to $\mathbf{x}^{(n)} = (x_i^{(n)}, x_{-i}^{(n)})$. Since the sum of its two components must be equal to M, it holds that $x_{-i}^{(n)} = M - x_i^{(n)}$, and the state of the Newsfeed of user n can be further reduced to a single integer: $\mathbf{x}^{(n)} = (x_i^{(n)})$. As a result, the associated Markov chain is a birth-and-death process with values ranging from 0 to M.

Similarly, the aggregated model associated with the Wall of user n is a birth-and-death process with a state $\mathbf{y}^{(n)} = (y_i^{(n)})$ ranging from 0 to K. All continuous-time 1-dimensional Markov Chains for the aggregated model are given in Fig. 2. A distinction needs to be made between the chains associated with particularized user i and the chains associated with other users $j \neq i$. The transition rates of these birth-and-death processes have an interpretation similar to those given in Section III-C.

From the stationary probabilities of the chain associated with the Newsfeed of user n, we can derive the steady-state probabilities $p_i^{(n)}$:

$$p_i^{(n)} = \sum_{x_i^{(n)}=0}^{M} \pi(x_i^{(n)}) \frac{x_i^{(n)}}{M}. \quad (6)$$

Similarly, from the stationary probabilities of the chain associated with user's n Wall, we can derive the $q_i^{(n)}$:

$$q_i^{(n)} = \sum_{y_i^{(n)}=0}^{K} \pi(y_i^{(n)}) \frac{y_i^{(n)}}{K}. \quad (7)$$

As in the decomposed model developed in Section III-C, the solution of the aggregated model relies on a fixed-point. The transition rates associated with the Newsfeed of a user n are expressed as functions of the Newsfeed probabilities $p_i^{(k)}$ for all users $k = 1, \dots, N$. The $p_i^{(k)}$ values in their turn depend on the steady-state probabilities π of the Markov chain associated with the Newsfeed of user k, through (6). A fixed-point should thus be solved for the Newsfeed probabilities. Once these are defined, the Wall probabilities can be derived,

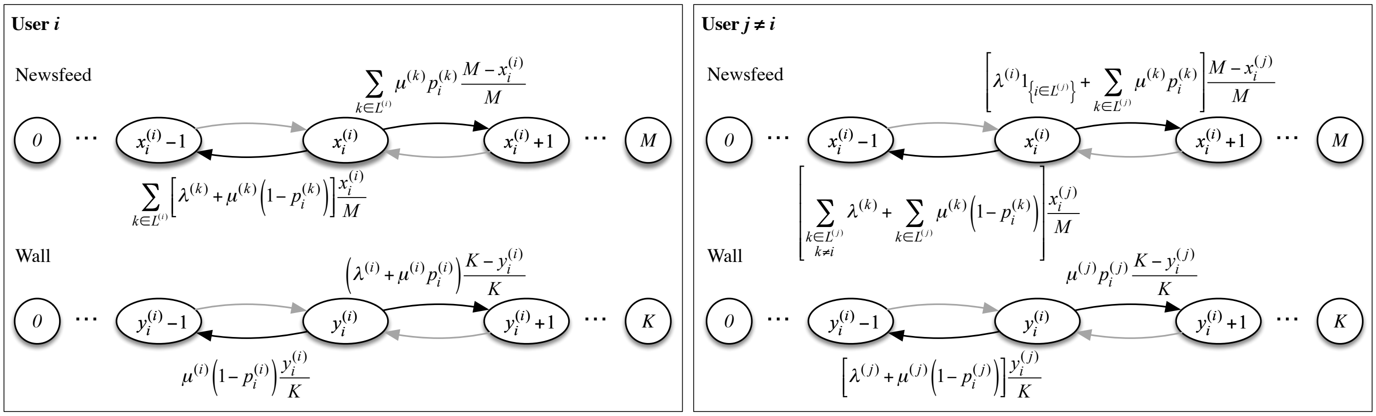
User *i*
Newsfeed
$\sum_{k\in L^{(i)}} \mu^{(k)} p_i^{(k)} \frac{M - x_i^{(i)}}{M}$
0
$x_i^{(i)} - 1$
$x_i^{(i)}$
$x_i^{(i)} + 1$
M
$\sum_{k\in L^{(i)}} \left[\lambda^{(k)} + \mu^{(k)}\left(1 - p_i^{(k)}\right)\right] \frac{x_i^{(i)}}{M}$
Wall
$\left(\lambda^{(i)} + \mu^{(i)} p_i^{(i)}\right) \frac{K - y_i^{(i)}}{K}$
0
$y_i^{(i)} - 1$
$y_i^{(i)}$
$y_i^{(i)} + 1$
K
$\mu^{(i)}\left(1 - p_i^{(i)}\right) \frac{y_i^{(i)}}{K}$
User *j* ≠ *i*
Newsfeed
$\left[\lambda^{(i)} 1_{\left\{i\in L^{(j)}\right\}} + \sum_{k\in L^{(j)}} \mu^{(k)} p_i^{(k)}\right] \frac{M - x_i^{(j)}}{M}$
0
$x_i^{(i)} - 1$
$x_i^{(i)}$
$x_i^{(i)} + 1$
M
$\left[\sum_{\substack{k\in L^{(j)}\\ k\neq i}} \lambda^{(k)} + \sum_{k\in L^{(j)}} \mu^{(k)}\left(1 - p_i^{(k)}\right)\right] \frac{x_i^{(j)}}{M}$
Wall
$\mu^{(j)} p_i^{(j)} \frac{K - y_i^{(j)}}{K}$
0
$y_i^{(i)} - 1$
$y_i^{(i)}$
$y_i^{(i)} + 1$
K
$\left[\lambda^{(j)} + \mu^{(j)}\left(1 - p_i^{(j)}\right)\right] \frac{y_i^{(j)}}{K}$

Fig. 2: Aggregated Markov Chain model.

since the transition rates associated with the Wall are functions of p's and not q's. The resolution of this fixed-point is detailed in the next section.

IV. Closed Form solution

A. Solution of the aggregated model

We first consider the particularized user i of the aggregated model presented in Section III-D, and we focus on the Markov chain associated with his Newsfeed i (see top left part of Fig. 2). To solve this birth-and-death process we define the quantity

$$\sigma_i := \frac{\sum_{k\in\mathcal{L}^{(i)}} \mu^{(k)} p_i^{(k)}}{\sum_{k\in\mathcal{L}^{(i)}} \left[\lambda^{(k)} + \mu^{(k)}(1 - p_i^{(k)})\right]}.$$

Note that σ_i depends on all probabilities $p_i^{(k)}$ for $k \in \mathcal{L}^{(i)}$.

The steady-state probability of the Markov chain associated with the Newsfeed of user i can then be derived using σ_i:

$$\pi(x_i^{(i)}) = \pi(0) \left(\begin{array}{c} M \\ x_i^{(i)} \end{array} \right) \sigma_i^{x_i^{(i)}},$$

where $\pi(0)$ is obtained by normalization (thanks to the Binomial formula):

$$\pi(0) = \frac{1}{(1+\sigma_i)^M}.$$

Then, applying this result in (6) we get:

$$\begin{aligned} p_i^{(i)} &= \frac{1}{(1+\sigma_i)^M} \sum_{m=1}^{M} \left(\begin{array}{c} M \\ m \end{array} \right) \sigma_i^m \frac{m}{M} \\ &\overset{m' := m-1}{=} \frac{\sigma_i}{(1+\sigma_i)^M} \sum_{m'=0}^{M-1} \left(\begin{array}{c} M-1 \\ m' \end{array} \right) \sigma_i^{m'} \\ &= \frac{\sigma_i}{1+\sigma_i}. \end{aligned}$$

By replacing the expression for σ_i, we obtain the following very simple expression:

$$p_i^{(i)} \sum_{k\in\mathcal{L}^{(i)}} \left(\lambda^{(k)} + \mu^{(k)}\right) = \sum_{k\in\mathcal{L}^{(i)}} \mu^{(k)} p_i^{(k)} \tag{8}$$

Following a similar reasoning, we get for the Newsfeed equation of any user $j \neq i$:

$$p_i^{(j)} \sum_{k\in\mathcal{L}^{(j)}} \left(\lambda^{(k)} + \mu^{(k)}\right) = \lambda^{(i)} \mathbf{1}_{\left\{i\in\mathcal{L}^{(j)}\right\}} + \sum_{k\in\mathcal{L}^{(j)}} \mu^{(k)} p_i^{(k)}. \tag{9}$$

As a result, the set of N equations (8) and (9) constitute the new equations of the fixed point, whose solution gives the required Newsfeed probabilities $p_i^{(n)}$ for $n = 1, \ldots, N$.

In the same fashion, the steady-state probabilities for the Wall can be directly derived from the steady-state probabilities for the Newsfeed as (remember $\lambda^{(n)} + \mu^{(n)} > 0$, $\forall n$):

$$q_i^{(i)} = \frac{\lambda^{(i)}}{\lambda^{(i)} + \mu^{(i)}} + \frac{\mu^{(i)}}{\lambda^{(i)} + \mu^{(i)}} \cdot p_i^{(i)}, \tag{10}$$

$$q_i^{(j)} = \frac{\mu^{(j)}}{\lambda^{(j)} + \mu^{(j)}} \cdot p_i^{(j)}. \tag{11}$$

The above analysis gives an important structural property of the steady-state solution as a side product.

Theorem 1 (Insensitivity in list size). *The steady-state probabilities to find posts from user i on the Wall of any user n ($q_i^{(n)}$, $n = 1, \ldots, N$) as well as on the Newsfeed of any user n ($p_i^{(n)}$, $n = 1, \ldots, N$), depend neither on the size M of the Newsfeed, nor on the size K of the Wall.*

Proof. The proof directly comes from the set of $2N$ equations (8)-(9) for the Newsfeeds and (10)-(11) for the Walls, that depend neither on M nor on K. □

B. Linear system

We can re-write (8)-(9) and (10)-(11) for posts with label i in a compact form and summarize our findings as follows.

Theorem 2 (Linear System). *The unknown column vectors $\mathbf{p}_i := (p_i^{(1)}, \ldots, p_i^{(N)})^T$ and $\mathbf{q}_i := (q_i^{(1)}, \ldots, q_i^{(N)})^T$ are the solution of the following linear system*

$$\mathbf{p}_i = \mathbf{A} \cdot \mathbf{p}_i + \mathbf{b}_i \tag{12}$$

$$\mathbf{q}_i = \mathbf{C} \cdot \mathbf{p}_i + \mathbf{d}_i. \tag{13}$$

In the above, $\mathbf{A}$ and $\mathbf{C}$ are $N \times N$ matrices independent of i, whereas $\mathbf{b}_i$ and $\mathbf{d}_i$ are N-column vectors that depend on i.

Hence, a standard linear system should be resolved for each i. The entries of the above matrices and vectors are summarised in Table I. It is interesting to note that $a_{j,j} = 0$ for all j, $b_{i,i} = 0$, $\mathbf{C}$ is diagonal, and also there is a unique positive $d_{i,j}$ entry for $i = j$.

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

TABLE I: Entries for the matrices/vectors of the linear system.

The matrix $\mathbf{A}$ is non-negative. In addition, it is row sub-stochastic, meaning that the sum of all its rows is less or equal to 1, with at least one row sum strictly less than 1 (if we reasonably assume that at least one user injects self-posts). Another interesting property is that $\mathbf{A}$ is a weighted version of the Follower-matrix $\mathbf{F} = \mathbf{L}^T$, so that if $\mathbf{1}_{\{j\in\mathcal{F}^{(k)}\}} = \mathbf{1}_{\{k\in\mathcal{L}^{(j)}\}} = 0 \Rightarrow a_{j,k} = 0$. There are cases however where j follows ℓ, but $a_{j,\ell} = 0$ in the matrix $\mathbf{A}$, because $\mu^{(\ell)} = 0$. Hence, users that never re-post from their Newsfeed alter the possibilities of post propagation in the graph. This is why we call $\mathbf{A}$, the *propagation matrix.*

C. Closed-form solution

We would like to know under which conditions a solution to the linear system in (12) - and as a consequence (13) - exists. To this aim we first recall the following known Lemma, where $\mathbf{I}_N$ is the $N \times N$ identity matrix. It relates the solution of our system to the spectral radius of $\mathbf{A}$, denoted by $\rho(\mathbf{A})$.

Lemma 1. *[15, Chapter 6, Lemma 2.1] Given a nonnegative matrix $\mathbf{T} \in \mathbb{R}_+^{N\times N}$, its spectral radius is $\rho(\mathbf{T}) < 1$ if and only if $(\mathbf{I}_N - \mathbf{T})^{-1}$ exists, which can be written as the series*

$$(\mathbf{I}_N - \mathbf{T})^{-1} \quad = \quad \sum_{n=0}^{\infty} \mathbf{T}^n \geq 0. \tag{14}$$

From the specific structure of the non-negative matrix $\mathbf{A}$ we have the following property.

Lemma 2. $\rho(\mathbf{A}) \leq 1$. *Strict inequality is guaranteed in the following non-exclusive non-exhaustive cases (cs):*

(cs1) $\lambda^{(n)} > 0$, $\forall n \in \mathcal{N}$.

(cs2) For every cycle in the Leader-graph, at least one participating user has a leader k with positive self-post rate.

Proof. Let us denote the row sums of $\mathbf{A}$ by $r(j)$, $j = 1 \ldots N$. Then $r(j) \leq 1$ by definition from Table I. It is known that ([17, Theorem 8.1.22]) the following bounds are valid for the spectral radius of a non-negative matrix: $\min_{j=1}^N r(j) \leq \rho(\mathbf{A}) \leq \max_{j=1}^N r(j)$. The right-hand side in our case is 1 and the first part is proven.

(cs1) When $\lambda^{(n)} > 0$, $\forall n$, then $\forall j$ and $k \in \mathcal{L}^{(j)}$, $a_{j,k} < \mu^{(k)}/\sum_{\ell\in\mathcal{L}^{(j)}}\mu^{(\ell)}$, so that $r(j) < 1$, $\forall j$. Then the matrix is *strictly* sub-stochastic, and $\rho(\mathbf{A}) \leq \max_{j=1}^N r(j) < 1$.

(cs2) In this case, suppose the length of a particular cycle is $\gamma > 1$ and the participating nodes are $n_1, \ldots, n_\gamma$. Then at least one row sum $r(j) < 1$, $j \in \{n_1, \ldots, n_\gamma\}$. By direct application of the Al'pin, Elsner, van den Dreissche bound [16, Theorem A], we conclude that $\rho(\mathbf{A}) < 1$. An additional condition for this bound is that $r(j) > 0$, $\forall j$, which is satisfied when $\mathcal{L}^{(j)} \neq \emptyset$, $\forall j \in \mathcal{N}$ and not all leaders of some user have $\mu^{(k)} = 0$. □

Remark 1. *A special instance of (cs2) is when $\mathbf{A}$ is irreducible and $\lambda^{(j)} > 0$ for at least one $j \in \mathcal{N}$.*

Theorem 3 (**Solution**)**.** *For the two cases of Lemma 2, the solution of the linear system (12)-(13) is unique, and given by*

$$\mathbf{p}_i \quad = \quad (\mathbf{I}_N - \mathbf{A})^{-1}\mathbf{b}_i \tag{15}$$

$$\mathbf{q}_i \quad = \quad \mathbf{C}(\mathbf{I}_N - \mathbf{A})^{-1}\mathbf{b}_i + \mathbf{d}_i. \tag{16}$$

Proof. Lemma 2 guarantees that $\rho(\mathbf{A}) < 1$ in both cases, so that from Lemma 1 the inverse $(\mathbf{I}_N - \mathbf{A})^{-1} \geq 0$ exists and the solution is unique. □

An interesting observation is that the inverse $(\mathbf{I}_N - \mathbf{A})^{-1}$ involved in the derivation of $\mathbf{p}_i$ (relation (15)) is independent of i. Thus, in the solution process the inverse should be calculated only once, and then applied to the expressions in (15)-(16) for labels $i = 1, \ldots, N$.

D. Fixed-point algorithm

For large N it can be practically very difficult to calculate the inverse $(\mathbf{I}_N - \mathbf{A})^{-1}$. A different way to proceed in order to solve the system (12) is to use an iterative approach.

Theorem 4. *For the two cases of Lemma 2 and any initialization vector $\mathbf{p}_i(0)$, the discrete-time linear system (17) converges towards the fixed-point solution (15) when $t \to \infty$.*

$$\mathbf{p}_i(t) = \mathbf{A} \cdot \mathbf{p}_i(t-1) + \mathbf{b}_i \tag{17}$$

Proof. We first write $\mathbf{p}_i(t)$ as a function of $\mathbf{p}_i(0)$ and t,

$$\mathbf{p}_i(t) = \mathbf{A}^t\mathbf{p}_i(0) + \left(\sum_{n=0}^{t-1}\mathbf{A}^n\right)\mathbf{b}_i.$$

We need to find the limiting value $\mathbf{p}_i := \lim_{t\to\infty}\mathbf{p}_i(t)$. For the two cases in Lemma 2 we have $\rho(\mathbf{A}) < 1$, so that from [17, pp.137–138, or Theorem 5.6.12] it holds $\mathbf{A}^\infty := \lim_{t\to\infty}\mathbf{A}^t = \mathbf{0}$. Additionally, from Lemma 1 the limit of the matrix series for $t \to \infty$ converges to $(\mathbf{I}_N - \mathbf{A})^{-1}$. Hence, the iteration converges to the solution (15), and is independent of the initialisation $\mathbf{p}_i(0)$. □

It is reminiscent of the DeGroot model [9]. Note that once the Newsfeed-vector $\mathbf{p}_i := \lim_{t\to\infty}\mathbf{p}_i(t)$ has been obtained, the Wall-vector $\mathbf{q}_i$ can be calculated from relation (13). The performance value Ψ_i is then directly derived from (1).

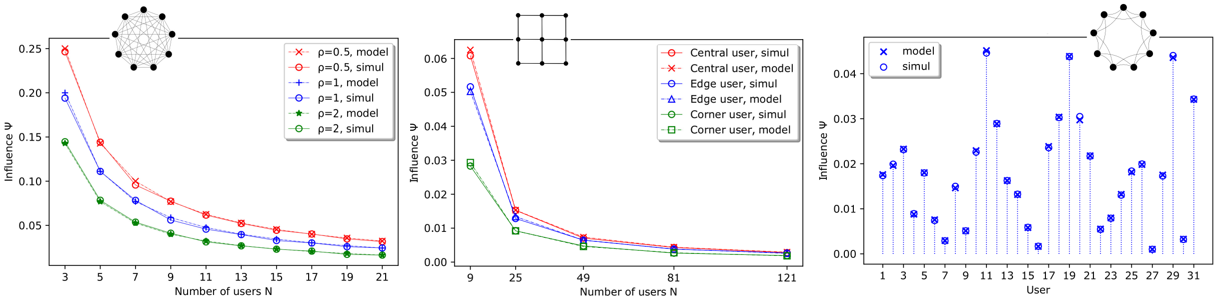

Fig. 3: Validation of the model: (a) left: Complete graph, (b) middle: Grid graph, (c) right: Ring graph.

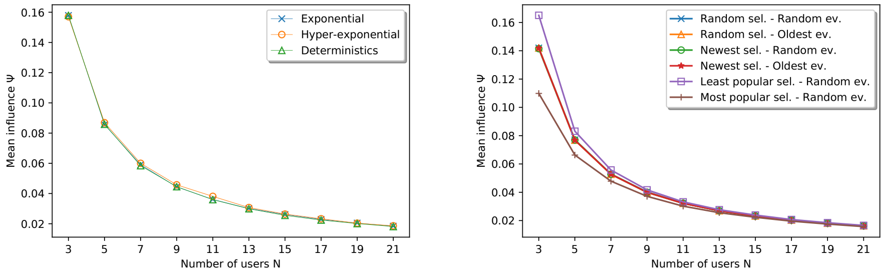

Fig. 4: Sensitivity with respect to modeling assumptions: (a) left: Inter-arrival distributions, (b) right: Selection/Eviction policies.

V. Numerical Evaluation

We have shown that the linear system in Theorem 2 can be solved either by matrix inversion, or by using a fixed-point iteration algorithm. We have programmed both methods in Python for any social-graph input and plan to make the code freely available. This code will be used for the numerical evaluation. Additionally, we have developed our own discrete event simulator to validate the mathematical analysis through simulation, and furthermore to evaluate the robustness of the modeling assumptions presented in Section III-A. Unlike our model, the simulator precisely implements the behavior of the generic OSP as described in Section II: i) The global state description consists of dynamic lists (of length K for Walls and M for Newsfeeds); ii) A variety of selection and eviction policies are implemented (random, in a first phase, and newest, oldest, popular later to evaluate robustness); iii) Self- and re-posts can be generated according to Poisson or other processes. As such, the simulator *does not* decouple the state space, *does not* estimate average probabilities, and *does not* rely on Markovian assumptions. For each simulation we set $M = 20$ and $K = 10$ and ran long enough simulations to reach the steady-state with small confidence intervals. More specifically, in all experiments, we let the simulator run for a total of 300 000 events (self- and re-posts).

A. Validation

We compare the values of the influence metric (1) resulting from the numerical model with those obtained by simulation. We use three different configurations for the user graph: complete graph, grid and ring. The results are given in Fig. 3.

a) Complete graph: In this case, each user follows all other users. All users have the same activity tuple (λ, μ). As the network is totally symmetric, we plot the influence of any user over the network for three different values of $\rho := \frac{\lambda}{\mu}$ (0.5, 1 and 2), as a function of the network size N. As shown on Fig. 3(a), there is a very good fit between model and simulation, with a maximal relative error of about 0.5%. As a qualitative result, we observe that the influence of a given user decreases as the size of the network increases. This is reasonable since the more the users in the network, the larger the competition between users to influence the Wall of eachother, thus the smaller the influence per user. Furthermore, we observe that the smaller the ρ, the higher the influence. This result shows that in a fully symmetric network, when everyone decreases his self-post activity, there is more space left on his Wall for being influenced by others through re-posting.

b) Grid graph: Depending on his position in the grid, a user may have 4, 3 or 2 leaders. All users have the same activity tuple (λ, μ), here set to $(5, 3)$. On Fig. 3(b), we plot the influence metric for three different types of users : the central user (with 4 leaders), a user at the middle of an edge (with 3 leaders) and a user at a corner (with 2 leaders), as a function of the network size, $N = 9, 25, 49, 81, 121$. We again obtain a very good match between the results of the model and simulation, with a maximal absolute error of about 10^{-3}. We observe that the corner user has less influence than the edge user, who in turn has less influence than the central user. This is an obvious qualitative result, considering the different numbers of followers each user has, but here we quantify the impact of the position on the chosen influence metric.

c) Ring graph: Users are arranged on a circle and each user i has R_i leaders on his right ($i+1$, ..., $i+R_i$) and R_i leaders on his left ($i-1$, ..., $i-R_i$), R_i being denoted as the *radius* of user i. For the comparison we use $N = 31$ users. Each one has been given a random uniform radius in $\{1, ..., 15\}$ and a random uniform activity tuple (λ_i, μ_i) in $[0.1, 10] \times [0.1, 10]$. We plot the influence for each user in Fig.3(c). Once again, we observe a very good fit between model and simulation, even for the user who has the least accurate estimation (user "20" with a relative error of 2.5%).

B. Robustness

We further evaluate the robustness of the model with respect to the modeling assumptions: Poisson arrival processes, random selection and random eviction policies. For this purpose, we modified our simulator to take into account alternative inter-arrival distributions, as well as alternative selection and eviction policies. In each simulation of this subsection, we chose a complete graph with a varying number N of users, and we set $(\lambda, \mu) = (10, 5)$.

a) Inter-arrival times: First we evaluate the robustness of the model with respect to the Poisson arrival assumption. In Fig. 4(a), we plot the evolution of the average influence $\sum_i \Psi_i/N$ with three different inter-arrival distributions for both self-post and re-post processes of all users: exponential (corresponding to the original Poisson assumption), hyper-exponential (resulting in a process having more variability than the Poisson process) and deterministic (process with no variability). As can be seen on the figure, the three curves almost coincide. We have confirmed this observation with many different tests, highlighting the fact that the model is almost insensitive to the Poisson assumption.

b) Policies: We programmed in our simulator alternative policies based on age and popularity. In the "newest selection" policy, each user always chooses the most recent post on his Newsfeed list to re-post on his Wall (instead of a random one). Hence, this policy uses extra information on the view order of posts. With the "least popular (resp. most popular) selection", the post to be re-posted is selected among the ones with the maximal (resp. minimal) number of re-posts. This policy also uses extra information on re-post history. "Oldest eviction" means that when a post has to be evicted from a Wall or a Newsfeed, it is the oldest in the list that is chosen (instead of a random one). We observe on Fig. 4(b) that replacing random selection by newest selection and/or replacing random eviction by oldest eviction, has almost no influence on performance. Now, in the case of a selection policy based on popularity, the difference with a random policy becomes higher. This is not surprising, as selection policies based on re-post history, drastically alter the studied system. However, the effect is mainly significant for small networks. We can thus generally conclude that our model is also very robust with regard to the choice of selection and eviction policies.

C. Model exploitation

Having demonstrated the accuracy and the robustness of our model, we now investigate how the influence of a given user is related to his position in the graph and to the relative values of his own activity compared to the activity of other users.

a) Direct vs. indirect influence: We first consider a ring with $N = 31$ users, each one having the same radius $R = 3$, and we plot the influence of user "1" on the other users. Since $R = 3$, user "1" has six followers: users $\{$"2", "3", "4", "29", "30", "31"$\}$. In Fig. 5(a) we represent three curves corresponding to three different values of the ratio $\rho = \frac{\lambda}{\mu}$, namely $\rho = 0.1$, 1, and 10, assuming that each user in the network has the same self-post rate λ and the same re-post rate μ. First, we observe that all the curves are symmetrical, which comes from the symmetry of the user graph. More importantly, we see that user "1" has a greater influence on his direct followers, than on the other users. And obviously, the greater the distance from user "1", the less influence of user "1" on the considered user. Here user "16" is the one that is less influenced by user "1". Interestingly, the difference between the "direct influence", i.e., the influence of a user on one of his follower, and the "indirect influence", i.e., the influence of a user on a node that is not one of his follower, is closely related to ρ. The smaller the ρ, the larger the influence diffusion in the network.

b) Influence and graph position: Here we consider a grid graph with 400 users, each one having the same activity $(\lambda, \mu) = (10, 10)$. Each square represents a user and is colored according to his influence Ψ_i over the network. As expected, the peripheral users, i.e., the users located on the outer edges, and in a more pronounced manner, the corner users are the ones with the smallest influence. This is due to the fact that these users have 3 or 2 direct followers, whereas all others have 4. But contrary to intuition, the central users are not the most influential. In fact, the users with the highest influence are located in the first inner ring, i.e., at one hop from the peripheral ones. And more precisely, the most influential users are the four diagonal neighbors of the corner users. We have verified this property on different graphs. As an example, on a tree, leaves are the less influential users, whereas the parents of leaves are the most influential users. As a conclusion, in a social network where all users have the same activity, being a leader of users with few other leaders increases one's influence. Obviously this is partly due to our definition of influence, but alternative metrics have shown to follow the same trend.

c) Influence and activity: We now want to see if a user with a position that gives him a low influence in the network can counterbalance his bad placement by increasing his posting activity. To this aim we consider again the grid of the previous subsection. We set $(\lambda, \mu) = (10, 10)$ for all users except for the south-west corner user who is given the same re-post rate, $\mu_{corner} = 10$, but that can adjust his self-post rate λ_{corner}. In Fig. 5(c), we let only λ_{corner} vary from 0.1 to 100, keeping (λ, μ) and μ_{corner} fixed, and plot the activity of the considered corner user, as well as the activity of the central user and of the diagonal neighbor of the corner user (the most influential in a network with symmetric activities).

First of all, we observe that the corner user becomes more

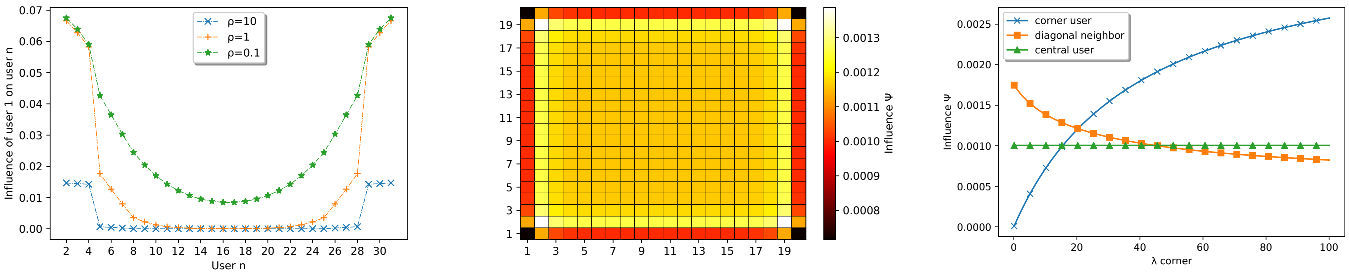

Fig. 5: Model exploitation: (a) left: Ring graph, (b) middle: Grid - fixed symmetric activity, (c) right: Grid - influence evolution.

and more influential as his posting rate increases, and he eventually becomes the most influential user in the network. The central user is too far away from the corner user to be affected by the evolution of his posting rate, so his influence remains constant. However, observe that the raise of λ_{corner} causes a drop in the influence of his diagonal direct neighbor. As a conclusion, the answer is positive: one can counterbalance his bad position by increasing his self-posting activity.

VI. Conclusions

In this work we have introduced an original mathematical model that analyzes the diffusion of posts in a generic social platform and quantifies the influence of a given user over any other within the entire network. By resolving it we have derived closed-form expressions for metrics of influence. These results constitute a novel powerful toolbox that can be further exploited to understand and design social platforms. They should be further compared and combined with network data analytics to highlight their importance. Our model can be used to derive policies for optimal user activity. But, most importantly we hope that it can lead to platform design rules that offer fair and unbiased access and post circulation for all.

References

[1] S. Goel, D.J. Watts, D.G. Goldstein. The structure of Online Diffusion Networks. *13th ACM Conference on Electronic Commerce (EC)*, Valencia, Spain, 2012.

[2] P. Alex Dow, L.A. Adamic, A. Friggeri. The Anatomy of Large Facebook Cascades. *7th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2013.

[3] D. Kempe, J. Kleinberg, É. Tardos. Maximizing the Spread of Influence Through a Social Network. *ACM KDD'03, New York, NY, USA*, pp.137–146, Aug. 2003.

[4] J. Cheng, L.A. Adamic, P.A. Dow, J. Kleinberg, J. Leskovec. Can cascades be Predicted? *23rd international conference on World wide web (WWW)*, Seoul, Korea, 2014.

[5] S. Zannettou, M. Sirivianos, J. Blackburn, N. Kourtellis. The Web of False Information: Rumors, Fake News, Hoaxes, Clickbait, and Various Other Shenanigans. *arxiv:1804.03461v1*, 2018.

[6] R. A. Holley, T. M. Liggett. Ergodic Theorems for Weakly Interacting Infinite Systems and the Voter Model. *Journal of the American Statistical Association*, Vol.3, No.4, pp.643–663, 1975.

[7] E. Yildiz, A. Ozdaglar, D. Acemoglu, A. Saberi, A. Scaglione. Binary Opinion Dynamics with Stubborn Agents. *ACM Transactions on Economics and Computation*, Vol.1, No.4, Article 19, pp.19:1–19:30, Dec. 2013

[8] V. S. Varma, I.-C. Morarescu, Y. Hayel. Continuous time opinion dynamics of agents with multi-leveled opinions and binary actions. *INFOCOM*, Honolulu, USA, 2018.

[9] M.H. DeGroot. Reaching a Consensus. *Journal of the American Statistical Association*, Vol.69, No.345, pp.118–121, Mar. 1974.

[10] M. Grabisch, A. Mandel, A. Rusinowska, and E. Tanimura. Strategic Influence in Social Networks. *J. Mathematics of Operations Research*, Vol.43, no.1, pp. 29–50, 2018.

[11] A. Silva. Opinion Manipulation in Social Networks. *Network Games, Control, and Optimization (NETGCOOP)*, Springer, pp. 187–198, 2017.

[12] F. Baccelli, A. Chatterjee, S. Vishwanath. Pairwise stochastic bounded confidence opinion dynamics: Heavy tails and stability. *INFOCOM*, pp. 1831–1839, 2015.

[13] N. Spasojevic, Z. Li, A. Rao, P. Bhattacharyya. When-To-Post on Social Networks. *ACM KDD'15, Sydney, NSW, Australia*, pp.2127–2136, Aug. 2015

[14] M. R. Karimi, E. Tavakoli, M. Farajtabar, L. Song, M. Gomez Rodriguez. Smart Broadcasting: Do You Want to Be Seen? *ACM KDD'16, San Francisco, CA, USA*, pp.1635–1644, Aug. 2016

[15] A. Berman, R. J. Plemmons. Nonnegative Matrices in the Mathematical Sciences. *SIAM Classics in Applied Mathematics; 9*, 1994.

[16] L. Elsner, P. van den Driessche. Bounds for the Perron root using max eigenvalues. *Linear Algebra and its Applications* 428, pp. 2000–2005, 2008.

[17] E. A. Horn, C. A. Johnson. Matrix Analysis. *Cambridge University Press*, 1985.