Collective Influence Maximization in Mobile Social Networks

Xudong Wu, Luoyi Fu, Shuaiqi Wang, Bo Jiang, Xinbing Wang, and Guihai Chen

Abstract—The omnipresence of information cascading process in mobile social networking applications makes the identification of a small set S of influential users, which is widely believed to trigger the information outbreak, always an crucial issue in various applications such as the mobile advertising and viral marketing. Formulated as Influence maximization (IM) in 2003, this NP-hard problem has received a multitude of studies with diverse angles. However, these works often unable to provide reliable solutions, due to the loss of an exact metric for evaluating users' contributions on information cascading in the state-of-the-art sampling based IM schemes. In this paper, we evaluate users in IM based on the collective influence (CI), a metric on the structural features of the users in network graph that reflects the contributions of the users' neighborhoods on shaping collective dynamics of the users over the whole network. For conducting the influencer identification under probabilistic diffusion model based on the CI, we specify a quantified structural feature of the most influential users from the scope of diffusion over the whole network, and reveal that the structural influence power (CI value) of each user is a weighted cumulation of the diffusion probabilities from neighbors within certain hops. Utilizing CI, we design a novel algorithm which identifies the influencers via iteratively choosing the users with top CI values. Moreover, we point out that directly computing CI values requires to traverse the network which is originally represented by a high-dimensional matrix, and leads to huge complexity of influencer identification. To improve scalability, we further trade precision for efficiency by incorporating network embedding, a dimensionality reduction technology for networks, into algorithm design, and propose a minor variant, where CI is jointly recapitulated by low-dimensional user representations and user degrees. The superiority of our algorithms is empirically validated over 8 datasets, with an increment in influence size up to 50 percent and a comparable or even less running time comparing with existing baselines.

Index Terms—Mobile social network, influence maximization, collective influence, network embedding

1 Introduction

With the popularization of intelligent mobile devices, diverse social networking medias (e.g., Twitter, Wechat, Facebook, and TikTok) have developed their mobile apps for enabling the instant diffusion of informations and contents among users, and thus give rise to the mobile social networks [2], [3], [4], [5]. Due to the instant diffusions in mobile social networks, and the resulting occurrence of rapid global-scale explosions of some informations from a small set of seeds, the importance of viral spreading has been widely recognized [6], [7], [8]. The seeds, or in other words 'superspreaders', not only determine the information diffusion scale, but also define the collective dynamics of a large population [6]. Their identification is at the heart of an abundance of applications, such as mobile advertising [2], points-of-interest promotion [9], viral marketing [8], [10], behavior adoption [11], and innovation diffusion [12]. Due to the direct relevance of the identification of 'superspreaders' and a wide range of applications, the problem of locating 'superspreaders' in different settings is becoming increasingly important in recent years [13].

- *The authors are with Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: {xudongwu, yiluofu, wangshuaiqi, bjiang, xwang8}@sjtu.edu.cn,, gchen@cs.sjtu.edu.cn.*

Manuscript received 21 August 2020; revised 15 June 2021; accepted 21 June 2021. Date of publication 25 June 2021; date of current version 6 January 2023. (Corresponding author: Xinbing Wang.)
Digital Object Identifier no. 10.1109/TMC.2021.3092434

The search for 'superspreaders' was first formulated by Kempe *et al.* [14] in 2003 as *Influence Maximization* (IM) problem, an NP-hard problem whose essence is to pick a given number of most influential users to maximize the influence diffusion size. Since then, numerous approaches [15], [16], [17] have been conceived to approximate the optimal solution from diverse perspectives, and try to balance between the algorithm scalability and the performance of selected seed users. The common methods in current seed selection framework are designed to adopt a certain number of samples to estimate the expected influence diffusion size from users, say the influences of them, and iteratively select the user with the highest estimated influence among the non-seed users in a greedy manner [16], [17], [18]. Existing works claim that the above common method can achieve an approximation ratio of $(1-\frac{1}{e}-\varepsilon)$ with probability $(1-\delta)$ by adopting a sufficient size of samples [16], [17], [18]. However, not only is it difficult to determine the appropriate sample size over the diverse mobile social networks with heterogeneous sizes and topologies, but also adopting a large size of samples is computationally challenging over current mobile social networks with millions of users. Furthermore, without an accurate estimation of influences of users, the sampling based methods potentially miss some 'superspreaders', resulting in the loss of influence diffusion size.

As it is difficult to reliably estimate the influence of users, is there any other reliable metric to evaluate different users' contributions on enlarging cascading scale? To this end, we

note that the influences of users may correlate with the structural features of them in network. For example, a hub node of several communities, if getting influenced, potentially further influences a large scale of nodes. On the contrast, a leaf node in a tree-like network has much weaker power in the influence diffusion. With such insight, in this paper, we seek to evaluate the influences of users in network based on their structural features in network, and we adopt the metric named *Collective Influence* (CI) in the influence evaluation. The concept of collective influence, which is first proposed by Morone and Makse [19], measures the significance of each node or the power of each node's neighborhood on shaping collective dynamics of the nodes over the whole network, and is quantified by a cost function collectively possessed by the node and its neighbors within certain hops. CI has proved capable of capturing the power of each node on shaping collective dynamics among all the nodes from the structural features of them, and thus an effective and reliable indicator of node's strategic importance in a rising number of observations in diverse disciplines, including academic accomplishment evaluation [13], economic assessment [20], and biology [21].

Inspired by the CI, which is a novel influence metric in science community, we investigate the IM problem from the structural features of nodes characterized by their neighborhoods in mobile social network. Particularly, in the IM problem, the collective dynamic noted above refers to the state of getting influenced, and we consider the Independent Cascading (IC) diffusion model, where each newly influenced user has one single chance to activate his uninfluenced neighbors with certain probabilities [14], [16], [17], [22]. Different from [19] which solely focuses on the limit case to determine the required minimum seed fraction for a network-wide outbreak, we aim to design a more general algorithm which is applicable to any given seed portion q due to budget concerns. In seed selection, we measure user contribution to influence diffusion by CI value, a metric quantifying the collective diffusion power of the neighborhood centered around the user itself.

Different from traditional methods, the incorporation of CI brings a few new challenges. To conduct IM based on CI, one difficulty lies in exploring the exact formula of the CI value under the probabilistic IC diffusion model to enable the CI can reflect the contributions of users on cascading. Moreover, how to efficiently quantify CI and perform CI-based seed selection over today's enormous networks (million-scale or even larger) remains a challenging task.

To overcome the aforementioned challenges, we propose completed solutions that embraces comprehensiveness and scalability. By mapping the IM to optimal percolation, we present that the most influential users are those who, if seeded, can minimize the largest eigenvalue of a linear matrix that stores the weighted topological interaction among users. Derived from such feature, we elaborate the formula of such largest eigenvalue via the power of each non-seed user in influence diffusion which is expressed by the CI value of them that takes the form as a cumulation of the diffusion probabilities from them within a certain number of hops. Utilizing CI, maximizing the potential influenced size is most efficiently conducted by iteratively seeding a given proportion of users with the largest CI value. However, the computation of exact CI value leads to prohibitive costs of seed selection due to unavoidable traversals for the acquisition of global structural knowledge, especially under large hops. In an effort to enhance scalability while preserving as much the accuracy, we introduce a novel collective influence embedding method to depict each user's CI with a low-dimensional latent representation (a vector of d elements with $d \ll |V|$, $|V|$ is the network size). The key idea is to generate for each user the *collective influence context*, a set of reachable neighbors with distinct accumulated weights randomly chosen from different paths, and then to use those CI contexts as observations to output the low-dimensional representations for the computation of CI values via maximum likelihood estimation (MLE) that returns provable error convergence. The CI embedding thus facilitates the design of a scalable algorithm CIM-ESS that iteratively selects the user with the highest CI value generated by the prescribed low-dimensional representations.

While unfolding the details of our solution in later sections, we summarize below its key contributions:

1) We are the first to study the collective influence based IM under IC diffusion model, and for the first time propose complete solution. Via optimal percolation, we quantify the features for the "superspreaders" from a full impact of seed portion q and the formula of CI value that serves as a reliable metric to evaluate users' contributions on enlarging cascading scale. We reveal that under IC model, the CI value of each user takes the form as a cumulation of the diffusion probabilities from the user within a certain number of hops.
2) To overcome the high complexity in computing CI values over the network originally represented by a high-dimensional matrix, we further propose the CIM-ESS algorithm, which incorporates network embedding technology into the CI based seed selection. The key is to reinterpret CI, originally composed by edge weights and user degrees, with low-dimensional latent representations of users.
3) We perform extensive experiments on 8 social network datasets, the biggest of which contains 4.8 million nodes. Even by only acknowledging the structural knowledge of immediate neighbors, CI-based seed selection apparently outperforms competing methods in terms of the influence diffusion size. Remarkably, CIM-ESS always achieves the top influence diffusion size in a near linear running time thanks to a parallel implementation enabled by the low-dimensional representations.

We organize the rest of this paper as follows: We review the background and the related works on IM in Section 2. The problem formulation is given in Section 3. In Section 4, we explore the formula of the CI in the IM problem under IC model, and give the CI based seed selection algorithm. The collective influence embedding scheme is presented in Section 5 which is followed by the experimental results in Section 6. We conclude the paper in Section 7.

2 Background and Related Works

In this section, we review the background of the Influence Maximization (IM) problem and present the main idea as well as the limitations of existing solutions.

Given a mobile social network, the fundamental problem for IM is to select the optimal seed users under the given budget who can maximize the diffusion size. The budget constrains the size of selected seeds. Kempe *et al.* [14] first formalize the seed selection as a combinatorial optimization problem. Specifically, they propose to firstly estimate the final influenced size starting from a set of seed users, and then select the optimal ones with the maximum estimated influence. They further prove that the IM problem is NP-hard. Most early works on IM (e.g., [14], [15]) estimate the influenced size through generating certain times of Monte-Carlo simulations of the IC diffusion model and taking the average. Then a greedy algorithm is adopted to select the users with the maximum expected influence. This method costs an $\Omega(K|V||E| \cdot \mathrm{poly}(\varepsilon^{-1}))$ complexity and achieves an approximation ratio of $(1-\frac{1}{e}-\varepsilon)$ [15] on the influenced size with probability $(1-\delta)$. One of the most efficient IM frameworks currently is the Reverse Reachable Sets (RR-sets) [16], [17], [22], which costs a time of $O(K(|V|+|E|)\log n/\varepsilon^2)$ for achieving a $(1-\frac{1}{e}-\varepsilon)$-approximate solution with probability $(1-\delta)$ [17], [22]. The RR-set based methodology contains two major steps: it first samples a number of nodes, and then selects seeds who can expectedly influence the most number of sampled nodes. Besides, some works [23], [24] which seek for high efficiency of seed selection estimate the influenced size of seed nodes based on the upper bounds derived from eigen analysis.

Furthermore, there also emerge some other variants of the IM problem. For instance, Nguyen *et al.* [16] consider maximizing the outward influence which refers to the influenced size minus seed size. The heterogeneous seeding costs of different users are studied in [25]. [26] focus on adaptively selecting seeds based on the diffusion feedbacks from previous seeds, and [27] study the multi-round seed selection to maximize the influenced size in a long run.

However, current sampling-based solutions usually miss some "superspreaders" due to the variance in sampling, and thus result in the loss of influence diffusion size. Recently, Morone *et al.* [19], [28], [29] pioneer the concept of collective influence, which regards one user's influence on diffusion by the structural features of them, to find the minimal set of influential users who can, if influenced, spread influence to the whole network or, if immunized, prevent an outbreak of influence. In the diffusion model in [19], a node u_i will get influenced only if his (d_i-1) neighbors get influenced, where d_i is the degree of u_i. Besides, there are many other works focusing on the effects of social network structure in diverse disciplines. For example, [30], [31] reveal the effects of social network structure on the collective action and the cooperation in large population. Centola *et al.* [32] explain the existence of cultural diversity from the structural perspective, and Sohn *et al.* [33] study how the social network structure affects the opinion distribution among users. Inspired by such works, we reinvestigate the IM problem to select the most influential seeds under a new look at users' influences from the novel structural perspective.

3 Problem Formulation

We model a mobile social network as a graph $G=(V,E)$, where V $(|V|=N)$ is the set of users and E is the set of directed edges that represent the social links among users. The topology of G is represented by an $N \times N$ matrix A, where $A_{ij}=1$ if there is an edge (i,j) from user u_i to u_j and $A_{ij}=0$ otherwise. We say u_j is a neighbor of u_i if $A_{ij}=1$. $\Gamma(i)$ denotes the set of all the neighbors of u_i. Each edge $(i,j) \in E$ is associated with a weight w_{ij} which indicates the probability that u_i can successfully influence u_j. Moreover, we adopt the Independent Cascading (IC) model [14], [16], [17], [22] to characterize the influence diffusion process started from seed users.

Definition 1. (Independent Cascading (IC) model.) *Initially, a set of seed users gets influenced and other users are uninfluenced. Such seed users then start the influence diffusion process in discrete steps. In each step, when a user u_i newly gets influenced, he then has a single chance to influence his each uninfluenced immediate neighbor u_j successfully with probability w_{ij} in the next step. Once a user gets influenced, he will remain influenced until the end. The influence diffusion process stops when there is no new user gets influenced. Then, the size of the influenced users at the end is called as the size of users that finally influenced by the seeds, as well as the influence of seeds.*

In reality, the influencing probability w_{ij} can be estimated from the interaction frequency or action logs of mobile social network applications [16]. When an uninfluenced user see a neighbor forwarding an information, he will possibly further forward such information and get influenced. Since users in mobile social network usually do not re-browse the informations they had browsed before, the IC model sets that each newly influenced user has a single chance to activate his uninfluenced neighbors. With the network and diffusion models, we formulate the influence maximization (IM) problem studied in this paper as follows.

Problem Formulation. Given a network $G=(V,E)$ and the seed fraction q, let $Q(S,q)$ denote the expected size of users that finally influenced by a set S of qN seed users under the IC model, the objective of the influence maximization (IM) problem is identifying the seed set S to maximize $Q(S,q)$.

Kempe *et al.* proved in [14] that the IM problem given above is NP-hard. The common solution in the existing studies on the IM problem (e.g., Monte-Carlo simulation [14] and Reverse Reachable Sets [22]) evaluates the contributions of users on enlarging the influence diffusion size via sampling, and iteratively selecting the users who can maximize the influenced size in a greedy manner. However, as we noted before, over social networks with diverse sizes, topologies and edge weights, not only is it difficult to determine the appropriate number of required samples for guaranteeing the accuracy of influence estimation, but also adopting excessive samples is computationally challenging over current networks with millions of users.

To tackle this dilemma in the existing solutions on IM, in this paper, we evaluate the influences of users relying on the structural features of them instead of via sampling. The power of each node on enlarging influence diffusion size derived from structural feature is called as the *Collective Influence (CI)* of the node. The concept of CI, which is first proposed by Morone *et al.* in [19], generally specifies the power of each node's neighborhood on shaping collective dynamics of the nodes over the whole network via the paths in network, and it has different formulas in different applications. In the IM

problem, the collective dynamic refers to the the state of getting influenced.

Collective Influence (CI). The collective influence (CI) is the metric we adopt in this paper to evaluate the contributions of users on maximizing the influence diffusion size under IC model from their structural features in network.

Challenges. The first challenge we meet lies on exploring the formula of CI in the IM problem under IC diffusion model. Furthermore, when we algorithmically select seed users for maximizing the influenced size $Q(S,q)$, another challenge is how to efficiently compute the CI values of users over large scale networks. Next, we move to address the first challenge.

4 Collective Influence Based Seed Selection

In order to evaluate the contributions of users on cascading based on the structural features of them, we intend to characterize the influence diffusion process under IC model via the network structure and capture the structural features of the "superspreaders" we need to identify (in Section 4.1). Such features then provide the basis for exploring the formula of the collective influence of users in the IM problem under IC model (in Section 4.2).

4.1 Quantifying Features For Superspreaders

For capturing the features of "superspreaders", aka most influential users, we borrow the ideas of solutions in [19], which maps finding the minimum required fraction of seeds for a network-wide outbreak to the optimal percolation.

4.1.1 Mapping IM to the Optimal Percolation

Definition 2. (Optimal percolation.) *The optimal percolation is the problem of identifying the minimal set of core nodes which, if removed, can minimize the size of the giant connected component in a given network.*

As presented in Definition 2, the optimal percolation identifies the core nodes through minimizing the size of the giant connected component in a given network. Similarly, under the IC model, maximizing the expected influenced size $Q(S,q)$ is equivalent to maximizing the probabilities of all the users being influenced, as well as minimizing the probabilities of not being influenced. In the following, we present that, since sharing similar objectives, the IM problem can also be mapped onto the optimal percolation.

On one hand, in the optimal percolation, [19] uses $v_{i\to j}$ to denote whether u_i belongs to the giant connected component when his one neighbor u_j is disconnected from the network, and uses $n_i = 0$ (resp. $n_i = 1$) to indicate u_i is removed (resp. is present in network). Then, the node u_i belongs to the giant connected component under the condition that at least one of its neighbor nodes except u_j belongs to the giant connected component. With this condition, over the locally-tree like network, $v_{i\to j}$ can be given by $v_{i\to j} \approx n_i\big(1 - \Pi_{m\in\Gamma(i)\setminus j}(1 - v_{m\to i})\big)$. Moreover, the objective of optimal percolation that minimizing the size of the giant connect component is equivalent to minimizing the sum $\sum_{(i,j)\in E} v_{i\to j}$ [19].

On the other hand, for the influence maximization under IC model studied in this paper, we use $I_i = 1$ (resp. $I_i = 0$) to indicate u_i is a seed (resp. is not a seed), and use $X_i^t = 1$ (resp. $X_i^t = 0$) to represent that u_i is influenced (resp. is uninfluenced) until the end of the time step t during the diffusion under IC model. Moreover, the random variable W_{ki}^t denotes whether the user u_k influences his neighbor u_i at time step t successfully ($W_{ki}^t = 1$) or not ($W_{ki}^t = 0$). Then, for the tth step ($t \geq 1$) during the diffusion under IC model, we have the following lemma.

Lemma 1. *Let x_i^t (resp. $\overline{x_i^t}$) denote the event $X_i^t = 1$ (resp. $X_i^t = 0$), and let $\Pr\Big(x_i^t|\overline{x_i^{t-1}}|\overline{x_j^t}\Big)$ denote the probability that a non-seed user u_i gets influenced in the tth step under the condition that his one neighbor u_j has not been influenced, we have*

$$\Pr\Big(x_i^t|\overline{x_i^{t-1}}, \overline{x_j^t}\Big) = (1 - I_i)\mathbb{E}\left[1 - \prod_{u_k\in\Gamma(i)\setminus u_j}\Big(1 - X_k^{t-1}\overline{X_k^{t-2}}W_{ki}^t\Big)|\overline{x_i^{t-1}}, \overline{x_j^t}\right]. \tag{1}$$

The proof of Lemma 1 is in Appendix A in the supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/TMC.2021.3092434. Recall the IC diffusion model in Definition 1, once a user gets influenced, he can try to influence his uninfluenced neighbors in the next step. Then, the intuition of Eq. (1) is that, under the IC model, when $I_i = 0$ (u_i is not a seed) and u_j is uninfluenced, u_i gets influenced at step t in the condition that at least one of its neighbor nodes except u_j is influenced at the $(t-1)$th step but is not influenced by u_i, and such neighbor node then successfully influences u_i. Furthermore, as the objective of IM is maximizing the size of users that expectedly get influenced during all the diffusion steps under IC model, maximizing the expected influenced size $Q(S,q)$ is approximately equivalent to maximizing the sum of the probabilities $\sum_t\sum_{(i,j)\in E}\Pr\Big(x_i^t|\overline{x_i^{t-1}}, \overline{x_j^t}\Big)$ (The proof is in Appendix A, available online in the supplementary material). Then, since IM has similar objective with the optimal percolation (i.e., minimizing $\sum_{(i,j)\in E} v_{i\to j}$ and maximizing $\sum_t\sum_{(i,j)\in E}\Pr\Big(x_i^t|\overline{x_i^{t-1}}, \overline{x_j^t}\Big)$ respectively) and $\Pr\Big(x_i^t|\overline{x_i^{t-1}}, \overline{x_j^t}\Big)$ has analogous formula with $v_{i\to j}$, we can conclude that the IM problem under IC model can be mapped to the optimal percolation.

With this mapping, we then quantify the features for the most influential users in IM borrowing the solution for the optimal percolation. [19], [21] present that the size of the giant connected component in a given network decreases with the decrease of the leading eigenvalue of the *non-backtracking matrix*, which is widely used for analyzing structural characteristics of networks [34] and will be introduced later. Thus, the solution of the optimal percolation is identifying the nodes who, if removed, can minimize the leading eigenvalue of the non-backtracking matrix. The elements in the non-backtracking matrix are defined over the non-backtracking paths of length 2 in network.

Definition 3. (Non-backtracking path.) *Let (k,h) denote a directed edge from node u_k to node u_h. For two edges (k,h) and (i,j), if $i = h$ and $j \neq k$, then $(k,h) \to (i,j)$ forms a non-*

backtracking path of length 2 in network. Furthermore, we say $(u_1, u_2) \to (u_2, u_3) \to \ldots \to (u_l, u_{l+1})$ *forms a non-backtracking path of length* l*, if* $u_{i-1} \neq u_{i+1}, \forall i \in [l]$.

Corresponding to the influence diffusion, the non-backtracking path specifies that each newly influenced user does not return to influence the user that influenced him before. Recalling the IC model in Definition 1, if u_i gets influenced from u_k, then u_i could further influence his uninfluenced neighbors but does not return to influence u_k. That is, under the IC model, the influence actually diffuses via non-backtracking paths in the network. Such characteristic further justifies our usage of the non-backtracking paths to characterize the influence diffusion. Furthermore, the non-backtracking matrix $\mathbf{M}$ has a dimension of $|E| \times |E|$. Each element in $\mathbf{M}$ is defined over a pair of edges in the set E. Specifically, for an element $\mathbf{M}_{khij}$, if the two edges $(k,h) \to (i,j)$ form a non-backtracking path, $\mathbf{M}_{khij}$ quantifies the evolution of the collective dynamic over the corresponding non-backtracking path $(k,h) \to (i,j)$. For example, in the optimal percolation, $\mathbf{M}_{khij} = \frac{\partial v_{i\to j}}{\partial v_{k\to h}}$.

Next, we first provide the formula of the non-backtracking matrix $\mathbf{M}$ in the IM problem under IC model, and then quantify the features for the most influential users in IM based on the non-backtracking matrix $\mathbf{M}$.

4.1.2 Formula of the Non-Backtracking Matrix $\mathbf{M}$ in IM

As the required collective dynamic in IM is the state of being influenced, we let each element $\mathbf{M}_{khij}$ in $\mathbf{M}$ characterize the evolution of the probability of user $u_i = u_h$ getting influenced via the non-backtracking path $(k,h) \to (i,j)$ during diffusion. Specifically, recalling the formula of the influencing probability $\Pr\left(x_i^t | \overline{x_i^{t-1}}, \overline{x_j^t}\right)$ in Eq. (1), since under the IC model, as $\mathbb{E}\left(W_{ki}^t\right) = w_{ki} \ll 1$ is assumed in most IM works [16], [22], we omit the high order components in the term $\mathbb{E}\Big[1 - \prod_{u_k \in \Gamma(i)\backslash u_j} \left(1 - X_k^{t-1}\overline{X_k^{t-2}}W_{ki}^t\right) | \overline{x_i^{t-1}}, \overline{x_j^t}\Big]$ and approximate Eq. (1) as

$$\begin{aligned} &\Pr\left(x_i^t | \overline{x_i^{t-1}}, \overline{x_j^t}\right) \\ &\approx (1 - I_i) \sum_{u_k \in \Gamma(i)\backslash u_j} \mathbb{E}\left[X_k^{t-1}\overline{X_k^{t-2}}W_{ki}^t | \overline{x_i^{t-1}}, \overline{x_j^t}\right] \\ &= (1 - I_i) \sum_{u_k \in \Gamma(i)\backslash u_j} \Pr\left(x_k^{t-1}\overline{x_k^{t-2}} | \overline{x_i^{t-1}}, \overline{x_j^t}\right) w_{ki}. \end{aligned} \tag{2}$$

Let v_{ij}^t serve as the abbreviation of $\Pr\left(x_i^t | \overline{x_i^{t-1}}, \overline{x_j^t}\right)$ and let v_{kij}^{t-1} serve as the abbreviation of $\Pr\left(x_k^{t-1}\overline{x_k^{t-2}} | \overline{x_i^{t-1}}, \overline{x_j^t}\right)$, we rewrite Eq. (2) into

$$v_{ij}^t = \sum_{u_k \in \Gamma(i)\backslash u_j} (1 - I_i) v_{kij}^{t-1} w_{ki} \cdot A_{ij}A_{ki}(1 - \mathbb{1}_{\{k=j\}}). \tag{3}$$

Here, $A_{ij}A_{ki}$ controls the existence of the two edges (i,j) and (k,i), and $(1 - \mathbb{1}_{\{k=j\}})$ controls that $(k,i) \to (i,j)$ forms a non-backtracking path. As the values of v_{ij}^t and v_{kij}^{t-1} in Eq. (3) both evolve with the seed set S, we take the value of v_{ij}^t as the function of $v_{kij}^{t-1}(\forall u_k \in \Gamma(i)\backslash u_j)$.

With above diffusion process via non-backtracking paths, we give below the formula of the non-backtracking matrix $\mathbf{M}$, whose elements characterize the influence diffusion under IC model over the non-backtracking paths of length 2 in network.

Definition 4. (Formula of Non-backtracking matrix $\mathbf{M}$ in IM.) *For each element* $\mathbf{M}_{khij}$*, if* $(k,h) \to (i,j)$ *forms a non-backtracking path, then* $\mathbf{M}_{khij} = \frac{\partial v_{ij}^t}{\partial v_{khj}^{t-1}}$ *characterizes the evolution of the influencing probability* v_{ij} *via the non-backtracking path* $(k,h) \to (i,j)$ *during the diffusion under IC model. With* v_{ij}^t *in Eq. (3), we have*

$$\begin{aligned} \mathbf{M} &= \begin{pmatrix} \mathbf{M}_{khij} & \cdots \\ \vdots & \ddots \end{pmatrix}, \mathbf{M}_{khij} \\ &= (1 - I_i) w_{ki} A_{ij} A_{kh} \mathbb{1}_{\{i=h\}} (1 - \mathbb{1}_{\{k=j\}}), \end{aligned} \tag{4}$$

where we multiply the factor $\mathbb{1}_{\{i=h\}}$ *due to* $u_i = u_h$ *in non-backtracking paths. Otherwise, if* $(k,h) \to (i,j)$ *do not form a non-backtracking path, then* $\mathbf{M}_{khij} = 0$.

In the Definition 4 above, we use the partial derivative $\mathbf{M}_{khij} = \frac{\partial v_{ij}^t}{\partial v_{kij}^{t-1}}$ to quantify that, when the conditional probability v_{kij}^{t-1} of u_k getting influenced first varies with the seed set S, how v_{kij}^{t-1} further varies the conditional probability v_{ij}^t of u_i getting influenced. That is, $\mathbf{M}$ controls the diffusion via the non-backtracking paths of length 2 after seeding a certain number of users. Moreover, for the powers of $\mathbf{M}$, say $\mathbf{M}^l$, taking the second power as an example, we have $\mathbf{M}_{khxy}^2 = \sum_{ij} \mathbf{M}_{khij}\mathbf{M}_{ijxy} = \sum_{ij} \frac{\partial v_{ij}^t}{\partial v_{kij}^{t-1}} \frac{\partial v_{xy}^{t+1}}{\partial v_{ijy}^t}$, which characterizes the evolution of the influencing probability v_{xy} via the non-backtracking path $(k,h) \to (i,j) \to (x,y)$ in two-hop diffusion. That is, the powers of $\mathbf{M}$ controls the multi-hop diffusion via the non-backtracking paths.

Further, referring to the optimal percolation which finds the core nodes by minimizing the leading eigenvalue of the non-backtracking matrix [19], we also characterize the feature of the most influential users based on the leading eigenvalue of $\mathbf{M}$. From the formula of $\mathbf{M}$, we can also see that if a user u_i is selected as a seed, all the elements related to $(1 - I_i)$ in $\mathbf{M}$ then become zero. Thus seeding a set of users results in the decrease of the leading eigenvalue of $\mathbf{M}$. In this paper, we use $\lambda(S,q)$ to denote the leading value of $\mathbf{M}$ after seeding a set S of $q|V|$ users.

4.1.3 Feature of the Most Influential Users in IM

With the formula of the non-backtracking matrix $\mathbf{M}$ in IM given in Definition 4, here, we will show that the most influential $q|V|$ users in IM are those who, if seeded, can minimize the leading eigenvalue $\lambda(S,q)$ of $\mathbf{M}$.

Concretely, as illustrated before, the objective of the IM problem is maximizing the expected influenced size $Q(S,q)$ and is approximately equivalent to maximizing the value of $\sum_t \sum_{(i,j)\in E} v_{ij}^t$. Let $\mathbf{Z}(S_1, q_1) = [\ldots, v_{ij}, \ldots]^T$ denote the initial values of $v_{ij}(\forall (i,j) \in E)$ after seeding a set S_1 of users with the size of $q_1 N$. Then, if seeding one more user $u_m \notin S_1$, the values of $v_{mn}(\forall u_n \in \Gamma(m))$ all become 1. We take such variations of the values of $v_{mn}(\forall u_n \in \Gamma(m))$ after seeding $u_m \notin S_1$ as a noise ϵ added to $\mathbf{Z}(S_1, q_1)$.

Moreover, as the power $\mathbf{M}^l$ of $\mathbf{M}$ characterizes the evolution of $v_{ij}(\forall (i,j) \in E)$ during the diffusion over network,

the increment of the expected diffusion size after seeding u_m can be quantified by a polynomial of $\mathbf{M}^l\epsilon$. Thus, we have $Q(S_1 \cup u_m, q_1 + \frac{1}{|V|}) - Q(S_1, q_1) \propto \mathbf{M}^l\epsilon$. Then, since the inner product of the increment $\mathbf{M}^l\epsilon$ can be approximated by $\epsilon^T(\mathbf{M}^l)^T\mathbf{M}^l\epsilon \approx \lambda(S_1, q_1)^{2l}\epsilon^T\epsilon$, there is

$$Q(S_1 \cup u_m, q_1 + \frac{1}{|V|}) - Q(S_1, q_1) \propto \lambda(S_1, q_1)^l.$$

Furthermore, we prove in Appendix A, available online in the supplementary material that, for two seed sets $S_1 \subseteq S_2$, if $u_m \notin S_1 \cup S_2$, then $Q(S_1, q_1) \leq Q(S_2, q_2)$ and $Q(S_1 \cup u_m, q_1 + \frac{1}{|V|}) - Q(S_1, q_1) \geq Q(S_2 \cup u_m, q_2 + \frac{1}{|V|}) - Q(S_2, q_2)$. Then, since $Q(S_1 \cup u_m, q_1 + \frac{1}{|V|}) - Q(S_1, q_1) \propto \lambda(S_1, q_1)^l$, we have $\lambda(S_1, q_1)^l \geq \lambda(S_2, q_2)^l$ and $\lambda(S_1, q_1) \geq \lambda(S_2, q_2)$. Thus, a larger expected influence diffusion size emerges with a smaller leading eigenvalue, and we can conclude that maximizing $Q(S, q)$ is equivalent to minimizing the leading eigenvalue $\lambda(S, q)$ of the non-backtracking matrix $\mathbf{M}$.

Feature of the most influential users. Based on the analysis above, we conclude that, in the IM problem, the most influential $q|V|$ seed users who can maximize the influenced size $Q(S, q)$ are the users who, if seeded, can minimize the leading eigenvalue $\lambda(S, q)$ of the non-backtracking matrix $\mathbf{M}$. Such feature enables us to evaluate the contributions of users on maximizing the influence diffusion size by their contributions on minimizing the leading eigenvalue.

4.2 Formula of Collective Influence (CI) in IM Problem

As we can evaluate users' contributions on cascading from their contributions to the leading eigenvalue of $\mathbf{M}$, in the following, we first give the formula of the leading eigenvalue and then, based on which, quantify the formula of collective influences (CIs) of users.

4.2.1 Formula of the Leading Eigenvalue

We compute the leading eigenvalue of $\mathbf{M}$ based on the power method. Let $\mathbf{B}_0 = \mathbf{1}$ denote an initial $|E|$-dimensional vector, and let l denote the order of the power. Referring [19] which computes the leading eigenvalue as the starting point of a series that serves as exact solution for optimal percolation, we have

$$\lambda \approx \lim_{l\to\infty}\left[\frac{\mathbf{B}_0^T\mathbf{M}^{2l}\mathbf{B}_0}{\mathbf{B}_0^T\mathbf{B}_0}\right]^{\frac{1}{2l}} = \lim_{l\to\infty}\left[\frac{(\mathbf{B}_0^T\mathbf{M}^l)(\mathbf{M}^l\mathbf{B}_0)}{\mathbf{B}_0^T\mathbf{B}_0}\right]^{\frac{1}{2l}}. \tag{5}$$

$(\mathbf{B}_0^T\mathbf{M}^l)$ and $(\mathbf{M}^l\mathbf{B}_0)$ are $|E|$-dimension vectors, and their elements are indexed by the sequences $\{kh\}, \forall(k,h) \in E$.

Since $\mathbf{B}_0^T\mathbf{B}_0$ is a constant that equals $|E|$, we then transfer quantifying users' contributions to the leading eigenvalue to quantifying their contributions to the value of $\mathbf{B}_0^T(\mathbf{M})^{2l}\mathbf{B}_0$. Consequently, we can quantify the CI value of each user as his contribution to the value of $\mathbf{B}_0^T(\mathbf{M})^{2l}\mathbf{B}_0$.

Further, we uncover in Lemma 2 that each user's contribution to the value of $\mathbf{B}_0^T(\mathbf{M})^{2l}\mathbf{B}_0$ is structurally determined by a *Ball* of radius l centered at this user. An example of the ball is given in Fig. 1. $Ball(k, l)$ denotes the ball of radius l centered at node u_k, and consists of the nodes that u_k can reach through a non-backtracking path with the length not larger than l. In addition, $\partial Ball(k, l)$ is the frontier of the ball $Ball(k, l)$ and consists of the nodes that u_k can reach via a non-backtracking path of length l. We use $p(k, i)$ to denote the non-backtracking path from u_k to u_i.

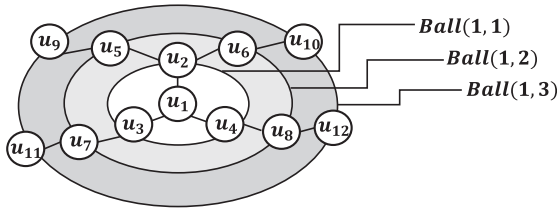


Fig. 1. An example of ball. $Ball(1,1), Ball(1,2), Ball(1,3)$ respectively denote the set of nodes that u_1 can reach with in $1, 2, 3$ hops. On the frontiers, $\partial Ball(1,1) = \{u_2, u_3, u_4\}$, $\partial Ball(1,2) = \{u_5, u_6, u_7, u_8\}$ and $\partial Ball(1,3) = \{u_9, u_{10}, u_{11}, u_{12}\}$. The path from u_1 to u_9 is $p(1,9) = \{(1,2), (2,5), (5,9)\}$, where we use $(1,2)$ to denote the edge from u_1 to u_2 and so on.

Lemma 2. *The inner product* $\mathbf{B}_0^T\mathbf{M}^{2l}\mathbf{B}_0$ *is equal to*

$$\mathbf{B}_0^T\mathbf{M}^{2l}\mathbf{B}_0 = \sum_{u_k\in V}\overline{I}_k \sum_{u_h\in\partial Ball(k,l)}\left(\Pi_{(x,y)\in p(k,h)}w_{xy}\overline{I}_x\overline{I}_y\right)(d_h-1)\overline{I}_h,$$

given the initial vector $\mathbf{B}_0 = \mathbf{1}$. *Here,* $\overline{I}_x$ *is the abbreviation of* $(1 - I_x)$ *and* d_h *denotes the degree of user* u_h.

Proof. We prove Lemma 2 by the induction of the formulas of the right and left vectors in $\mathbf{B}_0^T\mathbf{M}^{2l}\mathbf{B}_0$, i.e., $\mathbf{B}_0^T\mathbf{M}^l$ and $\mathbf{M}^l\mathbf{B}_0$. When $l = 1$, the first order right vector is $\mathbf{M}\mathbf{B}_0$. Taking the formula of $\mathbf{M}$ in Eq. (4) into $\mathbf{M}^l\mathbf{B}_0$, the khth element in $\mathbf{M}^l\mathbf{B}_0$ is equal to

$$\begin{aligned}\mathbf{M}\mathbf{B}_0|_{kh} &= \sum_{ij}\mathbf{M}_{khij} \qquad (\mathbf{B}_0 = \mathbf{1})\\ &= \sum_{ij}(1-I_i)A_{ij}A_{kh}\mathbb{1}_{ih}(1-\mathbb{1}_{kj})w_{ki}\\ &= \sum_{hj}(1-I_h)A_{hj}A_{kh}(1-\mathbb{1}_{kj})w_{kh}\\ &= A_{kh}w_{kh}(1-I_h)\sum_{j,j\neq k}A_{hj}\\ &= A_{kh}w_{kh}(1-I_h)(d_h-1).\end{aligned} \tag{6}$$

Similarly, for the first order left vector $\mathbf{B}_0^T\mathbf{M}$, we have

$$\begin{aligned}\mathbf{B}_0^T\mathbf{M}|_{kh} = \sum_{ij}\mathbf{M}_{ijkh} &= \sum_{ij}(1-I_k)A_{ij}A_{kh}\mathbb{1}_{jk}(1-\mathbb{1}_{ih})w_{ik}\\ &= (1-I_k)A_{kh}\sum_{i,i\neq h}A_{ik}w_{ik}.\end{aligned} \tag{7}$$

Combining Eqns. (6) and (7), we have

$$\mathbf{B}_0^T\mathbf{M}^2\mathbf{B}_0 = \sum_{kh}\overline{I}_i\overline{I}_k\overline{I}_h\left(\sum_{ik,i\neq h}A_{ik}w_{ik}\right)A_{kh}w_{kh}(d_h-1),$$

where $\overline{I}_i = (1 - I_i)$.

Furthermore, assume the lth order left vector has the form

$$\mathbf{B}_0^T\mathbf{M}^l|_{kh} = A_{kh}\overline{I}_k\overline{I}_h \sum_{i:d(i,k)=l, h\notin p(i,k)} \left(\Pi_{(x,y)\in p(i,k)} w_{xy}\overline{I}_x\overline{I}_y\right), \tag{8}$$

where $p(i,k)$ is the non-backtracking path with length l between u_i and u_k. Then, for the $(l+1)$th left vector, where $\mathbf{B}_0^T\mathbf{M}^{l+1} = (\mathbf{B}_0^T\mathbf{M}^l)\mathbf{M}$, we have

$$\begin{aligned}
&\mathbf{B}_0^T\mathbf{M}^{l+1}|_{kh} = \sum_{ab}(\mathbf{B}_0^T\mathbf{M}^l)|_{ab}\mathbf{M}_{abkh}\\
&= \sum_{ab} A_{ab}\overline{I}_a\overline{I}_b \left(\sum_{i:d(i,a)=l, b\notin p(i,a)} \left(\Pi_{(x,y)\in p(i,a)} w_{xy}\overline{I}_x\overline{I}_y\right)\right)\cdot\\
&(1-I_k)w_{ab}A_{kh}A_{ab}\mathbb{1}_{\{b=k\}}(1-\mathbb{1}_{\{a=h\}})\\
&= A_{kh}\overline{I}_k\overline{I}_h \sum_{i:d(i,k)=l+1, h\notin p(i,k)} \left(\Pi_{(x,y)\in p(i,k)} w_{xy}\overline{I}_x\overline{I}_y\right),
\end{aligned}$$

which shows that the formula of the left vector given in Eq. (8) also holds in the case of $(l+1)$.

Accordingly, the formula of the lth order right vector is

$$\begin{aligned}
&\mathbf{M}^l\mathbf{B}_0\\
&= \overline{I}_k\overline{I}_h A_{kh}w_{kh} \sum_{i:d(h,i)=l, k\notin p(h,i)} \left(\Pi_{(x,y)\in p(h,i)} w_{xy}\overline{I}_x\overline{I}_y\right)(d_i-1).
\end{aligned}$$

Together with the formulas of the elements in $\mathbf{B}_0^T\mathbf{M}^l$ and $\mathbf{M}^l\mathbf{B}_0$, we obtain the formula of $\mathbf{B}_0^T\mathbf{M}^{2l}\mathbf{B}_0$ in Lemma 2. □

Recalling Section 4.1, the most influential users are those who, if seeded, can minimize the leading eigenvalue of $\mathbf{M}$. Then, as the leading eigenvalue is computed as $\lambda \approx \lim_{l\to\infty}\left[\frac{\mathbf{B}_0^T\mathbf{M}^{2l}\mathbf{B}_0}{\mathbf{B}_0^T\mathbf{B}_0}\right]^{\frac{1}{2l}}$ and $\mathbf{B}_0^T\mathbf{B}_0$ is a constant, minimizing the leading eigenvalue of $\mathbf{M}$ is equivalent to minimizing the term $\mathbf{B}_0^T\mathbf{M}^{2l}\mathbf{B}_0$. By this, based on the formula of $\mathbf{B}_0^T\mathbf{M}^{2l}\mathbf{B}_0$ given in Lemma 2, we are ready to give the formula of CI, which we adopt to evaluate the contributions of users on maximizing the influenced size.

4.2.2 Formula of Collective Influences in IM

From Lemma 2, $\mathbf{B}_0^T\mathbf{M}^{2l}\mathbf{B}_0$ is the sum of the polynomial

$$\sum_{u_h\in\partial Ball(k,l)} \left(\Pi_{(x,y)\in p(k,l)} w_{xy}\overline{I}_x\overline{I}_y\right)(d_h-1)\overline{I}_h,$$

of all the non-seed nodes. Thus, to minimize the value of $\mathbf{B}_0^T\mathbf{M}^{2l}\mathbf{B}_0$, the most effective way is seeding the user with the largest value of

$$\sum_{u_h\in\partial Ball(k,l)} \left(\Pi_{(x,y)\in p(k,l)} w_{xy}\overline{I}_x\overline{I}_y\right)(d_h-1)\overline{I}_h.$$

Given the initial network where no user has been seeded, the contribution of each user u_i for the value of $\mathbf{B}_0^T\mathbf{M}^{2l}\mathbf{B}_0$ can be evaluated as

$$\sum_{u_h\in\partial Ball(k,l)} \left(\Pi_{(x,y)\in p(k,h)} w_{xy}\right)(d_h-1).$$

Replace u_k and u_h by u_i and u_j, we obtain the formula of the collective influence of user u_i as below.

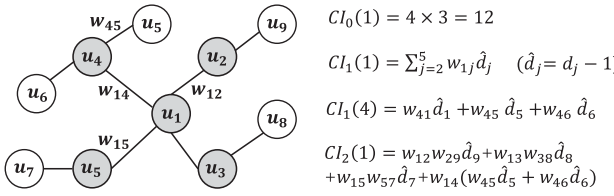


Fig. 2. An example of collective influence (CI). For u_1, we have $\partial Ball(1,1) = \{u_2, u_3, u_4, u_5\}$ and $\partial Ball(1,2) = \{u_5, u_6, u_7, u_8, u_9\}$. According to the CI formulation in Eq. (9), $CI_1(1) = \sum_{j=2}^{5} w_{ij}(d_j - 1)$ and $CI_2(1) = \sum_{j=5}^{9}(\Pi_{(x,y)\in p(1,j)} w_{xy})(d_j - 1)$. Also, $CI_1(4) = \sum_{u_j\in\partial Ball(4,1)} w_{4j}(d_j - 1)$. Specifically, $CI_0(i) = d_i(d_i - 1)$.

Formula of the Collective Influence (CI) in IM. In the IM problem under IC model, let l denote the power when computing the leading eigenvalue of $\mathbf{M}$, the collective influence (CI) of each user u_i is equal to

$$CI_l(i) = \sum_{u_j\in\partial Ball(i,l)} (\Pi_{(x,y)\in p(i,j)} w_{xy})(d_j - 1). \tag{9}$$

Here, $(x,y) \in p(i,j)$ denotes the edges on the non-backtracking path $p(i,j)$, w_{xy} is the influencing probability on edge (x,y) in IC model, and d_j denotes the degree of u_j. Then l controls the radius of the ball $Ball(i,l)$ considered in the computation of CI values.

Remark. The formula of the collective influence given in Eq. (9) quantifies the power of each user's neighborhood within l hops on shaping network collective dynamics, which specifically refers to the influence diffusion in this paper. In addition, the effect of the parameter l on the CI based seed selection will be analyzed in Section 4.2.3.

Fig. 2 presents a mini example of the formula of CI given in Eq. (9). The CI value of user u_i with radius l combines the reaching probabilities of u_i to the users on the frontier of $Ball(i,l)$ and the degrees of such users. Specifically, when $l = 0$, there are d_i paths starting from u_i and their endpoints are still node u_i. Since the probability of u_i influencing himself is obviously equal to 1 ($w_{ii} = 1$), $CI_0(i) = d_i \cdot 1 \cdot (d_i - 1)$. Notably, we can take $(\Pi_{(x,y)\in p(i,j)} w_{xy})$ as the weight of edges in the path $p(i,j)$ and $(d_j - 1)$ as the weight of endpoint u_j. With the weights of paths, the CI value of a node u_i given in the Eq. (9) can also be figured out as:

$$CI_l(i) = \text{The weighted sum of the non-backtracking paths starting from } u_i \text{ with length } l.$$

4.2.3 Effect of Parameter l on Seed Selection

Now, we move to explore the effect of the parameter l on the seed selection? Since maximizing $Q(S,q)$ is equivalent to minimizing $\lambda(S,q)$ as we presented in Section 4.1.3, we explore the effect of l via their performance on decreasing the leading eigenvalue of $\mathbf{M}$ to a given threshold 1. The reason behind is that if increasing the value of l can improve the performance of seed selection, then we can decrease $\lambda(S,q)$ to a given threshold by seeding less users under larger l. Here, we use q_c to denote the minimum required seed fraction to make $\lambda(S,q) = 1$.

Referring to the approximation formula of the leading eigenvalue in Eq. (5), if the term $\hat{\lambda} = \frac{\sum_{u_i \in V} CI_l(i)}{|E|} = 1$, then the leading eigenvalue of $\mathbf{M}$ is equal to 1. When $l = 0$, $CI_0(i) = \sum_i d_i(d_i - 1)$ and $\hat{\lambda}$ is initially equal to $\hat{\lambda} = \frac{\sum_{u_i \in V} d_i(d_i - 1)}{|E|}$. When $l > 0$, $\hat{\lambda} = \frac{\sum_{u_i \in V} CI_l(i)}{|E|}$. The formula of $\hat{\lambda}$ tells us that the value of q_c depends on the degree distribution of the network. Since it is generally hard to obtain the closed form expression of degree distribution in an arbitrary network, in the present work we derive the asymptotic formula of q_c on three common network models, i.e., Erdos-Renyi (ER) model, power-law degree distributed model and Stochastic Block Model (SBM) with multiple equal-sized communities. Then, in the Lemmas 3, 4 and 5 as below, we give the values of q_c under $l = 0$ and $l = 1$ over such three common network models for exploring the effect of parameter l on seed selection.

Lemma 3. *In an ER graph where each pair of nodes are connected at random with a given probability p. We have $q_c = \Theta(1 - \frac{1}{w\eta} - \frac{3}{\sqrt{\eta}})$ $(l = 0)$ and $q_c = \Theta\Big(\sqrt{\frac{1}{\eta}(1 - \frac{1}{w\eta})}\Big)(l = 1)$. Here, w denotes the weights of edges in G and $\eta = Np$.*

Lemma 4. *Given a network with the power-law degree distribution, i.e., $P(d = x) = a \cdot x^{-\gamma}$, q_c scales as $q_c = \Theta\big(K^{(1-\gamma)}\big)$ $(l = 0)$ and $q_c = \Theta\Big(K^{\overline{\gamma-2}}\Big)(l = 1)$. Here, $K = \Theta(1), K > 1$, and $2 < \gamma < 3$.*

Lemma 5. *The SBM characterizes networks into C equal-sized communities, and if two nodes belongs to a same community, they connect at random with probability p, otherwise, with probability q. The q_c for such SBM network scales as $q_c = \Theta\Big(1 - \frac{1}{w\eta} - \frac{3}{\sqrt{\eta}}\Big)$ $(l = 0)$ and $q_c = \Theta\Big(\sqrt{\frac{1}{\eta}(1 - \frac{1}{w\eta})}\Big)$ $(l = 1)$. Here, $\eta = \frac{N}{C}p + (N - \frac{N}{C})q$.*

The proofs for Lemmas 3, 4 and 5 are provided in Appendix A, available online, in the supplementary material. Comparing the values of q_c derived under $l = 0$ and $l = 1$, we can see that we can use fewer seed users to make $\lambda(S, q) \leq 1$ under $l = 1$, indicating that the seed users selected under $l = 1$ are more influential. The reason behind is that the larger l renders the CI values more structural information, and enables us to evaluate the influences of users more exactly. In Section 6, we further experimentally justify that the seeds selected under the larger value of l have the better performance on influence diffusion. Thus the larger l can help us select the better qN seed users for the IM problem.

Algorithm 1. CIM-SS Algorithm

Input: Graph $G = (V, E)$, Radius l;
Output: Seed set S;
1: $S = \emptyset$;
2: **while** $|S| < q|N|$ **do**
3: Compute $CI_l(i)$ $(\forall u_i \in V \backslash S)$ (Eq. (9));
4: $S^* = \arg\max_{u \in V \backslash S} CI_l(i)$;
5: Remove S^* from G;
6: $S = S \cup S^*$;
7: **end**
8: return S.

4.3 Collective Influence Based Seed Selection

4.3.1 Algorithm Design

With the formula of CI, a naive idea for seed selection based on node CI is iteratively selecting users with the highest CI value as shown in Algorithm 1. In each iteration, Algorithm 1, called CIM-SS, first selects the user with the highest $CI_l(i)$ as the new seed, and updates the CI value of each user with Eq. (9) after removing the new seed from network. The rationale of removing new seed from network is that, recalling Lemma 2, if user u_i is seeded, then $\overline{I}_i = 0$, indicating that seeding u_i is equivalent to removing it from the network when computing $\mathbf{B}_l^T \mathbf{B}_l$. Notably, as we remove the user from the network after selecting a seed and then update the CI values of remaining users to select the following seeds, we can avoid the overlap among the influences of seeds.

4.3.2 Performance Analysis

When quantifying CI above, we do not take into account the *circles*, which refer to the non-backtracking paths where a same node shows up more than once. In social networks, when considering the non-backtracking paths with length l, the contributions of the circles scale as a fraction of $\Theta\big(\frac{1}{N}\big)$ [15]. Thus the CI formula we give in Eq. (9) is actually a $\Theta\big(\frac{1}{N}\big)$-multiplicative error estimation. Moreover, since the leading eigenvalue $\lambda(S, q)$ is a monotonic and submodular function of S as shown in Proposition 1, such greedy seed selection algorithm for the NP-hard problem has an approximation ratio of $\big(1 - \frac{1}{e}\big)$. Together with the estimation error of CI, the approximation ratio of CIM-SS algorithm is $\big(1 - \Theta\big(\frac{1}{N}\big)\big)\big(1 - \frac{1}{e}\big)$.

Proposition 1. *The leading eigenvalue $\lambda(S, q)$ is a monotonic and submodular function.*

The proof of Proposition 1 is in Appendix A, available online, in the supplementary material. Furthermore, the complexity of Algorithm 1 is mainly focused on two tasks: the one is iteratively selecting seed users and the other is computing and updating users' CIs. In each iteration, the first task costs a complexity of $O(|V| \log |V|)$ when ranking the remaining users by their CI values. Thus, the first task costs a complexity of $O(|S||V| \log |V|)$. In the second task, Algorithm 1 needs to traverse all the edges and nodes that u_i can reach within l hops to compute the reaching probabilities via each path with length l. Under the IC model, we define $\Pi_{(x,y) \in p(i,j)} w_{xy}$ as the reaching probability via path $p(i, j)$ and it is different for different paths. However, referring to the famous "Six Degrees of Separation" theory [35], any user can reach almost all the other users through a few hops. Fig. 3 further justifies the theory. We present the numbers of the users that a given user can reach over two famous social networks LiveJournal and Wikipedia in Fig. 3, from which we can see that a source user can reach almost all the other users through 4 to 5 hops. This phenomenon means Algorithm 1 needs to traverse almost all of the nodes and edges in the network for computing one CI value. Thus computing and updating the CIs costs a complexity of $O\big(|S||V|(|E| + |V|)\big)$. Combing the two tasks together, we summarize in the following corollary the complexity.

Corollary 1. (Complexity of Algorithm 1.) *Algorithm 1 has a complexity of $O(|S||V|(|V| + |E|))$.*

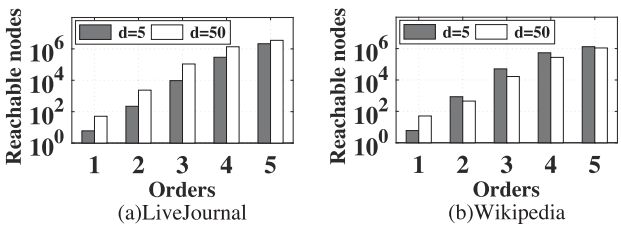


Fig. 3. The number of reachable nodes versus hops. We show the number of reachable nodes starting from two different nodes with the degree being 5 and 50.

Corollary 1 suggests that the complexity of Algorithm 1 goes up to the third power of network size. Such high complexity of Algorithm 1 motivates us to seek for more efficient solutions to improve the scalability of CI based seed selection over large scale social networks.

5 Embedding Collective Influence Maximization Scheme

Our solution for improving the scalability is presenting an embedding collective influence maximization scheme which embeds the network into a low-dimensional space, and then conducts the seed selection over the low-dimensional representations of users.

Notably, the network is originally represented by a graph $G = (V, E)$, which is then represented by an $N \times N$ matrix. Such original network representation results the challenge of high complexity in many classical network analysis based tasks, such as clustering, link prediction and classification [36]. The network embedding, which embeds network into a low-dimensional space, is a popular and general technology for coping with the challenge of high complexity [36], [37]. In this paper, relying on the metric of collective influence, we will present how to incorporate the network embedding technology into the task of influence maximization.

The key of the network embedding here is learning the low-dimensional representations of users, at the same time, preserving the CIs in the original network. Originally, each user is represented by an N-dimensional vector. As shown in Fig. 4, the specific objective of the network embedding task in this paper is learning the d-dimensional $(d \ll N)$ representation vector of each user (i.e., $\Phi : v \in V \rightarrow \mathbb{R}^d$) such that the low-dimensional representations can characterize the CI value, which is originally represented by

$$CI_l(i) = \sum_{u_j \in \partial Ball(i,l)} (\Pi_{(x,y)\in p(i,j)} w_{xy})(d_j - 1), \tag{10}$$

in the original network, as

$$CI_l(i) \propto \sum_{u_j \in V \setminus u_i} \Phi(u_i)^T \Phi(u_j)(d_j - 1). \tag{11}$$

Since we intend to characterize the CIs based on the low-dimensional representations $\Phi(u_i)\,(u_i \in V)$ of users, we call the low-dimensional representation learning in this paper as collective influence embedding, and present the methodologies in Section 5.1. With the low-dimensional user representations, we reformulate the IM problem as the Embedding IM problem for seeking scalable solutions.

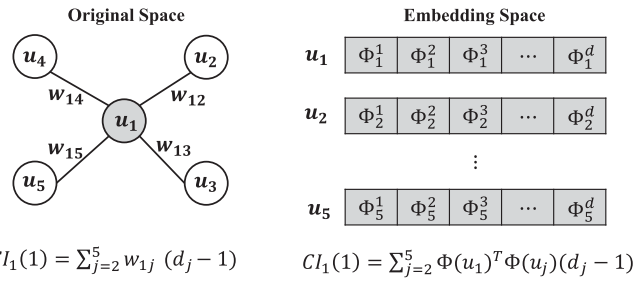


Fig. 4. A sketch of collective influence embedding. When $l = 1$, the CI value of u_1 originally equals $CI_1(1) = \sum_{j=2}^{5} w_{1j}(d_j - 1)$. The collective influence embedding is learning the d-dimensional representations $\Phi(u_i)$ of users, which satisfy $CI_1(1) = \sum_{j=2}^{5} \Phi(u_1)^T \Phi(u_j)(d_j - 1)$.

Embedding IM Problem. Given a network with N users where the users are represented by $\Phi(u_i)\,(1 \le i \le N)$, the aim of the Embedding IM problem is selecting a set S of qN seed users to maximize the influenced size $Q(S, q)$.

Coping with the Embedding IM problem, we propose the CIM-ESS algorithm which purses the CI based seed selection over $\Phi(u_i)\,(1 \le i \le N)$ in Section 5.2. The CIM-ESS algorithm can solve the Embedding IM problem with high efficiency. Next, we first move to the methods of collective influence embedding.

5.1 Collective Influence Embedding

We give the outline of the collective influence embedding in Fig. 5. The input is the original social network and the CI formulation in Eq. (10). Comparing with Eq. (10) and Eq. (11), obviously, we need to correlate the reaching probabilities (e.g., $(\Pi_{(x,y)\in p(i,j)} w_{xy})$) in Eq. (10) with users' low-dimensional representations. However, as illustrated before, it is time-consuming to compute the exact value of the reaching probabilities in the original network. Our solution is generating observations from a distribution parameterized by the reaching probabilities among users. As the observations need to identify the users belonging to the Ball of a given user and distinguish the reaching probabilities via different paths, we call the observations as *collective influence contexts* which are denoted by $\mathbf{Y}$. The generating method for $\mathbf{Y}$ will be unfolded in Section 5.1.1.

Furthermore, since we intend to characterize the reaching probabilities by the low-dimensional representations, the collective influence contexts can also be taken as the observations generated from a distribution parameterized by the low-dimensional representations of users. Such relation enables us to conduct the low-dimensional representation learning via *maximizing the likelihood* $P(\mathbf{Y} \mid \Phi(u_1), \Phi(u_2), \ldots, \Phi(u_N))$ (Section 5.1.2). In addition, through the convergence analysis in Section 5.1.3, we prove that the output low-dimensional representations converge to the maximizer of such likelihood.

5.1.1 Generating Collective Influence Context

We utilize a random walk approach to generate the observations from the reaching probabilities among users. In network

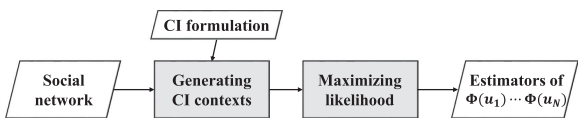


Fig. 5. Outline of collective influence embedding.

embedding, random walk is widely used to capture the structural regularities [37], information spread [38] and local community structure [39] in networks. Here, we adopt the random walk to capture the reaching probabilities among users, and conduct the random walk through non-backtracking paths in network. Specifically, given a source user u_i, the random walk randomly chooses one neighbor u_j of u_i to visit with the probability proportional to the weight w_{ij}. When the random walk arrives at u_j, it then chooses one of u_j's neighbor $u_k(u_k \neq u_i)$ to visit with the probability proportional to the weight w_{jk}. One random walk stops when the length of the sequence of the visited nodes meets a preset threshold L. Specially, when a random walk arrives at a node with no other edges other than the one such random walk arrived from, we will terminate the random walk . We conduct R random walks starting from each user in G. Note that if the reaching probability from u_i to u_m is larger than that from u_i to u_j, u_i and u_m may coexist in more random walk sequences than u_i and u_j.

The advantage of random walk is that it can not only incorporate the reachable users in the balls with different values of radius l, but also distinguish the heterogeneous reaching probabilities among different pairs of users via the coexisting times among them in the random walk sequences. Thus, the random walk can well capture the reaching probabilities among users. Next, we present how to extract the collective influence context $\mathbf{Y}$ from the $R|V|$ random walk sequences.

The context $\mathbf{Y}$ is the set of the collective influence context of each user u_i, say $\mathbf{Y}_i$. Concretely, $\mathbf{Y}_i$ consists of the users that line up behind u_i in all the $r|V|$ random walk sequences. For example, let $r_1 = \{u_1, u_2, \ldots, u_L\}$ be one random walk sequence starting from user u_1, then $u_2, \ldots, u_L$ are added into $\mathbf{Y}_1$. In addition, given one sequence $r_i = \{u_i, u_1, u_j \ldots, u_L\}(i \neq 1)$ that does not start from u_1, we also add $u_j \ldots, u_L$ into $\mathbf{Y}_1$. Different from the sampling methods adopted in current IM solutions, we generate the contexts of users also from the collective perspective. Specifically, consider the case that $u_j \in \mathbf{Y}_i$, $u_m \notin \mathbf{Y}_i$ and $u_m \in \mathbf{Y}_j$, we then also incorporate u_m into context $\mathbf{Y}_i$. As the random walk preferentially crosses the edges with higher weights, the larger the influence from u_i to u_j, the more times that u_j will appear in $\mathbf{Y}_i$. Also, if a node u_i has larger collective influence $CI_l(i)$, then $\mathbf{Y}_i$ will contain a larger size of nodes. We use s_{ij}, say coexisting times between u_i and u_j, to denote the times that u_j appears in $\mathbf{Y}_i$, and formalize the context $\mathbf{Y}_i$ of u_i as the set of the coexisting times between u_i and others, i.e.,

$$\mathbf{Y}_i = \{(u_1, s_{i1}), \ldots, (u_{i-1}, s_{i(i-1)}), (u_{i+1}, s_{i(i+1)}), \ldots, (u_N, s_{iN})\}.$$

5.1.2 Low-Dimensional Representation Learning

With the collective influence context $\mathbf{Y}$, the goal of the representation learning is to learn the low-dimensional representations of users that can maximize the likelihood $P(\mathbf{Y} \mid \Phi(u_1), \Phi(u_2), \ldots, \Phi(u_N))$. To this end, we need to figure out a probability distribution parameterized by the representations $\Phi(u_i)(\forall u_i \in V)$ for the collective influence context $\mathbf{Y}$. For this, we define $\Phi(u_i) = \{\theta_i, x_i\}$, where θ_i denotes the representation of user u_i as the source and x_i is the representation of u_i as the target. We then aim at utilizing the inner product $\langle \theta_i, x_j \rangle$ to represent the reaching probability from $u_i$to u_j. Here, we further define x_i as the preset vector generated from $\mathcal{N}(0, \mathbf{I})$. By this, we can have diverse destination vectors of different users, and further reduce the problem of representation learning to learning the estimators $\hat{\theta}_i(\forall u_i \in V)$ of the source vectors $\theta_i(\forall u_i \in V)$. Our scheme is also compatible with other generation methods of the destination vector x_i.

Distribution of Context $\mathbf{Y}$. We take each coexisting time s_{ij} as a random variable distributed around the value of $\langle \theta_i, x_j \rangle$, i.e., $s_{ij}|x_j \sim \mathcal{N}(\langle \theta_i, x_j \rangle, \sigma^2)$, where σ^2 is assumed as the variance of the observation noise in random walk. In practice, the variance σ^2 can be determined from the context $\mathbf{Y}$. We present how to compute variance σ^2 in Appendix B, available online, in the supplementary material. We adopt the Gaussian distribution here is for convenient computation and analysis. Furthermore, we correlate $\forall \mathbf{Y}_i \in \mathbf{Y}$ with the representations $\theta_i(\forall u_i \in V)$ via the following mixture of regression model

$$\mathbf{Y}_i | \, \Phi(u_1), \ldots, \Phi(u_N) \sim \frac{1}{N} \sum_{u_j \in V \setminus u_i} \mathcal{N}(\langle \theta_i, x_j \rangle, \sigma^2), \tag{12}$$

for jointly learning the estimators $\hat{\theta}_i$ of low-dimensional representations $\theta_i(1 \leq i \leq N)$ through maximizing the likelihood $P(\mathbf{Y}_i | \, \Phi(u_1), \ldots, \Phi(u_N))$ as follows.

EM Solutions. We conduct the learning of $\hat{\theta}_i \, (\forall u_i \in V)$ based on the Expectation-Maximization (EM) algorithm, which is considered as one of the most effective approaches for Maximizing Likelihood Estimation (MLE). The main idea of the EM is iteratively maximizing a log likelihood to obtain new parameters, and reevaluating the value of the log likelihood under the new parameters. The updating procedure is as follows. Given the updated representation θ_i^{t-1} at the $(t-1)$th iteration, the tth iteration in EM algorithm consists of the following E (expectation)-step and M(maximization)-step:

E-step: Computing the log likelihood, say $L(\theta_i' | \theta_i^{t-1})$, under the existing parameter θ_i^{t-1}. With the mixture of regressions model, we assume each coexisting times in $\mathbf{Y}_i$ is drawn i.i.d. from the mixture probability density in Eq. (12). Then, we define the likelihood function $L(\theta_i' | \theta_i^{t-1})$ as

$$L(\theta_i' | \theta_i^{t-1}) = \frac{1}{N} \sum_{j=1}^{N} \left(\sum_{j'=1}^{N} P(x_{j'} | s_{ij}, \theta_i^{t-1}) \log P(x_{j'}, s_{ij} | \theta_i') \right). \tag{13}$$

Here, $P(x_{j'} | s_{ij}, \theta_i^{t-1})$ denotes the posterior probability that the coexisting times s_{ij} is generated from the distribution parametrized by $x_{j'}$ and θ_i^{t-1}, and is given by

$$\begin{aligned} & P(x_{j'} | s_{ij}, \theta_i^{t-1}) \\ & = w_{\theta_i^{t-1}}(x_{j'}, s_{ij}) = \frac{\exp\left(-\frac{(s_{ij} - \langle \theta_i^{t-1}, x_{j'} \rangle)^2}{2\sigma^2} \right)}{\sum_{j'=1}^{N} \exp\left(-\frac{(s_{ij} - \langle \theta_i^{t-1}, x_{j'} \rangle)^2}{2\sigma^2} \right)}. \end{aligned} \tag{14}$$

Taking Eq. (14) and the formula of $P(x_{j'}, s_{ij} | \theta_i')$ into Eq. (13), we further simplify the likelihood $L(\theta_i' | \theta_i^{t-1})$ as

$$L\big(\theta_i'|\theta_i^{t-1}\big) = -\frac{1}{N}\sum_{j=1}^{N}\left(\sum_{j'=1}^{N} w_{\theta_i^{t-1}}(x_{j'}, s_{ij})\frac{\big(s_{ij} - \langle\theta_i', x_{j'}\rangle\big)^2}{2\sigma^2}\right). \tag{15}$$

It can be observed from From Eq. (15) that, by maximizing $L(\theta_i'|\theta_i^{t-1})$, we can reduce the deviation $(s_{ij} - \langle\theta_i', x_j\rangle)^2$, and maximize the likelihood $P(\mathbf{Y}_i|\,\Phi(u_1), \ldots, \Phi(u_N))$. Also, in function $L\big(\theta_i'|\theta_i^{t-1}\big)$, the value of $w_{\theta_i^{t-1}}(x_{j'}, s_{ij})$ attaches higher weights to the terms with smaller deviations, and thus enables the MLE to be convergent.

M-step: Updating the parameter θ_i by maximizing the likelihood $L\big(\theta_i'|\theta_i^{t-1}\big)$. In the tth iteration, let the updated parameter $\theta_i^t = M_{\mathbf{Y}_i}(\theta_i^{t-1}) = \arg\max_{\theta_i'\in\mathbb{R}^d} L(\theta_i'|\theta_i^{t-1})$, given the formula of $L(\theta_i'|\theta_i^{t-1})$ in Eq. (15), we have

$$\theta_i^t = M_{\mathbf{Y}_i}(\theta_i^{t-1}) = \frac{\sum_{j=1}^{N}\sum_{j'=1}^{N} w_{\theta_i^{t-1}}(x_{j'}, s_{ij})x_{j'}s_{ij}}{\sum_{j=1}^{N}\sum_{j'=1}^{N} w_{\theta_i^{t-1}}(x_{j'}, s_{ij})\langle x_{j'}, x_j\rangle}. \tag{16}$$

The EM algorithm iteratively conducts the above E-step and M-step until the parameters converge or the iterating times meet a preset threshold. Assuming there are T iterations in EM algorithm, the output of which is $\hat{\theta}_i = \theta_i^T = M_{\mathbf{Y}_i}(\theta_i^{T-1})$.

In summary, the CI embedding is conducted by first generating the collective influence contexts $\mathbf{Y}$, and then maximizing the likelihood of $\mathbf{Y}$ via EM algorithm. To ensure the reproducibility of our work, we also algorithmically present the procedures of the CI embedding scheme, generating context $\mathbf{Y}$ and computing variance σ^2 in Appendix B, available online, in the supplementary material.

5.1.3 Performance Analysis of Representation Learning

We now move to justify that the learned representations $\hat{\theta}_i(\forall u_i \in V)$ can maximize the likelihood of context $\mathbf{Y}$. To be more precise, we prove that the estimator $\hat{\theta}_i(\forall u_i \in V)$ converges to the maximizer $\theta_i(\forall u_i \in V)$ for the likelihood $P\big(\mathbf{Y}_i\,|\,\Phi(u_1), \Phi(u_2), \ldots, \Phi(u_N)\big)$. For simplification, in the following of this section, we omit the subscript i and use θ to represent θ_i since the performance analysis is the same for the representation learning of each user.

For the maximizer of likelihood, say θ, [40] introduces the self consistency property on the EM based MLE, i.e., $\theta = \arg\max_{\theta'\in\mathbb{R}^d} L(\theta'|\theta)$. Since $M_{\mathbf{Y}_i}(\theta_i^{t-1}) = \arg\max_{\theta_i'\in\mathbb{R}^d} L(\theta_i'|\theta_i^{t-1})$, the self consistency property suggests that $||\theta - M_{\mathbf{Y}}(\theta)||_2 = 0$. With this property, Theorem 1 presents that under certain condition, the updated estimator θ^t will be closer to the maximizer θ after each iteration.

Theorem 1. (Convergence of the representation learning.) *Given the mixture of regressions model in Eq. (15) with a sufficiently large signal-to-noise ratio (SNR) $\frac{||\theta^*||_2}{\sigma^2}$, there is a constant $\xi \in (0,1)$ that, in each iteration,*

$$||M_{\mathbf{Y}}(\theta^t) - \theta||_2 \le \xi||\theta^t - \theta||_2, \tag{17}$$

holds for all θ^t if $||\theta^t - \theta||_2 \le \frac{||\theta||_2}{2}$.

Proof. (Sketch.) As mentioned in Eq. (16), the EM operator $M(\cdot)$ has the form $M_{\mathbf{Y}_i}(\theta_i^{t-1}) = \frac{\sum_{j=1}^{N}\sum_{j'=1}^{N} w_{\theta_i^{t-1}}(x_{j'}, s_{ij})x_{j'}s_{ij}}{\sum_{j=1}^{N}\sum_{j'=1}^{N} w_{\theta_i^{t-1}}(x_{j'}, s_{ij})\langle x_{j'}, x_j\rangle}$.

By taking the expectation of the operator $M(\cdot)$ over the distribution of the pair $(\mathbf{Y}, \mathbf{X}) \in \mathbb{R}\times\mathbb{R}^d$, we have $M_{\mathbf{Y}}(\theta^t) = \mathbb{E}[w_{\theta^t}(\mathbf{Y}, \mathbf{X})\mathbf{Y}\mathbf{X}]$. Define the notations that $\Delta_w(\mathbf{X}, \mathbf{Y}) = w_{\theta^t}(\mathbf{X}, \mathbf{Y}) - w_\theta(\mathbf{X}, \mathbf{Y})$ and $\Delta = \theta^t - \theta$, Eq. (17) is equivalent to

$$||\mathbb{E}[\Delta_w(\mathbf{X}, \mathbf{Y})\mathbf{Y}\mathbf{X}]||_2 \le \xi||\Delta||_2.$$

Note that $\mathbf{Y} = \langle\mathbf{X}, \theta\rangle + v$, given any $\tilde{\Delta}$, Eq. (17) can be further transferred to

$$\begin{aligned}&\Big\langle\mathbb{E}[\Delta_w(\mathbf{X}, \mathbf{Y})\mathbf{Y}\mathbf{X}], \tilde{\Delta}\Big\rangle \le \xi||\Delta||_2||\tilde{\Delta}||_2;\\ &\mathbb{E}\Big[\Delta_w(\mathbf{X}, \mathbf{Y})\langle\mathbf{X}, \theta\rangle\langle\mathbf{X}, \tilde{\Delta}\rangle\Big] + \mathbb{E}\Big[\Delta_w(\mathbf{X}, \mathbf{Y})v\langle\mathbf{X}, \tilde{\Delta}\rangle\Big]\\ &\le \xi||\Delta||_2||\tilde{\Delta}||_2.\end{aligned}$$

In Appendix C, available online, in the supplementary material, we respectively prove that $\mathbb{E}\big[\Delta_w(\mathbf{X}, \mathbf{Y})\langle\mathbf{X}, \theta\rangle\langle\mathbf{X}, \tilde{\Delta}\rangle\big] \le \frac{\xi}{2}||\Delta||_2||\tilde{\Delta}||_2$ and $\mathbb{E}\big[\Delta_w(\mathbf{X}, \mathbf{Y})v\langle\mathbf{X}, \tilde{\Delta}\rangle\big] \le \frac{\xi}{2}||\Delta||_2||\tilde{\Delta}||_2$. Thus, with the upper bounds of such two terms, we can obtain the conclusion in Theorem 1. □

Since $\hat{\theta} = \theta^T = M_{\mathbf{Y}}(\theta^{T-1})$, by Theorem 1, we have

$$\begin{aligned}||\hat{\theta} - \theta||_2 = ||\theta^T - \theta||_2 = ||M_{\mathbf{Y}}(\theta^{T-1}) - \theta||_2 \le \xi||\theta^{T-1} - \theta||_2\\ \le \xi^2||\theta^{T-2} - \theta||_2 \le \ldots \le \xi^T||\theta^0 - \theta||_2.\end{aligned}$$

Thus the learned low-dimensional representations converge to the maximizer of the likelihood after sufficient iterations.

Algorithm 2. CIM-ESS Algorithm

Input: $\Phi(u_1), \ldots, \Phi(u_N)$, User degrees $d_1, \ldots, d_N$;
Output: Seed set S;
//Offline computing phase
for $1 \le i \le N$ **do**
  $CI(i) = 0$;
  for $1 \le j \le N$ **do**
    Compute $CI(i) = CI(i) + \langle\theta_i, x_j\rangle(d_j - 1)$;
  end
end
//Online selecting phase
$S = \emptyset$;
while $|S| \le qN$ **do**
  $s^* = \arg\max_{v_i\in V\setminus S} CI(i)$, $S = S \cup s^*$;
  $\theta_{i^*} \leftarrow$ source vector of s^*;
  for *each* $j : \langle\theta_{i^*}, x_j\rangle > 0$ **do**
    for *each* $i : \langle\theta_i, x_j\rangle > 0$ **do**
      $\langle\theta_i, x_j\rangle = \max\big\{(\langle\theta_i, x_j\rangle - \langle\theta_{i^*}, x_j\rangle), 0\big\}$;
      Update $CI(i)$;
    end
  end
end
return S.

5.2 Embedding CI Based Seed Selection

We algorithmically present the seed selection over low-dimensional representations in Algorithm 2 called CIM-ESS algorithm. With the learned low-dimensional representations,

we are able to compute the collective influences of users without the traverse of the network. At the beginning, CIM-ESS computes the CI values for each user, e.g., $CI(i)$, based on the low-dimensional representations. Then CIM-ESS iteratively selects the user with the highest value of $CI(i)$ as seed until the seed set size meets qN. After seeding one user, CIM-ESS updates the value of $\langle \theta_i, x_j \rangle$ by $\langle \theta_i, x_j \rangle = \max\{(\langle \theta_i, x_j \rangle - \langle \theta_{i^*}, x_j \rangle), 0\}$ and then updates the CIs based on the updated inner products. Here, we update the value of $\langle \theta_i, x_j \rangle$ by $\langle \theta_i, x_j \rangle = \max\{(\langle \theta_i, x_j \rangle - \langle \theta_{i^*}, x_j \rangle), 0\}$ is for avoiding the overlap among the influences of seeds and keeping the coexisting times $\langle \theta_i, x_j \rangle$ nonnegative.

Lemma 6 presents that CIM-ESS can purse the CI based seed selection with much less complexity comparing with CIM-SS. Regrettably, due to the errors of the estimated CI values, we are unable to provide the quantified performance guarantee of CIM-ESS.

Lemma 6. *The complexity of CIM-ESS scales as $O(d|S||V|^2)$, where $d(d \ll |V|)$ is the dimension of user representations.*

Proof. In the offline phase, CIM-ESS computes the inner products among the $|V|$ d-dimensional vectors. Then the complexity of the offline phase is $O(d|V|^2)$. During the online phase, in each iteration, CIM-ESS adopts a nested *For* loop to update the estimated CI values of users and costs a complexity of $O(d|V|^2)$. In addition, each iteration costs a complexity of $O(|V|\log|V|)$ to sort the users by their updated CI values. Since there are $q|V| = |S|$ iterations in the online phase, the complexity of the online phase is upper bounded by $O\Big(d|S|(|V|^2 + |V|\log|V|)\Big)$. In summary, the complexity of CIM-ESS is $O(d|S||V|^2)$. □

Remark. Although the collective influence embedding incurs more complexity, this is a one-shot task. Whereas the seed selection needs to be conducted for multiple times in real applications, such as the multi-round IM [26] and the location-aware IM that promotes different points of interests in different IM campaigns [9]. Thus, we propose the collective influence embedding scheme to improve the efficiency of seed selection in IM.

5.3 A Summary of Sections 4 and 5

In this paper, we study the problem of identifying a given size of seed users which can maximize the influence diffusion over the given mobile social network. For addressing the drawbacks of previous sampling based solutions, we take the collective influence (CI) as the metric, which evaluates the contributions of users on influence diffusion from their structural features. We uncover in Section 4 that, under the independent cascading diffusion model, the collective influence of each user u_i can be formulated as the weighted sum of non-backtracking paths starting from u_i, as presented in Eq. (9). Moreover, since directly computing the CI values needs to repeatedly traverse the whole network and thus induces huge complexity, we propose the collective influence embedding scheme in Section 5, for learning the low-dimensional representations of users that can characterize their collective influences over original network. The low-dimensional user representations then enables us to efficiently compute the CI values and select the seed users.

TABLE 1
Statistics of Datasets

Datasets	# of Nodes	# of Edges	Description
LiveJournal	4.85M	69M	Real social network
Wikipedia	1.79 M	28.5M	Wikipedia hyperlinks
Twitter	81K	1.8M	Real social network
Epinions	75K	0.5M	Real social network
Citation	1.5M	7M	Paper citations
ER	100K	/	ER network
PL	10K	/	Power-law
SBM	100K	/	Stochastic block

Remark. The percolation theory originally considers the networks where nodes are arranged in a regular grid like pattern [41]. We here map the IM problem under IC model into the optimal percolation with the assumption of locally-tree like network. Such assumptions on structural regularity may affect the performance of our solutions, when applied into dense social networks. We evaluate our solutions on several common social network datasets in Section 6.

6 Experiments

We will experimentally examine the performance of our solutions to IM problem. Specifically, we study the following four issues: (1) Are the seeds who can bring larger drops to the leading eigenvalue of $\mathbf{M}$ really more influential? (2) Do the two algorithms, i.e., CIM-SS and CIM-ESS outperform the current IM solutions on the effectiveness which is measured by the influence diffusion size of selected seed users? (3) Can the low-dimensional representations of users accurately capture the collective influences among them? (4) How is the structural feature of the seeds selected based on the CI. Next, we first introduce the experimental settings.

6.1 Experimental Settings

Dataset. We use 8 network datasets in experiments, as shown in Table 1, for evaluating our proposed solutions on influence maximization. Specifically, the Livejoural, Twitter and Epinions are real social networks where the nodes represent users and the edges represent social links among them, and are downloaded from the open social network dataset collection SNAP [42]. The Wikipedia dataset is also downloaded from SNAP, where the nodes represent articles in Wikipedia and the edges represent hyperlinks among them. The Citation network dataset is from the open academic dataset collection Acenap [43], and contains $1.5M$ papers (nodes) in the Machine Learning area and the citation relations (edges) among them. The three synthetic networks are respectively generated as ER model, power-law degree distribution (PL) and Stochastic Block Model (SBM). Since we generate each synthetic network under four different settings, we do not list the edge sizes in Table 1.

Baselines. We compare the CIM-SS and CIM-ESS algorithms with the following four baselines.

(1) IMM [22]: IMM is one of the most popular IM solutions based on the Reverse Reachable sets (RR-sets) framework. The main idea of the IMM lies in first sampling a sufficient number of RR-sets for the influenced size estimation, and

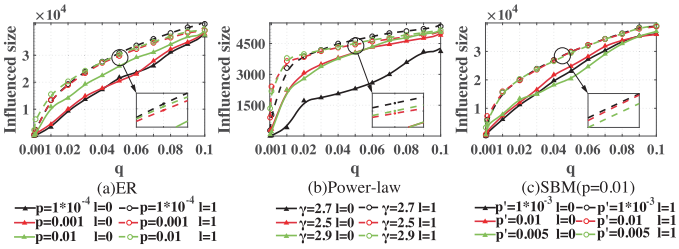


Fig. 6. Influenced size over ER, power-law and SBM graphs versus q. In ER (a) p means the connecting probability between each pair of users. In power-law (b), γ means the exponent of the power-law degree distribution. In SBM (c), p and p' respectively denote the probability of any user connecting to the users in a same community and other communities. q means the fraction of seed users.

then iteratively selecting seed users who can cover the most number of RR-sets.

(2) SKIM [18]: The main idea of the SKIM is repeatedly sampling the RR-sets from the network, and iteratively selecting the users who can firstly cover a preset number of RR-sets as the seeds.

(3) K-core: A popular percolation method to find core nodes. The main idea is deleting nodes in turn with the degrees being $1, 2, 3, \ldots$. After the deleting process, the remaining nodes are taken as the core nodes. Then the seeds in K-core are randomly selected from the remaining nodes.

(4) Maximum degree: Iteratively selecting the user with the highest degree. Since CIM-SS $(l = 0)$ is equivalent to selecting users with highest degrees, we call Maximum degree as CIM-SS $(l = 0)$ later.

Parameter Settings. We set the weight w_{ij} of edge (i, j) as $w_{ij} = \Theta(\frac{1}{d_j})$. This setting is widely used in [16] [22] [17].

Environment. We implement algorithms with Python 2.7 on a computer running Ubuntu 16.04 LTS with 40 cores 2.30 GHz (Intel Xeon E5-2650) and 128 GB memory.

6.2 Validation of the Quantifications

In Section 4, we prove in Lemmas 3, 4 and 5 that the seeds selected under $l = 1$ are more influential since they can decrease $\lambda(S, q)$ to 1 with fewer seed fraction. Here, we validate if the seeds selected under $l = 1$ can really influence more users comparing with those selected under $l = 0$. From Fig. 6, we can see that the seeds selected under $l = 1$ obviously outperform those selected under $l = 0$ on the influence diffusion size. Thus we validate that the seeds which bring larger drops to the leading eigenvalue of $\mathbf{M}$ are actually more influential.

In addition, Figs. 6a and 6c show that the influenced size increases with the connecting probability p and q on ER and SBM network, and Fig. 6b justifies that, over power-law network, the effectiveness of the seed users selected via CI increases with the increase of γ.

6.3 Performance On Seed Selection

Effectiveness Study. Table 2 presents the influence diffusion size starting from the seeds returned by CIM-SS, CIM-ESS, and the four baseline algorithms under IC diffusion model. Since CIM-SS ($l \geq 3$) and CIM-ESS measure the influences of users based on more structural context information as we discussed before, the seeds returned by them always influence the most users. Besides, CIM-SS ($l = 1, 2$) has the comparable performance with the baselines (i.e., SKIM and IMM). Notably, the influenced sizes of CIM-SS under $l = 5, 6$ are sometimes less than that under $l = 4$. This is due to the variance in the simulation of influence diffusion process when counting influenced size. Such variance further indicates that current sampling based methods cannot reliably estimate the influences of users. In addition, the Maximum degree (CI $(l = 0)$) and the K-core always have the poorest effectiveness due to the large overlap among the influences of the seeds returned by them.

Efficiency Study. Table 3 reports the running time of the seed selection algorithms. The CIM-SS $(l = 1)$ which has the comparable influenced size of the SKIM and IMM just costs a fraction of the running time of them. From $l = 2$, we can see the running time of CIM-SS largely increases with l since the computation of CIs needs to traverse almost all the nodes and edges in the network. Notably, due to the prohibitive running time, when conducting CIM-SS under $l = 4, 5, 6$, we only select the seeds from the top $1/500$ users with the highest out-degrees. For CIM-ESS, most of the running time is spent on computing the inner products among the low-representations of users. Fortunately, with the help of the low-dimensional representations, the CIs of users can

TABLE 2
Influenced Size Versus q

Algorithm	LiveJournal			Wikipedia			Citation			Epinions			Twitter		
	$q = 5 \times 10^{-6}$	10^{-5}	10^{-4}	5×10^{-6}	10^{-5}	10^{-4}	5×10^{-6}	10^{-5}	10^{-4}	10^{-5}	5×10^{-4}	10^{-3}	10^{-5}	5×10^{-4}	10^{-3}
SKIM	$8.3k$	$10.8k$	$53.7k$	$44k$	$50k$	$87k$	$22.4k$	$27k$	$83.4k$	171	202	$1.8k$	931	$2.4k$	$6.6k$
IMM	$23.8k$	$33.4k$	$64.4k$	$63.4k$	$71k$	$122.3k$	$19.1k$	$29.5k$	$85.2k$	32	204	$2k$	$3.5k$	$4.5k$	$9.4k$
K-core	297	384	933	$2.8k$	$3.7k$	$14k$	217	257	$1.7k$	27	50	658	$1.1k$	$1.8k$	$5.1k$
CIM-SS $(l = 0)$	$1.4k$	$9.2k$	$31.7k$	$3.2k$	$45k$	$19.2k$	$14k$	$19k$	$77.9k$	87	194	1048	$1.3k$	$1.8k$	$5.3k$
CIM-SS $(l = 1)$	$9.4k$	$19k$	$48k$	$16k$	$28k$	$90.4k$	$14.5k$	$27k$	$82.9k$	785	942	$2.2k$	$2.4k$	$3.5k$	$7.7k$
CIM-SS $(l = 2)$	$22.6k$	$38.4k$	$62k$	$15.3k$	$41.4k$	$130k$	$15.4k$	$32.1k$	$91.8k$	$1.1k$	$1.5k$	$2.7k$	$3.6k$	$5.5k$	$11.5k$
CIM-SS $(l = 3)$	$56.4k$	$64.3k$	$150k$	$77k$	$90.7k$	$232k$	$23.2k$	$36.5k$	$99.1k$	$1.3k$	$1.5k$	$2.9k$	$3.5k$	$4.8k$	$12k$
CIM-SS $(l = 4)$	**58.9k**	**68k**	**166.7k**	**90k**	**110k**	**240k**	**25.5k**	**36.8k**	**105k**	**1.3k**	**1.6k**	**2.9k**	**3.4k**	**4.8k**	**11.8k**
CIM-SS $(l = 5)$	**58.3k**	**66.4k**	**160k**	**90k**	**110k**	**240k**	**17.7k**	**30.6k**	**102k**	–	–	–	–	–	–
CIM-SS $(l = 6)$	**56.9k**	**64k**	**159k**	**89k**	**108k**	**225k**	**26.2k**	**34.8k**	**101k**	–	–	–	–	–	–
CIM-ESS	**59.5k**	**72.6k**	**163k**	**71k**	**97k**	**233k**	**18.5k**	**33.2k**	**101k**	**1.3k**	**1.5k**	**2.9k**	**3.5k**	**5k**	**11.7k**

Here, k means 10^3

TABLE 3
Running Time of Seed Selection (s) Versus q

	LiveJournal			Wikipedia			Citation			Epinions			Twitter		
Algorithm	$q=5\times10^{-6}$	10^{-5}	10^{-4}	5×10^{-6}	10^{-5}	10^{-4}	5×10^{-6}	10^{-5}	10^{-4}	10^{-5}	5×10^{-4}	10^{-3}	10^{-5}	5×10^{-4}	10^{-3}
SKIM	$41.0k$	$48.0k$	$97.0k$	$0.4k$	$0.4k$	$0.4k$	$31.0k$	$37.0k$	$85.0k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$
IMM	$0.8k$	$1.3k$	$14.0k$	$0.3k$	$0.5k$	$2.3k$	$0.1k$	$0.2k$	$1.0k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$
K-core	$46.8k$	$46.8k$	$46.8k$	$2.8k$	$2.8k$	$2.8k$	$1.5k$	$1.5k$	$1.5k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$
CIM-SS ($l=0$)	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$
CIM-SS ($l=1$)	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$
CIM-SS ($l=2$)	$0.1k$	$0.1k$	$0.5k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$
CIM-SS ($l=3$)	$0.3k$	$0.4k$	$4.0k$	$0.1k$	$0.1k$	$0.2k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.1k$	$0.2k$	$0.5k$	$0.7k$	$1.6k$
CIM-SS ($l=4$)	$0.8k$	$1.4k$	$15.0k$	$0.1k$	$0.1k$	$0.6k$	$0.1k$	$0.1k$	$0.1k$	$1.2k$	$1.8k$	$3.6k$	$1.3k$	$2.4k$	$5.0k$
CIM-SS ($l=5$)	$1.7k$	$2.6k$	$27.0k$	$0.1k$	$0.1k$	$1.1k$	$0.1k$	$0.1k$	$0.5k$	–	–	–	–	–	–
CIM-SS ($l=6$)	$2.2k$	$3.2k$	$30.0k$	$0.1k$	$0.1k$	$1.7k$	$0.1k$	$0.1k$	$0.6k$	–	–	–	–	–	–
CIM-ESS	**3.0k**	**3.2k**	**3.8k**	**0.3k**	**0.3k**	**0.4k**	**0.3k**	**0.3k**	**0.3k**	**0.1k**	**0.2k**	**0.2k**	**0.2k**	**0.4k**	**0.6k**

Here, k means 10^3

be easily conducted in parallel, thus largely improving the efficiency.

6.4 Performance of Collective Influence Embedding

Now, we present the performance of the collective influence embedding, whose objective is learning the low-dimensional representations $\{\theta_i, x_i\}$ of users to characterize the CI values of them. We set the random walk length as $L=50$ and conduct 10 random walk processes starting from each node. Since we further quantify the CI values by the coexisting times (e.g., s_{ij}) in the context $\mathbf{Y}$ and focus on using the inner product $\langle\theta_i, x_j\rangle$ to represent s_{ij}, we define the $error = \frac{|s_{ij}-\langle\theta_i,x_j\rangle|}{s_{ij}}$ to quantify the performance of the CI embedding. The smaller difference between the values of s_{ij} and $\langle\theta_i, x_j\rangle$ means the better performance of CI embedding. We adopt the relative error $\frac{|s_{ij}-\langle\theta_i,x_j\rangle|}{s_{ij}}$ here is for uniformly evaluating the performance of quantifying the coexisting times with different magnitudes. We provide the relative errors in Fig. 7, and from which we can see that the 80 percent of the relative errors are below 40 percent.

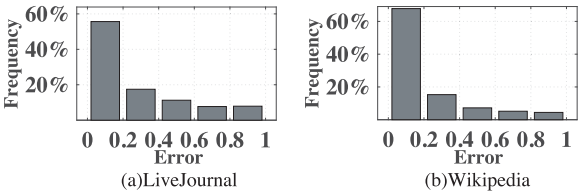


Fig. 7. The errors of collective influence embedding.

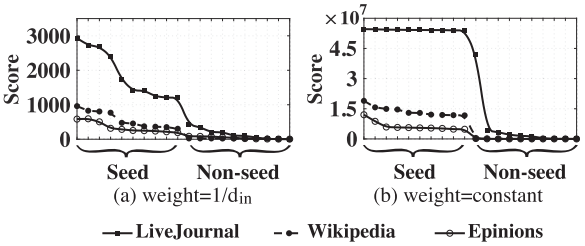


Fig. 8. The scores of seed and non-seed users under two weight settings. The "score" defined in Section 6.5 is used to quantify the structural features of seeds.

6.5 Structural Features of Superspreaders

At last, we move to the structural features of the seeds returned by CIM-SS in general graphs. Under the IC diffusion model, the influence diffusion over a given network also depends on the weights of edges. Thus we explore the features of seeds under two widely adopted weight settings, i.e., the one is setting the weights as $\Theta(1/d_{in})$ (d_{in} and d_{out} respectively refer to the in-degree and out-degree of a user) and the other is setting weights as constant. We find that, when the weight of each edge are set as $\Theta(1/d_{in})$, the seeds are those who can reach high out-degree users via low in-degree intermediate neighbors. On the other hand, when the weights are set as constants, the seeds are the users surrounded by high out-degrees nodes. We use a concept of "score" to quantify such structural features. We define $score = d^0_{out}\frac{d^1_{out}}{d^2_{in}}\frac{d^2_{out}}{d^2_{in}}$ when weights are set as $\Theta(1/d_{in})$, and define $score = d^0_{out}\cdot d^1_{out}\cdot d^2_{out}$ when weights are set as constants. Here, d^i_{out} and d^i_{in} respectively denote the mean out-degree and in-degree of the users' i-th-hop neighbors. Fig. 8 shows the scores of the seed and non-seed users under the two weight settings, and presents that the scores of seeds are much higher than those of non-seed users. The score gap between seed and non-seed users justifies our findings of the structural features.

7 Conclusion

In this paper, we take the first attempt to study the CI based influence maximization under the IC diffusion model, where we evaluate the influences of users based on the their structural features in mobile social networks. By mapping the studied problem to the optimal percolation, we present that the optimal seeds are those who can minimize the leading eigenvalue of a non-backtracking matrix, and quantify the formula of node CI which serves as a novel metric to evaluate users' contribution on cascading. Furthermore, we propose a novel CI embedding method to characterize the node CIs in a low-dimensional space, and over which we can purse the CI based seed selection with high efficiency. At last, experimental results on both real and synthetic network dataset demonstrate the superiority of our solution.

ACKNOWLEDGMENTS

This work was supported by National Key R&D Program of China under Grant 2018YFB2100302, and NSF China under Grants 61960206002, 61822206, 62020106005, 61829201, 61832013, and 42050105. The early version of this paper is in Proc. ACM MobiHoc Posters 2019 [1].

REFERENCES

[1] X. Wu, L. Fu, K. Wu, B. Jiang, X. Wang, and G. Chen, "Collective influence maximization," in *Proc. 20th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2019, pp. 385–386.

[2] S. Tang, J. Yuan, and V. Mookerjee, "Optimizing ad allocation in mobile advertising," in *Proc. 21st Int. Symp. Theory, Algorithmic Found., Protocol Des. Mobile Netw. Mobile Comput.*, 2020, pp. 181–190.

[3] M. N. Soorki, W. Saad, M. H. Manshaei, and H. Saidi, "Social community-aware content placement in wireless device-to-device communication networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 8, pp. 1938–1950, Aug. 2019.

[4] G. Tong, W. Wu, S. Tang, and D. Du, "Adaptive influence maximization in dynamic social networks," *IEEE/ACM Trans. Netw.*, vol. 25, no. 1, pp. 112–125, Feb. 2017.

[5] Z. Lu, Y. E. Sagduyu, and Y. Shi, "Integrating social links into wireless networks: Modeling, routing, analysis, and evaluation," *IEEE Trans. Mobile Comput.*, vol. 18, no. 1, pp. 111–124, Jan. 2019.

[6] J. Tang *et al.*, "Efficient approximation algorithms for adaptive seed minimization," in *Proc. Int. Conf. Manage. Data*, 2019, pp. 1096–1113.

[7] X. Li, J. D. Smith, T. N. Dinh, and M. T. Thai, "Why approximate when you can get the exact? Optimal targeted viral marketing at scale," in *Proc. IEEE Conf. Comput. Commun.*, 2017, pp. 1–9.

[8] K. Han, C. Xu, F. Gui, S. Tang, H. Huang, and J. Luo, "Discount allocation for revenue maximization in online social networks," in *Proc. 18th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2018, pp. 121–130.

[9] X. Wu, L. Fu, Y. Yao, X. Fu, X. Wang, and G. Chen, "GLP: A novel framework for group-level location promotion in Geo-social networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2870–2883, Dec. 2018.

[10] Z. Lu, Y. Wen, and G. Cao, "Information diffusion in mobile social networks: The speed perspective," in *Proc. IEEE Conf. Comput. Commun.*, 2014, pp. 1932–1940.

[11] D. Centola, "The spread of behavior in an online social network experiment," *Science*, vol. 329, no. 5996, pp. 1194–1197, 2010.

[12] E. M. Rogers, *Diffusion of Innovations*, New York, NY, USA: Simon and Schuster, 2010.

[13] S. Pei, F. Morone, and H. Makse, "Theories for influencer identification in complex networks," in *Complex Spreading Phenomena in Social Systems*. Berlin, Germany: Springer, 2018, pp. 125–148.

[14] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2003, pp. 137–146.

[15] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, "Maximizing social influence in nearly optimal time," in *Proc. 25th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2014, pp. 946–957.

[16] H. T. Nguyen, T. P. Nguyen, T. N. Vu, and T. N. Dinh, "Outward influence and cascade size estimation in billion-scale networks," in *Proc. Int. Conf. ACM Meas. Anal. Comput. Syst.*, 2017, pp. 1–30.

[17] Y. Lin, W. Chen, and J. C. Lui, "Boosting information spread: An algorithmic approach," in *Proc. IEEE 33rd Int. Conf. Data Eng.*, 2017, pp. 883–894.

[18] E. Cohen, D. Delling, T. Pajor, and R. F. Werneck, "Sketch-based influence maximization and computation: Scaling up with guarantees," in *Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage.*, 2014, pp. 629–638.

[19] F. Morone and H. Makse, "Influence maximization in complex networks through optimal percolation," *Nature*, vol. 524, no. 7563, 2015, Art. no. 65.

[20] S. Luo, F. Morone, C. Sarraute, M. Travizano, and H. Makse, "Inferring personal economic status from social network location," *Nat. Commun.*, vol. 8, 2017, Art. no. 15227.

[21] F. Morone, K. Roth, B. Min, H E. Stanley, and H. Makse, "Model of brain activation predicts the neural collective influence map of the brain," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 15, pp. 3849–3854, 2017.

[22] Y. Tang, Y. Shi, and X. Xiao, "Influence maximization in near-linear time: A martingale approach," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2015, pp. 1539–1554.

[23] Q. Liu, B. Xiang, E. Chen, H. Xiong, F. Tang, and J. X. Yu, "Influence maximization over large-scale social networks: A bounded linear approach," in *Proc. ACM 23rd Int. Conf. Inf. Knowl. Manage.*, 2014, pp. 171–180.

[24] K. Jung, W. Heo, and W. Chen, "IRIE: Scalable and robust influence maximization in social networks," in *Proc. 12th Int. Conf. Data Mining*, 2012, pp. 918–923.

[25] Y. Yang, X. Mao, J. Pei, and X. He, "Continuous influence maximization: What discounts should we offer to social network users?," in *Proc. Int. Conf. Manage. Data*, 2016, pp. 727–741.

[26] S. Lei, S. Maniu, L. Mo, and R. Cheng, and P. Senellart, "Online influence maximization," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2015, pp. 645–654.

[27] L. Sun, W. Huang, P. S. Yu, and W. Chen, "Multi-round influence maximization," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2249–2258.

[28] S. Pei, X. Teng, J. Shaman, F. Morone, and H. Makse, "Efficient collective influence maximization in cascading processes with first-order transitions," *Sci. Reports*, vol. 7, 2017, Art. no. 45240.

[29] F. Morone, B. Min, L. Bo, R. Mari, and H. Makse, "Collective influence algorithm to find influencers via optimal percolation in massively large social media," *Sci. Reports*, vol. 6, 2016, Art. no. 30062.

[30] D. M. Centola, "Homophily, networks, and critical mass: Solving the start-up problem in large group collective action," *Rationality Soc.*, vol. 25, no. 1, pp. 3–40, 2013.

[31] A. D. Stefano *et al.*, "Quantifying the role of homophily in human cooperation using multiplex evolutionary game theory," *PLoS One*, vol. 10, no. 10, pp. 1–21, 2015.

[32] D. Centola, J. C. Gonzalez-Avella , V. M. Eguiluz, and M. San Miguel , "Homophily, cultural drift, and the co-evolution of cultural groups," *J. Conflict Resolution*, vol. 51, no. 6, pp. 905–929, 2007.

[33] D. Sohn and N. Geidner, "Collective dynamics of the spiral of silence: The role of ego-network size," *Int. J. Public Opinion Res.*, vol. 28, no. 1, pp. 25–45, 2016.

[34] E. Abbe, S. Kulkarni, and E. J. Lee, "Nonbacktracking bounds on the influence in independent cascade models," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1407–1416.

[35] Wikipedia, "Six degrees of separation," 2012. [Online]. Available: https://en.wikipedia.org/wiki/Six_degrees_of_separation.

[36] H. Chen, H. Yin, W. Wang, H. Wang, Q. V. H. Nguyen, and X. Li, "PME: Projected metric embedding on heterogeneous networks for link prediction," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1177–1186.

[37] B. Perozzi, R. Al-Rfou , and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 701–710.

[38] S. Feng, G. Cong, A. Khan, X. Li, Y. Liu, and Y. M. Chee, "Inf2vec: Latent representation model for social influence embedding," in *Proc. IEEE 34th Int. Conf. Data Eng.*, 2018, pp. 941–952.

[39] S. Cavallari, V. W. Zheng, H. Cai, K. C. Chang, and E. Cambria, "Learning community embedding with community detection and node embedding on graphs," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 377–386.

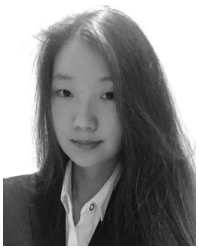
[40] S. Balakrishnan, M. J. Wainwright, and B. Yu, "Statistical guarantees for the EM algorithm: From population to sample-based analysis," *Ann. Statist.*, vol. 45, no. 1, pp. 77–120, 2017.

[41] L. Fu *et al.*, "Percolation degree of secondary users in cognitive networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 10, pp. 1994–2005, Nov. 2012.

[42] L. Jure and K. Andrej, "SNAP Datasets: Stanford large network dataset collection," Accessed: Jun. 2014. [Online]. Available: http://snap.stanford.edu/data

[43] X. Wang and L. Fu, "Acenap Datasets: SJTU large network dataset collection," 2017. [Online]. Available: http://acenap.sjtu.edu.cn/

Xudong Wu received the BE degree in information and communication engineering from the Nanjing Institute of Technology, China, in 2015. He is currently working toward the PhD degree at the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. His research of interests include area of social networking and big data, machine learning and combinatorial optimization.



Luoyi Fu received the BE degree in electronic engineering from Shanghai Jiao Tong University, China, in 2009, and the PhD degree in computer science and engineering from the Shanghai Jiao Tong University, China, in 2015. She is currently an associate professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. Her research of interests include the area of social networking and big data, scaling laws analysis in wireless networks, connectivity analysis and random graphs. She has been a member of the Technical Program Committees of several conferences including ACM MobiHoc 2018-2020, and the IEEE INFOCOM 2018-2020.

Shuaiqi Wang is working toward the BE degre at the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. He is working as a research intern supervised by Dr. Luoyi Fu. His research interests include social networking and big data.

Bo Jiang received the PhD degree in computer science from the University of Massachusetts Amherst, in 2015. He is currently an associate professor in john hopcroft center for computer science with Shanghai Jiao Tong University, China. His research of interests include the area of modeling, analysis and algorithm design for social and computer networks. He has been a member of the Technical Program Committees of several conferences including ACM Sigmetrics 2020-2021.

Xinbing Wang received the BS degree (with hons.) from the Department of Automation, Shanghai Jiaotong University, Shanghai, China, in 1998, the MS degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2001, and the PhD degree, major in the Department of Electrical and Computer Engineering, Minor, Department of mathematics,North Carolina State University, Raleigh, in 2006. Currently, he is a professor with the Department of Electronic Engineering, Shanghai Jiaotong University, Shanghai, China. He has been an associate editor for the *IEEE/ACM Transactions on Networking* and *IEEE Transactions on Mobile Computing*, and member of the Technical Program Committees of several conferences including ACM MobiCom 2012, 2018-2021, ACM Sigmetrics 2021, ACM MobiHoc 2012-2019, and the IEEE INFOCOM 2009-2020.

Guihai Chen received the BS degree from Nanjing University, the ME degree from Southeast University, and the PhD degree from the University of Hong Kong. He visited the Kyushu Institute of Technology, Japan, in 1998, as a research fellow, and the University of Queensland, Australia, in 2000, as a visiting professor. From 2001 to 2003, he was a visiting professor with Wayne State University. He is currently a distinguished professor and a deputy chair with the Department of Computer Science, Shanghai Jiao Tong University, China. He has published more than 200 papers in peer-reviewed journals and refereed conference proceedings in the areas of wireless sensor networks, high-performance computer architecture, peer-to-peer computing, and performance evaluation. He is a member of the *IEEE Computer Society*. He has served on technical program committees of numerous international conferences.