

# Exploring the Role of Intrinsic Nodal Activation on the Spread of Influence in Complex Networks



Arun V. Sathanur, Mahantesh Halappanavar, Yi Shi, and Yalin Sagduyu

**Abstract** In many complex networked systems, such as online social networks, activity originates at certain nodes and subsequently spreads on the network through influence. In this work, we consider the problem of modeling the spread of influence and the identification of influential entities in a complex network when nodal activation can happen via two different mechanisms. The first mechanism of activation stems from factors that are intrinsic to the node. The second mechanism comes from the influence of connected neighbors. After introducing the model, we provide an algorithm to mine for the influential nodes in such a scenario by modifying the well-known influence maximization algorithm. We sketch a proof of the submodularity of the influence function under the new formulation and demonstrate the same on larger graphs. Based on the model, we explain how influential content creators can drive engagement on social media platforms. Using additional experiments on a Twitter dataset, we then show how the formulation can be applied to real-world social media datasets. Finally, we derive a centrality metric that takes into account both the mechanisms of activation and provides for an accurate, computationally efficient, alternate approach to the problem of identifying influencers under intrinsic activation.

**Keywords** Complex networks · Influence maximization · Social influence · Intrinsic activation · Independent cascade · Monte carlo · Centrality · Spectral methods

---

A. V. Sathanur (✉)

Pacific Northwest National Laboratory, Seattle, WA, USA

e-mail: [arun.sathanur@pnnl.gov](mailto:arun.sathanur@pnnl.gov)

M. Halappanavar

Pacific Northwest National Laboratory, Richland, WA, USA

e-mail: [hala@pnnl.gov](mailto:hala@pnnl.gov)

Y. Shi · Y. Sagduyu

Intelligent Automation, Inc., Rockville, MD, USA

e-mail: [yshi@i-a-i.com](mailto:yshi@i-a-i.com); [ysagduyu@i-a-i.com](mailto:ysagduyu@i-a-i.com)

## 1 Introduction and Related Work

The advent and rapid adoption of social media platforms allows people to self-organize into complex social networks with rich dynamics. Users can disseminate their views, opinions, and other content while simultaneously consuming and reacting to the content created by friends, people, and organizations they follow. The success of such platforms depends on the myriad of content creators, the quality of their content, and the activities their audiences generate because of the various types of engagement possible with the posted content. These actions can be attributed to influence. The dynamics of influence and resulting diffusion of information in complex networks has been the subject of intense scrutiny for researchers and practitioners in many fields with particular attention to the identification of central or influential nodes on the network. One rigorous approach to finding influential users with motivations originating in viral marketing is the approach based on *influence maximization*.

We can define the influence maximization problem as follows: Consider a directed graph  $G = (V, E)$  that abstracts a complex network, where  $V$  is the set of nodes  $V = \{v_1, v_2, v_3 \dots\}$  and  $E$  is the set of directed edges  $\{(v_u, v_w) \mid v_u, v_w \in V\}$ . The directed edge  $(v_u, v_w)$  implies that  $v_u$  can influence  $v_w$  and not the other way round. However, it is possible that both  $(v_u, v_w)$  and  $(v_w, v_u)$  are valid edges. We denote by  $|V|$  the total number of nodes and by  $|E|$  the total number of edges in the graph  $G$ . Further, the nodes are labeled as either *Passive* or *Active*, denoting the state of the vertex. A necessary but not sufficient condition for an active vertex  $v_u$  to activate a passive vertex  $v_w$  is that  $(v_u, v_w) \in E$ . Other conditions come from the nature of the diffusion model. Given that it is possible to initially activate  $k$  nodes, the influence maximization problem aims to find the particular set of  $k$  *seed nodes*, called the *seed set*  $S$ . When the nodes in the set  $S$  are activated, the spread of influence results in maximal activations on the network among all possible such sets of  $k$  nodes. Note that in the subsequent discussions, we use the terms *reachability*, *number of activations on the network*, and *influence spread* synonymously to denote the total number of active nodes on the network after running the diffusion models, starting from the initial set of active nodes, until no more activations are possible.

Starting with the landmark paper by Kempe et al. [10], several works have explored newer diffusion models and variations to the ones studied in the work by Kempe et al., namely, the independent cascade (IC) model and the linear threshold (LT) model. These models explicitly address the various sociological aspects of influence. Li et al. in [15] consider influence dynamics and influence maximization under a general voter model with positive and negative edges. In a follow-up work, Kempe et al. [11] discuss a diverse set of models including the so-called decreasing cascade model where attempts by multiple neighbors to activate a node result in decreasing probability of activation, as the size of the set of neighbors trying to activate the node increases. The authors in Ref. [23] propose a general diffusion model that takes into account different granularities of influence, namely, pair-wise, local neighborhood, etc. The authors in [4], consider influence maximization under

the scenario where negative opinions may emerge and propagate. In [8], the authors consider the problem of identifying the individuals whose strong positive opinion about a product will maximize the overall positive opinion about the product. In the process, the authors leverage the social influence model proposed by Friedkin and Johnsen [7]. For a comprehensive survey on the various models of influence, we refer the reader to the paper by Zhang et al. [26].

Next, we consider the models that address two different types of activation: *intrinsic* and *influenced*. The interplay of these two mechanisms is exemplified by three different scenarios outlined below.

- Users posting content on social media due to their own initiative constitutes *intrinsic* activation. Actions such as sharing, retweeting, commenting constitute *influenced* activation.
- Posting behavior that is external to a given network can be considered to be *intrinsic* activation. This would include watching a video on a website from a shared e-mail link and then sharing it on Twitter. From the perspective of just the Twitter network, it appears that such users are *intrinsically* activated.
- In a traditional social network, such as a physical community, that is not an OSN (online social network), *intrinsically* activated users would be those who take the initiative to start an activity, for example, a *campaign for social good*. The same can then spread through *word of mouth*, *flyers*, etc.

Myers, Zhu, and Leskovec investigate the diffusion of information, with origins external to that of a social network, through the internal social influence mechanism [17]. In a recent work [6], the authors recognize that the events on social media can be categorized as exogenous and endogenous and model the overall diffusion through a multivariate Hawke's process to address activity shaping in social networks. In another recent work, the authors in [18] propose a novel diffusion model based on factor graphs and graphical models where the node potentials can correspond to the notion of *intrinsic* activation in our case. However, the focus of *their work is on the diffusion model itself*, not on the aspects of *intrinsic* activation. While being similar in spirit to these works, our work is geared toward modeling the spread of influence and *mining influential nodes in scenarios with intrinsic and influenced activation-aspects* that have not been studied in existing literature.

We make the following contributions in this work:

1. Our approach results in a probabilistic model for two different types of nodal activations, namely, *intrinsic* and *influenced* mechanisms found in real-world networked systems, such as OSNs.
2. We examine these mechanisms in the context of influencer mining from *two different perspectives*: the well-known combinatorial influence maximization perspective and a generalized centrality perspective.
3. We define a *modified influence spread function*, sketch a *proof of its submodularity*, and provide a modified version of the influence maximization algorithm to maximize the new influence spread function.

4. We examine the nature of content creators and consumers on a social network in light of the two activation mechanisms.
5. Carefully chosen experiments on synthetic and real-world graphs are used to illustrate various aspects of the model and compare it to the independent cascade model.
6. We derive a new centrality metric from the activation model and show that this metric can accurately identify influential users in a computationally efficient manner.

The initial aspects of this work was published in [21]. The present version is a significant extension of the above work where we have extensively examined the content creation and content spreading mechanisms, formally sketched a proof of the submodularity of the modified influence function, added an extensive set of experiments on a real-world Twitter graph, and improved the overall narrative by means of several smaller additions.

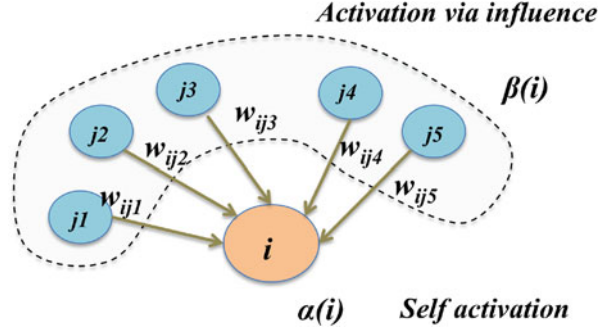
## 2 Modified Influence Maximization Approach

### 2.1 Formulation

Considering that nodal activation can originate from two different mechanisms, *Intrinsic* and *Influenced*, allows us to effectively model the so-called *self-evolving* systems (e.g., OSNs) that are comprised of *content creators* (higher probability of getting activated intrinsically) and *content spreaders* (activated via influence and spreading the information). Recognizing that most of the users are in some sense both activity creators and content spreaders (typically also the content consumers) at the same time, we introduce a real-valued parameter  $\alpha \in [0, 1]$  that models the probability of self-activation. The total probability for activation of a given node (user)  $i$  is composed of the probability of activation from the two different mechanisms. The parameter  $\alpha(i)$  denotes the probability of intrinsic activation, and  $\beta(i)$  denotes the probability through influence with  $\alpha(i) \geq 0$  and  $\beta(i) \geq 0$ . The influenced part of the probability for activation is comprised of the activation probabilities due the 1-hop neighbors of the user under consideration.

Note that there are many interaction models that are studied under influence maximization as pointed out in Sect. 1. Our model with intrinsic activation is based on one of the most widely studied models, namely, the independent cascade model, and this will be the focus of the current work. Specifically, in this work, we do not consider developing the variants of other models incorporating intrinsic activation. Similar to the IC model, the weights  $w_{ij}$  ( $0 \leq w_{ij} \leq 1$ ) when multiplied by  $\beta(i)$  denote the probability of user  $j$  activating user  $i$ , given that user  $j$  is activated by either of the above means. Figure 1 describes these mechanisms and the associated

**Fig. 1** A concise representation of the self and influenced mechanisms of activation of a node  $i$



coefficients. The described probabilistic formulation has similarities to the Friedkin-Johnsen social influence model for opinion change [7] where the authors recognize that the dynamics of opinion change are governed by two mechanisms—the intrinsic opinion and the influenced opinion.

We also adopt the weighted-cascade version of the IC model by normalizing the edge probabilities [10], so that the expected number of nodes influencing a given node is 1. Henceforth, in this work, when we refer to the IC model, we imply the weighted cascade version of the IC model. However, this is not a limitation of the model since our model can also be used in the pure IC model setting. Thus, if  $\mathbf{W}$  denotes the sparse weight matrix that characterizes the IC edge probabilities, we require that  $\mathbf{W}$  be row stochastic. That is,  $\sum_{j, (j,i) \in E} w_{ij} = 1$ . Further by assuming that the nodes are not *lazy* and are activated by either of the two mechanisms that we outline, we set  $\beta(i) = (1 - \alpha(i))$ . This will render the overall IC probability between nodes  $j$  and  $i$  to be  $(1 - \alpha(i))w_{ij}$ . While the changed probabilities denote a departure from the weighted-cascade model, when the effect of intrinsic activation is added back, the expected number of nodes activating a given node is still 1.

Note that all the model parameters discussed can be efficiently determined either by a maximum-likelihood-based approach (as in this work) or by alternative methods such as the expectation-maximization (EM) approach followed in reference [19]. For example, the proportion of tweets by a user  $i$  that are intrinsic in nature can quantify  $\alpha(i)$ , while a particular weight  $w_{ij}$  can be determined by the proportion of user  $i$ 's retweets (or influenced activity) having their origin in the activity of user  $j$  that user  $i$  follows. While these *local influence models* can be determined in alternate ways, our goal is to find the overall influencers once these model parameters are estimated.

Our formulation addresses the problem of identifying influential nodes on a network without explicit seeding. The original influence maximization approach with roots in viral marketing explicitly activates the seed nodes while in our formulation, the system is self-evolving in that nodes get activated intrinsically with a probability (content creation) and subsequently these activations spread (content consumption and spreading) on the network. This is the focus of the next section which describes the modifications to the original influence maximization algorithm necessary to identify the influencers under intrinsic activation.

## 2.2 Algorithm for Mining Influential Nodes Under Intrinsic Activation

We propose a simple modification to the classic influence maximization framework using the greedy hill-climbing optimizer [10], working with the IC model, to incorporate the self-activation mechanism. Let us assume that we are seeking  $k$  influential nodes out of a total of  $N$  nodes on the network. Let  $S^p$  be the set of influential nodes at step  $p \leq k$ . The greedy hill-climbing optimizer expands the set to size  $(p + 1)$  by polling each of the nodes not in  $S^p$  and augmenting those nodes, one at a time to form the set  $S^p \cup \{v\}$  and looking for the best marginal gain in terms of the activations. At each such step  $p$ , instead of setting each of the nodes in  $S^p \cup \{v\}$  to be activated and then computing the activations according to the IC model, we probabilistically activate each node in  $S^p \cup \{v\}$  with a probability given by the corresponding  $\alpha$  values to simulate the intrinsic activation process. This modification is depicted in line 9 of Algorithm 1. Given the probabilistic nature of the algorithms, the overall activation numbers are obtained by running the diffusion model in a Monte Carlo fashion by invoking  $n$  independent trials involving randomized graphs with corresponding edge weights.

Note that this algorithm results in the computation of a modified influence spread objective function  $\sigma(S)$  (same as  $a_{best}$  in the algorithm), which gives us the total number of activations on the network attributable to the multi-hop influence of

---

**Algorithm 1** Selects a set of  $k$  influential nodes that cause maximal activations on a network, following the independent cascade (IC) model with self-activation (IC-Int). The inputs are a directed graph ( $G = (V, E)$ ), set of edge probabilities ( $P = \{w_{ij} : (ji) \in E\}$ ), vector of alpha values ( $\alpha = \{\alpha_v : v \in V\}$ ), number of samples ( $n$ ), and number of influential nodes to be identified ( $k$ )

---

```

1: procedure IC-INT( $G, P, \alpha, k, n$ )
2:   Generate  $n$  random numbers  $r_{uv}^1 \dots r_{uv}^n$  for each edge in  $E$  and generate a set  $SG$  containing
    $n$  subgraphs such that in subgraph  $i$ ,  $w_{ij} \geq r_{uv}^i$ 
3:    $S \leftarrow \emptyset$  ▷ Set of influential nodes to be mined
4:   while  $|S| < k$  do
5:      $v_{best} \leftarrow \emptyset, a_{best} \leftarrow 0$ 
6:     for each node  $v$  in  $V \setminus S$  do
7:        $a \leftarrow 0$ 
8:       for each  $G_i \in SG$  in parallel do
9:          $\hat{S} \leftarrow$  active nodes in  $S \cup \{v\}$  based on  $\alpha$ 
10:        Compute number of nodes,  $\hat{a}$ , in  $V \setminus \hat{S}$  that are reachable from the  $\hat{S}$ 
11:         $a \leftarrow a + \hat{a}$  ▷ Synchronized update
12:       if  $a \geq a_{best}$  then
13:          $v_{best} \leftarrow v$ 
14:          $a_{best} \leftarrow a$ 
15:       if  $v_{best} \neq \emptyset$  then
16:          $S \leftarrow S \cup \{v_{best}\}$ 
17:   return  $S$ 

```

---

nodes in the set  $S$  when the corresponding nodes are activated intrinsically, in accordance with their  $\alpha$  values. Thus, during this process, at the step denoted by line 10, the nodes in the set  $(V \setminus S^p \cup \{v\})$  are not activated intrinsically, instead they are activated via influence. These aspects are discussed further in Sects. 2.3 and 2.4.

The running time of Algorithm 1 depends on the  $\alpha$  values since they affect whether a particular node is active in the given sample or not (line 9 of the algorithm). If the node is not active, then reachability will not be computed from that node. The worst-case complexity of the algorithm is the same as that for the independent cascade model and can be derived to be  $O(nk^2|V||E|)$  where  $n$  is the number of Monte Carlo samples,  $k$  is the number of influential nodes sought,  $|V|$  is the number of nodes (vertices) in the graph, and  $|E|$  is the number of edges in the graph. The approach to solve the problem as detailed in Algorithm 1 is based on the classic greedy algorithm to maximize monotone submodular functions. There are two ways to make the algorithm scalable. One is to accelerate the outer greedy optimization loop, and the second method is to improve the scalability of the reachability computation. There is prior work on both the areas. For example, the work presented in [14] uses lazy evaluations to speed up the greedy algorithm while reference [16] uses a stochastic version of the greedy algorithm to improve the scalability. References [5] and [3] on the other hand use techniques to speed up the reachability evaluations on the sampled subgraphs. Because our objective function is also monotone submodular (more on this in Sect. 2.4) and uses reachability computations, it can benefit from these algorithms to scale to networks with millions of nodes.

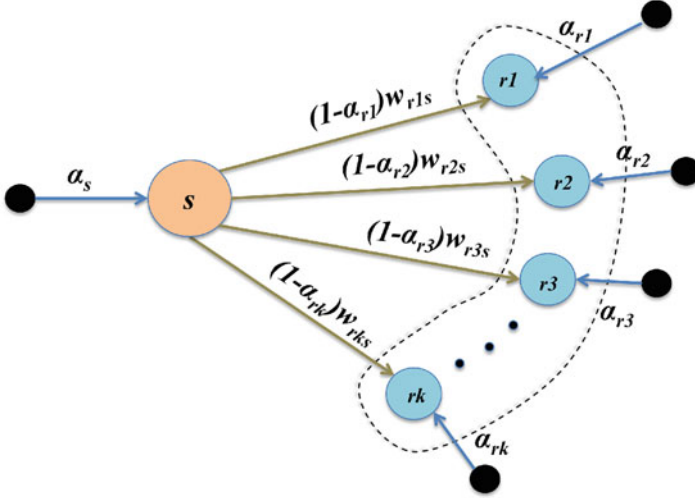
### 2.3 Content Creators and Engagement in Online Social Networks

With the help of the described activation model, we examine aspects of content creation, consumption, and content spreading in OSNs, and how these are tied to the success of the platform as a whole. Figure 2 shows the out-links around a source node ( $s$ ) and the various receiver (follower) nodes ( $r_1 \dots r_k$ ) with the  $\alpha$  and  $w$  values.

Using the described activation model, it is evident that by restricting ourselves to one hop, we can write the modified influence function  $\sigma(s)$  that denotes the expected number of nodes activated by node  $s$  as follows:

$$\sigma(s) = \alpha_s \sum_{k=1}^{d_s^o} (1 - \alpha_{r_k}) w_{r_k s} \quad (1)$$

Here,  $d_s^o$  denotes the out-degree of the node  $s$ . We are only modeling the activation through influence when the source node  $s$  gets activated intrinsically.



**Fig. 2** The out-links from a source node ( $s$ ) to the receiver nodes ( $r_1 \dots r_k$ ), and the associated node and edge attributes

We do not add the term  $\sum_{k=1}^{d_s^o} \alpha_{r_k}$  that denotes the intrinsic activation of the nodes ( $r_1 \dots r_k$ ). This is because activation via influence creates engagement on the social network platform (e.g., sharing, commenting, and liking). Thus  $\sigma(s)$  can be viewed as a surrogate for engagement. The set of influential nodes (users) that we wish to compute by following Algorithm 1 denotes the set of influential content creators that are able to maximize this engagement (by which we mean the influence spread under intrinsic activation of the creator nodes), and are quite valuable to the platforms.

The scenario in which all nodes have large values of  $\alpha$  will result in nodes creating high volume of content on their own, and there is not much spreading of the content through different forms of engagement. Alternatively, all nodes having rather small values of  $\alpha$  means that while the nodes are eager to spread the content, there is not much content created in the first place, again reducing the engagement. Therefore, we hypothesize there is an optimal assignment of the  $\alpha$  values for a given assignment of the interaction probabilities and the network topology that can maximize the spread of influence under intrinsic activation. While we provide evidence of this with experiments on a real-world Twitter dataset in Sect. 4, solving an actual optimization problem is beyond the scope of this work.

Equation (1) provides a quick preview of the distribution of the  $\alpha$  values that can lead to maximizing this engagement. The objective function  $\sigma(s)$  favors a source node with large  $\alpha_s$  and high out-degree, connected to receivers with low  $\alpha_{r_k}$  values who easily engage with the intrinsic activity of the source node (higher value of the IC probability along these edges and lower value of the receiver  $\alpha$ ). In practice when users can have arbitrary  $\alpha$  values, Algorithm 1 is able to seamlessly identify such influential content creators by simulating the two mechanisms of activation.



The same will not be possible with the independent cascade model because every node that is selected to be a part of the seed set is necessarily activated, thereby overestimating a given node's influence.

## 2.4 Optimality of the Influence Maximization Algorithm with Intrinsic Activation

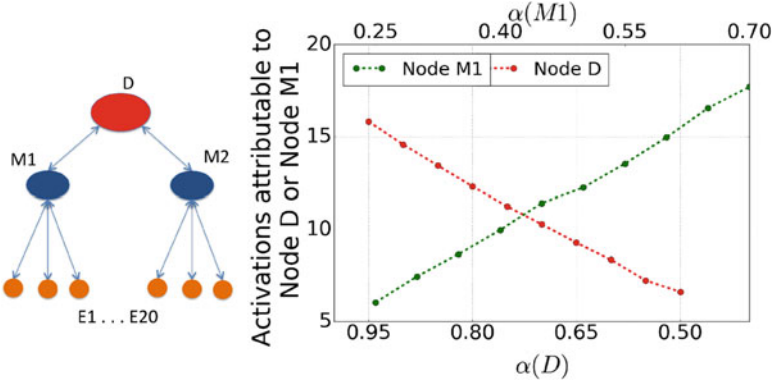
For the classic influence maximization problem with the IC model, the greedy hill-climbing optimizer is shown to be optimal in the sense that it provides  $(1 - \frac{1}{e} - \epsilon)$  approximation guarantee on the expected influence spread function. This is because the expected influence spread  $\sigma(S)$  is a monotone submodular function [10, 12]. The greedy algorithm expands the seed set  $S$  by the addition of nodes with the highest marginal gain in terms of the number of activations. For the case of intrinsic nodal activation, we have nodes activated intrinsically, as well as through influence. Thus, it appears that an influence function defined by the total number of activations on the network is not submodular. However, given that we are only interested in the total number of activations caused by the spread of intrinsic activations ( $a_{best}$  in Algorithm 1), the submodularity property can be shown to remain valid.

For each node on the network  $i \in V$ , we can introduce an edge pointing from a newly created dummy node  $i_D$  to the actual node  $i$  with an activation probability equal to  $\alpha_i$ . This process is illustrated in Fig. 2. Let  $V_D$  denote the set of dummy nodes. Note that there is a one-to-one correspondence between the nodes in  $V_D$  and  $V$ . Also, because every node  $i_D$  in  $V_D$  has a single outgoing edge to the corresponding node  $i$  in  $V$ ,  $i_D$  cannot be activated by  $i$ . On the other hand, given a large number of samples  $n$ , the expected number of times the edge between any pair of nodes ( $i_D \in V_D, i \in V$ ) is activated is  $n\alpha_i$  leading to us to represent the IC probability between  $i_D$  and  $i$  to be  $\alpha_i$ . This is represented by line 9 of Algorithm 1. Thus, the original influencer mining problem can now be transformed to mining for influential nodes in the set  $V_D$  under the IC model. Given that the influence (cumulative reachability) function is submodular under the IC model [10], the influence function in the case of intrinsic activation being present, namely,  $a_{best}$  in Algorithm 1, is also submodular.

## 3 Synthetic Experiments

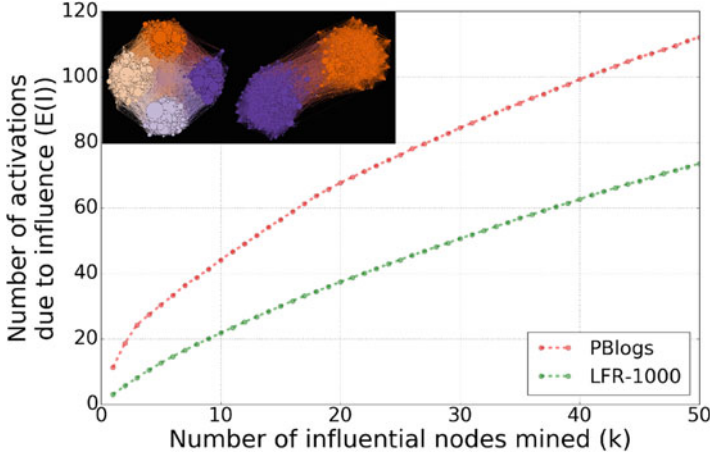
### 3.1 Small Organization Tree

We first consider a small directed and weighted network with 23 nodes, organized in a tree-like fashion. The graph is depicted on the left side of Fig. 3. In this experiment, we consider the tree-like network to depict a small organization with



**Fig. 3** The small organizational tree network (left) and the behavior of the influence functions with the various  $\alpha$  values

a director (Node D), two managers (M1 and M2), and 20 employees (E1–E20), with 10 employees each working under the two managers. We set  $\alpha^0(D) = 0.95$ , signifying that the director almost exclusively acts intrinsically. We also set  $\alpha^0(M1) = \alpha^0(M2) = 0.25$ . All Employees have an  $\alpha$  of 0.25 as well. As for the weights (same as the activation probabilities in the IC model), the edges ending at node D receive weights of 0.5 each (when the director chooses to be influenced, the director gets influenced equally by the two managers). As for the managers, they have a weight of 0.5 each on the edges that are incoming from D and the remaining 0.5 is split equally among the edges originating at the 10 employees each. All Employees carry a weight of 1.0 on the edges originating from the managers. We then perturb this baseline case to mimic a situation where the director starts becoming more susceptible to influence, while the manager M1 becomes inflexible. This is simulated by setting  $\alpha(D) = \alpha^0(D) - \delta$  and  $\alpha(M1) = \alpha^0(M1) + \delta$ . We then sweep  $\delta$  from 0.05 to 0.45. The results are shown in the right panel of Fig. 3, where we can see that D begins as the most influential node as expected, but then M1 becomes more influential than D at a certain value of  $\delta$  and will eventually have reach over most of the employees on the network. Note that the activation numbers plotted on the y-axis are the expected numbers over a Monte Carlo analysis with  $n = 3200$  samples. This simple experiment shows that the nature of influence spread on social networks is sensitive to the extent of intrinsic activation of the key nodes. Clearly, these scenarios cannot be easily captured by the IC model, where the concept of intrinsic activation with a continuous probability value ( $\alpha \in [0, 1]$ ) does not exist.



**Fig. 4** Submodular nature of the influence function under self-activation. *Inset: The PBlogs (left) and LFR-1000 (right) networks visualized in Gephi [2].*  $x$ -Axis refers to the number of influential nodes mined and  $y$ -axis refers to the expected number of activations achieved due to influence. This is represented as  $E(I)$

### 3.2 Larger Graphs and the Influence Function

Our next experiment involves two larger graphs where the topology of one is from a real-world dataset, while the other is synthesized. In both cases, the node  $\alpha$  values are drawn from a uniform distribution  $U[0, 1]$ , and the  $w_{ij}$  values are also drawn from  $U[0, 1]$  and then normalized as described earlier in Sect. 2.1. The graphs under consideration are described below:

- LFR-1000 graph with 1000 nodes and 11,433 edges is a synthetic network that follows the generative LFR model that mimics real-world graphs [13].
- The PBlogs graph [1] represents a real-world blogs network and has 1095 nodes and 12,597 edges.

Further details of these graphs are discussed in [9].

When we applied the modified influence maximization approach given by Algorithm 1 to the LFR-1000 and the PBlogs graphs and requested for 50 seeds, we observed (Fig. 4) that the cumulative influence spread (total number of influenced activations) showed a **submodular character** as evidenced by the diminishing gains in the total number of activations for each new seed added to the set.

## 4 Experiments on a Real-World Twitter Dataset

In this section, we consider the various aspects of intrinsic activation on an interaction graph constructed from Twitter data.

## 4.1 Data Collection

We first build a directed follower/friend graph from Twitter data using the public Twitter API [24], where each user is a vertex and a directed edge  $(u, v)$  from user  $u$  to user  $v$  means that  $v$  follows  $u$ . Our goal is to capture a portion of the Twitter graph such that there are enough interactions between the nodes to estimate the  $\alpha$  and the  $w_{ij}$  parameter values with reasonable confidence as required by our model. Algorithm 2 depicts the graph construction details.

Algorithm 2 starts by adding a seed twitter user  $u_0$  to the set  $S$  (line 2) and by adding followers. In order to improve the density of the graph (as measured by  $\frac{|E|}{|V|^2}$ ), we pick up to  $k_{in}$  ( $k_{in}$  is set to 15 in our experiment) users with highest in-degree values in the set  $S$  to form a new set  $S' \subseteq S$ . If  $k_{in} > |S|$ , then we will just pick all the nodes in set  $S$ . New vertices and edges are added accordingly (lines 9–15). Note that new vertices are added if the users are being introduced for the first time. Low out-degree nodes (based on the threshold  $k_{out}$ , which is set to 11 in our experiments) are excluded in the graph construction (line 18). The process is repeated until a required number of vertices have been added to  $G$ . Random seeds are added to  $S$  when it becomes empty (line 17).

---

**Algorithm 2** Generate a directed graph,  $G = (V, E)$ , from the given Twitter dataset. Unique users in the dataset are represented as nodes, and the notion of a follower is represented as a directed edge. If user  $v$  follows user  $u$ , then add a directed edge  $(u, v)$ . The desired number of nodes  $n$  and sampling parameters  $k_{in}$  and  $k_{out}$  are provided as inputs to the algorithm

---

```

1: procedure TWITTER-GRAPHGEN( $G = (V, E), k_{in}, k_{out}, n$ )
2:    $S = \{u_0\}$  ▷ Seed User
3:    $V(G) \leftarrow \{u_0\}$ 
4:    $E(G) \leftarrow \emptyset$ 
5:   while  $|V| < n$  do ▷ Graph is less than the desired size
6:      $S' \subseteq S$  ▷ Select  $k_{in}$  users from set  $S$  such as based on top in-degree
7:      $S \leftarrow S \setminus S'$ 
8:      $F \leftarrow \emptyset$ 
9:     for each user  $u$  in  $S'$  do
10:       $F \leftarrow F \cup \{\text{all followers of user } u\}$ 
11:      for each node  $v$  in  $F$  do
12:        if  $v \notin V(G)$  then
13:           $V \leftarrow V \cup \{v\}$ 
14:           $S \leftarrow S \cup \{v\}$ 
15:           $E \leftarrow (u, v)$  ▷ Add a directed edge  $(u, v)$ 
16:     if  $S = \emptyset$  then
17:        $S = \{u_{rand}\}$  ▷ Randomly select a user from the dataset
18:   Recursively remove nodes with out-degree  $< k_{out}$ . Goto Line 5
19: return  $G$ 

```

---

Starting with the user “PurdueEngineers” as the *seed\_user*  $u_0$  to collect data, and following Algorithm 2, we obtain a graph with 1167 nodes and 10,292 edges. We then generate an interaction graph from the follower/friend graph by assigning a weight on each edge  $(u, v)$  based on the number of interactions, where interactions refer to replies, retweets, or mentions. We define weight  $\gamma_{(u,v)}$  for a directed edge  $(u, v)$  as one plus the number of times user  $v$  replies to, retweets, or mentions user  $u$ . Note that we need to define positive weights and thus we define weight prior to normalization by one plus the number of interactions.

## 4.2 Influence Spread Results

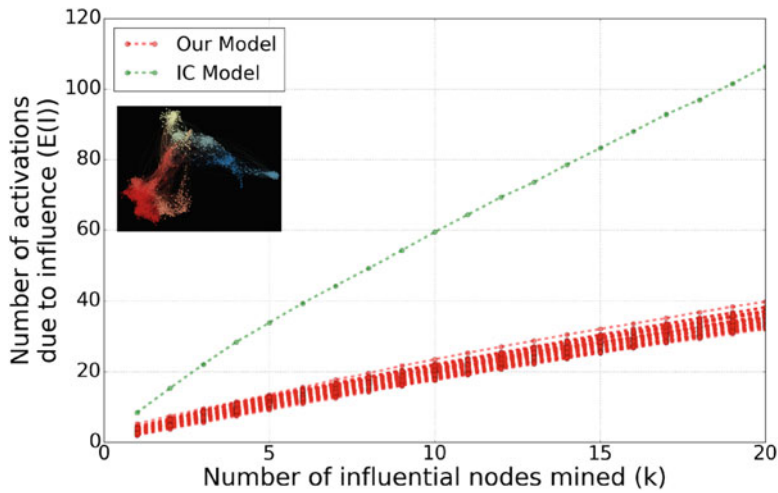
Given the dataset with tweets and interactions in the form of retweets, replies, and mentions, we estimate the node-specific parameters  $\alpha_i$  and the edge-specific parameters  $w_{ij}$  as below.

$$\begin{aligned}\gamma_i &= \sum_j \gamma_{(j,i)}, \\ \alpha_i &= \frac{k_i}{\gamma_i + k_i}, \\ \beta_i &= 1 - \alpha_i, \\ w_{ij} &= \frac{\gamma_{(j,i)}}{\gamma_i}.\end{aligned}$$

Here,  $k_i$  refers to the total number of intrinsic tweets from user  $i$  and  $\gamma_i$  is the total number of interactions that user  $i$  participated in. Meanwhile  $\gamma_{(j,i)}$  breaks this up according to the interactions with the users that user  $i$  follows.

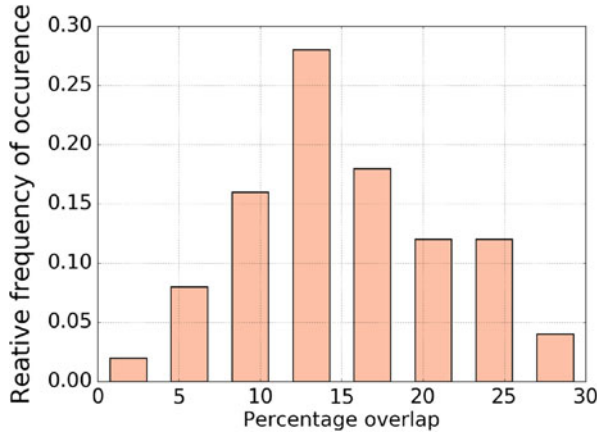
Note that the nature of interactions between two users can be highly complex and may be dependent on a host of features. However, in this work, we are not concerned with the complexities of the interactions. We simply compute the parameters based on counts of tweets and interactions in a *maximum likelihood manner* without regard to other features, such as topics and sentiment strength.

On the same Twitter graph, we compare the activations achieved by the IC model and the present model, incorporating intrinsic activation. We retain the interaction probabilities as the same between the two models while noting that any interaction probability  $w_{ij}$  becomes  $(1 - \alpha_i)w_{ij}$  for our model. We then randomized the intrinsic activation parameter  $\alpha$  for each of the users to observe if the influenced activations can match that of the IC model over a number of trials. Figure 5 illustrates the results from a Monte Carlo analysis with 50 trials. Here the influence spread curves corresponding to the Monte Carlo runs for the model with intrinsic activation (IC-Int) are all well below the influence spread curve for the IC model as expected. Furthermore, these results are in line with the observation that the authors make in [18] where they show that the IC model significantly overestimates the activations.

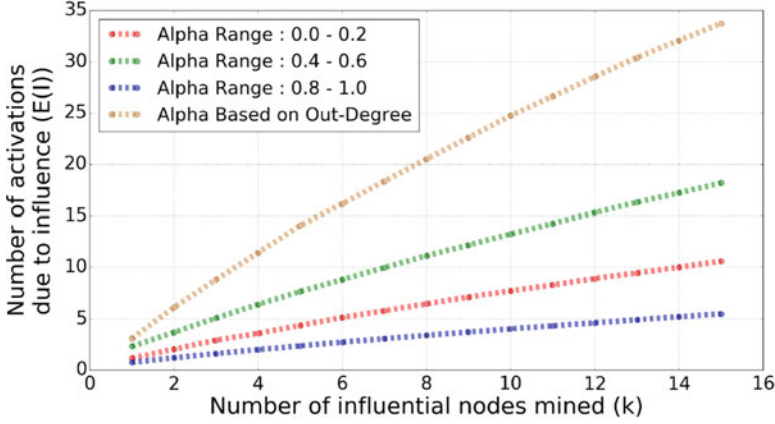


**Fig. 5** Comparing the influence spread with the IC model (Green) and the Monte Carlo runs on our model with intrinsic activation (Red). *Inset: The constructed Twitter graph is visualized in Gephi*

**Fig. 6** Distribution of the percentage overlap between the influencers identified by the IC model and the IC model with intrinsic activation in a Monte Carlo run



Next, for each of the Monte Carlo runs, we identify the percentage overlap between the sets of influential seeds identified by the IC model and our model with intrinsic activation. These results are shown in Fig. 6 for 50 Monte Carlo runs and for top 30 influential nodes. Note that while a small number of runs show nearly no overlap, more than 25% of the runs show 20% or more overlap. This is due to the fact that both the IC model and our model with intrinsic activation favor nodes with large out-degrees. However, the IC model with intrinsic activation also requires that such nodes have a high enough  $\alpha$  value to be influential along the lines of the discussions in Sect. 2.3.



**Fig. 7** The mean activation curves for three different ranges of  $\alpha$  assignments and the fourth with  $\alpha$  values proportional to the node out-degree

Continuing our discussions that began in Sect. 2.3 regarding maximizing the *engagement*, we consider four different cases of assigning  $\alpha$  values to the nodes. In the first three cases, the random  $\alpha$  values are drawn from three different intervals in a uniform manner. In the first case, all the  $\alpha$  values are drawn according to the distribution  $U[0, 0.2]$ . Meanwhile in the second case, they were drawn from  $U[0.4, 0.6]$  and finally in the third case, from the distribution  $U[0.8, 1.0]$ .

The mean number of influenced activations over three 30-run Monte Carlo analyses is plotted in Fig. 7. Note that each of the sub-problems corresponding to one realization of the  $\alpha$  values for all the nodes involved one run of Algorithm 1 with 1000 random samples ( $n$ ). For the fourth case, the  $\alpha$  values for the nodes are set deterministically in proportion to the node out-degree values. Clearly, it can be seen that the cases where all of the  $\alpha$  values are either all small or all large fall short of the number of influenced activations for the case corresponding to the middle range of  $\alpha$  values. The fourth case where the  $\alpha$  values are proportional to the out-degree far outperforms the rest. This observation is in line with the discussions presented in Sect. 2.3.

## 5 A Centrality Metric Incorporating Intrinsic Activation

In this final section, we examine the influencer mining on networks with intrinsic and influenced nodal activations from a slightly different perspective. By collecting the various probabilities together and recognizing the recursive nature of influence spread on a social network, we arrive at a generalized PageRank-type spectral influence measure that was first presented in [22]. As demonstrated in [22], when considering activity on an OSN, this approach is a better measure of influence spread than a purely topological metric such as PageRank.

For a given node  $i$ , from Fig. 1, the total probability of activation  $p_A^T(i)$  can be written as

$$p_A^T(i) = \alpha(i) + \left( 1 - \prod_{j, (j,i) \in \mathcal{E}} (1 - \beta(i)w_{ij}p_A^T(j)) \right). \quad (2)$$

where  $p_A^T(i)$  denotes the total probability of activation for node  $i$  (intrinsic and influenced). The parameter  $\alpha(i)$  denotes probability of node  $i$  getting activated intrinsically. The quantity  $\beta(i)w_{ij}$  denotes the probability of node  $j$  activating node  $i$  through influence as before.

Equation (2) summarizes the total activation probability for any given node in terms of the activation probabilities of the neighbors and whether the respective connecting edges are *live* or not. For the IC model, each activated node has a single shot probability of activating its neighbor and the activation along each edge is independent of the other edges. The second term in the large parentheses represents the probability of some neighboring node activating node  $i$  through influence. When we write out Eq. (2) for all the nodes, we will be dealing with a large system of coupled nonlinear equations, whose solution can be computationally expensive to obtain. By retaining the leading-order terms, we get a linearized version of Eq. (2). The goal of this approximation is to get the equation in a linear form so that we can use mature linear algebraic methods to quickly compute the set of influential nodes. This is valid to a large extent because, given the weighted-cascade version of the IC model that we are employing, the  $w_{ij}$  values will be small and we can neglect the higher-order terms to linearize Eq. (2) as follows.

$$p_A^T(i) = \alpha(i) + \beta(i) \sum_{j, (j,i) \in \mathcal{E}} w_{ij} p_A^T(j). \quad (3)$$

We set  $\beta(i) = (1 - \alpha(i))$  as explained in Sect. 2.1 and extend Eq. (3) to the entire network with  $N$  nodes to obtain a matrix-vector equation as follows:

$$\mathbf{p}_A^T = \boldsymbol{\alpha} \mathbf{1} + ((\mathbf{I} - \boldsymbol{\alpha}) \mathbf{W}) \mathbf{p}_A^T. \quad (4)$$

In Eq. (4),  $\mathbf{p}_A^T$  is a vector of size  $N \times 1$ , denoting respectively the total probability of activation for all the nodes on the network.  $\mathbf{I}$  denotes the identity matrix of size  $N \times N$ .  $\boldsymbol{\alpha}$  denotes the diagonal matrix with entries corresponding to the intrinsic activation probability for all the nodes on the network;  $\mathbf{W}$  denotes the sparse, stochastic weight matrix with entries given by the weights  $w_{ij}$  discussed earlier; and  $\mathbf{1}$  is the all-ones vector of size  $N \times 1$ .

We can then express the total activation probabilities as

$$\mathbf{p}_A^T = \mathbf{1}^T \mathbf{G}; \mathbf{G} = (\mathbf{I} - (\mathbf{I} - \boldsymbol{\alpha}) \mathbf{W})^{-1} \boldsymbol{\alpha}, \quad (5)$$



where  $\mathbf{1}^T$  provides for the column-sum of  $\mathbf{G}$ . We also note that because the matrix  $\mathbf{W}$  is a row stochastic matrix, the matrix  $\mathbf{G}$  is also row stochastic.

Consider the quantity  $C_A(i)$ , specific to node  $i$  as defined below.

$$C_A(i) = \left( \sum_{j=1, i \neq j}^N G_{ji} \right). \quad (6)$$

$C_A(i)$  corresponds to the sum of the entries in column  $i$  of  $\mathbf{G}$  with the exception of the corresponding diagonal term and represents the expected number of hosts activated by node  $i$  getting intrinsically activated and is a measure of influence. In our experiments with the PBlogs and the LFR-1000 graphs, discussed in Sect. 3.2, the  $\alpha$  and  $\mathbf{W}$  entries were randomized with entries drawn from the uniform distribution over  $[0, 1]$  and  $\mathbf{W}$  was converted to a row-stochastic matrix. We then compare the sets of top- $k$  influencers identified by both the methods on two larger graphs in our dataset. The comparison is carried out with respect to two measures: (1) Jaccard similarity and (2) Rank-Biased Overlap (RBO). RBO considers ordering with higher weights given to matches that happen at the top [25]. These results are presented in Table 1 where we see good agreement between the sets of influential nodes obtained by both methods.

The behavior of the Jaccard index is not necessarily monotone as a larger number of influencers are considered. As we move away from the top influencing nodes (increasing  $k$ ), we encounter many nodes that are of a similar influence. Since the centrality-based method is an approximation, the relative positions can change a lot and it is easily possible that going from  $k = 30$  to  $k = 50$ , we may not get a proportional increase in the overlap between the two sets. Hence, it can result in non-monotone behavior. The RBO-based comparison can also exhibit a similar non-monotone behavior. However, this measure is known to be stable because of the weighting by the rank. The same is observed in our experimental results as well.

Thus, the proposed centrality metric, which includes the intrinsic activation mechanism, represents a computationally more viable alternative to the full-scale influence maximization framework. It retains the essence of the model and the influential nodes can be mined by solving a linear system involving a sparse matrix.

**Table 1** Correlations, two ways, between the proposed approaches for the two inputs PBlogs and LFR1000 for different sizes of seed sets (10, 20, 30, and 50)

Correlation type	Input	$k = 10$	$k = 20$	$k = 30$	$k = 50$
Jaccard	PBlogs	0.538	0.818	0.875	0.818
RBO	PBlogs	0.817	0.846	0.851	0.868
Jaccard	LFR1000	0.818	0.905	0.765	0.818
RBO	LFR1000	0.979	0.963	0.947	0.937

Closer the metric to one, the better

## 6 Conclusions

In this work, we introduce the notion of nodes in a complex network getting activated by two mechanisms: intrinsically and through influence as commonly observed in online social networks. Using a modified version of the influence maximization algorithm and working with a suitable influence spread objective function, we show how it is possible to identify influential users whose intrinsic content spreads maximally through influence. We also sketch a short proof on the submodularity of the modified influence function, allowing for approximation guarantees on the algorithm. We utilized several synthetic and real-world datasets to examine various aspects of the proposed activation model. We also explain why some assignments of the intrinsic activation probability ( $\alpha$  values) to the various nodes can result in much higher activations than other assignments, which is also demonstrated on a Twitter graph. We finally derive a novel centrality metric from the activation model that can provide for a computationally faster and accurate method to identify influential users on a social network where activations can be intrinsic or influenced.

Building on this work, we are exploring multiple facets of this problem in our ongoing research including the exploration of how a social network can be successful in the long run by balancing the two modes of activation discussed here. To achieve this objective, we are considering development of variants of other interaction models with intrinsic activation as well extending the notion of intrinsic activation to more fine-grained user behavior. We are also extending these methods to other complex systems, such as for attack modeling in cyber networks [20].

**Acknowledgements** This research was supported in part by the High Performance Data Analytics Program (HPDA) and in part by the Control of Complex Systems Initiative (CCSI) at the Pacific Northwest National Laboratory (PNNL). HPDA is a collaboration led by Pacific Northwest National Laboratory (PNNL) with partners Mississippi State University, University of Washington, and Georgia Institute of Technology. CCSI is a Laboratory Directed Research and Development (LDRD) program at the PNNL. PNNL is operated by Battelle for the U.S. Department of Energy under Contract DE-AC05-76RL01830.

## References

1. Adamic, L.A., Glance, N.: The political blogosphere and the 2004 us election: divided they blog. In: *Proceedings of the 3rd International Workshop on Link Discovery*, pp. 36–43. ACM, New York (2005)
2. Bastian, M., Heymann, S., Jacomy, M.: Gephi: an open source software for exploring and manipulating networks (2009). [www.gephi.org](http://www.gephi.org)
3. Borgs, C., Brautbar, M., Chayes, J., Lucier, B.: Maximizing social influence in nearly optimal time. In: *SODA '14 Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 946–957. SIAM, Philadelphia (2014)

4. Chen, W., Collins, A., Cummings, R., Ke, T., Liu, Z., Rincon, D., Sun, X., Wang, Y., Wei, W., Yuan, Y.: Influence maximization in social networks when negative opinions may emerge and propagate. In: *SIAM Data Mining*, pp. 379–390 (2011)
5. Cohen, E., Delling, D., Pajor, T., Werneck, R.F.: Sketch-based influence maximization and computation: scaling up with guarantees. In: *CIKM '14 International Conference on Conference on Information and Knowledge Management*, pp. 629–638. ACM, New York (2014)
6. Farajtabar, M., Du, N., Gomez-Rodriguez, M., Valera, I., Zha, H., Song, L.: Shaping social activity by incentivizing users. In: *Advances in Neural Information Processing Systems*, pp. 2474–2482 (2014)
7. Friedkin, N.E., Johnsen, E.C.: Social influence networks and opinion change. *Adv. Group Process.* **16**(1), 1–29 (1999)
8. Gionis, A., Terzi, E., Tsaparas, P.: Opinion maximization in social networks. In: *SIAM Data Mining Conference*, pp. 387–395. SIAM, Philadelphia (2013)
9. Halappanavar, M., Sathanur, A., Nandi, A.: Accelerating the mining of influential nodes in complex networks through community detection. In: *CF'16 Proceedings of the 13th ACM International Conference on Computing Frontiers*, Como, May 16–18, 2016, pp. 64–71 (2016)
10. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: *Proceedings of ACM SIGKDD*, pp. 137–146. ACM, New York (2003). <https://doi.org/10.1145/956750.956769>
11. Kempe, D., Kleinberg, J., Tardos, É.: Influential nodes in a diffusion model for social networks. In: *Automata, Languages and Programming*, pp. 1127–1138. Springer, Berlin (2005)
12. Krause, A., Golovin, D.: Submodular function maximization. In: *Tractability: Practical Approaches to Hard Problems*, vol. 3(19), p. 8. Cambridge University Press, Cambridge (2012)
13. Lancichinetti, A., Fortunato, S.: Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys. Rev. E* **80**(1), 016118 (2009)
14. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: *Proceedings of ACM SIGKDD*, pp. 420–429. ACM, New York (2007)
15. Li, Y., Chen, W., Wang, Y., Zhang, Z.L.: Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships. In: *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, pp. 657–666. ACM, New York (2013)
16. Mirzasoleiman, B., Badanidiyuru, A., Karbasi, A., Vondrak, J., Krause, A.: Lazier than lazy greedy. In: *Twenty-Ninth AAAI Conference on Artificial Intelligence* (2015)
17. Myers, S.A., Zhu, C., Leskovec, J.: Information diffusion and external influence in networks. In: *ACM SIGKDD*, pp. 33–41. ACM, New York (2012)
18. Quach, T.T., Wendt, J.D.: A diffusion model for maximizing influence spread in large networks. In: *Social Informatics: 8th International Conference, SocInfo 2016, Bellevue, November 11–14, 2016. Proceedings, Part I*, pp. 110–124. Springer International Publishing, Berlin (2016)
19. Saito, K., Nakano, R., Kimura, M.: Prediction of information diffusion probabilities for independent cascade model. In: *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pp. 67–75. Springer, Berlin (2008)
20. Sathanur, A.V., Haglin, D.J.: A novel centrality measure for network-wide cyber vulnerability assessment. In: *IEEE Symposium on Technologies for Homeland Security (HST)*, 2016, pp. 1–5. IEEE, Washington (2016)
21. Sathanur, A.V., Halappanavar, M.: Influence maximization on complex networks with intrinsic nodal activation. In: *Social Informatics: 8th International Conference, SocInfo 2016, Bellevue, November 11–14, 2016. Proceedings, Part II*. Springer International Publishing, Berlin (2016)
22. Sathanur, A.V., Jandhyala, V., Xing, C.: Physense: scalable sociological interaction models for influence estimation on online social networks. In: *IEEE International Conference on Intelligence and Security Informatics*, pp. 358–363. IEEE, Washington (2013)

23. Srivastava, A., Chelmiss, C., Prasanna, V.K.: Influence in social networks: a unified model? In: 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 451–454. IEEE, Washington (2014)
24. The Twitter public API. <https://dev.twitter.com/rest/public>
25. Webber, W., Moffat, A., Zobel, J.: A similarity measure for indefinite rankings. *ACM Trans. Inf. Syst.* **28**(4), 20 (2010)
26. Zhang, H., Mishra, S., Thai, M.: Recent advances in information diffusion and influence maximization in complex social networks. In: *Opportunistic Mobile Social Networks*, p. 37. CRC Press, Boca Raton (2014)