Collective Influence Maximization in Random Networks: Quantifications, Algorithms and Experiments

ABSTRACT

1 INTRODUCTION

With the development and popularity of online social networks (e.g., Facebook, Twitter and Wechat), viral marketing, which leverages the "word-of mouth" effect over social networks, has become a powerful mode for advertising. For example, a company can give free products to some initial users (e.g., u_i) in hope of them sharing the positive comments online and attracting their friends (e.g., u_j) to actually buy the product. Then u_j may in return influence his own friends. Viral marketing is thus modeled as influence diffusion process over social networks. The fundamental problem of viral marketing is to identify the most influential users who can maximize the influence diffusion size. This problem motivates the research on the Influence Maximization (IM) problem in the past decade.

In the seminal work on IM, Kempe *et al.* [1] proposed probabilistic diffusion models to characterize the influence diffusion process since whether one user can influence his friends are considered as random events from sociology literature. The main procedure of solving IM problem proposed in [1] is consisted of two main steps: Firstly, estimating the expected influence diffusion size of seeds based on samplings of networks; Secondly, selecting a set of seed users who can maximize the expected diffusion size. Kempe *et al.* [1] proved that finding an optimal solution for IM under the above procedure is NP-hard and proposed a greedy framework that can provide an approximation guarantee. However, the greedy framework in [1] is well known computationally, it runs for days even over networks with only a few thousand nodes. Then plenty of works on IM proposed every year to improve the efficiency of the greedy framework.

There are two common limitations of current IM solutions: (1) Most of them are restricted to the greedy framework whose performance is restricted to an approximation ratio of $(1 - \frac{1}{e} - \varepsilon)$. Empirically, the performance of greedy greedy even sometimes is not better than the heuristic that selects the users with highest degrees. (2) Those works on the efficiency improvement is usually on the trade off between performance guarantee and efficiency, further lowering the performance of the selected seeds.

To fundamentally improve of the IM especially over nowadays large scale networks with thousands of users, in this paper, we investigate the novel Collective Influence (CI) framework which selects the influential users from the network structural perspective, instead of from maximizing the estimated diffusion size based on sampling. As is well-known, in the social network area, there is a famous "Six Degrees of Separation" theory describing that each user can reach almost all the other users via at most six hops. So that a social network can be modeled as a giant connected component from the structural perspective. Morone *et al.* [2] first showed that the most influential users are those maintain the global connectivity of the network, or equivalently, dismantle the network in disconnected pieces, if removed. With the objective of vanishing the giant connected component, finding the seed users can be mapped to the optimal percolation.

Our work is focused on the Collective Influence (CI) framework over random networks under the consideration that the influence diffusion among users is stochastic, and correspondingly models the networks as weighted graphs. To the best of our knowledge, this is the first work that investigates the CI under the more realistic weighted graphs. In our model, the influence diffusion is thus characterized by the Independent Cascading (IC) model which is one of the most prominent model proposed in [1]. By taking the initial network as a giant connected component of uninfluenced users, we transverse maximizing the influenced size to minimizing the size of the uninfluenced giant connect component. Correlating the power of multiple spreaders' influence with network integrity, we take the influence of a given user and his reachable users on vanishing the giant uninfluenced connected component as the given user's collective influence. We prove that the goal of the CI based IM is to select the users with the highest CIs. Furthermore, we find that the CI framework is scalable when the immediate neighbors are considered in computing CIs. To jointly improve the comprehensiveness in computing CIs and the scalability of the CI framework, we further proposed the embedding based CI solution. We design a network embedding algorithm that learns the low-dimensional representations of users, at the same time, preserves the CIs in the original network represented by the $N \times N$ high-dimensional matrix (N is the size of users). Through the convergence analysis, we prove that the learned low-dimensional representations can exactly capture the original CIs. Then a more efficient CI algorithm which lies on the low-dimensional representations is proposed.

Thus in CI framework, we can compute the CI of users from the from the structural perspective and thus avoid much sampling workload, which is the most time consuming task in classical greedy framework. In addition, the CI framework take advantages of the integrity of users' own influence and the influences of their reachable users in seed selection. The overlapping of different users' influences is coincidentally the root of the NP-hardness in classical IMs which treat each user as independent agent. So that the CI framework can break through the restrictions of greedy framework and achieve near optimal solutions. Our main contributions are summarized as follows.

Quantifications. Through the in-depth study of the interactions among users over weighted graphs, we qualify the condition for minimizing the uninfluenced size and formalize its relation with the CI of users under the IC model. Then we explore the required minimum seed size for realizing the stable condition of the non-existence of giant uninfluenced connected component.

Algorithms. Under the usual settings of IM that the seed set is bounded by a constant K, we first proposed the CI seed selection algorithm that selects users with highest CIs with the qualified condition. For the high performance on large scale networks, we then embed the networks which represented by an $N \times N$ matrix into an $N \times d(d \ll N)$ space and propose a novel embedding CI framework lying on the low-dimensional representations of users.

Experiments. We evaluate the performance of our proposed algorithms on 9 social network datasets (6 real social networks dataset and 3 synthetic networks). The experimental results present that, even just considering the immediate neighbors, the CI algorithm achieves comparative or even better diffusion size comparing with the baselines and costs much less running times. The embedding CI algorithm has much better effectiveness and still costs just a fraction of the time of other baselines. The results also justify our theoretical results on the minimum required seed size for realizing non-existence of giant uninfluenced connected component.

We organize the rest of this paper as follows: In Section 2, we review the background and the relation works on the IM problem. The network and diffusion models, as well as the CI problem is given in Section 3. In Section 4, we give the qualified conditions for minimizing the uninfluenced size and propose the CI based IM algorithms. The network embedding method for preserving CI and the embedding CI algorithm are proposed in Section 5 which is followed by the experimental results in Section 6. We conclude the paper in Section **??**.

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.

The fundamental problem for the IM is to select the optimal seed users who can expectedly maximize the diffusion of a product or an idea over a given social network. Kempe *et al.* [1] first formalized the seed selection as a combinatorial optimization problem for seeking effective solutions.

Definition 2.1. (**Influence Maximization (IM) problem.**) Let a graph $G = (V, E)$ denote a given social network where V and E respectively represent the set of users and their social links, and $I(G, S)$ denote the size of the users that can expectedly be influenced by a set S of seed users. The objective of the IM problem is to select the seed set S with the size being K from V to maximize $I(G, S)$, i.e.,

$$S = S : \arg\max_{S \subseteq V} I(S, G), |S| = K. \tag{1}$$

To quantify the expected influence $I(G, S)$, Kempe *et al.* [1] proposed the Independent Cascading (IC) and the Linear Threshold (LT) model to describe the influence diffusion process among users. In the two classical models, each edge $(i, j) \in E$ is correlated with a Bernouli random variable that follows $\mathcal{B}(w_{ij})$. That is user u_i can successfully influence user u_j with the probability being w_{ij}. Kempe *et al.* [1] further proved that the IM problem is NP-hard under both IC and LT models.

With the stochastic diffusion process, most early works on the IM (e.g. [1][3]) estimate the influence $I(S, G)$ through generating sufficient times of Monte-Carlo simulations and then taking the average. However, this method is known to be computationally expensive. It costs an $\Omega(K|V||E| \cdot \text{poly}(\epsilon^{-1}))$ time complexity for conducting a greedy solution with an approximation ratio of $(1 - \frac{1}{e} - \epsilon)$ [3]. One of the most efficient IM frameworks currently is the Reverse Reachable Sets (RR-sets) framework [4] [5][6] whose main idea is briefly reviewed as follows. Associate with each edge $(i, j) \in E$ a biased coin that lands heads with probability w_{ij}, the RR-sets framework first flip each coin to generate graph samples. If one coin (i, j) lands head in a sample, the corresponding edge (i, j) is live, i.e, u_i can successfully influence u_j. Then the RR-sets framework randomly selects a certain number of nodes, and from each of them (e.g., u_j), starts the reverse breath first search via live edges and includes all searched nodes into corresponding RR-set (e.g., R_j). If u_i in the RR-set R_j, say u_i covers u_j, u_i is considered that can influence u_j successfully in diffusion. Thus selecting the seed users who can maximize $I(G, S)$ is equivalent to selecting those who can cover the most number of RR-sets. The RR-sets framework costs a time of $O(K(|V| + |E|) \log n/\varepsilon^2)$ for achieving a $(1 - \frac{1}{e} - \varepsilon)$-approximate solution [5][6] .

In addition, there are some works study other variants of the IM problem. Nguyen *et al.* [4] considered the outward influence to maximize the reward $I(G, S) - |S|$. The heterogeneous seeding cost of different users is studied in [7][8]. Furthermore, [9][10] focus on adaptively selecting the seeds based on the influence diffusion feedbacks started by previously selected seeds. In summary, the current IM solutions noted above are all restricted to the greedy framework whose performance is limited to the $(1 - \frac{1}{e} - \varepsilon)$ approximation ratio.

Due to the great demand of the high performance of IM in various applications, a natural question is: Is there a new framework that can break through the restriction of the greedy framework, as well as achieve higher efficiency comparing with existing sample-based influence estimation methods? In this paper, we study the collective influence, which is a new IM framework that selects seeds from the structural perspective to overcome the two main problems of IM.

3 PROBLEM FORMULATION

In this paper, we study the Collective Influence (CI) based IM problem in random networks.

3.1 Network and diffusion models.

We model a given social network by a weighted graph denoted by $G = (V, E, W)$, where V is the set of users, E is the set of their social links, and $W : E \longrightarrow (0, 1)$ is a function that assigns each edge an weight to quantify the succeeding probability of the influence diffusion via it. With the weighted graph, we adopt the Independent Cascading (IC) model to model the influence diffusion process over social networks.

Definition 3.1. (**Independent Cascading (IC) model.**) Initially, a given set S of selected seed users is influenced at the beginning and they start the influence diffusion process in discrete steps. In each step, each newly influenced user u_i remains influenced until the end, and has a single chance to activate his uninfluenced neighbor u_i successfully with probability w_{ij}, which is equal to the weight of the edge in the network. The influence diffusion process stops when there is no new users get influenced. Then the influence $I(G, S)$ of seed set S is the expected size of influenced users at the end of the above influence diffusion process.

Under the IC model, given a a set of seed users S, the probability of user u_j being influenced, say $I(S, u_j)$ can be given by

$$I(S, u_j) = 1 - \Pi_{u_i \in \Gamma(j)} \left(1 - w_{ij} I(S, u_i)\right),$$

where $\Gamma(j)$ denotes the set of the neighbors of u_j. Then the influence of S over network G is formalized as

$$I(S, G) = \sum_{u_j \in V} I(S, u_j).$$

The objective IM is to select the seed set S with size K in hope to maximize the expected influence $I(S, G)$.

3.2 Problem formulation

In this paper, we propose to solve the IM problem under IC model based on the Collective Influence (CI) framework. The main idea of the CI is to select the users from a given network, if removed, can vanish the giant connected component in the network. For the network $G = (V, E, W)$, its topology is represented by an $N \times N$ matrix A where $A_{iJ} = 1$ where there is an edge from user u_i to u_j. Each element A_{ij} is also associated with a weight w_{ij}. Initially, all the users belong to the giant connected component of uninfluenced users. Then an influence diffusion process starts from a set of seed users and evolves as the IC model. Let $Q(S, G)$ denote the size of the giant connected component of uninfluenced users after seeding a set S of users, we transfer the objective of IM, i.e., maximizing $I(S, G)$ to the minimization of $Q(S, G)$.

Definition 3.2. (**Collective influence based IM problem.**) Given a social network $G = (V, E, W)$, where all the users remain uninfluenced initially, the goal of the collective influence based IM problem is to select a set of seed users from V who can minimize the size of the giant connected component of uninfluenced users $Q(S, G)$ under the IC influence diffusion model.

In Section 4, we will present our methodology for minimizing the $Q(S, G)$ with the collective influence framework under IC model. Furthermore, in Section 5, we proposed a novel embedding based CI framework which embed the network represented by the $N \times N$ matrix into a $N \times d$ ($d \ll N$) space. The aim of considering the embedded representation is to seek the more efficient solution for seed selection in the CI framework, with the help of the low-dimensional representing vectors of users.

4 COLLECTIVE INFLUENCE FRAMEWORK UNDER IC MODEL

The aim of the seed selection in the CI framework is to minimize the size of the uninfluenced users $Q(S, G)$. Intuitively speaking, the minimum value of the $Q(S, G) = 0$ when all the users in G are influenced. In this section, we first explore the stable condition for $Q(S, G) = 0$ under the IC influence diffusion model, and then present the seed selection algorithm to select the users who can minimize $Q(S, G)$.

4.1 Condition For Minimizing Uninfluenced Size

We characterize the stable condition that the minimum value of $Q(S, G)$ satisfies from the node, edge and network levels, respectively. On the *node* level, let v_i denote the probability that the node u_i is influenced, then $Q = (S, G) = 0$ is conditioned on $v_i = 1 (\forall u_i \in V)$. On the *edge* level, we need to consider the question that for an edge (i, j), if one endpoint u_j is not influenced, how about another endpoint u_i? Let v_{ij} denote the probability that u_i is influenced while u_j remains uninfluenced. Based on the definition of the IC model, v_{ij} can be formulated as:

$$v_{ij} = 1 - \Pi_{k \in \Gamma(i) \backslash j} (1 - w_{ki} v_{ki}), \tag{2}$$

where $\Gamma(i) \backslash j$ denotes the neighbors of u_i excluding u_j. From Eqn. (2), we can see that the formulation of $v_{ij} (\forall (i, j) \in E)$ has the iterative property. That is v_{ij} and $v_{ki} (k \in \Gamma(i) \backslash j)$ iteratively influences each other. Jointly consider the v_{ij} via all the edges in a network, on the *network* level, the stable condition for $Q(S, G) = 0$ can be characterized by Jacobian matrix $\mathbf{M}$ of $v_{ij} (\forall (i, j) \in E)$, i.e.,

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}_{klij} & \cdots \\ \vdots & \ddots \end{pmatrix}, \mathbf{M}_{klij} = A_{ij} A_{kl} \delta_{il} (1 - \delta_{kj}) \frac{\partial v_{ij}}{\partial v_{kl}}, \tag{3}$$

where the term $A_{ij} A_{kl} \delta_{il} (1 - \delta_{kj})$ controls that $\mathbf{M}_{klij}$ probably has non-zero value only when (k, l) and (i, j) forms a non-backtracking path, that is, both edges (k, l) and (i, j) exist in the network and $k \neq j, l = i$.

With the Jacobian matrix M of $v_{ij} (\forall (i, j) \in E)$ in Eqn. (3), Lemma 4.1 presents the stable condition for $Q(S, G) = 0$ under the IC influence diffusion model.

Lemma 4.1. *For the influence diffusion under the IC model, the stable condition for* $Q(S, G) = 0$ *is that the leading eigenvalue* λ *of the Jacobian matrix* M *given in Eqn. (3) is smaller than* 1.

Proof. We use a function $f(\cdot)$ to denote the iterative property among the formulations of $v_{ij} (\forall (i, j) \in E)$, and use a vector $\mathbf{v}$ to denote all the variables v_{ij}. From the intuition of convergence, the solution is assigned to $\mathbf{v}$ until $\mathbf{v} = f(\mathbf{v})$.

When the value of $\mathbf{I}$ comes to the stable solution $\mathbf{v}^*$ with an error denoted by ϵ_0, the function $f(\cdot)$ can be approximated linearly by its Jacobian matrix $\mathbf{M}$ whose element is $M_{ij} = \frac{\partial f_{ij}}{\partial v_{ij}} |_{v_{ij}=0}, \forall (i, j) \in E$, we have

$$f(\epsilon_0) = \mathbf{M} \epsilon_0 + o(\epsilon_0).$$

After repeating the iterations for l times, the inner product of the l-th error vector can be formalized as

$$\langle \epsilon_l, \epsilon_l \rangle = (1 + o(1)) \epsilon_0^T (\mathbf{M}^T \mathbf{M})^l \epsilon_0. \tag{4}$$

In Eqn. (4), the inner product of errors can be approximated by the power of the leading eigenvalue of $\mathbf{M}$. When $l \longrightarrow \infty$, if the leading eigenvalue which is denoted by λ satisfies $\lambda < 1$, we have

$$\lim_{l \to \infty} \langle \epsilon_l, \epsilon_l \rangle = \lim_{l \to \infty} \epsilon_0^T (\mathbf{M}^T \mathbf{M})^l \epsilon_0 = \lim_{l \to \infty} \lambda^{2l} \langle \epsilon_0, \epsilon_0 \rangle = 0.$$

Thus after the enough rounds of the iterations, the error vector $\langle \epsilon_l, \epsilon_l \rangle$ will eventually come to 0, meaning that the solution of the function $f(\mathbf{v})$ is stable.

Now we move to the case that the leading eigenvalue $\lambda > 1$. In this case, we decompose the Jacobian matrix $\mathbf{M}$ into its Jordan canonical form $\mathbf{J}$ which contain the eigenvalues with value $\lambda > 1$ on the diagonal. Then such diagonal elements will become λ^l in $\mathbf{J}^l$ and will diverge when $l \longrightarrow \infty$. As a result, the inter product of the errors $\langle \epsilon_l | \epsilon_l \rangle$ will increase continuously until ϵ_l leaves the neighborhood in which the derivative is taken to approximate function $f(\cdot)$. Thus $\mathbf{v}$ will not approach to $\mathbf{v}^*$.

Based on the analysis above, we can conclude that the condition $v_{ij} = 0 (\forall (i,j) \in E)$ is stable as long as the leading eigenvalue $\lambda < 1$ since a variable $\mathbf{v}$ with a small error will eventually come to the stable solution after enough time of the iterations. Thus we end the proof for the Lemma 4.1. □

Lemma 4.1 shows that the stable condition of $Q(S,G) = 0$ is the leading eigenvalue of $\mathbf{M}$ smaller than 1. We proceed to explore the relation between the leading eigenvalue λ and the nodes in the network. This relation is further utilized to determine which users should be seeded for minimizing the uninfluenced size $Q(S,G)$.

4.2 Solutions For Minimizing Uninfluenced Size

4.2.1 Formulation of the leading eigenvalue λ. We computed the leading eigenvalue λ based on the power method. Given a $2|E|$-dimensional initial vector w_0, let $\langle \mathbf{w}_l, \mathbf{w}_l \rangle = \langle \mathbf{w}_0 (\mathbf{M}^l)^T, \mathbf{M}^l w_0 \rangle = \lambda^{2l} \langle \mathbf{w}_0, \mathbf{w}_0 \rangle$, we have

$$\lambda = \lim_{l \to \infty} \left[\frac{\langle \mathbf{w}_l, \mathbf{w}_l \rangle}{\langle \mathbf{w}_0, \mathbf{w}_0 \rangle} \right]^{\frac{1}{2l}}. \tag{5}$$

Then we have element $\mathbf{M}_{klij} = A_{ij} A_{kl} \delta_{il} (1 - \delta_{kj}) w_{kl}$. According to the IC model defined in Definition 3.1, we have the derivative $\frac{\partial v_{ij}}{\partial v_{kl}} = w_{kl}$ where w_{kl} is the probability that u_k can influence u_i as described in IC model. Now we move to the derivation of the inner product $\langle \mathbf{w}_l, \mathbf{w}_l \rangle$ given the initial vector $\mathbf{w}_0 = \mathbf{1}$.

When $l = 1$, the first order right vector is

$$\mathbf{w}_1 \rangle_{ij} = \sum_{kl} \mathbf{M}_{ijkl} | \mathbf{w}_0 \rangle_{kl} = A_{ij} \sum_{jk, k \neq i} A_{jk} w_{jk}. \tag{6}$$

$$_{ij} \langle \mathbf{w}_1 = \sum_{kl} \rangle \mathbf{w}_0 |_{kl} \mathbf{M}_{klij} = A_{ij} (d_i - 1) w_{ij}. \tag{7}$$

Then the first order inner product is

$$\langle \mathbf{w}_1, \mathbf{w}_1 \rangle = \sum_{ij} A_{ij} (d_i - 1) w_{ij} \left(\sum_{jk, k \neq i} A_{jk} w_{jk} \right). \tag{8}$$

We continue to derive the closed form expression of the inner product $\langle \mathbf{w}_l, \mathbf{w}_l \rangle = \langle \mathbf{w}_0 | (\mathbf{M}^l)^T, \mathbf{M}^l \mathbf{w}_0 \rangle$ by induction. Suppose the l-th order right vector has the form

$$\mathbf{w}_l \rangle_{ij} = A_{ij} \sum_{k: d(j,k) = l, i \notin p(j,k)} \left(\Pi_{(x,y) \in p(j,k)} w_{xy} \right), \tag{9}$$

where $p(j,k)$ is the path with length l between u_j and u_k. From Eqn. (9), we can see that Eqn. (6) is a special case of Eqn. (9) when $l = 1$. Then for $(l+1)$ where $\mathbf{w}_{l+1} \rangle = \mathbf{M} \mathbf{w}_l \rangle$, we have

$$\begin{aligned} \mathbf{w}_{l+1} >_{ij} &= \sum_{kl} \mathbf{M}_{ijkl} | \mathbf{w}_l >_{kl} \\ &= A_{ij} \sum_{k: d(j,k) = l+1, i \notin p(j,k)} \left(\Pi_{(x,y) \in p(j,k)} w_{xy} \right). \end{aligned}$$

Accordingly, the formulation of the l-th order left vector is

$$_{ij} \langle \mathbf{w}_l = \sum_{k: d(k,i) = l-1, j \notin p(k,i)} (d_k - 1) \left(\Pi_{(x,y) \in p(k,i)} w_{xy} \right) w_{ij}.$$

Therefore, in computing the inner product $\langle \mathbf{w}_l, \mathbf{w}_l \rangle$, the nodes that in a ball of radius l around the user u_i are taken into account, and the closed form of the inner product can be computed as

$$\langle \mathbf{w}_l, \mathbf{w}_l \rangle = \sum_i (d_i - 1) \sum_{j: d(i,j) = l} \left(\Pi_{(x,y) \in p(i,j)} w_{xy} \right). \tag{10}$$

From Eqn. (**??**), the inner product $\langle \mathbf{w}_l, \mathbf{w}_l \rangle$ is the sum of the polynomial $(d_i - 1) \sum_{j: d(i,j) = l} \left(\Pi_{(x,y) \in p(i,j)} w_{xy} \right)$ of all the nodes in the network. Since $\lambda = \lim_{l \to \infty} \left[\frac{\langle \mathbf{w}_l, \mathbf{w}_l \rangle}{\langle \mathbf{w}_0, \mathbf{w}_0 \rangle} \right]^{\frac{1}{2l}}$, minimizing the leading eigenvalue of the Jacobian matrix $\mathbf{M}$ is equivalent to selecting the users if removed can minimize the l-th power $\langle \mathbf{w}_l, \mathbf{w}_l \rangle = \sum_i (d_i - 1) \sum_{j: d(i,j) = l} \left(\Pi_{(x,y) \in p(i,j)} w_{xy} \right)$.

4.2.2 The minimum required seed size for $Q(S,G) = 0$. Given that the stable condition of $Q(S,G) = 0$ is the leading eigenvalue of λ the Jacobian matrix $\mathbf{M}$ smaller than 1, a natural question is how many seed users are needed to make $\lambda < 1$? Initially, all the users remain uninfluenced, then the CI based IM select a set of seed users and remove them from the giant connected component of the uninfluenced users until the the giant connected component vanish. Here, we consider the case that $l = 0$ in deriving the lower bound of required seed user size for $Q(S,G) = 0$. When $l = 0$, $\langle \mathbf{w}_l, \mathbf{w}_l \rangle = \sum_i (d_i - 1)$ and the leading eigenvalue λ can be written as $\lambda \frac{\sum_i d_i (d_i - 1)}{\sum_i d_i}$. The most effective way for reducing the value of $\langle \mathbf{w}_l, \mathbf{w}_l$ is removing the nodes with the highest degrees. Let q_c denote the fraction of the nodes required to be seeded for $Q(S,G) = 0$, then q_c satisfies that

$$\lambda = \frac{\sum_i d_i (d_i - 1) n_i}{\sum_i d_i} = \frac{(1 - q_c)(\overline{d}^{2\prime} - \overline{d}^{\prime})}{\overline{d}} = 1, \tag{11}$$

where $\overline{d}$ is the mean degree in the initial network, $\overline{d}^{\prime}$ and $\overline{d}^{2\prime}$ are the mean degree and square mean degree of the network after removing the $|V| q_c$ nodes. In this paper, we consider the most representative network models, i.e., ER, power-law degree distribution and stochastic block model (SBM) network to derive the value q_c.

LEMMA 4.2. *In an ER graph where each pair of nodes are connected at random with a given probability p and the degree distribution is approximated by $P(d = x) = \frac{(\lambda)^x e^{-\lambda}}{x!} (\lambda = np)$ under the large nodes size n. The value of the required fraction of users q_c for $Q(S,G) = 0$ can be scaled as $q_c = \Theta\left(\frac{\lambda - 1}{\lambda}\right)$.*

LEMMA 4.3. *Given a network with the power-law degree distribution, i.e., $P(d = x) = a \cdot x^{-\gamma}$, the required fraction of seed users q_c for $Q(S,G) = 0$ can be scaled as*

$$q_c = \begin{cases} \Theta(1) \quad (0 < \gamma < 1) \\ \Theta\left(\left(\frac{d_{max}}{d_{min}} \right)^{1-\gamma} \right) \quad (1 < \gamma < 2) \\ o(1) \quad (\gamma > 2), \end{cases}$$

where d_{max} and d_{min} respectively denotes the maximum and minimum degree in the initial network.

LEMMA 4.4. *Given a network characterized by the SBM which has C communities and the nodes belonging to different communities* *connect at random with probability q, the required fraction of seed users q_c for $Q(S,G) = 0$ can be scaled as $q_c = \Theta\left(1 - (1-q)^{(2q-1)n}\right)$.*

4.3 Seed Selection Algorithm For Minimizing Uninfluenced Size

In most cases of the IM problem, the size of seed users is preset as a constant $K(K \ll N)$ due to the limited budget for seeding users. Thus we proceed to give the algorithm for selecting K seed users who can minimize the uninfluenced size $Q(S,G)$ at the most. Theorem 4.5 presents the rule for selecting under the conclusion that minimizing the uninfluenced size $Q(S,G)$ is equivalent to minimizing the value of $\langle \mathbf{w}_l, \mathbf{w}_l \rangle$.

Theorem 4.5. *In the Collective Influence (CI) framework defined under the IC influence diffusion model, to maximize the influenced size $I(S,G)$, as well as minimize the uninfluenced size $Q(S,G)$, it is optimal to iteratively select the users with the maximum collective influence CI_l, i.e.,*

$$CI_l(i) = (d_i - 1) \sum_{j:d(i,j)=l} \left(\Pi_{(x,y)\in p(i,j)} w_{x,y}\right), \tag{12}$$

where $e \in p(i,j)$ denotes the edges on the shortest path from user u_i to u_j over the network.

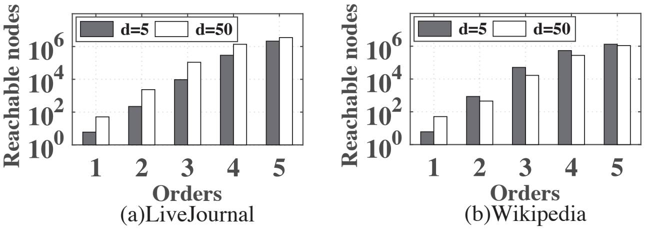
Proof. As presented in Lemma 4.1, the stable condition for $Q(S,G)$ is that the leading eigenvalue of the Jacobian matrix $\mathbf{M}$ is smaller than 1. Recalling Section 4.2, the leading eigenvalue λ can be computed by the power method, i.e., $\lambda = \lim_{l\to\infty}\left[\frac{\langle \mathbf{w}_l, \mathbf{w}_l\rangle}{\langle \mathbf{w}_0, \mathbf{w}_0\rangle}\right]^{\frac{1}{2l}}$, and minimizing the leading eigenvalue λ is equivalent to minimizing the value of $\langle \mathbf{w}_l, \mathbf{w}_l \rangle$. Furthermore, the formulation $\langle \mathbf{w}_l, \mathbf{w}_l \rangle = \sum_i (d_i - 1) \sum_{j:d(i,j)=l} \left(\Pi_{(x,y)\in p(i,j)} w_{xy}\right)$ in Eqn. (10) suggests that $\langle \mathbf{w}_l, \mathbf{w}_l \rangle$ can be taken as the sum of the collective influence $(d_i - 1) \sum_{j:d(i,j)=l} \left(\Pi_{(x,y)\in p(i,j)} w_{xy}\right)$ of all the users in the network. Thus the most effective way to minimize the leading eigenvalue is seeding the users with the maximum value of $(d_i - 1) \sum_{j:d(i,j)=l}$ $\left(\Pi_{(x,y)\in p(i,j)} w_{xy}\right)$. □

We call the value of $CI_l(i) = (d_i - 1) \sum_{j:d(i,j)=l} \left(\Pi_{(x,y)\in p(i,j)} w_{x,y}\right)$ as the collective influence of user u_i since it takes the influence of the users that u_i can reach through l hops as u_i's own influence. Based on the seed selection rule given by Theorem 4.5, Algorithm 1 presents the CI based seed selection for selecting K seed users in hope to maximize the influence diffusion size. With the input being the network $G = (V, E, W)$, Algorithm 1 iteratively selects the seed users with the largest CI. In each iteration, the CIs of the remaining users are first updated by removing the seed user selected in last iteration from the network. Thus the user with the largest updated CI, which is given by Theorem 4.5, among all the remaining users is selected as the new seed. The Algorithm 1 continues until the size of the seed set meets the preset constant K.

Performance of Algorithm 1. Algorithm 1 has an approximation ratio of $\Theta(\frac{1}{N})$ since in computing the CI, we do not the take into account of the cycles in the paths with length l. In the social networks, when considering the paths with length l, the contributions of the paths containing cycles can be scaled as a fraction of $\Theta(\frac{1}{N})$ [3]. In addition, It is noteworthy that, for computing the CI_l value of a given user, Algorithm 1 needs to traverse all the edges and nodes that u_i can reach within l hops. However, referring to the famous "Six Degrees of Separation" theory, any user can reach almost all the other users through a few hops. Also, Fig. 1 which presents the number of the users that a given user can reach over the two famous social networks LiveJournal and Wikipedia, further justifies the theory. From Fig.1, 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 for computing CI of a user. Thus the CI based seed selection algorithm costs a complexity of $O\big(K(\log n + |V|(|E| + |V|))\big)$. The high complexity of CI framework motivates us to seek for more efficient solutions to improve its scalability. In Section, we will present a novel network embedding based CI framework which embeds the original network represented by an $N \times N$ matrix into an $N \times d\,(d \ll N)$ space. With the help of the low-dimensional representation of the network, we further proposed a network embedding based CI algorithm for selecting seed users using polynomial time.

```
// Collective Influence (CI) based seed selection algorithm
Input: Graph $G = (V, E)$, the weight of edges $w_{ij}$, truncating
       length $l$ of CI, the size of seed set $K$;
Output: The set $S$ of seed users with size $K$;
$S = \emptyset$ ;
while $|S| < K$ do
    Remove $S^*$ from $G$;
    Update $CI_l(i) \forall u_i \in V \backslash S$;
    $S^* = \arg\max_{u \in V \backslash S} CI_l(i)$;
    $S = S \cup S^*$;
end
return $S$.
```

Algorithm 1: Collective Influence (CI) based seed selection algorithm.



(a)LiveJournal (b)Wikipedia

Figure 1: The number of reachable nodes vs. hops

5 EMBEDDING BASED COLLECTIVE INFLUENCE FRAMEWORK

As stated previously, it is difficult to purse the collective influence framework on current large scale social networks. Therefore, in this section, we propose a novel embedding framework for conducting the collective influence framework under IC model proposed in Section 4 more efficiently. Traditionally, a network/graph is represented by a $N \times N$ matrix, and each user/node is represented by a N-dimensional vector. Our main idea is first learning the d-dimensional $(d \ll N)$ representations, which can capture the collective influence derived in Theorem 4.5, for the users in a given network, and then designing a new seed selection algorithm based on the learned low-dimensional representations for pursing the CI based IM efficiently.

The classical network embedding technology is mainly consisted of two steps: (1) Generating the context that contains the users can be reached by a given user; (2) Learning the low-dimensional representations of the users that can maximize the likelihood of the context. Corresponding to the embedding based CI framework, in this section, we first present how to generate the context of each user $u_i \in V$ that contains the users u_i can reach within a few hops, at the same time reflects different reaching probability of u_i to different users which is described by the term $\left(\Pi_{(x,y)\in p(i,j)} w_{x,y}\right)$ in Theorem 4.5 (Section5.1). Thus we call the context generated in our embedding based CI framework as the collective influence context. Then we proceed to carry out the maximizing likelihood estimation method to learn the users' low-dimensional representations that can maximize the likelihood of the collective influence context (Section5.2). Section 5.3 presents the convergence analysis of the proposed representation learning method. In Section 5.4, we utilize the learned low-dimension representations of the users to generate the CIs given by Theorem 4.5, and then conduct the efficient seed selection based on the CIs.

5.1 Generating Collective Influence Context

In generating the collective influence context, depending on the diffusion process defined under the IC model, we utilize a random walk approach that samples the as many as possible users that are potentially influenced by a given source user. The advantage of the random walk approach in generating the influence context is that it can take into account the stochasticity brought by the IC model in influence diffusion. To the best of our knowledge, this is the first method that generates the context of users under the IC influence diffusion model.

Given a source user u_i in the network, the random walk randomly chooses one neighbor u_j to visit with the probability proportional to the weight w_{ij}. When the random walk arrives at the the user u_j, it then chooses one of u_j's neighbor to visit next. Particularly, we allow that the random walk can go back to u_i from u_j. The walk from u_i to u_j (resp. u_j to u_i) is a sample of the influence from u_i to u_j (resp. u_j to u_i). One random walk stops when the length of the sequence of the visited nodes meets a preset threshold L. We conduct r random walks of the length L starting from every user. Note that due to the bias of the transition among users in the random walk respect to weights of edges, if the reaching probability from u_i to user u_m is larger the that to u_j, u_i and u_m may coexist in more random walk sequences than u_i and u_j. Thus the random walk under the IC model can not only capture the reachable users but also distinguish the influences among users, further justifying our usage of it. In the sequel, the extract collective influence context from the $r|V|$ random walk sequences.

The collective influence context of user u_i, say $\mathbb{C}_i$, is consisted of the users that line up behind u_i in all the $r|V|$ random walk sequence.. For example, let $r_1 = \{u_1, u_2, \dots, u_L\}$ be one of the random walk sequences starting from user u_1, we have the users $u_2, \dots, u_L$ are added into $\mathbb{C}_1$. In addition, given one sequence $r_i = \{u_i, u_1, u_j \dots, u_L\}(i \neq 1)$ not started from u_1, we say that $u_j \dots, u_L$ are also added into $\mathbb{C}_1$. We believe that the larger the influence from u_i to u_j under the IC model, the more times that u_j will appear in $\mathbb{C}_i$.

5.2 Low-dimensional Representation Learning

In the generated collective influence context $\mathbb{C}_i(\forall u_i \in V)$, the larger the influence from user u_i to user u_j, the more times u_j will appear in the context $\mathbb{C}_i$. We use coexisting times, say s_{ij}, to denote the times that user u_i appears in the context $\mathbb{C}_i$. Then a set $\mathbf{Y}_i$ of the tuples which contain the users and their coexisting times with the user u_i are extracted from the context $\mathbb{C}_i$, i.e.,

$$\mathbf{Y}_i = \{(u_1, s_{i1}), (u_2, s_{i2}), \dots, (u_{|Y_i|}, s_{i|Y_i|})\}, \tag{13}$$

where the $S_i = |Y_i|$ is the number of users appearing in the context $\mathbb{C}_i$ for at least once.

The goal of the representation learning is to learn the low-dimensional representations of the users that can maximize the likelihood of the tuples set $\mathbf{Y}_i = \{(u_1, s_{i1}), (u_2, s_{i2}), \dots, (u_{|Y_i|}, s_{i|Y_i|})\}$ $(\forall u_i \in V)$. To be more precise, let θ_i denote the representation of user u_i as the source in the influence diffusion, x_i be the representation of u_j as the target. Here, we define the x_i as the preset vector with $\mathbf{X}\mathbf{X}^T = \mathbf{I}$. Given the times s_{ij} that u_i appears in the context $\mathbb{C}_i$, the goal of the representation learning is to learn the estimator of the source vector $\hat{\theta}_i$ as close to θ_i, which maximizes the likelihood of $\mathbf{Y}_i$, as possible.

Now we propose a probabilistic model with the parameter θ_i which can be inferred via the maximum likelihood estimating. Since the aim of the representations learning is for capturing the collective influences, we formulate the coexisting time as the result of a bootstrapping approach over the low-dimensional representations, i.e., the coexisting time s_{ij} and the representations θ_i and x_j are linked by the equation

$$s_{ij} = \langle \theta_i, x_j \rangle + v_i,$$

where $v_i \sim \mathcal{N}(0, \sigma^2)$ is assumed as the observation noise in the random walk generating method. Due to the randomness in the random walk, the coexisting time between a pair of users in the collective influence context may not be strictly proportional to influence among them over the given network. It is more reasonable that the coexisting times s_{ij} is a random variable which is distributed around the influence that is represented by the $\langle \theta_i, x_j \rangle$ in the low-dimensional space. With these notations, we can write the distribution of the coexisting times of u_j in the context $\mathbb{C}_i$ as:

$$s_{ij}|x_j \sim \mathcal{N}(\langle \theta_i, x_j \rangle, \sigma^2).$$

We further model the coexisting times of all the users in the context $\mathbb{C}_i$ as the following mixture model:

$$p_{\theta_i}(\mathbf{Y}_i) = \frac{1}{S_i} \sum_{u_j \in \mathbf{Y}_i} \mathcal{N}(\langle \theta_i, x_j \rangle, \sigma^2), \ (|\mathbf{Y}_i| = S_i). \tag{14}$$

Given the above mixture model, we propose to learn the representation θ_i based on the mixture of regressions model where we can model the coexisting times as a likelihood and estimate the representation θ_i by maximizing the likelihood. Furthermore, in such mixture of regressions model, we introduce a hidden variable

$\mathbf{Z}_i \in \{z_{ij}\}_{j=1}^{S_i}$ as an indicator of the underlying mixture component: say $j = n$ when one coexisting time s is generated from the distribution

$$s|x_n = s \sim \mathcal{N}(\langle\theta_i, x_n\rangle, \sigma^2).$$

Based on the mixture of regressions model, we formulate the representations learning of user u_i as the estimation of the parameter θ_i. To be more precise, the objective is to learn the optimal estimation θ_i^* that maximizes the likelihood of the coexisting times of all the users in the collective influence context $\mathbb{C}_i$. Next, we proceed to present the solutions to the likelihood maximization problem for learning the representations.

5.2.1 *EM solutions for representation learning.*

Our solutions for the Maximum Likelihood Estimation (MLE) is based on the Expectation-Maximization (EM) algorithm which is considered as one of the most effective approaches for solving the MLE. Here, we first review the procedures of the general EM algorithm and then present the EM based solutions for the representations learning. The main idea of the EM algorithms is iteratively maximizing the log likelihood to obtain the new parameters, and then reevaluating the value of the log likelihood at the new parameter. The updating procedure is as follows.

EM updates. Given the updated representation θ_i^{t-1} obtained at the $(t-1)$-th iteration, the t-th iteration of the EM algorithm is consisted of the following E (expectation)-step and M(maximization)-step:

- **E-step:** Computing the log likelihood under the existing parameter θ_i^{t-1};
- **M-step:** Updating the parameter θ_i by maximizing the log likelihood, i.e.,

$$\theta_i^t = \arg \max_{\theta_i' \in \mathbb{R}^d} Q\left(\theta_i' | \theta_i^{t-1}\right). \tag{15}$$

The EM algorithm iteratively conducts the above two steps until the parameter converges or the iterating times meet a preset threshold. To obtain the representation θ_i based on the EM algorithms, we now move to illustrate how to formulate the log likelihood and compute the maximizer in each iteration.

With the proposed mixture of regressions model, we assume that the coexisting times of each user in the tuples set $\mathbf{Y}_i$ is drawn i.i.d. from the mixture probability density in Eqn. (14). Under such assumption, the log likelihood function $Q(\theta_i'|\theta_i)$ is defined as:

$$Q\left(\theta_i'|\theta_i\right) = -\frac{1}{S_i}\sum_{j=1}^{S_i}\left(\sum_{j'=1}^{S_i} P(z_{ij'}|s_{ij})\left(s_{ij} - \langle\theta_i', x_{j'}\rangle\right)^2\right). \tag{16}$$

In Eqn. (16), $P(z_{ij'}|s_{ij})$ is the conditional probability that the coexisting time s_{ij} belongs to the user $u_{j'}$, and is defined as:

$$P(z_{ij'}|s_{ij}) = \exp\left(-\frac{(s_{ij} - \langle\theta_i, x_{j'}\rangle)^2}{2\sigma^2}\right)\left[\sum_{j'=1}^{S_i} \exp\left(-\frac{(s_{ij} - \langle\theta_i, x_{j'}\rangle)^2}{2\sigma^2}\right)\right]^{-1}. \tag{17}$$

From the formulation of Eqn. (17), we can see that the conditional probability $P(z_{ij'}|s_{ij})$ is the weight of the item $\left(s_{ij} - \langle\theta_i', x_{j'}\rangle\right)^2$ in computing the log likelihood. In order to simplify the formulation, we use $w_{\theta_i}(x_{j'}, s_{ij})$ to denote the conditional probability $P(z_{ij'}|s_{ij})$. Then the Eqn. (16) becomes

$$Q\left(\theta_i'|\theta_i\right) = -\frac{1}{S_i}\sum_{j=1}^{S_i}\left(\sum_{j'=1}^{S_i} w_{\theta_i}(x_{j'}, s_{ij})\left(s_{ij} - \langle\theta_i', x_{j'}\rangle\right)^2\right). \tag{18}$$

As mentioned before, the M-step in the EM algorithm is to maximize the log likelihood function in Eqn. (18) to obtain the updated parameter θ_i'. Thus we define the EM operator as $M_{\mathbf{Y}_i}(\theta_i) = \arg\max_{\theta_i' \in \mathbb{R}^d} Q(\theta_i'|\theta_i)$. Given the formulation of the $Q(\theta_i'|\theta_i)$ as shown in Eqn. (18), we have

$$M_{\mathbf{Y}_i}(\theta_i) = \frac{\sum_{j=1}^{S_i}\sum_{j'=1}^{S_i} w_{\theta_i}(x_{j'}, s_{ij}) x_{j'} s_{ij}}{\sum_{j=1}^{S_i}\sum_{j'=1}^{S_i} w_{\theta_i}(x_{j'}, s_{ij})\langle x_{j'}, x_j\rangle}. \tag{19}$$

5.2.2 *Representation Learning Algorithm.*

Based on the analysis above, we are ready to give the algorithm of the EM based representation learning in Algorithm 2. We first compute the variance σ^2 in the model based on the r random walks starting from each user. Given the user-coexisting times tuples set $\mathbf{Y}_i (1 \le i \le n)$, the EM algorithm is conducted to iteratively update the estimator of the representation θ_i by maximizing the log likelihood of the coexisting times.

```
// Representation learning $(V, \mathbf{Y}, s)$
Generating Collective Influence Context(G, R, l)
 (Algorithm 3);
Input: Users set: $V$; tuples set: $\mathbf{Y}$; coexisting times $s$;
Output: Representation of users: $\theta_i (1 \le i \le n)$
for $1 \le j \le n$ do
    for $1 \le i \le n$ do
        Calculate the variance of $\hat{\sigma}_{i,j}^2$ over $r$ sequences
        $\hat{\sigma}_{i,j}^2 = \frac{1}{R}\sum_{r=1}^{R}(s_{i,j,r} - \frac{1}{R}\sum_{r=1}^{R} s_{i,j,r})^2$ ;
    end
    Estimate the variance:$\hat{\sigma}_j^2 = \frac{1}{n}\sum_{i=1}^{n}\hat{\sigma}_{i,j}^2$;
    Generate the target vector $x_j$;
end
Estimate the model variance $\sigma^2 = \frac{1}{n}\sum_{j=1}^{n}\hat{\sigma}_j^2$;
for $1 \le i \le n$ do
    Initialize the representation $\theta_i^{old}$ ;
    while convergence condition is not satisfied do
        Compute the log likelihood based on Eqn. (18);
        Update the representation with Eqn. (19);
        $\theta_i^{new} = M_{\mathbf{Y}_i}(\theta_i^{old})$;
    end
    return $\theta_i^{new}$.
end
```

Algorithm 2: Representation learning algorithm.

Algorithm 3 presents the procedure for generating the collective influence context. The R random walks are first started from per user to simulate the influence diffusion process over the given network G. Then we generate the collective influence context by adding the users that line behind a given user v in the random walk sequences into the user v's context as described in Definition **??**. Notably, in generating the context $\mathbb{C}_{i,r}$, only the r-th random walk sequence starting from per user is taken into consideration.

```
//Generating Collective Influence Context(G, R, l)
Input: Network $G = (V, E)$, walks per user R, walk length $l$
Output: Collective influence context $\mathbb{C}_{i,r}$, $\mathbb{C}_i$, $(1 \le i \le n)$,
        $(1 \le r \le R)$.
for $1 \le r \le R$ do
    for $1 \le i \le n$ do
        Conduct the random walk from $u_i$ with size $l$: $C_{i,r}$;
    end
end
for $1 \le r \le R$ do
    for $1 \le i \le n$ do
        Generalize collective influence context $\mathbb{C}_{i,r}$;
        Count the coexisting time of each user in $\mathbb{C}_{i,r}$: $s_{i,j,r}$;
        return $\mathbb{C}_{i,r}$;
    end
end
for $1 \le i \le n$ do
    Generalize the collective influence $\mathbb{C}_i$ ;
    Extract the user-coexisting times tuples set $\mathbf{Y}_i$;
    return $\mathbb{C}_i$ , $\mathbf{Y}_i$.
end
```

Algorithm 3: Generating Collective Influence Context

5.3 Performance Analysis Of Representation Learning

We now move to the theoretical analysis for the representations learning. Our analysis is to uncover the conditions under which the distance between the representation θ_i returned by Algorithm 2 and the MLE θ_i' are bounded. For the simplification, in this subsection, we omit the subscript i and use θ to represent θ_i since the performance analysis is the same for the representations learning for all the users.

Given the maximizer of the log likelihood, say θ^*, [11] introduces the self-consistency property for the EM based maximum likelihood estimation, i.e.,

$$\theta^* = \arg\max_{\theta' \in \mathbb{R}^d} Q(\theta'|\theta^*). \tag{20}$$

Since the maximizer of the log likelihood is denoted by the $M_Y(\theta) = \arg\max_{\theta' \in \mathbb{R}^d} Q(\theta'|\theta)$, with the self-consistency shown in Eqn. (20), we have $\theta^* = M_Y(\theta^*)$. Under the property of the log likelihood function, Theorem 5.1 presents the property of the proposed mixture of regressions model in bounding the distance of the returned representation θ to the maximizer θ^*.

Theorem 5.1. ***(Convergence of the representation learning.)*** *Given the mixture of regressions model as presented in Eqn. (17) with a sufficiently large signal-to-noise ratio (SNR) $\frac{||\theta^*||_2}{\sigma^2}$, there is a constant $\lambda \in (0, 1/2]$ that*

$$||M(\theta) - \theta^*||_2 \le \lambda||\theta - \theta^*||_2, \tag{21}$$

holds for all θ if $||\theta - \theta^*||_2 \le \frac{||\theta^*||_2}{32}$.

Proof. As mention in Eqn. (19), the EM operator $M(\cdot)$ has the form $M_{\mathbf{Y}_i}(\theta_i) = \frac{\sum_{j=1}^{S_i} \sum_{j'=1}^{S_i} w_{\theta_i}(x_{j'}, s_{ij}) x_{j'} s_{ij}}{\sum_{j=1}^{S_i} \sum_{j'=1}^{S_i} w_{\theta_i}(x_{j'}, s_{ij}) \langle x_{j'}, x_j \rangle}$. 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

$$M(\theta) = \frac{1}{S}\mathbb{E}[w_\theta(\mathbf{Y}, \mathbf{X})\mathbf{Y}\mathbf{X}].$$

Define the notations that $\Delta_w(\mathbf{X}, \mathbf{Y}) = w_\theta(\mathbf{X}, \mathbf{Y}) - w_{\theta^*}(\mathbf{X}, \mathbf{Y})$ and $\Delta = \theta - \theta^*$, Eqn. (21) is equivalent to

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

Note that $\mathbf{Y} = \langle \mathbf{X}, \theta^* \rangle + v$, given any $\tilde{\Delta}$, Eqn. (21) can be further transferred to

$$\begin{gathered}\left\langle \mathbb{E}\left[\Delta_w(\mathbf{X}, \mathbf{Y})\mathbf{Y}\mathbf{X}\right], \tilde{\Delta} \right\rangle \le \lambda||\Delta||_2||\tilde{\Delta}||_2 \\ \underbrace{\mathbb{E}\left[\Delta_w(\mathbf{X}, \mathbf{Y})\langle \mathbf{X}, \theta^* \rangle \langle \mathbf{X}, \tilde{\Delta} \rangle\right]}_{L1} + \underbrace{\mathbb{E}\left[\Delta_w(\mathbf{X}, \mathbf{Y}) v \langle \mathbf{X}, \tilde{\Delta} \rangle\right]}_{L2} \le \lambda||\Delta||_2||\tilde{\Delta}||_2.\end{gathered} \tag{22}$$

Thus the Eqn (22) suggests that the proof of the Eqn. (21) is equivalent to bounding the two terms $L1$ and $L2$ in Eqn (22). Then we proceed to respectively provide the bound of the two terms.

1. Upper bound of the $L1 = \mathbb{E}\left[\Delta_w(\mathbf{X}, \mathbf{Y})\langle \mathbf{X}, \theta^* \rangle \langle \mathbf{X}, \tilde{\Delta} \rangle\right]$. It is noteworthy that the proof for the Eqn. (22) can be decomposed into the two sub problems, i.e., $L1 \le \frac{\lambda}{2}||\Delta||_2||\tilde{\Delta}||_2$ and $L2 \le \frac{\lambda}{2}||\Delta||_2||\tilde{\Delta}||_2$. Here, we first show that $L1$ is upper bounded by $\frac{\lambda}{2}||\Delta||_2||\tilde{\Delta}||_2$. Our proof is started by applying the Taylor's property to the function $\Delta_w(\mathbf{X}, \mathbf{Y})$. Since $\frac{d}{d_\theta} w_\theta(\mathbf{X}, \mathbf{Y}) = \frac{2}{S\sigma^2} \cdot \frac{\mathbf{X}, \mathbf{Y}}{\exp\left(\frac{2\mathbf{Y}\langle \mathbf{X}, \theta \rangle}{\sigma^2}\right)}$, according to the Taylor's series with the integral form, we have

$$\Delta_w(\mathbf{X}, \mathbf{Y}) = \frac{1}{S\sigma^2} \int_0^1 \frac{2\mathbf{Y}\langle \mathbf{X}, \Delta \rangle}{\exp\left(\frac{2Z_u}{\sigma^2}\right)} du, \tag{23}$$

where $Z_u = \mathbf{Y}\langle \mathbf{X}, \theta^* + u\Delta \rangle (u \in [0, 1])$. By taking the Eqn. (23) into $L1$, the objective here becomes to prove

$$\frac{1}{S} \int_0^1 \mathbb{E}\left[\frac{2\mathbf{Y}\langle \mathbf{X}, \theta^* \rangle}{\sigma^2 \exp\left(\frac{2Z_u}{\sigma^2}\right)} \langle \mathbf{X}, \Delta \rangle \langle \mathbf{X}, \tilde{\Delta} \rangle\right] du \le \frac{\lambda}{2}||\Delta||_2||\tilde{\Delta}||_2. \tag{24}$$

We set $M = \mathbb{E}\left[\frac{2\mathbf{Y}\langle \mathbf{X}, \theta^* \rangle}{S\sigma^2 \exp\left(\frac{2Z_u}{\sigma^2}\right)} \langle \mathbf{X}, \Delta \rangle \langle \mathbf{X}, \tilde{\Delta} \rangle\right]$. By $\theta^* = \theta_u - u\Delta$, we decompose M into the following two terms, i.e.,

$$\begin{aligned} M1 &= \mathbb{E}\left[\frac{2\mathbf{Y}\langle \mathbf{X}, \theta_u \rangle}{S\sigma^2 \exp\left(\frac{2Z_u}{\sigma^2}\right)} \langle \mathbf{X}, \Delta \rangle \langle \mathbf{X}, \tilde{\Delta} \rangle\right] \\ M2 &= -\mathbb{E}\left[\frac{2\mathbf{Y}u\langle \mathbf{X}, \Delta \rangle}{S\sigma^2 \exp\left(\frac{2Z_u}{\sigma^2}\right)} \langle \mathbf{X}, \Delta \rangle \langle \mathbf{X}, \tilde{\Delta} \rangle\right], \end{aligned}$$

as $M = M1 + M2$. According to the Cauchy-Schwartz Inequality, $M1$ and $M2$ can respectively be upper bounded by

$$M1 \le \sqrt{\mathbb{E}\left[\frac{\mathbf{Y}^2\langle \mathbf{X}, \theta_u \rangle^2}{S^2\sigma^4 \exp\left(\frac{4Z_u}{\sigma^2}\right)}\right]} \cdot \sqrt{\mathbb{E}\left[4\langle \mathbf{X}, \Delta \rangle^2 \langle \mathbf{X}, \tilde{\Delta} \rangle^2\right]}$$

$$M2 \le \sqrt{\mathbb{E}\left[\frac{\mathbf{Y}^2}{S^2\sigma^4 \exp\left(\frac{4Z_u}{\sigma^2}\right)}\right]} \cdot \sqrt{\mathbb{E}\left[4u^2\langle \mathbf{X}, \Delta \rangle^4 \langle \mathbf{X}, \tilde{\Delta} \rangle^2\right]}.$$

Combining Lemma 5.2 which shows that $\sqrt{\mathbb{E}\left[\frac{\mathbf{Y}^2\langle\mathbf{X},\theta_u\rangle^2}{S^2\sigma^4\exp\left(\frac{4Z_u}{\sigma^2}\right)}\right]} \leq \frac{\gamma}{14}$ and $\sqrt{\mathbb{E}\left[\frac{\mathbf{Y}^2}{S^2\sigma^4\exp\left(\frac{4Z_u}{\sigma^2}\right)}\right]} \leq \frac{\gamma}{32}$, we have $M1 \leq \frac{\lambda}{14}\sqrt{\mathbb{E}\left[4\langle\mathbf{X},\Delta\rangle^2\langle\mathbf{X},\tilde{\Delta}\rangle^2\right]}$ and $M2 \leq \frac{\lambda}{32}\sqrt{\mathbb{E}\left[4u^2\langle\mathbf{X},\Delta\rangle^4\langle\mathbf{X},\tilde{\Delta}\rangle^2\right]}$. The Lemma 5 in [11] shows that for a given random vector $\mathbf{X} \sim \mathcal{N}(0,\mathbf{I})$, there is $\mathbb{E}[\langle\mathbf{X},u\rangle^2\langle\mathbf{X},v\rangle^2] \leq 3||u||_2^2||v||_2^2$ and $\mathbb{E}[\langle\mathbf{X},u\rangle^4\langle\mathbf{X},v\rangle^2] \leq 15||u||_2^2||v||_2^2$. By the two upper bounds, we can further bound the $M1$ and $M2$ as

$$M1 \leq \frac{\lambda}{14}\sqrt{12}||\Delta||_2||\tilde{\Delta}||_2 \leq \frac{\lambda}{4}||\Delta||_2||\tilde{\Delta}||_2$$
$$M2 \leq \frac{\lambda}{32}\sqrt{60}||\Delta||_2^2||\tilde{\Delta}||_2 \leq \frac{\lambda}{4}||\Delta||_2||\tilde{\Delta}||_2 \quad (||\Delta||_2 \leq 1). \tag{25}$$

Then we have $M = M1 + M2 \leq \frac{\lambda}{2}||\Delta||_2||\tilde{\Delta}||_2$. Taking the value of the M into Eqn. (24), we can conclude that $L1 \leq \frac{\lambda}{2}||\Delta||_2||\tilde{\Delta}||_2$. Notably, in Eqn. (25), we make an assumption that $||\Delta||_2 \leq 1$. Next, we will proceed to proved that under the condition that $||\Delta||_2 \geq 1$, $L1 \leq \frac{\lambda}{2}||\Delta||_2||\tilde{\Delta}||_2$ still holds.

Our exploration of the $L1$'s value under the condition that $||\Delta||_2 = ||\theta - \theta^*||_2 \geq 1$ is based on the condition expectation. Given any random variable a, its conditional expectation under an event Ξ is defined as $\Psi(\Xi) = \mathbb{E}[a|\Xi]P(\Xi)$. Based on the parameters in the formulation of $L1$, we define the following four events:

- Ξ_1: For a given constant $\tau = C_\tau\sqrt{\log||\theta^*||_2}$, event Ξ_1 refers that $\Xi_1 = \{|v| \geq \tau/2\}$. Since $v \sim \mathcal{N}(0,\sigma^2)$, we have $P(\Xi_1) \leq 2e^{-\frac{\tau^2}{2\sigma^2}}$. (Event Ξ_2 is defined in the proof of Lemma 5.2.)
- Ξ_3: $\Xi_3 = \{|\langle\mathbf{X},\theta^*\rangle| \geq \tau\} \cap \{|\langle\mathbf{X},\theta_u\rangle| \geq \tau\} \cap \{|v| \leq \frac{\tau}{2}\}$. According to the property that for $\mathbf{X} \sim \mathcal{N}(0,\sigma^2)$, $P(|\mathbf{X}| \geq \tau) \leq 2e^{-\frac{\tau^2}{2\sigma^2}}$ and $P(|\mathbf{X}| \leq \tau) \leq \sqrt{\frac{2}{\pi}}\frac{\tau}{\sigma}$, we have $P(\Xi_3) \leq \frac{4\sqrt{2}}{\sqrt{\pi}}\frac{\tau}{\sigma}e^{-\frac{\tau^2}{2||\theta_u||^2}}e^{-\frac{\tau^2}{2||\theta^*||^2}}$.
- Ξ_4: $\Xi_4 = \{|\langle\mathbf{X},\theta_u\rangle| \leq \tau\}$.Similar to probability bound of the event Ξ_3, we have $P(\Xi_4) \leq \frac{\tau}{||\theta_u||_2}$.
- Ξ_5: $\Xi_5 = \{|\langle\mathbf{X},\theta^*\rangle| \leq \tau\}$. Similar to event Ξ_4, $P(\Xi_5) \leq \frac{\tau}{||\theta^*||_2}$.

Let us define the conditional expectation of $L1$ on a given event Ξ as $\Psi(\Xi) = \mathbb{E}[\Delta_w(\mathbf{X},\mathbf{Y})\langle\mathbf{X},\theta^*\rangle\langle\mathbf{X},\tilde{\Delta}\rangle|\Xi]P(\Xi)$, $L1$ is decomposed as

$$L1 = \Psi(\Xi_3) + \Psi(\Xi_1) + \Psi(\Xi_4) + \Psi(\Xi_5). \tag{26}$$

We now turn our attention to the values of the four terms in Eqn. (26) in turn.

$\Psi(\Xi_3)$: We first reformulate the $\Psi(\Xi_3)$ by the Cauchy-Schwartz Inequality as

$$\begin{aligned}\Psi(\Xi_3) &\leq \mathbb{E}\left[\left|\Delta_w(\mathbf{X},\mathbf{Y})\langle\mathbf{X},\theta^*\rangle\langle\mathbf{X},\tilde{\Delta}\rangle\right||\Xi_1\right]P(\Xi_1)\\ &\leq |||\mathbb{E}[\mathbf{X}\mathbf{X}^T]|||_{op}||\tilde{\Delta}||_2||\theta^*||_2\cdot\left(\frac{4\sqrt{2}}{\sqrt{\pi}}\frac{\tau}{\sigma}e^{-\frac{\tau^2}{2||\theta_u||^2}}e^{-\frac{\tau^2}{2||\theta^*||^2}}\right)\\ &\overset{(i)}{\leq} \frac{8\sqrt{2}}{\sqrt{\pi}}\frac{\tau}{\sigma}||\tilde{\Delta}||_2||\theta^*||_2\left(e^{-\frac{\tau^2}{2||\theta_u||^2}}e^{-\frac{\tau^2}{2||\theta^*||^2}}\right).\end{aligned}$$

The inequality (i) is based on the Lemma 8 in [11] which justifies that $|||\mathbb{E}[\mathbf{X}\mathbf{X}^T]|||_{op} \leq 2$ for $\mathbf{X} \sim \mathcal{N}(0,\mathbf{I})$.

$\Psi(\Xi_1)$: Based on the independence of $\mathbf{X}$ and v, we have

$$\begin{aligned}\Psi(\Xi_1) &\leq \mathbb{E}\left[\left|\Delta_w(\mathbf{X},\mathbf{Y})\langle\mathbf{X},\theta^*\rangle\langle\mathbf{X},\tilde{\Delta}\rangle\right||\Xi_1\right]P(\Xi_1)\\ &\leq |\mathbb{E}[\mathbf{X}\mathbf{X}^T]|||\tilde{\Delta}||_2||\theta^*||_2\cdot 2e^{-\frac{\tau^2}{2\sigma^2}}\\ &\leq 2||\tilde{\Delta}||_2||\theta^*||_2e^{-\frac{\tau^2}{2\sigma^2}}\end{aligned}$$

$\Psi(\Xi_4)$: Conditioned on the event Ξ_4, we have that

$$\langle\mathbf{X},\theta^*\rangle^2 \leq 2\langle\mathbf{X},\theta_u\rangle^2 + 2\langle\mathbf{X},\Delta\rangle^2 \leq 2\tau^2 + 2\langle\mathbf{X},\Delta\rangle^2.$$

With such property, we are ready to give the bound of $\Psi(\Xi_4)$. Since $P(\Xi_4) \leq \frac{\tau}{||\theta_u||_2}$, then

$$\begin{aligned}\Psi(\Xi_4) &\leq \mathbb{E}\left[\left|\Delta_w(\mathbf{X},\mathbf{Y})\langle\mathbf{X},\theta^*\rangle\langle\mathbf{X},\tilde{\Delta}\rangle\right||\Xi_4\right]P(\Xi_4)\\ &\leq \frac{\tau}{||\theta_u||_2}\sqrt{\mathbb{E}[\langle\mathbf{X},\tilde{\Delta}\rangle^2|\Xi_4]}\sqrt{\mathbb{E}[\langle\mathbf{X},\theta^*\rangle^2|\Xi_4]}\\ &\leq \frac{\tau\sqrt{2||\tilde{\Delta}||_2^2}\sqrt{2\tau^2+2\langle\mathbf{X},\Delta\rangle^2}}{||\theta_u||_2} \overset{(i)}{\leq} \frac{2\tau||\tilde{\Delta}||_2||\Delta||_2\sqrt{\tau^2+2}}{||\theta_u||_2}.\end{aligned}$$

Here, the inequality (i) is based on the Lemma 8 in [11] and the assumption that $||\Delta||_2 \geq 1$.

$\Psi(\Xi_5)$: Taking the conditions described in event Ξ into $L1$, we can directly obtain the following bound that

$$\Psi(\Xi_5) \leq \frac{\tau}{||\theta^*||_2}\sqrt{\mathbb{E}[\langle\mathbf{X},\tilde{\Delta}\rangle^2|\Xi_5]}\sqrt{\mathbb{E}[\langle\mathbf{X},\theta^*\rangle^2|\Xi_5]} \leq \frac{\sqrt{2}\tau^2||\tilde{\Delta}||_2}{||\theta^*||_2}.$$

Putting the bounds of the four terms together, we are ready to give the bounds of $L1$ which is formulated in Eqn. (26) under the case that $||\Delta||_2 \geq 1$, i.e.,

$$L1 \leq (\beta_1+\beta_2+\beta_3+\beta_4)\,||\tilde{\Delta}||_2||\Delta||_2 \tag{27}$$
$$\beta_1 = \frac{8\sqrt{2}}{\sqrt{\pi}}\frac{\tau}{\sigma}||\theta^*||_2e^{-\frac{\tau^2}{2||\theta_u||^2}-\frac{\tau^2}{2||\theta^*||^2}};\quad \beta_2 = 2||\theta^*||_2e^{-\frac{\tau^2}{2\sigma^2}};$$
$$\beta_3 = \frac{2\tau\sqrt{\tau^2+2}}{||\theta_u||_2};\quad \beta_4 = \frac{\sqrt{2}\tau^2}{||\theta^*||_2}.$$

By setting appropriate SNR value $\frac{||\theta^*||}{\sigma}$ and the constant C_τ, we can derive the conclusion that $L1 \leq \frac{\lambda}{2}||\tilde{\Delta}||_2||\Delta||_2$ in the case that $||\Delta||_2 \geq 1$. As mentioned before, the proof for the Theorem 5.1 is decomposed into the two parts that $L1 \leq \frac{\lambda}{2}||\tilde{\Delta}||_2||\Delta||_2$ and $L2 \leq \frac{\lambda}{2}||\tilde{\Delta}||_2||\Delta||_2$ in Eqn. (22). Now we have finished the proof for the $L1 \leq \frac{\lambda}{2}||\tilde{\Delta}||_2||\Delta||_2$, and then we move to the analysis of $L2$.

2. Upper bound of the $L2 = \mathbb{E}\left[\Delta_w(\mathbf{X},\mathbf{Y})v\langle\mathbf{X},\tilde{\Delta}\rangle\right]$. Similar to the analysis of $L1$, our analysis on the $L2$ is also started from applying the Taylor's theory. As $\frac{d}{d\theta}w_\theta(\mathbf{X},\mathbf{Y}) = \frac{2}{S\sigma^2}\cdot\frac{\mathbf{X},\mathbf{Y}}{\exp\left(\frac{2\mathbf{Y}\langle\mathbf{X},\theta\rangle}{\sigma^2}\right)}$, proving $L2 \leq \frac{\lambda}{2}||\tilde{\Delta}||_2||\Delta||_2$ is equivalent to showing that

$$\frac{1}{S}\int_0^1\mathbb{E}\left[\frac{2\mathbf{Y}v\langle\mathbf{X},\Delta\rangle\langle\mathbf{X},\tilde{\Delta}\rangle}{\sigma^2\exp\left(\frac{2Z_u}{\sigma^2}\right)}\right]du \leq \frac{\lambda}{2}||\Delta||_2||\tilde{\Delta}||_2. \tag{28}$$

By further applying the Cauchy-Schwartz Inequality, we have

$$\mathbb{E}\left[\frac{2\mathbf{Y}v\langle\mathbf{X},\Delta\rangle\langle\mathbf{X},\tilde{\Delta}\rangle}{S\sigma^2\exp\left(\frac{2Z_u}{\sigma^2}\right)}\right] \leq \sqrt{\mathbb{E}\left[\frac{4\mathbf{Y}^2}{S^2\sigma^4\exp\left(\frac{4Z_u}{\sigma^2}\right)}\right]}\sqrt{\mathbb{E}\left[v^2\langle\mathbf{X},\Delta\rangle^2\langle\mathbf{X},\tilde{\Delta}\rangle^2\right]}$$

$$\overset{(i)}{\leq} \sqrt{\mathbb{E}\left[\frac{4\mathbf{Y}^2}{S^2\sigma^4 \exp\left(\frac{4Z_u}{\sigma^2}\right)}\right]} \sqrt{3||\Delta||_2^2||\tilde{\Delta}||_2^2} \overset{(ii)}{\leq} \frac{\sqrt{3}\lambda}{16}||\Delta||_2||\tilde{\Delta}||_2,$$

where the inequality (i) is based on the Lemma 5 in [11] and the inequality (ii) is from the Eqn. (30) in Lemma 5.2. It is noteworthy that the Eqn. (30) in Lemma 5.2 holds only under the condition that $||\Delta||_2 \leq 1$. How is the value of $L2$ when $||\Delta||_2 \geq 1$? To answer this question, we proceed to explore the value of $L2$ under $||\Delta||_2 \geq 1$.

When $||\Delta||_2 \geq 1$, according to the Cauchy-Schwartz Inequality, we have

$$\begin{aligned}
&\mathbb{E}\left[\Delta_w(\mathbf{X},\mathbf{Y})v\langle \mathbf{X}, \tilde{\Delta}\rangle\right] \leq \mathbb{E}\left[|\Delta_w(\mathbf{X},\mathbf{Y})v\langle \mathbf{X}, \tilde{\Delta}\rangle|\right] \\
&= \mathbb{E}[|\Delta_w(\mathbf{X},\mathbf{Y})\frac{v}{\langle \mathbf{X}, \theta^*\rangle}\langle \mathbf{X}, \theta^*\rangle\langle \mathbf{X}, \tilde{\Delta}\rangle|] \\
&\leq \mathbb{E}[|\Delta_w(\mathbf{X},\mathbf{Y})\langle \mathbf{X}, \theta^*\rangle\langle \mathbf{X}, \tilde{\Delta}\rangle|] \cdot \mathbb{E}[|\frac{v}{\langle \mathbf{X}, \theta^*\rangle}|] \\
&\overset{(i)}{\leq} \frac{\lambda}{2}||\Delta||_2||\tilde{\Delta}||_2\sqrt{\frac{\mathbb{E}[v^2]}{\mathbb{E}[\langle \mathbf{X}, \theta^*\rangle^2]}} \\
&\leq \frac{\lambda}{2}||\Delta||_2||\tilde{\Delta}||_2\sqrt{\frac{\sigma^2}{2||\theta^*||_2^2}} \leq \frac{\lambda}{2}||\Delta||_2||\tilde{\Delta}||_2.
\end{aligned}$$

The Inequality (i) holds since we have proved that $\mathbb{E}[|\Delta_w(\mathbf{X},\mathbf{Y})\langle \mathbf{X}, \theta^*\rangle \langle \mathbf{X}, \tilde{\Delta}\rangle|] \leq \frac{\lambda}{2}||\Delta||_2||\tilde{\Delta}||_2$ in bounding the $L1$.

Therefore, with the upper bound $L1 \leq \frac{\lambda}{2}||\Delta||_2||\tilde{\Delta}||_2$ and $L2 \leq \frac{\lambda}{2}||\Delta||_2||\tilde{\Delta}||_2$, we can draw the conclusion that $\left\langle \mathbb{E}\left[\Delta_w(\mathbf{X},\mathbf{Y})\mathbf{Y}\mathbf{X}\right], \tilde{\Delta}\right\rangle \leq \lambda||\Delta||_2||\tilde{\Delta}||_2$ holds and complete the proof of Theorem 5.1. □

Lemma 5.2. *There is a $\lambda \in [0, 1/2)$ that for $\forall u \in [0, 1]$ satisfies*

$$\sqrt{\mathbb{E}\left[\frac{\mathbf{Y}^2\langle \mathbf{X}, \theta_u\rangle^2}{S^2\sigma^4 \exp\left(\frac{4Z_u}{\sigma^2}\right)}\right]} \leq \frac{\gamma}{14}, \quad (\theta_u = \theta^* + u\Delta) \tag{29}$$

$$\sqrt{\mathbb{E}\left[\frac{\mathbf{Y}^2}{S^2\sigma^4 \exp\left(\frac{4Z_u}{\sigma^2}\right)}\right]} \leq \frac{\gamma}{32} \quad (||\Delta||_2 \leq 1). \tag{30}$$

Proof. We divide our proof for the Eqn. (29) and Eqn. (30) accordingly.

Bounding Eqn. (29). Let us define the conditional expectation function that $\Psi(\Xi) = \mathbb{E}\left[\frac{\mathbf{Y}^2\langle \mathbf{X}, \theta_u\rangle^2}{S^2\sigma^4 \exp\left(\frac{4Z_u}{\sigma^2}\right)}|\Xi\right] P(\Xi)$, for $\Psi(\Xi) = \mathbb{E}\left[\frac{\mathbf{Y}^2\langle \mathbf{X}, \theta_u\rangle^2}{S^2\sigma^4 \exp\left(\frac{4Z_u}{\sigma^2}\right)}\right]$ in Eqn. (29), we thus have

$$\mathbb{E}\left[\frac{\mathbf{Y}^2\langle \mathbf{X}, \theta_u\rangle^2}{S^2\sigma^4 \exp\left(\frac{4Z_u}{\sigma^2}\right)}\right] = \Psi(\Xi_1) + \Psi(\Xi_1^c \cap \Xi_2) + \Psi(\Xi_3), \tag{31}$$

in which the events Ξ_1, Ξ_2 and Ξ_3 are defined as follows.

- Ξ_1: For a given constant $\tau = C_\tau\sqrt{\log||\theta^*||_2}$, event Ξ_1 refers that $\Xi_1 = \{|v| \geq \tau/2\}$. Since $v \sim \mathcal{N}(0, \sigma^2)$, we have $P(\Xi_1) \leq 2e^{-\frac{\tau^2}{2\sigma^2}}$.
- Ξ_2: $\Xi_2 = \{|\langle \mathbf{X}, \theta^*\rangle| \leq \tau\} \cap \{|\langle \mathbf{X}, \theta_u\rangle| \leq \tau\}$. According to the property that for $\mathbf{X} \sim \mathcal{N}(0, \sigma^2)$, $P(|\mathbf{X}| \leq \tau] \leq \sqrt{\frac{2}{\pi}}\frac{\tau}{\sigma}$, we have $P(\Xi_2) \leq \frac{\tau}{||\theta^*||_2\sigma} + \frac{\tau}{||\theta_u||_2\sigma}$.
- Ξ_3: $\Xi_3 = \{|\langle \mathbf{X}, \theta^*\rangle| \geq \tau\} \cap \{|\langle \mathbf{X}, \theta_u\rangle| \geq \tau\} \cap \{|v| \leq \frac{\tau}{2}\}$. Given the probability bound of the events Ξ_1 and Ξ_2, we have $P(\Xi_3^c) \leq \frac{\tau}{||\theta^*||_2\sigma} + \frac{\tau}{||\theta_u||_2\sigma} + 2e^{-\frac{\tau^2}{2}}$.

Upon giving the definitions of the three events, we now move to explore the values of the three terms $\Psi(\Xi_1)$, $\Psi(\Xi_2)$ and $\Psi(\Xi_3)$.

$\Psi(\Xi_1)$: Note that

$$\frac{\mathbf{Y}^2\langle \mathbf{X}, \theta_u\rangle^2}{S^2\sigma^4 \exp\left(\frac{4Z_u}{\sigma^2}\right)} \leq \sup_{t\geq 0}\frac{t^2}{S^2\exp 4t} \leq \frac{1}{S^2}\frac{1}{4e^2}. \tag{32}$$

We have that $\Psi(\Xi_1) \leq \frac{1}{2S^2e^2}e^{-\frac{\tau^2}{2}}$.

$\Psi(\Xi_1^c \cap \Xi_2)$: By Eqn. (32) and the probability bound of the event Ξ_2, we have

$$\Psi(\Xi_1^c \cap \Xi_2) \leq \frac{1}{4S^2e^2}P(\Xi_2) \leq \frac{1}{4S^2e^2}\left[\frac{\tau}{||\theta^*||_2\sigma} + \frac{\tau}{||\theta_u||_2\sigma}\right]. \tag{33}$$

$\Psi(\Xi_3)$: Conditioned on the event Ξ_3 that $\{|\langle \mathbf{X}, \theta^*\rangle| \geq \tau\} \cap \{|\langle \mathbf{X}, \theta_u\rangle| \geq \tau\} \cap \{|v| \leq \frac{\tau}{2}\}$, there is $|\mathbf{Y}| = |\langle \mathbf{X}, \theta^*\rangle + v| \geq |\langle \mathbf{X}, \theta^*\rangle| - |v| \geq \frac{\tau}{2}$, and $|\mathbf{Y}\langle \mathbf{X}, \theta_u\rangle| \geq \frac{\tau^2}{2}$. Thus

$$\frac{\mathbf{Y}^2\langle \mathbf{X}, \theta_u\rangle^2}{S^2\sigma^4 \exp\left(\frac{4Z_u}{\sigma^2}\right)} \leq \sup_{t\geq\frac{\tau^2}{2}}\frac{t^2}{S^2\exp 4t} \leq \frac{t^2}{S^2\exp 4t}\Big|_{t=\frac{\tau^2}{2}} = \frac{\tau^4}{4S^2e^{2\tau^2}}. \tag{34}$$

Therefore, combining the upper bound of the three terms as shown above yields

$$\begin{aligned}
&\mathbb{E}[\frac{\mathbf{Y}^2\langle \mathbf{X}, \theta_u\rangle^2}{S^2\sigma^4 \exp\left(\frac{4Z_u}{\sigma^2}\right)}] \\
\leq &\frac{1}{2S^2e^2}e^{-\frac{\tau^2}{2}} + \frac{1}{4S^2e^2}\left[\frac{\tau}{||\theta^*||_2\sigma} + \frac{\tau}{||\theta_u||_2\sigma}\right] + \frac{\tau^4}{4S^2e^{2\tau^2}}.
\end{aligned}$$

Setting the constant C_τ and the SNR sufficiently large, we can obtain the claim in Eqn. (29).

Bounding Eqn. (30). We also prove the Eqn. (30) based on the conditional expectation function that $\Psi(\Xi) = \mathbb{E}\left[\frac{\mathbf{Y}^2}{S^2\sigma^4 \exp\left(\frac{4Z_u}{\sigma^2}\right)}|\Xi\right] P(\Xi)$. Here, we define an event about the value of $\langle \mathbf{X}, \theta_u\rangle$.

- Ξ_4: $\Xi_4 = \{|\langle \mathbf{X}, \theta_u\rangle| \leq \tau\}$. Similar to the probabilistic bounding of the event Ξ_2, we have $P(\Xi_4) \leq \frac{\tau}{||\theta_u||_2\sigma}$.

Given the event Ξ_4, the Eqn. (30) can be decomposed as

$$\mathbb{E}\left[\frac{\mathbf{Y}^2}{S^2\sigma^4 \exp\left(\frac{4Z_u}{\sigma^2}\right)}\right] = \Psi(\Xi_4) + \Psi(\Xi_4^c). \tag{35}$$

$\Psi(\Xi_4)$: Since $P(\Xi_4) \leq \frac{\tau}{||\theta_u||_2\sigma}$, the term $\Psi(\Xi_4)$ is upper bounded by $\Psi(\Xi_4) \leq \frac{\tau}{S^2\sigma^5||\theta_u||_2}\mathbb{E}[\mathbf{Y}^2]$. With the definition that $\mathbf{Y} = \langle \mathbf{X}, \theta^*\rangle + v$, we have

$$\mathbb{E}[\mathbf{Y}^2|\Xi_4] \leq \mathbb{E}[2\langle \mathbf{X}, \theta^*\rangle^2 + 2v^2|\Xi_4].$$

For $\langle \mathbf{X}, \theta^*\rangle^2$, conditioned on the event that $|\langle \mathbf{X}, \theta_u\rangle| \leq \tau$, it becomes $\langle \mathbf{X}, \theta^*\rangle^2 \leq 2\tau^2 + 2\langle \mathbf{X}, \Delta\rangle^2$. Referring the Lemma 8 in [11], the second moment of the matrix $\mathbf{X}$ is upper bounded by $|||\mathbb{E}[\mathbf{X}\mathbf{X}^T]|||_{op} \leq$

2, the $\langle \mathbf{X}, \theta^* \rangle^2$ further becomes $\langle \mathbf{X}, \theta^* \rangle^2 \leq 2\tau^2 + 4$. We can find that

$$\mathbb{E}[\mathbf{Y}^2 | \Xi_4] \leq 4\tau^2 + 8 + 2\sigma^2 \quad (||\Delta||_2 \leq 1).$$

Then we have $\Psi(\Xi_4) \leq \frac{\tau(4\tau^2+8+2\sigma^2)}{S^2\sigma^5||\theta_u||_2}$.

$\Psi(\Xi_4^c)$: Under the event that $|\langle \mathbf{X}, \theta_u \rangle| \leq \tau$, we have

$$\Psi(\Xi_4^c) \leq \mathbb{E}\left[\frac{\mathbf{Y}^2}{S^2\sigma^4 \exp(\frac{4Z_u}{\sigma^2})} | \Xi_4^c\right] \leq \frac{1}{4e\tau^2 S^2}.$$

The rationale behind the inequality is that for $f(t) = \frac{t^2}{\exp(\mu t)}$, there is $\sup_{t \in [0,\infty)} f(t) = \frac{4}{(e\mu)^2}$. Combining the two conditional expectation $\Psi(\Xi_4)$ and $\Psi(\Xi_4^c)$, we have

$$\mathbb{E}\left[\frac{\mathbf{Y}^2}{S^2\sigma^4 \exp\left(\frac{4Z_u}{\sigma^2}\right)}\right] \leq \frac{\tau(4\tau^2 + 8 + 2\sigma^2)}{S^2\sigma^5||\theta_u||_2} + \frac{1}{4e\tau^2 S^2}.$$

With the relation that $||\theta_u||_2 \geq ||\theta^*||_2 - ||\Delta||_2 \leq \frac{||\theta^*||_2}{2}$, we further have

$$\mathbb{E}\left[\frac{\mathbf{Y}^2}{S^2\sigma^4 \exp\left(\frac{4Z_u}{\sigma^2}\right)}\right] \leq \frac{2\tau(4\tau^2 + 8 + 2\sigma^2)}{S^2\sigma^5||\theta^*||_2} + \frac{1}{4e\tau^2 S^2}.$$

By setting the SNR $\frac{||\theta^*||_2}{\sigma^2}$ and the constant C_τ sufficiently large, we can obtain the claim in Eqn. (30). Thus we end the proof for Lemma 5.2. □

5.4 Embedding Seed Selection Algorithm

Next, we introduce how to select the seed users based on the learned low-dimensional representations for capturing the collective influences. Algorithm 4 presents the procedure of the seed selection based on the low-dimensional representation. Since the inner product $\langle \theta_i, x_j \rangle$ represents the collective influence of the users u_i on u_j, we select the user with the maximum $\sum_{j=1}^{W} \hat{S}_{ij}$ as the next seed in each iteration. In the k-th iteration in the online selecting phase, upon selecting the k-th seed, we update the collective influence $\hat{S}_{ij}$ for each node. The updating of the $\hat{S}_{ij}$ here is similar to updating the derivative matrix $CI_l(i)$, which represents the collective influence computed in the original space, in Algorithm 1.

```
//Embedding Seed Selection($\Theta$, $\mathbf{X}$, $K$)
Input: Source vector: $\Theta$, Target vector: $\mathbf{X}$, Seed size: $K$;
Output: Seed set $S$.
//Offline computing phase
Randomly choose $W$ vectors from $\mathbf{X}$;
for $1 \leq j \leq W$ do
    for $1 \leq i \leq N$ do
        Compute $\hat{S}_{ij} = \langle \theta_i, x_j \rangle$;
    end
end
//Online selecting phase
for $1 \leq k \leq K$ do
    $S_k = \arg\max_{v_i \in V} \sum_{j=1}^{W} \hat{S}_{ij}$, $S = S \cup S_k$;
    $\theta_{i^*} \longleftarrow$ source vector of $s_k$;
    for each node $v_1 \in V \backslash S$ do
        Update $\hat{S}_{ij} = \max(\hat{S}_{ij} - \langle \theta_{i^*}, x_j \rangle, 0)(1 \leq j \leq W)$ ;
    end
end
return $S$.
```

Algorithm 4: Embedding Seed Selection

6 EXPERIMENTS

We will experimentally show that our proposed CI framework and embedding CI framework are better than existing IM solutions on both the effectiveness, which is measured by the influence diffusion size of selected seed users, and the efficiency measured by the running time of seed selection process. We further justify our theoretical results on the minimum required seed size of minimizing uninfluenced size. Our experiments are on both real social, academic and synthetic networks.

6.1 Experimental Settings

Dataset. We use 9 datasets in the experiments, i.e., 6 real-world networks,and 3 synthetic networks generated as the ER, power-law degree distribution (PL) and Stochastic Block Model (SBM). The 9 datasets are summarized in Table 1. The LiveJournal, Wikipedia, Twitter and Epinions are downloaded from the open social network dataset collection SNAP. The Co-author and Citation are the academic networks downloaded from the open academic dataset collection Acenap. The Co-author network is consisted of the coauthorship among 1.7M scholars in the Computer Science area, and the Citation network contains the citation relations among 1.5M papers belonging to the Machine Learning topic. The ER, PL and SBM are the synthetic network we used to validate our theoretical results in Section 4.2.2. Since we generate each of the three models under four different settings, we do not list the number of edges in them in Table 1.

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
Co-author	1.7M	50M	Co-authorship of scholars
Citation	1.5M	7M	Citations among papers
Twitter	81K	1.8M	Real social network
Epinions	75K	0.5M	Real social network
ER	100K	/	ER network
PL	10K	/	Power-law distribution
SBM	100K	/	stochastic block model

Baselines. We compare the CI and embedding CI framework with the following four baselines on IM.

(1) IMM [5]: IMM is one of the most IM solution based on the Reverse Reachable sets (RR-sets) framework. Its main idea lies on first sampling sufficient the number of RR-sets to ensure the accuracy of estimating the expected influence, and then iteratively selecting seed users who can cover the most number of RR-sets.

(2) SKIM [12]: 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 L of RR-sets as seeds.

(3) K-core. K-core is a popular percolation method which is used to find the cores of a given network. It deletes the nodes in turn with the degree being $1, 2, 3, \ldots$, until the remaining node size meet a preset threshold. The remaining nodes after the deletion is

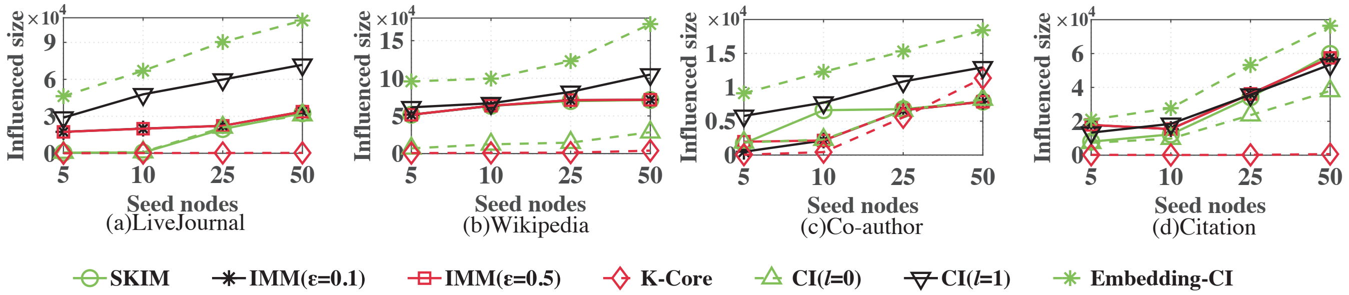


Figure 2: Influence diffusion size vs. K.

	LiveJournal				Wikipedia				Co-author				Citation			
Algorithm	$K = 5$	10	25	50	5	10	25	50	5	10	25	50	5	10	25	50
SKIM	3937	3945	4460	5769	1311	1884	1946	2010	2882	3460	3682	4252	1125	1130	1614	2204
IMM ($\varepsilon = 0.1$)	194	1940	2009	2693	99	532	712	1023	69	300	498	895	52	224	299	496
IMM ($\varepsilon = 0.5$)	110	786	1939	2485	51	444	620	900	45	285	425	513	27	187	290	466
K-core	46.8k	46.8k	46.8k	46.8k	2788	2788	2788	2788	3553	3553	3553	3553	1453	1453	1453	1453
CI ($l = 0$)	0.7	1.5	3.5	6.8	0.35	0.63	1.46	2.9	0.4	0.6	1.6	3.3	0.25	0.49	1.2	2.4
CI ($l = 1$)	347	485	591	681	7	19	45	73	9.3	11.5	17.3	27	43	63	92	119
Embedding-CI	2628	2644	2668	2692	1419	1433	1446	1460	1206	1220	1236	1248	1003	1020	1026	1043

Table 2: Running time of seed selection (s) vs. K

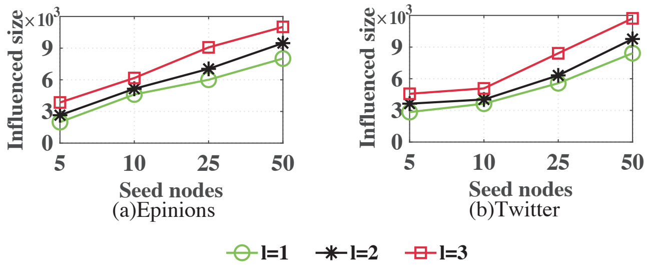


Figure 3: Effect of l on Influence diffusion size.

	Epinions				Twitter			
Algorithm	$K = 5$	10	25	50	5	10	25	50
$l = 1$	2.6	4.4	6.8	10	1.8	3.9	7.8	12.6
$l = 2$	26	39	52	69	22	44	70	88
$l = 3$	193	287	498	795	231	315	570	1015

Table 3: Effect of l on running time (s)

considered as the core of a given network. Then the seed users are randomly selected from the cores.

(4) Maximum degree. Maximum degree iteratively selects the users with the highest degrees. Coincidentally, according to the formulation of the CI given by Theorem 4.5, when $l = 0$, the CI algorithm is equivalent to selecting the users with highest degrees. Thus we called the Maximum degree as CI ($l = 0$) later.

Parameter settings. We set the weight edge w_{ij} of edge (i, j) as the value of the $w_{ij} = \frac{1}{d_{in}(u_j)}$ where $d_{in}(u_j)$ is the in-degree of node u_j. This setting is widely used in [4] [5] [6]. We compute the influence diffusion size of the seed users returned by the algorithms by running the Monte-Carlo simulation of IC model.

Environment. We implement algorithms with Python 2.7 and conduct experiments on a computer running Ubuntu 16.04 LTS with 40 cores 2.30 GHz (Intel Xeon E5-2650) and 126 GB memory.

6.2 Performance On Seed Selection

6.2.1 Effectiveness study. In the experiments, we set the seed size as $K = 5, 10, 25, 50$ and the effectiveness study is conducted over the four large scale networks, i.e., LiveJournal, Wikipedia, Co-author and Citation. Fig. 2 presents the influence diffusion size of the seed users returned by the baselines and our CI and Embedding-CI algorithm. From Fig. 2, we can see when we just consider the immediate neighbors of the users in network in computing CI, the seed users returned by CI algorithm has better performance on influence diffusion size than other baselines in most cases. The classical CI algorithm is not scalable when $l \geq 2$ since a given user can reach almost all the other users during a few hops. Thus we just conduct CI algorithm on $l = 0$ and $l = 1$ over the four large scale networks. The effect of l will be discussed in Section 6.2.3.

The embedding CI always has the best performance since it comprehensively considers the collective of the users and their immediate and multi-hop neighbors. By incorporating the power of multiple spreaders and thus correlating the influence with network integrity, the Embedding-CI is obtains more information of the seed users on the performance of influence diffusion. So that Embedding-CI has much better effectiveness than CI under $l = 1$. In conducting the Embedding CI, we set the value of r as fine, i.e., each user is taken as the source node of 5 random walk sequences. The dimension of the learned representations is set as 50. Since the dimension 50 is much smaller the size of any network, Embedding-CI is scalable when considering the CI of users via multi-hops.

For IMM and SKIM, due to the greedy framework they rely on has a approximation ratio that is limited to $(1-\frac{1}{e}-\varepsilon)$, their performances

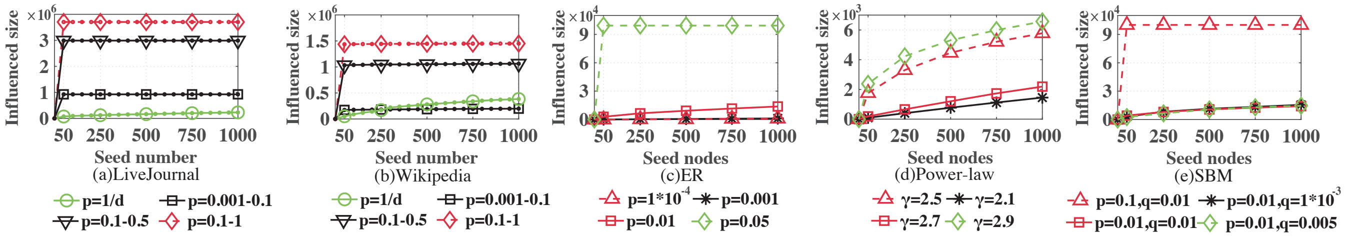


Figure 4: Influence diffusion size vs. K.

are always restricted to a sub-optimal value. Another reason is that the traditional greedy IM solutions limit their attention to individuals that are treated as non-interacting and independent agent, resulting in the selected seeds only have local influence in a network. The Maximum degree (CI ($l = 0$)) and K-core always have the poorest effectiveness. The reason behind is that they do not optimize a global function of influence in seed selection. As a result, the selected users may have much overlapping influence, making the growing of seed size brings little gain of influence diffusion size.

6.2.2 Efficiency study. Table 2 reports the running time of the seed selection algorithms. We can see from Table 2 that $CI(l = 1)$ costs just a fraction of the time of the SKIM, IMM since $CI(l = 1)$ does not require much sampling from the network. It just computes the collective influence of the users and their immediate neighbors. For Embedding-CI, most running time of is spent on the computation of the inner products among the low-representations of users at the beginning. Fortunately, with the help of the low-dimensional representations, the collective influences of users can be easily conducted in parallel. The K-core costs the most time because it deletes the lowest degree node one by one, at the same time, updates the network structure after deleting one node. For $CI(l = 0)$, it heuristically selects the users with highest degrees and costs the minimum time. In summary, both $CI(l = 1)$ is scalable to large scale networks and have the better performance on seed selection.

6.2.3 Discussion of the parameter l. Now, we proceed to discuss the effect of the value of l on the performance of CI algorithm. Since Ci algorithm is time-consuming in the large scale networks when $l \geq 2$, we study the effect of l on two small networks, i.e., Twitter and Epinions. Fig. 3 reports the effect of l on the influence diffusion size. With the increase of l, the CI algorithm can obtain more comprehensive collective influences of users since the powers of the users who are multi hops away are also taken into consideration. However, as presented in Table 3, the increase of l substantially improves the running time of the CI algorithm. In addition, the CI algorithm needs to read in all the data of a network even for selecting one seed, making it difficult to be conducted in parallel. The unscalability of the CI algorithm motivates our study on the embedding case.

6.3 Validation of the condition for minimizing uninfluenced size

At last, let us validate the condition for minimizing uninfluenced size, i.e., $Q(S, G) = 0$. In the Fig. 4 (a) and Fig. 4 (b), we explore the size of the required for realizing $Q(S, G) = 0$ under different settings of edge weights in the LiveJournal and Wikipedia. With the increase of the edge weights, we can see the phase transition from the curves of influence diffusion size, meaning that the giant uninfluenced connected component becomes more "fragile". Also, we can see the seed users are more influential under the large edge weights in IC model.

In the Fig. 4 (c), Fig. 4 (d) and Fig. 4 (e), we validate our theoretical results derived in Section 4.2.2. We generate the ER, power-law degree distribution and SBM networks as follows:

- ER: Each pair of nodes are connected randomly with a given probability p.
- Power-law degree distribution: Each node comes into the network in turn and bring a constant m of new edges. Then it choose a node u_i to connect with a probability proportional to the value of $(d_i + k_0)$, where d_i is the current degree of u_i and $k_0 \geq -m$. Under the above evolution process, the network follows the degree distribution $P(d = x) \sim x^{-\gamma}, \gamma = 3 + \frac{k_0}{m}$.
- SBM: Each node is randomly assigned to one of the C communities. The nodes that belong to a same community connect each other with probability p, and connect to the members of other communities with probability q.

In Fig. 4 (c), we can see the required size of seeds decreases with the probability p in ER network as we give in Theorem 4.2. Furthermore, in power-law degree distribution network, Fig. 4 (c) justifies that the required seed size decrease with γ under the same m. Then as Theorem 4.4 describes, the seeds in the SBM becomes more influential with the increase of the value of q.

REFERENCES

[1] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proc. SIGKDD*, pages 137–146. ACM, 2003.

[2] F. Morone and H. Makse. Influence maximization in complex networks through optimal percolation. *Nature*, 524(7563):65, 2015.

[3] C. Borgs, M. Brautbar, J. Chayes, and booktitle=Proc. SODA pages=946–957 year=2014 organization=SIAM Lucier, B. Maximizing social influence in nearly optimal time.

[4] H. T. Nguyen, T. P. Nguyen, T. N. Vu, and T. N. Dinh. Outward influence and cascade size estimation in billion-scale networks. *Proc. Sigmetrics*, 1(1):20, 2017.

[5] Y. Tang, Y. Shi, and X. Xiao. Influence maximization in near-linear time: A martingale approach. In *Proc.SIGMOD*, pages 1539–1554. ACM, 2015.

[6] Y. Lin, W. Chen, and J. CS Lui. Boosting information spread: An algorithmic approach. In *Proc. ICDE*, pages 883–894. IEEE, 2017.

[7] Y. Yang, X. Mao, J. Pei, and X. He. Continuous influence maximization: What discounts should we offer to social network users? In *Proc. SIGMOD*. ACM, 2016.

[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. MobiHoc*, pages 121–130. ACM, 2018.

[9] S. Lei, S. Maniu, L. Mo, and P. Cheng, R.and Senellart. Online influence maximization. In *Proc. SIGKDD*, pages 645–654. ACM, 2015.

[10] J. Yuan and S. Tang. Adaptive discount allocation in social networks. In *Proc. MobiHoc*, page 22. ACM, 2017.

[11] S. Balakrishnan, M. J. Wainwright, and B. Yu. Statistical guarantees for the em algorithm: From population to sample-based analysis. *The Annals of Statistics*, 45(1):77–120, 2017.

[12] E. Cohen, D. Delling, T. Pajor, and R. F. Werneck. Sketch-based influence maximization and computation: Scaling up with guarantees. In *Proc. CIKM*, pages 629–638. ACM, 2014.

[13] P. ERDdS and A. R&WI. On random graphs i. *Publ. Math. Debrecen*, 6:290–297, 1959.

[14] A. L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

A PROOFS FOR LEMMAS 3.2, 3.3 AND 3.4

(Lemma 3.2.) Given an ER graph with the degree distribution being $P(d = x) = \frac{(\lambda)^x e^{-\lambda}}{x!}(\lambda = np)$, the convergence, which corresponds to the percolation over the entire network, of the CI algorithm is upper bounded by

$$q_c = \Theta\left(\frac{\lambda - 1}{\lambda}\right).$$

Proof. The ER graph was first introduced by Erdős and Rényi in 1959 [13], in which every pair of nodes are connected at random with a given probability p. Since then, a large number of studies has adopted the ER graph as one of the most representative random network models for exploring the properties including but not limited to information diffusion, community detection, and graph matching, etc. Given an ER graph with N nodes, the distribution of the degree is formalized as $P(d = x) = \binom{N-1}{x} p^x (1-p)^{N-1-x}$. Throughout the paper, we consider the influence maximization over the large sparse network and consequently, we assume that under such scenarios, the degree distribution of the ER graph is approximated by

$$P(d = x) = \frac{(np)^x e^{-np}}{x!} = \frac{(\lambda)^x e^{-\lambda}}{x!}(\lambda = np).$$

The convergence of the CI algorithm is indicated by the leading eigenvalue $\lambda = 1$. Recall Eqn. (11), the leading eigenvalue is a function of the degree distribution of a given network. Thus the proof for Lemma4.2 is based on the degree distribution. We say that d_{max} is the largest degree of nodes in the network, d_{min} is the lowest degree in the network, then $a \cdot \sum_{x=d_{min}}^{d_{max}} P(d = x) = 1$ where a is the normalization factor. Since the CI algorithm under $l = 0$ is equivalent to removing the nodes with the highest degrees, given ξ is the lowest degree of the removed network, we have

$$q_c = a \cdot \sum_{x=\xi}^{d_{max}} P(d = x).$$

In order to determine q_c, we proceed to explore the value of ξ based on the spectral condition indicated by the leading eigenvalue λ. The Eqn. (11) tells us that the leading eigenvalue is determined by the mean degree $\overline{d}'$ and square mean degree $\overline{d}^{2\prime}$ of the network after removing the fraction of q_c nodes. Then we first turn our attention to the value of $\overline{d}'$ and $\overline{d}^{2\prime}$.

After removing the Nq_c nodes with the highest degrees, the mean degree of the $N(1 - q_c)$ remaining degrees is given by

$$\begin{aligned}
\overline{d}' &= \frac{a}{1 - q_c} \sum_{x=d_{min}}^{\xi} \frac{\lambda^x}{x!} e^{-\lambda} x \\
&= \frac{1}{1 - q_c} \cdot \lambda e^{-\lambda} \cdot a \sum_{x=d_{min}}^{\xi} \frac{\lambda^{x-1}}{(x-1)!} e^{-\lambda} \\
&= \frac{1}{1 - q_c} \cdot \lambda e^{-\lambda} \cdot \left(e^{\lambda} - a \cdot \sum_{x=\xi+1}^{d_{max}} \frac{\lambda^{x-1}}{(x-1)!}\right).
\end{aligned}$$

Similarly, the $\overline{d}^{2\prime}$ can be given by

$$\begin{aligned}
\overline{d}^{2\prime} &= \frac{a}{1 - q_c} \sum_{x=d_{min}}^{\xi} \frac{\lambda^x}{x!} e^{-\lambda} x^2 \\
&= \frac{1}{1 - q_c} a \cdot \lambda e^{-\lambda} \sum_{x=d_{min}}^{\xi} \frac{x\lambda^{x-1}}{(x-1)!} \\
&= \frac{1}{1 - q_c} \lambda e^{-\lambda} \cdot a \left(\sum_{x=d_{min}}^{\xi} \frac{(x-1)\lambda^{x-1}}{(x-1)!} + \sum_{x=d_{min}}^{\xi} \frac{\lambda^{x-1}}{(x-1)!}\right) \\
&= \frac{1}{1 - q_c} \lambda e^{-\lambda} \cdot a \left(\sum_{l=d_{min}-1}^{\xi-1} \frac{\lambda^{l-1}}{(l-1)!} + \sum_{x=d_{min}}^{\xi} \frac{\lambda^{x-1}}{(x-1)!}\right) \\
&= \frac{1}{1 - q_c} \lambda e^{-\lambda} \cdot \left(\lambda e^{\lambda} - \lambda \sum_{l=\xi}^{d_{max}} \frac{\lambda^{l-1}}{(l-1)!} + e^{\lambda} - a \sum_{x=\xi+1}^{d_{max}} \frac{\lambda^{x-1}}{(x-1)!}\right) \\
&\equiv \frac{1}{1 - q_c} (\lambda^2 + \lambda).
\end{aligned}$$

Taking the value of the $\overline{d}'$ and $\overline{d}^{2\prime}$ into Eqn. (11), we have

$$\frac{(1 - q_c)\left[\frac{1}{1-q_c}(\lambda^2 + \lambda) - \frac{1}{1-q_c} \cdot \lambda e^{-\lambda} \cdot \left(e^{\lambda} - a \cdot \sum_{x=\xi+1}^{d_{max}} \frac{\lambda^{x-1}}{(x-1)!}\right)\right]}{\lambda} = 1$$

$$\frac{\Theta(\lambda^2 + \lambda) - \lambda\left(1 - a \sum_{x=\xi+1}^{d_{max}} \frac{\lambda^{x-1}}{(x-1)!} e^{-\lambda}\right)}{\lambda} \approx 1 \tag{36}$$

Since $q_c = a \cdot \sum_{x=\xi}^{d_{max}} P(d = x)$, from Eqn. (36), we have

$$q_c = \Theta(\frac{\lambda - 1}{\lambda}). \tag{37}$$

Thus we finish the proof of Lemma 4.2 □

Lemma 4.2 tells us that, in the ER network, the number of the nodes that need to be removed for making the giant connected component vanish increase with the connecting probability p. Upon finishing the analysis of the ER network , we now turn our attention to the networks with power-law degree distribution.

(Lemma 3.3.) Given a network with the power-law degree distribution, i.e., $P(d = x) = a \cdot x^{-\gamma}$, the convergence, which corresponds to the percolation over the entire network, of the CI algorithm is upper bounded by

$$q_c = \begin{cases} \Theta(1) \quad (0 < \gamma < 1) \\ \Theta\left(\left(\frac{d_{max}}{d_{min}}\right)^{1-\gamma}\right) \quad (1 < \gamma < 2) \\ o(1) \quad (\gamma > 2). \end{cases}$$

Proof. The power-law degree distribution is a popular phenomenon in modern social networks where only a little fraction of the users have the large degrees. As characterized by Barabási *et al* [14], the power-law degree distribution is originated from the preferential attachment rule which reflects that, in the network evolution, the newly added users are more willing to connect those with high degrees, making the high degree users more easily to have larger degrees. The aim of the convergence analysis on the power-law degree is to exploring how many nodes need to seeded for the percolation of the entire social network.

As the name suggests, the degree distribution function of the power-law degree distribution networks is formalized as

$$P(d = x) = a \cdot x^{-\gamma},$$

where a is the normalization factor and $\gamma > 0$.

Similar to the proof for Lemma 4.2, we also prove the Lemma 4.3 based on the analysis of the degree distribution. We also let the d_{max} and d_{min} denote the lowest and highest degree in the given social network, then there is $\sum_{x=d_{min}}^{d_{max}} P(d = x) = a \cdot x^{-\gamma} = 1$. Given the lowest degree of the nodes selected by the CI algorithm under $l = 0$, we have $q_c = a \sum_{x=\xi}^{d_{max}} x^{-\gamma}$. Taking the continuous limit of q_c, then q_c becomes the function of the ξ as

$$q_c = \int_{\xi}^{d_{max}} a \cdot x^{-\gamma} dx = \frac{d_{max}^{1-\gamma} - \xi^{1-\gamma}}{d_{max}^{1-\gamma} - d_{min}^{1-\gamma}}. \tag{38}$$

Here, $a = 1/d_{max}^{1-\gamma} - d_{min}^{1-\gamma}$. To obtain more precise value of the q_c, we now move to the spectral condition shown in Eqn. (11) for determining the ξ. After removing the Nq_c nodes with the highest degrees, the mean degree of the $N(1 - q_c)$ remaining degrees is given by

$$\overline{d}' = \frac{1}{1 - q_c} \int_{d_{min}}^{\xi} a \cdot x^{-\gamma} x dx = \frac{1}{1 - q_c} \frac{\gamma - 1}{\gamma - 2} \frac{\xi^{2-\gamma} - d_{min}^{2-\gamma}}{d_{max}^{1-\gamma} - d_{min}^{1-\gamma}}.$$

The square mean degree of the remaining $N(1 - q_c)$ nodes is given by

$$\overline{d}^{2\prime} = \frac{1}{1 - q_c} \int_{d_{min}}^{\xi} a \cdot x^{-\gamma} x^2 dx = \frac{1}{1 - q_c} \frac{\gamma - 1}{\gamma - 3} \frac{\xi^{3-\gamma} - d_{min}^{3-\gamma}}{d_{max}^{1-\gamma} - d_{min}^{1-\gamma}}.$$

Taking the $\overline{d}'$ and the $\overline{d}^{2\prime}$ into the Eqn. (11), we obtain the relationship between the ξ and d_{min}, d_{max} and γ, i.e.,

$$\frac{\gamma - 2}{\gamma - 3} \frac{\xi^{3-\gamma} - d_{min}^{3-\gamma}}{d_{max}^{2-\gamma} - d_{min}^{2-\gamma}} - \frac{\xi^{2-\gamma} - d_{min}^{2-\gamma}}{d_{max}^{2-\gamma} - d_{min}^{2-\gamma}} = 1, \tag{39}$$

since the mean degree of the nodes in the original network is $\overline{d} = \frac{\gamma-1}{\gamma-2} \frac{d_{max}^{2-\gamma} - d_{min}^{2-\gamma}}{d_{max}^{1-\gamma} - d_{min}^{1-\gamma}}$.

In order to determine the value of the ξ and q_c, we start the discussion on Eqn. (38) and Eqn. (39). From Eqn. (38) and Eqn. (39), we can see that the different value of the γ may bring significant different value of the ξ and q_c. Thus the discussion of the Eqn. (38) and Eqn. (39) is based on the value of γ.

1°$0 < \lambda < 2$: Before we proceed, let us comment on a simplifying assumption that we will implicitly introduced in the following. Suppose that the highest degree user has the connections with almost all the other users and the lowest degree user only connects to a few users, then d_{max} and d_{min} can be scaled as $d_{max} = \Theta(n)$ and $d_{min} = o(n)$ respectively. Under the assumption, we have

$$\begin{aligned}
&\frac{\gamma - 2}{\gamma - 3} \frac{\xi^{3-\gamma} - d_{min}^{3-\gamma}}{d_{max}^{2-\gamma}} - \frac{\xi^{2-\gamma} - d_{min}^{2-\gamma}}{d_{max}^{2-\gamma}} \approx 1 \\
&\frac{\gamma - 2}{\gamma - 3} \left(\frac{\xi}{d_{max}}\right)^{3-\gamma} - \left(\frac{\xi}{d_{max}}\right)^{2-\gamma} \approx 1 + \frac{\gamma - 2}{\gamma - 3} \left(\frac{d_{min}}{d_{max}}\right)^{3-\gamma} - \left(\frac{d_{min}}{d_{max}}\right) \\
&\frac{\gamma - 2}{\gamma - 3} \left(\frac{\xi}{d_{max}}\right)^{3-\gamma} - \left(\frac{\xi}{d_{max}}\right)^{2-\gamma} = \Theta(1) \\
&\xi = \Theta(d_{max})
\end{aligned}$$

Then q_c becomes

$$q_c = \begin{cases} \frac{d_{max}^{1-\gamma} - \xi^{1-\gamma}}{k^{1-\gamma}} = \Theta(1) \quad (0 < \gamma < 1) \\ \frac{d_{max}^{1-\gamma} - \xi^{1-\gamma}}{d_{min}^{1-\gamma}} = \Theta\left(\left(\frac{d_{max}}{d_{min}}\right)^{1-\gamma}\right) \quad (1 < \gamma < 2). \end{cases} \tag{40}$$

2° $2 < \gamma < 3$: In this case, the Eqn. (39) becomes:

$$\begin{aligned}
&\frac{\gamma - 2}{\gamma - 3} \frac{d_{min}^{3-\gamma} - \xi^{3-\gamma}}{d_{min}^{2-\gamma}} - \frac{d_{min}^{2-\gamma} - \xi^{2-\gamma}}{d_{min}^{2-\gamma}} \approx 1 \\
&\frac{\gamma - 2}{3 - \gamma} \frac{\xi^{3-\gamma}}{d_{min}^{2-\gamma}} + \left(\frac{\xi}{d_{min}}\right)^{2-\gamma} \approx 2 + \frac{\gamma - 2}{3 - \gamma} m \\
&\frac{\gamma - 2}{3 - \gamma} \frac{\xi^{3-\gamma}}{d_{min}^{2-\gamma}} = \Theta\left(\frac{\gamma - 2}{3 - \gamma} m\right) \\
&\xi = \Theta(d_{min})
\end{aligned}$$

Taking $\xi = \Theta(d_{min})$ into Eqn. (38), we have

$$q_c = \frac{\xi^{1-\gamma}}{d_{min}^{1-\gamma}} = o(1). \tag{41}$$

3° $\gamma > 3$: In this case, the Eqn. (39) becomes:

$$\begin{aligned}
&\frac{\gamma - 2}{\gamma - 3} \frac{d_{min}^{3-\gamma} - \xi^{3-\gamma}}{d_{min}^{2-\gamma}} - \frac{d_{min}^{2-\gamma} - \xi^{2-\gamma}}{d_{min}^{2-\gamma}} \approx 1 \\
&\frac{\gamma - 2}{\gamma - 3} \frac{\xi^{3-\gamma}}{d_{min}^{2-\gamma}} - \left(\frac{\xi}{d_{min}}\right)^{2-\gamma} = \Theta(d_{min}) \\
&\xi = \Theta(d_{min})
\end{aligned}$$

Taking $\xi = \Theta(d_{min})$ into Eqn. (38), we have

$$q_c = \frac{\xi^{1-\gamma}}{d_{min}^{1-\gamma}} = o(1). \tag{42}$$

Based on the discussion above, we are ready to summarize the nodes need to seeded for the percolation over the entire network with power-law degree distribution, i.e.,

$$q_c = \begin{cases} \Theta(1) \quad (0 < \gamma < 1) \\ \Theta\left(\left(\frac{d_{max}}{d_{min}}\right)^{1-\gamma}\right) \quad (1 < \gamma < 2) \\ o(1) \quad (\gamma > 2). \end{cases} \tag{43}$$

Now we have completed the proof for Lemma 4.3. □

(Lemma 3.4.) Given a network characterized by the SBM which has the degree distribution as shown in Eqn. (45), the fraction of the nodes that need to be seeded for the percolation over the entire network is upper bounded by

$$q_c = \Theta\left(1-(1-q)^{(2q-1)n}\right). \tag{44}$$

Proof. The Stochastic Block Model (SBM) is one of the most common models to characterize the communities in the networks. The analysis of the convergence of CI on the SBM can explore how many nodes need to be seeded for percolating the influences over the network with communities. Suppose the network is consisted of n nodes and Q communities, given a node that belongs to a community with size k, the probability distribution of its degree is formalized as

$$P(d=x) = \tag{45}$$
$$\begin{cases} \sum_{a=0}^{x} C_{k-1}^{a} C_{n-k}^{x-a} p^{a}(1-p)^{k-1-a} q^{x-a}(1-q)^{n-k-(x-a)} & (x \leq k-1) \\ \sum_{a=0}^{k-1} C_{k-1}^{a} C_{n-k}^{x-a} p^{a}(1-p)^{k-1-a} q^{x-a}(1-q)^{n-k-(x-a)} & (x \geq k-1) \end{cases}$$

where p denotes the probability that there is an edge between any pair of nodes in a same community and q denotes that of the edge between a pair of nodes that belong to different communities.

As mentioned at the beginning of this section, the fraction of the seeded nodes is given by $q_c = a \cdot \sum_{x=\xi}^{d_{max}} P(d=x)$. To compute the value of the q_c, we need to determine the two unknown parameters, i.e., a and ξ. Thus our proof is started by determining the value of the normalization factor a, followed by the discussion of the lowest degree of the seeded nodes ξ based on the spectral condition shown in Eqn. (11).

1. Computation for the normalization factor a. The normalization factor a satisfies that $a \cdot \sum_{d_{min}}^{d_{max}} P(d=x)$. To explore the value of a, we first compute the sum of the probabilities of the degrees lying into the interval $[d_{max}, d_{min}]$. Before we proceed, let us comment on a simplifying assumption that $d \geq K$. The reason behind is that one user may have the edges between almost all of his mates in a same community as well as a certain number of members in other communities. Then, we have

$$\begin{aligned} P(d=x) &= \sum_{a=0}^{k-1} C_{k-1}^{a} C_{n-k}^{x-a} p^{a}(1-p)^{k-1-a} q^{x-a}(1-q)^{n-k-(x-a)} \\ &= p^{k-1} q^{n-k} \sum_{a=0}^{k-1} (\frac{1}{p}-1)^{k-1-a} (\frac{1}{q}-1)^{n-k-(x-a)}. \end{aligned}$$

Let $\alpha = \frac{1}{p}-1$, $\beta = \frac{1}{q}-1$, and $t = \frac{\alpha}{\beta}$, then

$$P(d=x) = p^{k-1} q^{n-k} \alpha^{k-1} \beta^{n-k-x} \sum_{a=0}^{k-1} C_{k-1}^{a} C_{n-k}^{x-a} t^{a}$$

For $m \leq k-1$, we consider that

$$\sum_{a=0}^{k-1} C_{k-1}^{a} C_{n-k}^{x-a} = C_{n-1}^{x},$$

and

$$\frac{k-1}{x} C_{n-k+x}^{x} < \sum_{a=0}^{x} C_{k-1}^{a} C_{n-k}^{x-a} < C_{k-1}^{x} C_{n-k+x}^{x}.$$

Then we have

$$\frac{C_{k-1}^{x} C_{n-k+x}^{x}}{\frac{k-1}{x} C_{n-k+x}^{x}} = \frac{x}{k-1} C_{k-1}^{x} \in 2^{H(\frac{x}{k-1})(k-1)} \left[\frac{1}{k}, 1\right],$$

where $H(r) = -r\log r - (1-r)\log(1-r)$. Thus, we have

$$\sum_{a=0}^{x} C_{k-1}^{a} C_{n-k}^{x-a} \approx C_{k-1}^{x} C_{n-k+x}^{x}. \tag{46}$$

Now, we are ready to compute the sum of $P(d=x)$ in the interval $[d_{min}, d_{max}]$.

$$\begin{aligned} \sum_{x=d_{min}}^{d_{max}} P(d=x) &= p^{k-1} q^{n-k} \alpha^{k-1} \beta n - k \sum_{x=d_{min}}^{d_{max}} \left(\beta^{-x} \sum_{a=0}^{k-1} C_{k-1}^{a} C_{n-k}^{x-a} t^{a}\right) \\ &= p^{k-1} q^{n-k} \alpha^{k-1} \beta^{n-k} \sum_{x=d_{min}}^{d_{max}} \left(b^{-x} t^{k-1} \sum_{a=0}^{k-1} C_{k-1}^{a} C_{n-k}^{x-a} \left(\frac{1}{t}\right)^{k-1-a}\right). \end{aligned} \tag{47}$$

Furthermore, since

$$\begin{aligned} \sum_{a=0}^{k-1} C_{k-1}^{a} C_{n-k}^{x-a} \left(\frac{1}{t}\right)^{k-1-a} &= \sum_{a=0}^{k-1} C_{k-1}^{a} C_{n-k}^{x-a} + \left[\left(\frac{1}{t}\right) - 1\right] \left(\sum_{a=0}^{k-2} C_{k-1}^{a} C_{n-k}^{x-a}\right) \\ &+ \left[\left(\frac{1}{t}\right)^{2} - 1\right] \left(\sum_{a=0}^{k-3} C_{k-1}^{a} C_{n-k}^{x-a}\right) + \cdots + \left[\left(\frac{1}{t}\right)^{k-1} - 1\right] \left(\sum_{a=0}^{0} C_{k-1}^{a} C_{n-k}^{x-a}\right), \end{aligned}$$

$\sum_{x=d_{min}}^{d_{max}} P(d=x)$ can be formalized as

$$\begin{aligned} \sum_{x=d_{min}}^{d_{max}} P(d=x) &= p^{k-1} q^{n-k} \alpha^{k-1} \beta^{n-k} \\ &\cdot \sum_{x=d_{min}}^{d_{max}} \left[\beta^{-x} t^{k-1} \left(C_{n-1}^{x} + \sum_{a=1}^{k-1} \left[\left(\frac{1}{t}\right)^{a} - 1\right] C_{k-1}^{a} C_{n-k+x}^{k-a-1}\right)\right] \end{aligned} \tag{48}$$

In most social networks with n users, the number of the communities is scaled as $\Theta(\log n)$, thus the size of the communities k can be scaled as $k = \Theta(\frac{n}{\log n})$. Based on this property, Eqn (48) can be further formalized as

$$\begin{aligned} \sum_{x=m}^{n} P(d=x) &= (1-p)^{k-1}(1-q)^{n-k} \\ &\cdot \sum_{x=m}^{n} \left[\beta^{-x} t^{k-1} \left(C_{n-1}^{x} + \sum_{a=1}^{k-1} \left[\left(\frac{1}{t}\right)^{a} - 1\right] C_{k-1}^{a} C_{n-k+x}^{k-a-1}\right)\right]. \end{aligned} \tag{49}$$

Since $k = \Theta(\frac{n}{\log n})$, we have

$$\begin{aligned} C_{k-1}^{a} &\leq C_{k-1}^{\frac{k-1}{2}} < 2^{k-1} = O\left(2^{C_2 \frac{n}{\log n}}\right) = O\left(\left(\frac{n}{\log n}\right)^{C_4}\right) \\ C_{n-k+x}^{k-a-1} &< n^{k} < O\left(n^{\frac{n}{\log n}}\right). \end{aligned}$$

In Eqn. (49), let $M = (1-P)^{k-1}(1-q)^{n-k} \sum_{x=d_{min}}^{d_{max}} \{\beta^{-x} t^{k-1} \sum_{a=1}^{k-1} [(\frac{1}{t})^{a} - 1] C_{k-1}^{a} C_{n-k+x}^{k-a-1}\}$, then

$$M \leq (1-p)^{k-1}(1-q)^{n-k}(1-t^{k-1})(k-1)C_1 \left(\frac{n}{\log n}\right)^{C_4} C_2 n^{\frac{n}{\log n}} \sum_{x=d_{min}}^{d_{max}} \beta^{-x}.$$

Since $\beta = \frac{1}{q} - 1$, when $q < \frac{1}{2}$, we have $\beta > 1$, otherwise $\beta < 1$. Thus further analysis of the value of M is based on the discussion q.

1° $q < \frac{1}{2}$: $\beta > 1$, and $\sum_{x=d_{min}}^{d_{max}} \to \frac{\beta^{1-d_{min}}}{\beta^{-1}} = \Theta(1)$. Since $(n-k)\log\frac{1}{1-q} >> (\frac{n}{\log n}+C_4)\log n$ and $(k-1)(1-q)^{n-k} \cdot n^{\frac{n}{\log n}+C_4} \to 0$, then

$$(1-p)^{k-1}(1-q)^{n-k}(1-t^{k-1})(k-1)C_1C_2n^{\frac{n}{\log n}+C_4} \to 0.$$

Then we have, when $q < \frac{1}{2}$, $M \to 0$.

2° $q > \frac{1}{2}$: $\beta < 1$, and $\sum_{d_{min}}^{d_{max}} \beta^{-x}$ becomes

$$\sum_{d_{min}}^{d_{max}} \beta^{-x} = \frac{\beta^{1-d_{min}} - b^{1-d_{max}}}{\beta - 1} = \frac{q}{2q-1}\left[\left(\frac{q}{1-q}\right)^{d_{max}} - \left(\frac{q}{1-q}\right)^{d_{min}}\right].$$

In addition, $(1-q)^{n-k}\sum_{x=d_{min}}^{d_{max}} \beta^{-x} \to q^n(1-q)^{-k}$. Based on the observation, we have

$$M \to q^n \left(\frac{1-p}{1-q}\right)^k \left(\frac{1}{1-p}\right)(1-t^{k-1})(k-1)C_1C_2n^{\frac{n}{\log n}+C_4} \to 0.$$

With $M \to 0$, the Eqn. (49) becomes

$$\sum_{x=d_{min}}^{d_{max}} P(d=x) = (1-p)^{k-1}(1-q)^{n-k}t^{k-1}\sum_{x=d_{min}}^{d_{max}} \beta^{-x}C_{n-1}^x \quad (50)$$

We then move to the discussion of $\sum_{x=d_{min}}^{d_{max}} \beta^{-x}C_{n-1}^x$ in Eqn. (50).

Computing the value of $\sum_{x=d_{min}}^{d_{max}} \beta^{-x}C_{n-1}^x$. With the assumption that $d_{max} = \Theta(n)$ and $d_{min} = o(n)$, $\sum_{x=d_{min}}^{d_{max}} \beta^{-x}C_{n-1}^x$ can be approximated as $\sum_{x=d_{min}}^{d_{max}} \beta^{-x}C_{n-1}^x \approx \left(\frac{1}{\beta}+1\right)^{n-1} - \sum_{x=0}^{d_{min}} \beta^{-x}C_{n-1}^x$. Suppose $\sum_{x=0}^{d_{min}} \beta^{-x}C_{n-1}^x = (\frac{n-1}{d_{min}})^\theta(1+\frac{1}{\beta})^{d_{min}}$, then

$$\sum_{x=0}^{d_{min}} \frac{(n-1)!\beta^{-x}}{x!(n-1-x)!} = \left(\frac{n-1}{d_{min}}\right)^\theta \frac{d_{min}!\beta^{-x}}{x!(d_{min}-x)!}.$$

Let

$$\hat{\theta} = \arg\min\left(\left|\sum_{x=0}^{d_{min}}\left(\frac{(n-1)!}{(n-1-x)!} - \left(\frac{n-1}{d_{min}}\right)^\theta \frac{d_{min}!}{(d_{min}-x)!}\right)\beta^{-x}\right|\right),$$

we have

$$\begin{aligned}\theta \approx \hat{\theta} &\approx \arg\min\left(\sum_{x=0}^{d_{min}} \frac{n^\theta - n^x}{\beta^x}\right) \\ &= \arg\min\left(\frac{\beta^{d_{min}+1}-1}{\beta-1}n^\theta - \frac{n^{d_{min}+1}-\beta^{d_{min}}}{n-\beta}\right) \\ &= \log_n \frac{(\beta-1)(n^{d_{min}+1}-\beta^{d_{min}})}{(n-\beta)(\beta^{d_{min}+1}-1)} \\ &= d_{min} - \gamma. \end{aligned} \quad (51)$$

Here, $\gamma = \begin{cases} 0, \ \beta^{d_{min}} = O(n) \\ \lambda, \ \beta^{d_{min}} = cn^\lambda \end{cases}$. Thus $\sum_{x=d_{min}}^{d_{max}} \beta^{-x}C_{n-1}^x$ equals

$$\sum_{x=d_{min}}^{d_{max}} \beta^{-x}C_{n-1}^x = \left(\frac{1}{\beta}+1\right)^{n-1} - (\frac{n-1}{d_{min}})^{m-\gamma}(1+\frac{1}{\beta})^{d_{min}}. \quad (52)$$

Combining Eqn. (52) and Eqn. (52), we have

$$\begin{aligned}&\sum_{x=d_{min}}^{d_{max}} P(d=x) = (1-p)^{k-1}(1-q)^{n-k} \\ &\cdot\left[t^{k-1}\left(1+\frac{1}{\beta}\right)^{n-1} - t^{k-1}\left(\frac{d_{max}-1}{d_{min}}\right)^{d_{min}-\gamma}\left(1+\frac{1}{\beta}\right)^{d_{min}}\right] \\ &= \left(\frac{p}{q}\right)^{k-1}\left[1-(1-q)^{d_{max}-d_{min}-1}\left(\frac{d_{max}-1}{d_{min}}\right)^{d_{min}-\gamma}\right] \\ &\to \left(\frac{p}{q}\right)^{k-1}\left[1-(1-q)^{d_{max}}\left(\frac{d_{max}-1}{d_{min}}\right)^{d_{min}-\gamma}\right].\end{aligned}$$

As mentioned before, we assume that $m = o(n)$, hence $n\log(1-q) + (d_{min-\gamma}\log\left(\frac{d_{max}}{d_{min}}\right) \to -\infty$ and $(1-q)^{d_{max}}\left(\frac{d_{max}}{d_{min}}\right)^{d_{min}-\gamma} \to 0$. Therefore, $\sum_{x=d_{min}}^{d_{max}} P(d=x)$ is equal to

$$\sum_{x=d_{min}}^{d_{max}} P(d=x) = \Theta\left(\left(\frac{p}{q}\right)^{k-1}\right). \quad (53)$$

Under the condition that $a \cdot \sum_{x=d_{min}}^{d_{max}} P(d=x) = 1$, the value of normalization factor a is scaled as

$$a = \Theta\left(\left(\frac{q}{p}\right)^{k-1}\right). \quad (54)$$

Upon finishing the computation of the normalization factor a, we now can proceed to determine the lowest degree ξ of the seed nodes selected by the CI algorithm.

2. Determining the lowest seeding degree ξ. Similar to the derivation of the sum of the probability $\sum_{x=d_{min}}^{d_{max}} P(d=x)$, the first moment and the second moment of the degree distribution on the SBM is formalized as

$$\begin{aligned}&\sum_{x=d_{min}}^{d_{max}} P(d=x)x \to (d_{max}-1)(1-q)\left(\frac{p}{q}\right)^{k-1} \\ &= (d_{max}-1)(1-q)\sum_{x=d_{min}}^{d_{max}} P(x=d); \end{aligned} \quad (55)$$

$$\begin{aligned}&\sum_{x=d_{min}}^{d_{max}} P(d=x)x^2 \to (d_{max}-1)(1-q)\left[1+(d_{max}-2)(1-q)\right]\left(\frac{p}{q}\right)^{k-1} \\ &= (d_{max}-1)(1-q)\left[1+(d_{max}-2)(1-q)\right]\sum_{x=d_{min}}^{d_{max}} P(d=x). \end{aligned} \quad (56)$$

Based on the formalization of the first and second moments shown in Eqn (55) and Eqn. (56), recalling the spectral condition of ξ presented in Eqn. (11), we first have

$$\begin{aligned}&\sum_{x=d_{min}}^{\xi} P(d=x)x^2 - \sum_{x=d_{min}}^{\xi} P(d=x)x \\ &= (d_{max}-1)(d_{max}-2)(1-q)^{d_{max}+2}\left(\frac{p}{q}\right)^{k-1}\left(\frac{d_{max}}{\xi}\right)^{\xi-2-\gamma}. \end{aligned} \quad (57)$$

By taking the Eqns. (55) and (57) into Eqn. (11), we obtain the spectral condition over the SBM model, i.e.,

$$(d_{max}-2)(1-q)^{d_{max}+1}\left(\frac{d_{max}}{\xi}\right)^{\xi-2-\gamma} = 1$$

$$\begin{aligned}(\xi-2-\gamma)\log_n^{\left(\frac{d_{max}}{\xi}\right)} &= -\log_n^{d_{max}-2} - (d_{max}+1)\log_n^{1-q}\\ &\rightarrow -(d_{max}+1)\log_n^{1-q} - 1 \qquad (58)\end{aligned}$$

Let $\xi = C(d_{max}+3+\gamma)$, then

$$\begin{aligned}(\xi-2-\gamma)\log_n(\frac{d_{max}}{\xi}) &= (d_{max}+1)\left(1-\log_{d_{max}} d_{max}+3+\gamma-\log_{d_{max}} C\right)\\ &\rightarrow -(d_{max}+1)\log_n C. \qquad (59)\end{aligned}$$

Taking Eqn. (59) into Eqn. (58), we have

$$\begin{aligned}-(n+1)\log_n C &= -(n+1)\log_n(1-q) - 1\\ C &= (1-q)\cdot n^{\frac{1}{n+1}} \rightarrow 1-q\\ \xi &= (1-q)(n+3+\gamma) \qquad (60)\end{aligned}$$

To explore the precise value of ξ, we then discuss the γ in Eqn. (60) as follows.

Determining the γ in Eqn. (60). Recalling Eqn. (52), $\theta = d_{min} - \gamma$ in computing $\sum_{x=d_{min}}^{d_{max}} \beta^{-x} C_{n-1}^{x}$ where $d_{min} = o(n)$. However, the lowest degree among the deleted nodes is not always equal to $o(n)$. For this reason, we now recompute the parameter γ for the sum $\sum_{x=\xi}^{d_{max}} b_{-x} C_{n-1}^{x}$. The $\sum_{x=\xi}^{d_{max}} b_{-x} C_{n-1}^{x}$ can be approximated as $\sum_{x=\xi}^{d_{max}} b^{-x} C_{n-1}^{x} \approx \left(\frac{1}{\beta}+1\right)^{n-1} - \sum_{x=0}^{\xi} \beta^{-x} C_{n-1}^{x}$. Suppose $\sum_{x=0}^{\xi} \beta^{-x} C_{n-1}^{x} = \sum_{x=0}^{\xi} \left(\frac{n-1}{\xi}\right)^{\theta} (1+\frac{1}{\beta})^{\xi}$, then

$$\sum_{x=0}^{\xi} \frac{(n-1)!\beta^{-x}}{x!(n-1-x)!} = \sum_{x=0}^{\xi} \left(\frac{n-1}{\xi}\right)^{\theta} \left(1+\frac{1}{\beta}\right)^{\xi}.$$

Furthermore, we have

$$\begin{aligned}\theta \approx \hat{\theta} &= \arg\min\left(\left|\sum_{x=0}^{\xi}\left(\frac{(n-1)!}{(n-1-x)!} - \left(\frac{n-1}{\xi}\right)^{\beta} \frac{\xi!}{(\xi-x)!}\right)\beta^{-x}\right|\right)\\ &\approx \arg\min\left(\sum_{x=0}^{\xi} \frac{n^{\xi+1}-\beta^{\xi}}{n-\beta} - \left(\frac{n}{\xi}\right)^{\theta} \frac{\xi^{\xi+1}-\beta^{\xi}}{\xi-\beta}\right)\\ &= \log_{\left(\frac{n}{\xi}\right)} \frac{(n^{\xi+1}-\beta^{\xi})(\xi-\beta)}{(\xi^{\xi+1}-\beta^{\xi})(n-\beta)}\\ &\approx \xi\end{aligned}$$

Comparing the Eqn. (51), then $\gamma = 0$. Thus

$$\xi = (1-q)(n+3).$$

Based on all the analysis above, we are ready to give the fraction q_c of nodes that need to be seeded for the percolation over a SBM network.

3. Computing the fraction q_c. As mentioned in Eqn. (**??**), the fraction of seed nodes is computed as $q_c = a \cdot \sum_{x=\xi}^{d_{max}} P(d=x)$. Since $\xi = (1-q)(n+3)$, we have

$$\begin{aligned}q_c &= a\cdot\sum_{x=\xi}^{d_{max}} P(d=x) \rightarrow t^{k-1}(1-p)^{k-1}(1-q)^{n-k}\sum_{x=\xi}^{n-1}\beta^{-x}C_{n-1}^{x}\\ &= a\cdot\left(\frac{p}{q}\right)^{k-1}\left[1-(1-q)^{qn+3q-4}\left(\frac{n-1}{(1-q)(n+3)}\right)^{(1-q)(n+3)}\right]\\ &= a\cdot\left(\frac{p}{q}\right)^{k-1}\left[1-(1-q)^{(2q-1)n+6q-7}\right]\\ &\rightarrow a\cdot\left(\frac{p}{q}\right)^{k-1}\left[1-(1-q)^{(2q-1)n}\right]\\ &= \Theta\left(1-(1-q)^{(2q-1)n}\right).\end{aligned}$$

Therefore, we complete the proof for the Lemma 4.4. □