2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)

Influence Maximization at Community Level: A New Challenge with Non-submodularity

Lan N. Nguyen
CISE Department
University of Florida
Gainesville, FL, 32611
Email: {lan.nguyen}@ufl.edu

Kunxiao Zhou
Computer School
Dongguan University of Technology
Dongguan, China
Email: zhoukx@dgut.edu.cn

My T. Thai
CISE Department
University of Florida
Gainesville, FL, 32611
Email: mythai@cise.ufl.edu

***Abstract*—Motivated by various settings, we study a new Influence Maximization problem at the Community level (IMC) which aims at finding k users to maximize the benefit of *influenced communities* where a community is influenced iff the number of influenced users belong to this community exceeds its predefined threshold. In general, IMC objective function is not submodular nor supermodular, thereby making it very challenging to apply existing greedy solutions of the classic influence maximization (IM) where submodular function is required. Furthermore, the major challenge in the traditional methods for any related IM problem is the inefficiency in estimating the influence spread. IMC brings this difficulty to a higher level when considering influenced communities instead of influencing each individual user. In this paper, we propose different approximation algorithms for IMC: (1) Using Sandwich approach with a tight submodular function to bound the IMC objective function, (2) Activating the top-k influencing nodes found from network sampling. Furthermore, when the activated thresholds of communities are bounded by a constant, we propose an algorithm with performance guarantee tight to the inapproximability of IMC assuming the exponential time hypothesis. Each algorithm has its own strengths in a trade-off between effectiveness and running time, which are illustrated both in theory and comprehensive experimental evaluation.**

***Index Terms*—Influence Maximization; Approximation Algorithms; Online Social Networks; Optimization; Viral Marketing**

I. Introduction

Information propagation in Online Social Networks (OSNs) has become a popular research topic recently. Social networks have been shown to be a useful tool for large scale behavioral manipulation and a social network user can be influenced by their friends to adopt certain behaviors. In this context, the Influence Maximization (IM) problem [1] has been studied extensively: Given a network and an integer k, IM asks for k influential users that maximizes the expected influence spread. IM has been widely adopted in viral marketing: companies target top k users by offering them free products or services with the hope to trigger a chain reaction of product adoption. A lot of research [2], [3], [4], [5] (and references therein) has been done to develop a scalable solution with $1 - 1/e - \epsilon$ performance guarantee for IM.

In this paper, we study a new problem, Influence Maximization at the Community level (IMC). Given a social network and a collection of disjoint set of users, we call each set of users is a community. An *influenced community* is a community in which the number of influenced users in this community exceeds its predefined threshold. A community, if being influenced, will bring back some benefit. The objective is to identify k seed nodes that maximize the expected benefit of influenced communities.

The IMC problem is motivated by many different applications, where communities are disjoint. As mentioned in the "Collaborative Based" concept on viral marketing: the product may be useful for a single user but is much more useful when used in a group context [6]. In this type of viral marketing, a certain group of users must be influenced to some degree (threshold) so the product or service can be adopted. Moreover, with an increasing in integrating social network to power grid for energy management, IMC could also be applied in the information attack of a power grid through a social network [7]. In this kind of attack, an adversary tries to influence a certain amount of electric users of each community in order to cause inter-area oscillations, causing cascading failure and devastating impact on large geographical areas. In this context, communities represent geographical neighborhoods, hence they are disjoint. Another readily application of IMC is in the context of election where each community represents a state of population.

As IM is a special case of IMC, we ask what will happen if we adopted IM's solutions to IMC. Indeed, most of IM's solutions [1], [2], [3], [4], [8] exploited a trait that IM exhibits submodular behavior, a critical property on how the $(1 - 1/e - \epsilon)$-performance guarantee could be acquired. IMC problem, on the other hand, is shown to be neither submodular nor supermodular, making IMC more challenging to devise an efficient algorithm. Moreover, computing the influence spread of a given seed set is a challenging task and IMC elevates this challenge while considering the impact to communities rather than individual independently. Motivated by these observations, the main contributions of this work are:

- We show that IMC is inapproximable within $O(\mathtt{r}^{1/(2(\log\log \mathtt{r})^c)})$ ratio and does not exhibit submodular behaviors, where $\mathtt{r}$ is the number of communities and $c > 0$ is a universal constant independent of $\mathtt{r}$.
- We present a new sampling technique, modifying from the Reverse Influence Sampling (RIS) [5] so that we take communities as the main subject and randomly sample deterministic instances of the social network in which a community can be influenced. This sampling technique then helps us acquire an estimation on the benefit of influenced communities given an initial seed set.
- We propose three approximation algorithms to solve IMC: (1) Using sandwich approach to provide upper bounded submodular function for the objective function,

2575-8411/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDCS.2019.00040

(2) Classifying communities/users by their impact on influencing the other and activating the most influencing users, and (3) When the activation threshold of each community is bounded, we obtain an $O(\mathtt{r}^{1/2})$-approximation algorithm, which is tight to the inapproximable ratio result assuming the exponential time hypothesis [9].
- We conduct extensive experiments using real world datasets to show the effectiveness of our proposed algorithms and illustrate the above trade-offs.

Related work. Kempe et al [1] first formulated IM and proved its NP-hardness. In addition, computing the expected influence spread for a given seed set is #P-hard. Series of studies have been done to approximate the influence maximization problem [2], [3], [4], [5], [8]. However, different to our work, all of these studies exploited IM's submodularity to obtain the ratio of $1 - 1/e - \epsilon$ and focused on improving the time complexity.

Some versions of non-submodular related-influenced maximization have been studied recently. Most notably, Lin et al. [10] introduced a k-boosting problem which aims at finding k users to boost so to trigger a maximized "boosted" influence spread. Lu et al. [11] proposed the Comparative Independent Cascade model where the IM under this model is no longer submodular. Both works utilized the Sandwich Approximation strategy, finding a submodular function that bounds the original objective function; the derived results associated with these relaxed functions are then used as the estimated solution to the original non-submodular problem. Furthermore, Ma et al [12] proposed a seed selection strategy using network graphical properties in another non-submodular diffusion model, called the influence barricade model. However, this method provides no theoretical approximation guarantee. All of these models still focused on maximizing influence spread. Different from the above studies, we study how the spread of influence makes an impact to communities given a seed set. The difference in objective function requires us to devise new approaches.

Organization. Section II formally defines the IMC problem and its challenges, including its inapproximability. In Section III, we propose a new approach to estimate the benefit of influenced communities given an initial seed set. Sections IV and V present our three approximation solutions to IMC. The experimental evaluation of our methods is demonstrated in Section VI. Finally, Section VII concludes the paper.

II. Models and Problem Definition

A. Propagation Model and Problem Definition

We abstract a social network using a weighted graph $G = (V, E, w)$ with $|V| = n$ nodes and $|E| = m$ directed edges. Each edge $e \in E$ is associated with a weight $w_e \in [0, 1]$ which indicates the probability that u influences v. By convention, $w_e = 0$ if $e \notin E$. Note that for each edge $e = (u, v) \in E$, the notation w_e and $w(u, v)$ are used interchangeably. Technically, we can view G as a generative model for deterministic graphs. A deterministic graph $\mathcal{G} = (V, E_{\mathcal{G}})$ is generated from G by selecting each edge $(u, v) \in E$, independently, with probability $w(u, v)$. We refer to $\mathcal{G}$ as a *sample graph* of G.

We use the fundamental diffusion model namely Independent Cascade (IC) [1]. Assume that we have a set of seed nodes S, the propagation processes happen in discrete rounds. At round 0, all nodes in S are active (influenced) and the others are inactive. At round $t \geq 0$, when a node u gets activated, initially or by another node, it has a single chance to activate each inactive neighbor v with the successful probability proportional to the edge weight $w(u, v)$. An activated node remains active till the end of the diffusion process. Note that even we only use IC model, similar to [2], [13] the solution can be easily extended to the Linear Threshold model [1].

Given a collection of disjoint sets $\mathtt{Com} = \{\mathtt{C_i} | 1 \leq i \leq \mathtt{r}, \mathtt{C}_i \subset V\}$, $\mathtt{r} = |\mathtt{Com}|$. We call each set $\mathtt{C}_i$ a community. Let $\mathtt{h}_i$ be *activation threshold* of community $\mathtt{C}_i$. A community is influenced if the number of activated users in this community exceeds its activation threshold. Denote b_i as a benefit if the community $\mathtt{C}_i$ gets influenced. Let $\mathtt{c}(S)$ be the expected benefit of influenced communities under a seed set S and a given propagation model. We formally define IMC as follows:

Definition 1. (IMC) *Given a graph $G = (V, E, w)$ an integer $1 \leq k \leq |V|$, a propagation model and a set of communities $\mathtt{Com}$. The* IMC *problem asks for a seed set $S \subset V$ of k nodes that maximizes the expected benefit of influenced communities $\mathtt{c}(S)$ under the given propagation model.*

B. Challenges of the ICM Problem

We now provide key properties and challenges of the IMC problem. Because IM is a special case of IMC, IMC is also NP-hard and computing $\mathtt{c}(S)$ given S is #P-hard. Moreover, we show the following inapproximable result, indicating the complexity of IMC.

Theorem 1. *Assuming the exponential time hypothesis [9], there is no polynomial time algorithm that approximates IMC to within $O(\mathtt{r}^{1/2(\log\log \mathtt{r})^c})$ factor of the optimum, where $c > 0$ is a universal constant independent of $\mathtt{r}$.*

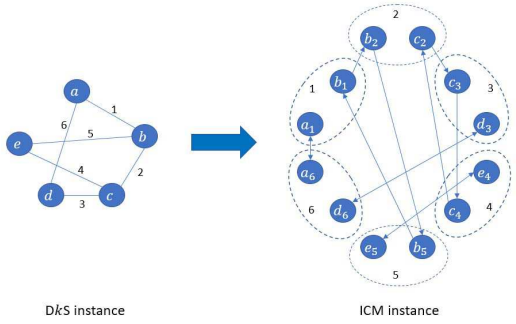


Fig. 1: Conversion from DkS instance to IMC instance. The labels on the edges of DkS graph are corresponding to the labels on communities of IMC instance.

Proof. We consider the following *Densest k-Subgraph* (DkS) problem: given a connected undirected graph G and an integer k, the goal is to find a set S of k-nodes that an induced subgraph of G from S contains maximum number of edges. Let $e(S)$ be the number of edges in induced subgraph of G from S. DkS has been shown to be inapproximable within factor of $O(n^{1/(\log\log n)^c})$ [14] assuming the exponential time hypothesis [9]. To reach the relation between the inapproximability of DkS and IMC, given an instance (G_D, k) of DkS, we construct an instance $(G_I, \mathtt{Com}, k)$ of IMC as follows: For each edge $e = \{a, b\} \in E$, we create a community $\mathtt{C}_e$ with activation threshold $\mathtt{h}_e = 2$ and two nodes a_e and b_e in $\mathtt{C}_e$. Node $a(b)$ is called *corresponding node* of $a_e(b_e)$. By that

construction, a multiple nodes in G_I would have the same corresponding node in G_D. Let $\mathtt{U}_a$ be a set of nodes having the same corresponding node a. We create edges connecting between nodes in $\mathtt{U}_a$ such that $\mathtt{U}_a$ is strongly connected and each edge is weighted by 1. Figure 1 shows a simple conversion from DkS instance to ICM instance. Let S_D and S_I are optimal solutions of the DkS and IMC instance. We have following observations:

- We create a solution S'_I for the IMC instance by: for each node $a \in S_D$, pick an arbitrary node in $\mathtt{U}_a$ and put into S'_I. Because $\mathtt{U}_a$ is strongly connected, all nodes in $\mathtt{U}_a$ will be activated. Hence, all communities, corresponding to the edges in the induced subgraph of S_D, are influenced. So we have: $e(S_D) = \mathtt{c}(S'_D) \leq \mathtt{c}(S_I)$
- We create a solution S'_D for the DkS instance by: for each node $u \in S_I$, pick its corresponding node in G_D and put into S'_D. Consider a community $\mathtt{C}_e$ which is influenced by S_I, both two node $a_e, b_e \in \mathtt{C}_e$ are activated, which means that at least one node in $\mathtt{U}_a$ and one node in $\mathtt{U}_b$ are in S_I. So a, b will be put into S'_D and edge $e = \{a, b\}$ is in the induced subgraph of S'_D. So we have: $\mathtt{c}(S_I) = e(S'_D) \leq e(S_D)$.

Hence, $e(S_D) = c(S_I)$. Assuming there is a $O(\mathtt{r}^{1/2(\log\log \mathtt{r})^c})$ approximation algorithm K for IMC problem. Denote S_I^{K} as K's solution for IMC instance. We create solution S_D^{K} for DkS instance by picking all corresponding node to S_I^{K}. It is trivial to see that:

$$e(S_D^{\mathsf{K}}) = \mathtt{c}(S_I^{\mathsf{K}}) \geq O(\mathtt{r}^{-1/2(\log\log \mathtt{r})^c})\mathtt{c}(S_I) \quad (1)$$

$$\geq O(\mathtt{r}^{-1/2(\log\log \mathtt{r})^c})e(S_D) \quad (2)$$

During the construction of two instances, we have $\mathtt{r} = |G_D.E| \leq O(n^2)$ while $\mathtt{r} = O(n)$. Replacing into equation (1), we observe that: By using K, we could achieve a solution S_D^{K} for the DkS problem which guarantees:

$$e(S_D^{\mathsf{K}}) \geq O(n^{-1/(\log\log n)^c})e(S_D)$$

which contradicts to the inapproximability of DkS problem. □

Non-submodularity of $\mathtt{c}(\cdot)$. In IM, the expected influence of the seed set S is a monotone and submodular function. Thus, a natural greedy algorithm returns a solution with $1-1/e$ approximation ratio. However, the objective function $\mathtt{c}(S)$ in IMC is *neither submodular nor supermodular* on the set S of initial activated nodes. Consider each community needs only one node to be influenced, then $\mathtt{c}(\cdot)$ exhibits a *submodular behavior*. On the other hand, when several nodes are activated initially, their benefit of influenced communities can be more than the sum of their individual activation effect, which is a *supermodular behavior*. To illustrate, consider an example in Fig. 2, each edge has weight 0.3 and each community has the activation threshold 2. Therefore, we have $\mathtt{c}(\emptyset) = 0, \mathtt{c}(\{a\}) = 0.327, \mathtt{c}(\{b\}) = 0.39, \mathtt{c}(\{a, b\}) = 1.09$. Hence, $\mathtt{c}(\{b\}) - \mathtt{c}(\emptyset) < \mathtt{c}(\{a, b\}) - \mathtt{c}(\{a\})$, which means $\mathtt{c}(\cdot)$ does not exhibit submodular behavior. The non-submodularity of $\mathtt{c}(\cdot)$ means that the seed set returned by the greedy algorithm may not have the $(1 - 1/e)$-approximation ratio.

Differences to IM. Two notable state-of-the-art influence maximization frameworks are Influence Maximization via Martigale (IMM) [4] and SSA/DSSA [2]. Both of them are based on the sampling method called Reverse Influence Sampling (RIS). Briefly, RIS captures the influence landscape of G by: for a node $v \in V$, a random set R of nodes is generated such that for any seed set S, the probability that $R \cap S \neq \emptyset$ equals the probability that v can be influenced by S. If v is selected uniform randomly from V, the expected influence of any seed set $S \subseteq V$ equals $n \cdot \mathbb{E}[\mathtt{1}_{R \cap S \neq \emptyset}]$ where expectation is taken over the randomness of R. Then, both SSA and IMM used the greedy algorithm for the max-coverage problem to find a seed set that covers the maximum number of sampling sets.

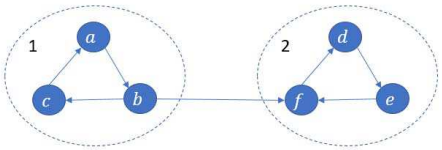


Fig. 2: Non-submodular example

However, the IMC problem is fundamentally different due to the objective function that considers influenced communities instead of individuals separately. RIS seems to be inapplicable in IMC because we find no relationship between the randomly selected node in RIS with the community it belongs to and how it can contribute to estimate the expected benefit of influenced communities. Although such challenge can be solved by subtly modifying RIS, the most challenging task is that IMC's objective function is non-submodular and inapproximable within $O(\mathtt{r}^{1/(2(\log\log \mathtt{r})^c)})$. Hence, using greedy solutions for the max-coverage problem given a collection of samples will not provide $1 - 1/e$ approximation ratio. Therefore, we need to devise efficient approximation algorithms based on the new sampling technique.

III. Benefit Estimation

In this section, we propose a sampling method, called *Reverse Influenceable Community* (RIC), to estimate the expected benefit of influenced communities given a seed set S.

We denote $\rho : \mathtt{Com} \to \mathbb{R}^{\geq}$ as a probability distribution among Com, where $\rho(\mathtt{C}_i) = \frac{b_i}{\mathtt{b}}$ and $\mathtt{b} = \sum_{j=1}^{\mathtt{r}} b_i$. Given a sample graph $\mathcal{G}$ of G, a community C is called being *touched* by $u \in V$ (or u *touches* C) in $\mathcal{G}$ if there exist a path in $\mathcal{G}$ that connects u to any nodes in C. The RIC sample is defined as follows:

Definition 2. *(RIC) Given $G = (V, E, w)$, Com, a random RIC sample g is generated by: 1) selecting a random community $\mathtt{C}_g$ given probability distribution ρ; 2) generating a sample graph $\mathcal{G}_g$ from G and 3) returning a set of nodes that touch $\mathtt{C}_g$ in $\mathcal{G}_g$. $\mathtt{C}_g$ is called the "source" community of g.*

Fig. 3 shows an example of RIC-sample g. The "source" community $\mathtt{C}_g$ is marked by the dotted circle and contains node $\{v_1, v_2, v_3, v_4\}$. $\mathtt{h}_g = 3$ is an activation threshold of $\mathtt{C}_g$. In this example, we need at least 3 activated users among $\{v_1, v_2, v_3, v_4\}$ to make g influenced. As definition 2, g contains all nodes (even outside $\mathtt{C}_g$) that connect to the nodes in $\mathtt{C}_g$, including v_5, v_6 and v_7. For simplicity, if $\mathtt{C}_g$ is touches by u in $\mathcal{G}_g$, we also call u touches g or g is touched by u.

Alg. 1 presents in detail how a RIC sample is generated. For each $u \in \mathtt{C}_g$, $\mathtt{R}_g(u)$ is a reachable set of node u, which is defined as a set of nodes $v \in V$ that can reach u in $\mathcal{G}_g$ (there exists a path from v to u). First, $\mathtt{C}_g$ is randomly selected given probability distribution ρ. Next, we mark each edge e in G with one of three states: (1) $\perp$ - e has potential to be in $\mathcal{G}_g$,

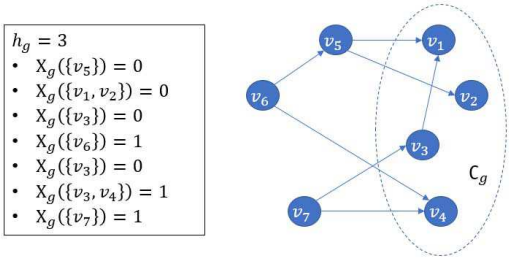


Fig. 3: Example of RIC-sample. Additionally, we also store $\mathtt{R}_g(\cdot)$ values of v_1, v_2, v_3, v_4, i.e $\mathtt{R}_g(v_1) = \{v_1, v_3, v_5, v_6, v_7\}$

(2) y - e is in $\mathcal{G}_g$, and (3) n - e is not in $\mathcal{G}_g$. Each edge's state is set to be $\perp$ initially and will be changed to y or n when we first process it. This state information is stored by $\mathtt{st}[\cdot]$ array. The algorithm then works in backward Breadth-First Search (BFS) manner as follows: First, we put all nodes in $\mathtt{C}_g$ into Q - a queue storing nodes we are going to visit. While Q is not empty, we take a node u out of Q and mark it as "visited" ($\mathtt{ck}[u] = \mathtt{true}$ if u is "visited", false otherwise). Next, we consider all nodes v that u has incoming edge ($e = (v, u)$) with. If $\mathtt{st}[e] = \perp$, edge e is never processed before, so we randomly generate edge $e = (v, u)$ with probability w_e. If edge $e = (v, u)$ exists in $\mathcal{G}_g$ and v does not exist in Q as well as was not visited before, we put v into Q. This process is repeated until Q is empty and we will have a deterministic sample graph $\mathcal{G}_g$ that contains paths by which the nodes in $\mathtt{C}_g$ could be activated. Finally, for all $u \in \mathtt{C}_g$, we determine $\mathtt{R}_g(u)$ by using Depth-First Search with graph $\mathcal{G}_g$.

Estimating c(S). We now present how we use the RIC-sampling to estimate $\mathtt{c}(\cdot)$. Let g be a given RIC sample with the source community $\mathtt{C}_g$ and the activation threshold $\mathtt{h}_g$. Given a seed set S, we say that g is influenced by S if S could reach at least $\mathtt{h}_g$ nodes in $\mathtt{C}_g$. In Fig. 3, g is influenced by $\{v_5, v_6\}$ or $\{v_7\}$ but not by $\{v_1\}$ or $\{v_1, v_4\}$. Define $\mathtt{X}_g(S) : 2^V \to \{0, 1\}$ as an indicator function whether g is influenced by S.

Algorithm 1 Generate a random RIC graph

Input A social graph G, set of communities Com
Output Source community $\mathtt{C}_g$ and its activation threshold $\mathtt{h}_g$, *reachable* set $\mathtt{R}_g(u) \quad \forall u \in \mathtt{C}_g$

```
Select source community C_g randomly
Create a graph G_g with all nodes in G and no edge
st[e_uv] = ⊥ ∀(u, v) ∈ E
Initiate empty queue Q
Put all nodes v ∈ C_g into Q
while Q is not empty do
    u ← Q.dequeue()
    ck[u] ← true
    for each in-comming edge e_vu of u do
        if st[e_vu] = ⊥ then
            st[e_vu] = y with prob w_e : n otherwise
            if st[e_vu] = y then create edge e_vu in G_g
        if st[e_vu] = y & !ck[v] & !Q.contain(v) then
            Q.enqueue(v)
for each node u ∈ C_g do
    Run DFS to find R_g(u)
```

Return $\mathtt{C}_g, \mathtt{h}_g, \mathtt{R}_g(u) \quad \forall u \in \mathtt{C}_g$

Lemma 1. *For any $S \subseteq V$, we have $\mathtt{c}(S) = \mathtt{b} \cdot \mathbb{E}[\mathtt{X}_g(S)]$ where the expectation is taken over the randomness of g.*

Proof. Because $\mathtt{X}_g(S)$ is an indicator variable, then $\mathbb{E}[\mathtt{X}_g(S)] = \Pr[\mathtt{X}_g(S) = 1]$. We have:

$$\Pr[\mathtt{X}_g(S) = 1] = \sum_{\mathtt{C}_i \in \mathtt{Com}} \Pr[\mathtt{C}_i = \mathtt{C}_g] \cdot \Pr[\mathtt{C}_i \text{ is influenced by } S]$$
$$= \sum_{\mathtt{C}_i \in \mathtt{Com}} \frac{b_i}{\mathtt{b}} \Pr[\mathtt{C}_i \text{ is influenced by } S] = \frac{1}{\mathtt{b}} \mathtt{c}(S)$$

which completes the proof. □

Let $\mathcal{R}$ be a set of independent random RIC samples, define:

$$\hat{\mathtt{c}}_{\mathcal{R}}(S) = \frac{\mathtt{b}}{|\mathcal{R}|} \cdot \sum_{g \in \mathcal{R}} \mathtt{X}_g(S) \quad \forall S \subseteq V \tag{3}$$

Intuitively, $\hat{\mathtt{c}}_{\mathcal{R}}(S)$ closely estimates $\mathtt{c}(S)$ for any $S \subseteq V$ if $|\mathcal{R}|$ is sufficiently large. Therefore, IMC could be solved by generating a large collection $\mathcal{R}$ of RIC samples, then finding a seed set S that influences the maximum number of RIC graphs in $\mathcal{R}$. We define a problem of finding S given a collection $\mathcal{R}$ of RIC samples - the MAXR problem, which is essentially different with the max-coverage problem solved in IM.

Definition 3. *(MAXR) Given a set $\mathcal{R}$ of RIC samples, find a seed set $S \subset V$ of k nodes that maximizes the number of influenced RIC samples in $\mathcal{R}$. The RIC sample g is called influenced by S if S can reach at least $\mathtt{h}_g$ nodes in $\mathtt{C}_g$.*

By this definition, we observe that the objective function of MAXR is equivalent to $\hat{\mathtt{c}}_{\mathcal{R}}(\cdot)$. So from now on, we will also call $\hat{\mathtt{c}}_{\mathcal{R}}(\cdot)$ the objective function of MAXR.

Clearly, MAXR is NP-hard as max-coverage is a special case. Moreover, the additional challenge of solving MAXR is presented by the following lemma, indicating that a natural greedy algorithm cannot solve MAXR either.

Lemma 2. *$\hat{\mathtt{c}}_{\mathcal{R}}(\cdot)$ is non-submodular.*

Proof. We consider a simple instance where $\mathcal{R}$ contains only one RIC graph g. The source community $\mathtt{C}_g$ has only 2 nodes u, v and $\mathtt{R}_g(u) = \{u\}, \mathtt{R}_g(v) = \{v\}$. We have $\hat{\mathtt{c}}_{\mathcal{R}}(\emptyset) = 0, \hat{\mathtt{c}}_{\mathcal{R}}(\{u\}) = 0, \hat{\mathtt{c}}_{\mathcal{R}}(\{v\}) = 0$ and $\hat{\mathtt{c}}_{\mathcal{R}}(\{u, v\}) = 1$. Therefore, $\hat{\mathtt{c}}_{\mathcal{R}}(\{u\}) - \hat{\mathtt{c}}_{\mathcal{R}}(\emptyset) < \hat{\mathtt{c}}_{\mathcal{R}}(\{u, v\}) - \hat{\mathtt{c}}_{\mathcal{R}}(\{v\})$ □

Overall, **two critical pieces** needed to solve IMC are:

- Solving the MAXR problem to find a seed set S that influences the maximum number of RIC samples.
- Generating a sufficiently large number of RIC samples such that the returned set S from the MAXR problem guarantees the bounded error.

IV. Algorithms for the MAXR problem

In this section, we tackle the first piece by devising three approximation algorithms to solve the MAXR problem.

A. Upper Bound Greedy (UBG)

The first algorithm is called **U**pper **B**ounded **G**reedy. Overall, UBG uses a concept of *Sandwich Approximation* [11] by bounding $\hat{\mathtt{c}}_{\mathcal{R}}(\cdot)$ with a submodular function. The most challenging task is to devise a submodular function that is tight to our objective function. In this solution, we propose submodular functions $\nu(\cdot), \nu_{\mathcal{R}}(\cdot)$ to bound $\mathtt{c}(\cdot), \hat{\mathtt{c}}_{\mathcal{R}}(\cdot)$ respectively and prove the tightness by observing that those functions overlap our objective function in a special case. The Sandwich Approximation is defined as follows:

Algorithm 2 UBG algorithm

Input $\mathcal{R}, k$
Output S

1: $S_\nu \leftarrow$ greedy selection with objective function $\nu_{\mathcal{R}}(\cdot)$
2: $S_{\mathtt{c}} \leftarrow$ greedy selection with objective function $\hat{\mathtt{c}}_{\mathcal{R}}(\cdot)$
3: $S \leftarrow \arg\max_{S_\nu, S_{\mathtt{c}}} \hat{\mathtt{c}}_{\mathcal{R}}(S)$

Return S

Theorem 2. *(*Sandwich Approximation [11]*) Let $\sigma : 2^V \rightarrow \mathbf{R}$ be a non-submodular. Let μ and ν be submodular and defined on the same ground set V such that $\mu(S) \leq \sigma(S) \leq \nu(S)$ for all $S \subseteq V$. That is $\mu(\cdot)(\nu(\cdot))$ is a lower (resp., upper) bound on σ everywhere. Consider the problem of maximizing σ subject to a cardinality constraint k: run the greedy algorithm on all three functions. It produces an approximate solution for μ and ν. Let S_σ, S_μ, S_ν be the solution obtained from σ, μ, ν respectively. Then, select the final solution to be*

$$S_{\mathtt{sand}} = \arg\max_{S \in \{S_\sigma, S_\mu, S_\nu\}} \sigma(S) \tag{4}$$

Then we have:

$$\sigma(S_{\mathtt{sand}}) \geq \max\left\{\frac{\sigma(S_\nu)}{\nu(S_\nu)}, \frac{\mu(S^*_\sigma)}{\sigma(S^*_\sigma)}\right\}(1-1/e)\sigma(S^*_\sigma) \tag{5}$$

Therefore, we have to find at least one of $\mu(\cdot)$ or $\nu(\cdot)$ that should be close to our objective function. In this work, we derive a submodular upper bound $\nu_{\mathcal{R}}(\cdot)$ of $\hat{\mathtt{c}}_{\mathcal{R}}(\cdot)$. $\nu_{\mathcal{R}}(\cdot)$ is considerably closer to $\hat{\mathtt{c}}_{\mathcal{R}}(\cdot)$ than any other submodular bounded functions we have tested. Define:

$$\nu(S) = \mathtt{b} \cdot \mathbb{E}\Big[\min(\frac{|\mathtt{I}_g(S)|}{\mathtt{h}_g}, 1)\Big] \tag{6}$$

$$\nu_{\mathcal{R}}(S) = \frac{\mathtt{b}}{|\mathcal{R}|}\sum_{g\in\mathcal{R}} \min\Big(\frac{|\mathtt{I}_g(S)|}{\mathtt{h}_g}, 1\Big) \tag{7}$$

where $\mathtt{I}_g(S)$ is the a set of nodes in $\mathtt{C}_g$ that can be connected by S in RIC sample g.

$\nu(S)$ presents the expected fractional benefit from being influenced of all communities while $\nu_{\mathcal{R}}(S)$ provides an estimation on $\nu(S)$ from the sample set $\mathcal{R}$. Recalling that RIC sample g is called influenced if $|\mathtt{I}_g(S)| \geq \mathtt{h}_g$. Therefore, to display the fractional value of being influenced, we take min value between $|\mathtt{I}_g(S)|/\mathtt{h}_g$ and 1.

Lemma 3. *For any set $S \subseteq V$, $\mathtt{c}(S) \leq \nu(S)$. Also, for a given collection $\mathcal{R}$ of* RIC *samples, $\hat{\mathtt{c}}_{\mathcal{R}}(S) \leq \nu_{\mathcal{R}}(S)$.*

Proof. First, let compare $\hat{\mathtt{c}}_{\mathcal{R}}(\cdot)$ and $\nu_{\mathcal{R}}(\cdot)$. Given $g \in \mathcal{R}$, if $\mathtt{X}_g(S) = 1$ then $|\mathtt{I}_g(S)| \geq \mathtt{h}_g$ and $\min\big(|\mathtt{I}_g(S)|/\mathtt{h}_g, 1\big) = 1$. Otherwise $\mathtt{X}_g(S) = 0$, but $\min\big(|\mathtt{I}_g(S)|/\mathtt{h}_g, 1\big) \geq 0$. Therefore, $\hat{\mathtt{c}}_{\mathcal{R}}(S) \leq \nu_{\mathcal{R}}(S), \quad \forall S \subseteq V$.

To prove $\nu(\cdot)$ is an upper bound of $\mathtt{c}(\cdot)$, recalling $\mathtt{c}(S) = \mathtt{b} \cdot \mathbb{E}[\mathtt{X}_g(S)]$ where:

$$\mathbb{E}[\mathtt{X}_g(S)] = \mathbb{E}[\mathbf{1}_{|\mathtt{I}_g(S)|\geq \mathtt{h}_g}] = \Pr[|\mathtt{I}_g(S)|/\mathtt{h}_g \geq 1]$$

Without loosing generality, we have:

$$\Pr[|\mathtt{I}_g(S)|/\mathtt{h}_g \geq 1] = \Pr[\min(|\mathtt{I}_g(S)|/\mathtt{h}_g, 1) \geq 1]$$

Using Markov inequality [15], we have:

$$\Pr[\min(|\mathtt{I}_g(S)|/\mathtt{h}_g, 1) \geq 1] \leq \mathbb{E}[\min(|\mathtt{I}_g(S)|/\mathtt{h}_g, 1)] \tag{8}$$

So $\mathtt{c}(\cdot)$ is upper bounded by $\nu(\cdot)$. □

Considering an RIC sample g, it is trivial that $|\mathtt{I}_g(\cdot)|$ is a *submodular* function, which also means $\nu_{\mathcal{R}}(S) = \frac{\mathtt{b}}{|\mathcal{R}|}\sum_{g\in\mathcal{R}} \min\Big(\frac{|\mathtt{I}_g(S)|}{\mathtt{h}_g}, 1\Big)$ is a *submodular* function. Furthermore, $\nu_{\mathcal{R}}(\cdot)$ overlaps $\hat{\mathtt{c}}_{\mathcal{R}}(\cdot)$ when the activation thresholds of communities are bounded by 1. We have the following lemma.

Algorithm 3 MAF algorithm

Input $\mathcal{R}, k$
Output S

1: Initiate $S_1, S_2 \leftarrow \emptyset$
2: SC $\leftarrow$ sorted list of Com in order of their appearance in $\mathcal{R}$
3: **while** SC *is not empty* **do**
4: $\quad$ C $\leftarrow$ take out 1st community of SC;
5: $\quad$ X $\leftarrow$ pick h nodes in C
6: $\quad$ **if** $|S_1 \cup \mathtt{X}| \leq k$ **then** $S_1 = S_1 \cup \mathtt{X}$
7: $S_2 \leftarrow k$ nodes that appear the most in $\mathcal{R}$
8: $S = \arg\max_{S' \in \{S_1, S_2\}} \hat{\mathtt{c}}_{\mathcal{R}}(S')$

Return S

Lemma 4. *For any set $S \subseteq V$, $\mathtt{c}(S) = \nu(S)$ and $\hat{\mathtt{c}}_{\mathcal{R}}(S) = \nu_{\mathcal{R}}(S)$ if $\mathtt{h}_g = 1$ for all $g \in \mathcal{R}$.*

Proof. If $\mathtt{h}_g = 1$ for all g, then

$$\min(\frac{|\mathtt{I}_g(S)|}{\mathtt{h}_g}, 1) = \min(|\mathtt{I}_g(S)|, 1)$$
$$= \mathbf{1}_{|\mathtt{I}_g(S)|\geq 1} = \mathtt{X}_g(S)$$

the lemma follows. □

Lemma 4 also implies that $\mathtt{c}(\cdot)$ and $\hat{\mathtt{c}}_{\mathcal{R}}(\cdot)$ are submodular if the activation thresholds are bounded by 1.

Our UBG algorithm to MAXR is presented in Alg. 2. Denote S_ν as a solution obtained from the greedy algorithm with objective function $\nu_{\mathcal{R}}(\cdot)$. By Theorem 2, UBG guarantees $\frac{\hat{\mathtt{c}}_{\mathcal{R}}(S_\nu)}{\nu_{\mathcal{R}}(S_\nu)}(1-1/e)$ approximation ratio to the MAXR problem.

*B. Most Appearance First (*MAF*)*

The second algorithm we present is called **M**ost-**A**ppearance-**F**irst (MAF). Accordingly, the idea of MAF is that we compute the frequency of appearance of communities or nodes in $\mathcal{R}$, then try to activate the most influential one. First, MAF considers the communities that appear the most in $\mathcal{R}$. The algorithm starts by determining frequency each community in Com appears in $\mathcal{R}$ then sorts these communities in descending order of their frequency. Let SC be a sorted list of communities. MAF then sequentially takes the community C with highest frequency out of SC. Let h be the activation threshold of C. MAF then randomly picks h nodes in C and puts into S_1. This process then repeats until $|S_1| = k$. Next, MAF considers nodes that appear the most in $\mathcal{R}$. A node v is considered to *appear* in RIC sample g if v is either in $\mathtt{C}_g$ or there exists $u \in \mathtt{C}_g$ that $v \in \mathtt{R}_g(u)$. MAF then picks k nodes that appear the most in $\mathcal{R}$ and put into S_2. Finally, MAF returns a solution $S \in \{S_1, S_2\}$ that influences the most RIC samples in $\mathcal{R}$. The full MAF is presented by Alg. 3.

Theorem 3. MAF *returns a $\frac{1}{\mathtt{r}}\lfloor\frac{k}{h}\rfloor$ approximate result to the* MAXR *problem, where $h = \max_{i=1}^{\mathtt{r}} \mathtt{h}_i$*

Proof. Denote $S_{\mathtt{OPT}}$ an optimal solution to MAXR, we have:

$$\hat{\mathtt{c}}_{\mathcal{R}}(S_1) = \frac{\mathtt{b}}{|\mathcal{R}|}\sum_{g\in\mathcal{R}} \mathtt{X}_g(S_k) \quad \geq \frac{\mathtt{b}}{|\mathcal{R}|}\cdot\frac{1}{\mathtt{r}}\cdot\Big\lfloor\frac{k}{h}\Big\rfloor\cdot|\mathcal{R}| \tag{9}$$

$$\geq \frac{1}{\mathtt{r}}\Big\lfloor\frac{k}{h}\Big\rfloor\cdot\frac{\mathtt{b}}{|\mathcal{R}|}\sum_{g\in\mathcal{R}} \mathtt{X}_g(S_{\mathtt{OPT}}) \geq \frac{1}{\mathtt{r}}\Big\lfloor\frac{k}{h}\Big\rfloor\hat{\mathtt{c}}_{\mathcal{R}}(S_{\mathtt{OPT}}) \tag{10}$$

The inequality (9) comes from the observation that MAF chooses to influence communities appearing the most in $\mathcal{R}$ and the "budget" to influence each community is at most

$h = \max_{i=1}^{\mathtt{r}} \mathtt{h}_i$. Therefore, there are at least $|\mathcal{R}|\frac{1}{\mathtt{r}}\lfloor\frac{k}{h}\rfloor$ RIC samples which are influenced by the seed set S_1.

S_2, on the other hand, does not provide any approximation guarantee. We consider an example as follows: Assume $\mathcal{R}$ contains 6 RIC samples $\{g_1, g_2, ..g_6\}$ in which $\mathtt{C}_{g_i}$s are disjoint sets, $|\mathtt{C}_{g_i}| = 3$ and $\mathtt{h}_{g_i} = 2$ for all $i \in [1, 6]$. Given a node $u \notin \mathtt{C}_{g_i}$ for $i \in [1, 6]$, there exists an edge in g_i that connects u with a node in $\mathtt{C}_{g_i}$ for $i \in \{1, 2, 3\}$. Similarly, given a node $v \notin \mathtt{C}_{g_i}$ for $i \in [1, 6]$, there exists an edge in g_j that connects v with a node in $\mathtt{C}_{g_j}$ for $j \in \{4, 5, 6\}$. Other than these, there is no other edges in g_i for $i \in [1, 6]$. Let $k = 2$. In this scenario, u, v are nodes that appear the most in $\mathcal{R}$, thus $S_2 = \{u, v\}$. However $\hat{\mathtt{c}}_{\mathcal{R}}(S_2) = 0$ since each community $\mathtt{C}_{g_i}$ required at least 2 nodes to be influenced while S_2 only guarantees to influence one node in each community. The optimal solution is $\hat{\mathtt{c}}_{\mathcal{R}}(S_{\mathtt{OPT}}) = 1$. Therefore, S_2 does not provide any approximation guarantee to MAXR. However, S_2 actually performs well in experiments. By selecting the best among S_1, S_2, we can bound the performance guarantee of MAF and integrate MAF into IMC framework, which will be presented in next section. □

C. Algorithm for Bounded Activation Threshold

Having considered two approximation algorithms to MAXR, we now propose solutions with a tight approximation ratio for a special case where the activation threshold of communities is bounded by 2, which means a community needs at least 2 or 1 activated users to be influenced. Then we extend our solution to a case that the input network has bounded activation thresholds by a constant $d \geq 2$.

Denote $\mathtt{I}_g(S)$ as a set of nodes in $\mathtt{C}_g$ that can be connected by S in the RIC sample g, $\mathtt{I}_g(S) = \{u \mid u \in \mathtt{C}_g, \mathtt{R}_g(u) \cap S \neq \emptyset\}$. Next, let $\mathcal{G}_{\mathcal{R}}(u)$ be a set of RIC samples in $\mathcal{R}$ that u can touch - $\mathcal{G}_{\mathcal{R}}(u) = \{g \mid g \in \mathcal{R} : \exists v \in \mathtt{C}_g, u \in \mathtt{R}_g(v)\}$; and $\mathcal{D}_{\mathcal{R}}(S, u)$ is a set of RIC samples in $\mathcal{G}_{\mathcal{R}}(u)$ that S can influence, $\mathcal{D}_{\mathcal{R}}(S, u) = \{g \mid g \in \mathcal{G}_{\mathcal{R}}(u), \mathtt{I}_g(S) \geq \mathtt{h}_g\}$. Before going further, we have the following lemma.

Lemma 5. *Given a seed set $S \subseteq V$, we have*

$$\max_{u \in S} |\mathcal{D}_{\mathcal{R}}(S, u)| \leq \sum_{g \in \mathcal{R}} \mathtt{X}_g(S) \leq \sum_{u \in S} |\mathcal{D}_{\mathcal{R}}(S, u)|$$

Proof. The first inequality is trivial because $\mathcal{D}_{\mathcal{R}}(S, u)$ is a subset of the RIC samples set that S can influence. The second inequality comes from observing that: a sample g, if being influenced by S, would appear on at least one set $\mathcal{D}_{\mathcal{R}}(S, u)$ for all $u \in S$. □

Lemma 5 gives us an observation that: the number of influenced RIC samples by S is lower bounded by the size of $\mathcal{D}_{\mathcal{R}}(S, u)$ with any $u \in S$ separately but being upper bounded by sum of $|\mathcal{D}_{\mathcal{R}}(S, u)|$ $\forall u \in S$. This motivates us to propose the BT algorithm, which is presented in Alg. 4.

The idea of BT is that: for each node $u \in V$, we find a seed set $\mathtt{K}(u)$ that maximizes the number of influenced RIC samples in $\mathcal{G}_{\mathcal{R}}(u)$. To find $\mathtt{K}(u)$, we first add u in $\mathtt{K}(u)$. Because every RIC sample $g \in \mathcal{G}_{\mathcal{R}}(u)$ is touched by u and $\mathtt{h}_g \leq 2$, to make g be influenced by $\mathtt{K}(u)$, we only need at most one more node in $\mathtt{C}_g$ to be connected by $\mathtt{K}(u)$ beside the one that is already connected by u. That brings us back to the case when the activation threshold is bounded by 1 and a natural greedy algorithm can return $(1 - 1/e)$-approximation result for $\mathtt{K}(u)$.

Algorithm 4 BT algorithm

Input $\mathcal{R}, k$
Output S

1: **for** *each node* $u \in V$ **do**
2: $\quad \mathtt{G} \leftarrow$ Copy of $\mathcal{G}_{\mathcal{R}}(u)$
3: $\quad$ **for** *each* RIC *sample* $g \in \mathtt{G}$ **do**
4: $\quad\quad$ **for** *each node* $v \in \mathtt{C}_g$ **do**
5: $\quad\quad\quad$ **if** $u \in \mathtt{R}_g(v)$ **then**
6: $\quad\quad\quad\quad$ Remove v out of g
7: $\quad\quad\quad\quad \mathtt{h}_g = \mathtt{h}_g - 1$
8: $\quad \mathcal{T} \leftarrow$ greedily select $k - 1$ nodes with collection $\mathtt{G}$
9: $\quad \mathtt{K}(u) \leftarrow \{u\} \cup \mathcal{T}$
10: $S = \arg\max_{\mathtt{K}(u); u \in V} |\mathcal{D}_{\mathcal{R}}(\mathtt{K}(u), u)|$

Return S

BT returns a seed set S which is $\mathtt{K}(u)$ that has the maximum $|\mathcal{D}_{\mathcal{R}}(\mathtt{K}(u), u)|$. The approximate guarantee of BT is obtained by the following theorem.

Theorem 4. BT *provides a $\frac{1}{k}(1 - \frac{1}{e})$ approximation guarantee to the* MAXR *problem if the activation threshold of each community is at most 2.*

Proof. Denote $S_{\mathtt{OPT}}$ as an optimal solution for the MAXR problem. For any $u \in S_{\mathtt{OPT}}$, the natural greedy algorithm guarantees

$$|\mathcal{D}_{\mathcal{R}}(\mathtt{K}(u), u)| \geq (1 - 1/e) \max_{T \subset V; |T| = k-1} |\mathcal{D}_{\mathcal{R}}(T \cup \{u\}, u)|$$
$$\geq (1 - 1/e)|\mathcal{D}_{\mathcal{R}}(S_{\mathtt{OPT}}, u)|$$

Denote $\mathtt{v} = \arg\max_{u \in S_{\mathtt{OPT}}} |\mathcal{D}_{\mathcal{R}}(S_{\mathtt{OPT}}, u)|$ and $\mathtt{u} = \arg\max_{u \in V} |\mathcal{D}_{\mathcal{R}}(\mathtt{K}(u), u)|$. According to BT, $S = \mathtt{K}(\mathtt{u})$. Therefore, we have:

$$\hat{\mathtt{c}}_{\mathcal{R}}(S_{\mathtt{OPT}}) = \frac{\mathtt{r}}{|\mathcal{R}|} \sum_{g \in \mathcal{R}} \mathtt{X}_g(S_{\mathtt{OPT}}) \leq \frac{\mathtt{r}}{|\mathcal{R}|} \sum_{u \in S_{\mathtt{OPT}}} |\mathcal{D}_{\mathcal{R}}(S_{\mathtt{OPT}}, u)|$$
$$\leq \frac{\mathtt{r}}{|\mathcal{R}|} k \cdot |\mathcal{D}_{\mathcal{R}}(S_{\mathtt{OPT}}, \mathtt{v})| \leq \frac{\mathtt{r}}{|\mathcal{R}|} \frac{k}{1 - 1/e} |\mathcal{D}_{\mathcal{R}}(\mathtt{K}(\mathtt{v}), \mathtt{v})|$$
$$\leq \frac{k}{1 - 1/e} \frac{\mathtt{r}}{|\mathcal{R}|} |\mathcal{D}_{\mathcal{R}}(\mathtt{K}(\mathtt{u}), \mathtt{u})| \leq \frac{k}{1 - 1/e} \hat{\mathtt{c}}_{\mathcal{R}}(S)$$

which completes the proof. □

In Alg. 4, BT starts by iterating through every node $u \in V$. For each node, BT creates a sample set G which contains all RIC samples that u can touch. For each sample $g \in \mathtt{G}$, BT removes nodes in $\mathtt{C}_g$ that u can connect to and decreases $\mathtt{h}_g$. After this step, each sample $g \in \mathtt{G}$ will have an activation threshold at most 1. Then the greedy algorithm is applied to find $k - 1$ nodes that maximizes the number of influenced samples in G. $\mathtt{K}(u)$ will contain all these nodes and u.

Extending to d-bounded thresholds: An interesting observation is that we can extend BT to solve IMC with activation thresholds are bounded by a constant d. Denote $\mathtt{BT}^{(d)}$ as an algorithm to solve IMC with bounded threshold d.

Assume we have $\mathtt{BT}^{(d-1)}$ in hand with α-approximation guarantee. We devise $\mathtt{BT}^{(d)}$ by modifying from Alg. 4 that the set $\mathcal{T}$ (line 8) is selected by running $\mathtt{BT}^{(d-1)}$ to find $k - 1$ nodes that maximize influenced RIC samples in $\mathcal{G}_{\mathcal{R}}(u)$. Using similar proof as Theorem 4, $\mathtt{BT}^{(d)}$ provides $\frac{\alpha}{k}$ approximation guarantee to IMC with d-bounded threshold. As the approximation ratio of $\mathtt{BT}^{(2)}$ is $\frac{1-1/e}{k}$, using induction, we bound the approximation guarantee of $\mathtt{BT}^{(d)}$ by $\frac{1-1/e}{k^{d-1}}$.

Combining with MAF: In subsection IV-B, we proposed the MAF algorithm which guarantees $\frac{\mathtt{r}h}{k}$ approximate ratio. Interestingly, if we combine MAF and BT by running both these two algorithms to get two different results - called $S_{\mathtt{MAF}}, S_{\mathtt{BT}}$ - then return $S = \arg\max_{S' \in \{S_{\mathtt{MAF}}, S_{\mathtt{BT}}\}} \hat{\mathtt{c}}_{\mathcal{R}}(S')$, we obtain a solution with a new approximation guarantee, *which is tight to the inapproximability result in Theorem 1*. Let us call this solution MB (MAF and BT).

Theorem 5. MB *provides* $\Theta(\sqrt{\frac{1-1/e}{\mathtt{r}}})$ *approximation guarantee to the* MAXR *problem if the activation threshold of each community is at most 2.*

Proof. By definition, we have:

$$\hat{\mathtt{c}}^2_{\mathcal{R}}(S) \geq \hat{\mathtt{c}}_{\mathcal{R}}(S_{\mathtt{MAF}}) \cdot \hat{\mathtt{c}}_{\mathcal{R}}(S_{\mathtt{BT}}) \geq \frac{1-1/e}{k} \cdot \frac{1}{\mathtt{r}} \cdot \left\lfloor \frac{k}{2} \right\rfloor \cdot \hat{\mathtt{c}}^2_{\mathcal{R}}(S_{\mathtt{OPT}})$$

$\frac{\lfloor k/2 \rfloor}{k} = \Theta(1)$ with sufficient large k. Therefore, $\hat{\mathtt{c}}_{\mathcal{R}}(S) \geq \Theta(\sqrt{\frac{1-1/e}{\mathtt{r}}})\hat{\mathtt{c}}_{\mathcal{R}}(S_{\mathtt{OPT}})$, which completes the proof. □

V. Algorithmic Framework to IMC

In this section, we solve the second required piece: finding the minimum number of RIC samples to provide a bounded-error guarantee ϵ, where ϵ is an arbitrary small number. Then, putting together with any algorithm for MAXR described in section IV, we provide an algorithmic framework IMCAF for the IMC problem such that: given an α-approximation algorithm to MAXR, the IMCAF returns an $\alpha(1-\epsilon)$-approximation guarantee to IMC with probability at least $1-\delta$, where δ is an arbitrary small number. Accordingly, we have three approximation algorithms to IMC with the following ratios: $\frac{1}{\mathtt{r}}\lfloor \frac{k}{h} \rfloor (1-\epsilon)$ using MAF, $\Theta(\sqrt{\frac{1-1/e}{2\mathtt{r}}})(1-\epsilon)$ using MB, and $\frac{\mathtt{c}(S_\nu)}{\nu(S_\nu)}(1-1/e-\epsilon)$ using UBG.

A. Minimum size of $\mathcal{R}$

To bound the error between $\hat{\mathtt{c}}_{\mathcal{R}}(\cdot)$ and $\mathtt{c}(\cdot)$ within an ϵ value, we utilize the following lemma, which is trivially derived from the martingale theory in [16].

Lemma 6. [16] *Given a collection $\mathcal{R}$ of* RIC *samples, a seed set S and $\epsilon > 0$, the following inequalities hold,*

$$\Pr[\hat{\mathtt{c}}(S) > (1+\epsilon)\mathtt{c}(S)] \leq \exp(\frac{-|\mathcal{R}|\epsilon^2}{3\mathtt{b}}\mathtt{c}(S)) \quad (11)$$

$$\Pr[\hat{\mathtt{c}}(S) < (1-\epsilon)\mathtt{c}(S)] \leq \exp(\frac{-|\mathcal{R}|\epsilon^2}{2\mathtt{b}}\mathtt{c}(S)) \quad (12)$$

Denote S^* as an optimal solution to IMC. We have the following observations:

Corollary 1. *Given $0 < \epsilon_1, \delta_1 < 1$, with the number of* RIC *samples satisfies*

$$|\mathcal{R}| \geq \frac{2\mathtt{b}\ln(1/\delta_1)}{\epsilon_1^2 \mathtt{c}(S^*)}$$

the following condition is guaranteed:

$$\Pr[\hat{\mathtt{c}}(S^*) \geq (1-\epsilon_1)\mathtt{c}(S^*)] \geq 1-\delta_1 \quad (13)$$

This corollary is trivially derived from equation (12).

Corollary 2. *Given $0 < \alpha, \epsilon_2, \delta_2 < 1$, with the number of* RIC *samples satisfies*

$$|\mathcal{R}| \geq \frac{3\mathtt{b}\ln(\binom{n}{k}/\delta_2)}{\alpha^2\epsilon_2^2\mathtt{c}(S^*)} \quad (14)$$

for any seed set S_k of k nodes, the following condition is guaranteed:

$$\Pr[\hat{\mathtt{c}}(S) \geq \mathtt{c}(S) + \alpha\epsilon_2\mathtt{c}(S^*)] \leq \delta_2 \quad (15)$$

Proof. Consider an arbitrary seed set of k nodes S_k, we have:

$$\Pr[\hat{\mathtt{c}}(S) \geq \mathtt{c}(S) + \alpha\epsilon_2\mathtt{c}(S^*)] = \Pr[\hat{\mathtt{c}}(S) \geq (1+\alpha\epsilon_2\frac{\mathtt{c}(S^*)}{\mathtt{c}(S)})\mathtt{c}(S)] \leq \exp(-\frac{|\mathcal{R}|}{3\mathtt{b}}\frac{\alpha^2\epsilon_2^2\mathtt{c}^2(S^*)}{\mathtt{c}^2(S)}\mathtt{c}(S)) \leq \exp(-\frac{|\mathcal{R}|\alpha^2\epsilon_2^2}{3\mathtt{b}}\mathtt{c}(S^*))$$

Using the union bound theory, to let condition (15) satisfy for any seed set S of k nodes, we have:

$$\Pr[\hat{\mathtt{c}}(S) \geq \mathtt{c}(S) + \alpha\epsilon_2\mathtt{c}(S^*)] \leq \binom{n}{k}\exp(-\frac{|\mathcal{R}|\alpha^2\epsilon_2^2}{3\mathtt{b}}\mathtt{c}(S^*))$$

The equation (15) follows by replacing $\binom{n}{k}\exp(-\frac{|\mathcal{R}|\alpha^2\epsilon_2^2}{3\mathtt{b}}\mathtt{c}(S^*)) \leq \delta_2$ □

Given an α-approximation algorithm to MAXR, we can now formally define the minimum number of RIC samples to get the bounded error guarantee ϵ as follows.

Theorem 6. *Given an α-approximation algorithm to the* MAXR *problem and $0 \leq \epsilon_1, \epsilon_2, \delta_1, \delta_2 \leq 1$, let $\epsilon \geq \epsilon_1 + \epsilon_2$ and $\delta \geq \delta_1 + \delta_2$. If the number of* RIC *graphs satisfies:*

$$|\mathcal{R}| \geq \frac{\mathtt{b}}{\mathtt{c}(S^*)}\max\left(\frac{2\ln(1/\delta_1)}{\epsilon_1^2}, \frac{3\ln(\binom{n}{k}/\delta_2)}{\alpha^2\epsilon_2^2}\right) \quad (16)$$

the returned seed set S will guarantee:

$$\Pr[\mathtt{c}(S) \geq \alpha(1-\epsilon)\mathtt{c}(S^*)] \geq 1-\delta \quad (17)$$

which means S is an $\alpha(1-\epsilon)$-approximate solution to the IMC instance with probability at least $1-\delta$.

Proof. Let $S_{\mathtt{OPT}}$ be an optimal solution to MAXR problem.

$$\mathtt{c}(S) \geq \hat{\mathtt{c}}(S) - \alpha\epsilon_2\mathtt{c}(S^*) \quad (18)$$

$$\geq \alpha\hat{\mathtt{c}}(S_{\mathtt{OPT}}) - \alpha\epsilon_2\mathtt{c}(S^*) \geq \alpha\hat{\mathtt{c}}(S^*) - \alpha\epsilon_2\mathtt{c}(S^*) \quad (19)$$

$$\geq \alpha(1-\epsilon_1)\mathtt{c}(S^*) - \alpha\epsilon_2\mathtt{c}(S^*) \quad (20)$$

$$\geq \alpha(1-\epsilon)\mathtt{c}(S^*) \quad (21)$$

Inequality (18) happens with probability $1-\delta_2$ while inequality (20) happens with probability $1-\delta_1$. Overall, $\mathtt{c}(S) \geq \alpha(1-\epsilon)\mathtt{c}(S^*)$ with probability at least $(1-\delta_1)(1-\delta_2) \geq 1-\delta$ □

A drawback of this threshold is that it depends on $\mathtt{c}(S^*)$, which is intractable. However, we can replace $\mathtt{c}(S^*)$ by its lower bound while still making our conditions be satisfied. As long as the input k is greater than at least one threshold $\mathtt{h_i}$ of any community, the optimal solution always guarantees that at least one community will be influenced. We then tighten this lower bound by observation that $\mathtt{c}(S^*) \geq \frac{\beta k}{h}$ - where $\beta = \min_{i=1}^{\mathtt{r}} b_i$ and $h = \max_{i=1}^{\mathtt{r}} \mathtt{h}_i$. Replacing this bound into (16), we achieve an official bound of $|\mathcal{R}|$ (denoted Ψ) that we will use in our entire algorithm.

$$\Psi = \frac{\mathtt{b}h}{\beta k}\max\left(\frac{2\ln(1/\delta_1)}{\epsilon_1^2}, \frac{3\ln(\binom{n}{k}/\delta_2)}{\alpha^2\epsilon_2^2}\right) \quad (22)$$

B. IMC Algorithm Framework

Since the bound Ψ in equation (22) can be a large number, generating Ψ samples then applying any MAXR algorithms should not be a good strategy. Therefore, we utilize the SSA method [2], [17], [18] in which we could reduce the number of generated samples but still keep $\alpha(1-\epsilon)$ approximation guarantee with probability at least $1-\delta$. In short, the SSA

Algorithm 5 IMC Algorithmic Framework (IMCAF)

Input $G, \mathtt{Com}, k, \epsilon, \delta$ and an α-approx alg κ to MAXR
Output An $\alpha(1-\epsilon)$-approximation solution S with probability at least $(1-\delta)$

1: Initiate $\epsilon_1, \epsilon_2, \delta_1, \delta_2$ such that $\epsilon_1 + \epsilon_2 \leq \epsilon$ and $\delta_1 + \delta_2 \leq \delta$
2: $\Psi = \frac{\mathtt{b}h}{\beta k} \max\left(\frac{2\ln(1/\delta_1)}{\epsilon_1^2}, \frac{3\ln(\binom{n}{k}/\delta_2)}{\alpha^2\epsilon_2^2}\right)$
3: Initiate $\varepsilon_1, \varepsilon_2, \varepsilon_3$ such that $\epsilon \geq \varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \varepsilon_1\varepsilon_2$
4: $\Lambda = (1+\varepsilon_1)(1+\varepsilon_2)\frac{3}{\varepsilon_3^2}\ln\frac{3}{2\delta}$
5: $\mathcal{R} \leftarrow$ generate Λ RIC samples using Alg. 1.
6: **do** *// stop* stage
7: $S \leftarrow \kappa(\mathcal{R}, k)$
8: **if** $\frac{|\mathcal{R}|}{\mathtt{r}}\hat{\mathtt{c}}_{\mathcal{R}}(S) \geq \Lambda$ **then**
9: $\mathtt{c}^* \leftarrow \mathtt{Estimate}(G, S, \varepsilon_2, \frac{\delta}{3\log\Psi/\Lambda}, |\mathcal{R}|\frac{1+\varepsilon_2}{1-\varepsilon_2}\frac{\varepsilon_3^2}{\varepsilon_2^2})$
10: **if** $\hat{\mathtt{c}}_{\mathcal{R}}(S) \leq (1+\varepsilon_1)\mathtt{c}^*$ **then return** S
11: Double size of $\mathcal{R}$ with new RIC samples
12: **while** $|\mathcal{R}| \leq \Psi$

method keeps generating samples and *stop* at *exponential check points* to verify if there is adequate statistical evidence on the solution quality for termination. The SSA method has been successfully applied on IM. But because IMC is different to IM, some modification should be done. Therefore, we provide an IMC algorithmic framework (IMCAF) presented in Alg. 5. Comparing to SSA, IMCAF has the following modifications:

- We setup the maximum number of samples by equation (22) (line 2)
- We use Alg. 1 to generate RIC samples (line 5 and 11)
- We find a candidate seed set S in *stop* stage by solving the MAXR problem, which we already proposed several approximation algorithms in section IV. (line 7)
- We use an Estimate procedure (Alg. 6) to verify whether the candidate set S is within $\alpha(1-\epsilon)$-approximate ratio to optimal solution. (line 9)

The first and second modification are to adapt the IMC objective which is to maximize the benefit of influenced communities. The third modification comes from the fact that finding a candidate set S given a collection of samples is defined as MAXR problem, which is different to the max coverage problem in SSA. (In SSA, S could be found by solving max coverage, which is submodular). The final modification is to utilize Dagum's estimation [19] to estimate $\mathtt{c}(S)$. An estimation is presented by Alg. 6. The key part of Estimate procedure is that we iteratively generate RIC samples until the number of influenced samples reaches Λ' value. Line 6-9 is used to verify whether a newly generated sample is influenced by S. Based on *Stopping Rule Algorithm* (Section 2.1 [19]), Estimate procedure guarantees an estimation $\mathtt{c}^*$ of $\mathtt{c}(S)$ such that $\mathtt{c}^* \geq (1-\epsilon')\mathtt{c}(S_k)$ with probability at least $1-\delta'$.

Theorem 7. *Given $0 < \epsilon, \delta < 1$ and an α-approximation algorithm to* MAXR, IMCAF *provides an $\alpha(1-\epsilon)$-approximate guarantee to IMC with probability at least $1-\delta$. Specifically,*

$$\Pr[\mathtt{c}(S) \geq \alpha(1-\epsilon)\mathtt{c}(S^*)] \geq 1-\delta \tag{23}$$

Proof. According to IMCAF, the seed set S would be returned when one of following conditions is satisfied: (1) $|\mathcal{R}| \geq \Psi$ or (2) $\frac{|\mathcal{R}|}{\mathtt{r}}\hat{\mathtt{c}}_{\mathcal{R}}(S) \geq \Lambda$ and $\hat{\mathtt{c}}_{\mathcal{R}}(S) \leq (1+\varepsilon_1)\mathtt{c}^*$.

By Theorem 6 and equation (22), we already proved that: if $|\mathcal{R}| \geq \Psi$, the seed set S guarantees an $\alpha(1-\epsilon)$-approximate ratio with probability at least $1-\delta$. So let us consider the second condition: Given $\frac{|\mathcal{R}|}{\mathtt{r}}\hat{\mathtt{c}}_{\mathcal{R}}(S) \geq \Lambda$, Lemma 6 in [2] proved that:

$$\Pr[\hat{\mathtt{c}}_{\mathcal{R}}(S^*) \leq (1-\varepsilon_3)\mathtt{c}(S^*)] \leq 2\delta/3 \tag{24}$$

Denote $S_{\mathtt{OPT}}$ as an optimal solution of MAXR given the collection $\mathcal{R}$. We have:

$$\hat{\mathtt{c}}_{\mathcal{R}}(S) \geq \alpha\hat{\mathtt{c}}_{\mathcal{R}}(S_{\mathtt{OPT}}) \geq \alpha\hat{\mathtt{c}}_{\mathcal{R}}(S^*) \tag{25}$$

Thus, combining equation (24) and equation (25) gives:

$$\Pr[\hat{\mathtt{c}}_{\mathcal{R}}(S) \leq \alpha(1-\varepsilon_3)\mathtt{c}(S^*)] \leq 2\delta/3 \tag{26}$$

Now, based on *Stopping Rule Algorithm* (Section 2.1 [19]), we have:

$$\Pr[\mathtt{c}^* \geq (1+\varepsilon_2)\mathtt{c}(S)] \leq \delta/3 \tag{27}$$

Combine equation (27) and the condition $\hat{\mathtt{c}}_{\mathcal{R}}(S) \leq (1+\varepsilon_1)\mathtt{c}^*$, we have:

$$\Pr[\hat{\mathtt{c}}_{\mathcal{R}}(S) \geq (1+\varepsilon_1)(1+\varepsilon_2)\mathtt{c}(S)] \leq \delta/3 \tag{28}$$

Finally, combining equation (26) and equation (28) gives

$$\Pr[\mathtt{c}(S) \geq \alpha\frac{1-\varepsilon_3}{(1+\varepsilon_1)(1+\varepsilon_2)}\mathtt{c}(S^*)] \geq 1-\delta$$

Since $\epsilon \geq \varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \varepsilon_1\varepsilon_2$ (line 3 alg 5), we have:

$$\Pr[\mathtt{c}(S) \geq \alpha(1-\epsilon)\mathtt{c}(S^*)] \geq 1-\delta$$

which completes the proof. □

Algorithm 6 Estimate procedure

Input $G, S, \epsilon', \delta', T_{max}$
Output Estimation of $\mathtt{c}(S)$ with error ϵ'

1: $\Lambda' = 1 + 4(e-2)\ln\frac{2}{\delta'}\frac{1}{(\epsilon')^2}(1+\epsilon')$
2: $\mathtt{Inf} = 0$
3: **for** T from 1 to T_{max} **do**
4: $\mathtt{C}, \mathtt{h}, \mathtt{R}(\cdot) \leftarrow$ generate a RIC sample using Alg. 1
5: $\mathtt{tmp} = 0$
6: **for** *each node* $v \in \mathtt{C}$ **do**
7: **if** $\mathtt{R}(v) \cap S \neq \emptyset$ **then** $\mathtt{tmp} = \mathtt{tmp} + 1$
8: **if** $\mathtt{tmp} \geq \mathtt{h}$ **then**
9: $\mathtt{Inf} = \mathtt{Inf} + 1$
10: **if** $\mathtt{Inf} \geq \Lambda'$ **then return** $\mathtt{r}\Lambda'/\mathtt{T}$
11: **Return** -1

How to integrate the MAXR algorithms? MAF or BT(MB) algorithms could be easily integrated into Alg. 5. However, UBG algorithm has a problem that: UBG's approximation guarantee depends on $\mathcal{R}$, which leads to inconsistency on approximate ratio of MAXR on each *stop* stage in Alg 5. To deal with this problem, we only consider an upper-bound function $\nu(\cdot)$. Running SSA with objective function $\nu(\cdot)$ would produce a solution S_ν that guarantees $1-1/e-\epsilon$ approximation ratio. By returning $S = S_\nu$, we have:

$$\mathtt{c}(S) = \frac{\mathtt{c}(S_\nu)}{\nu(S_\nu)}\nu(S_\nu) \geq \frac{\mathtt{c}(S_\nu)}{\nu(S_\nu)}(1-1/e-\epsilon)\nu(S_\nu^*)$$
$$\geq \frac{\mathtt{c}(S_\nu)}{\nu(S_\nu)}(1-1/e-\epsilon)\nu(S^*) \geq \frac{\mathtt{c}(S_\nu)}{\nu(S_\nu)}(1-1/e-\epsilon)\mathtt{c}(S^*)$$

where S_ν^*, S^* are optimal solutions to the objective functions $\nu(\cdot), \mathtt{c}(\cdot)$ respectively.

VI. Experimental Evaluation

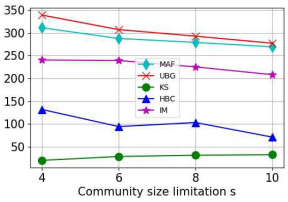
In this section, we compare the performance of our proposed algorithms with several intuitive heuristic solutions.

A. Experimental Settings

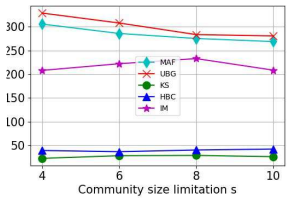
The experiments were conducted on a Linux machine with 2.3GHz Xeon 18 cores processor and 256GB of RAM. We

TABLE I: Statistics of datasets

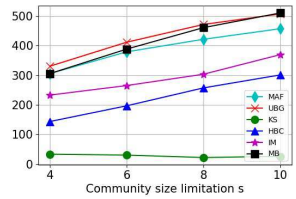
Data	Type	Nodes	Edges
Facebook	Undirected	747	60.05 K
Wiki-vote	Directed	7.1 K	103.6 K
Espinions	Directed	76 K	508.8 K
DBLP	Undirected	317 K	1.05 M
Pokec	Directed	1.6 M	30.6 M



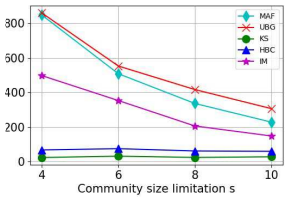
(a) Facebook, Random



(b) Facebook, Louvain



(c) Facebook, Louvain, bounded threshold



(d) DBLP, Louvain

Fig. 4: Quality of solution with different community structures. Y-axis is the expected benefit of influenced communities.

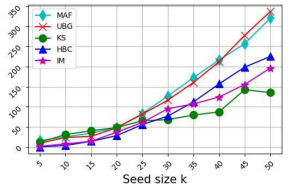
used the datasets from Stanford Network Analysis Project [20], summarized in Table I.

Settings. To partition the social networks into disjoint communities, we utilized the well-known Louvain algorithm [21], [22], which extracts communities to optimize the network modularity. In order to see the impact of communities selection on IMC, we also use Random algorithm as the baseline. In the Random algorithm, we fix the number of communities and randomly put nodes into communities. To prevent cases in which some communities are significantly larger than the others, we limited the community size by a certain value s. If a community C was larger than s, we split it into $\lceil |C|/s \rceil$ communities. We set $s = 8$ unless otherwise stated.

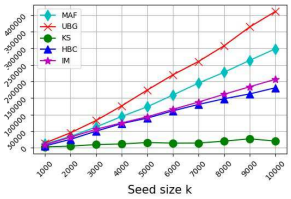
The benefit of a community was equal to its population ($b_i = |\mathtt{C}_i|$). For each edge $e = (u, v)$, let e's weight equal $\frac{1}{d(v)}$ where $d(v)$ is the in-coming degree of node v (in case of undirected graph, the edge $e = (u, v)$ will be considered as two directed edges (u, v) and (v, u)). The activation threshold of each community was set to be 2 in the experiments including MB algorithm. Otherwise, the activation threshold of each community was set to be 50% of its population ($\mathtt{h}_i = 0.5|\mathtt{C}_i|$). In all the experiments, we kept $\epsilon = \delta = 0.2$, $\epsilon_1 = \epsilon_2 = \epsilon/2$ and $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = \epsilon/4$.

Baselines: To our extent, no existing algorithm was applicable to the IMC problem. Thus, we compared our proposed algorithms with the following heuristic methods:

- HBC (High Beneficial Connection). HBC selects k nodes that have high beneficial connection to the communities set. A beneficial connection of node u was defined as $B(u) = \sum_{v \in N^-(u)} w(u, v) \cdot \frac{b_{\mathtt{C}(v)}}{\mathtt{h}_{\mathtt{C}(v)}}$ where $N^-(u)$ is u's out-coming neighbors; $\mathtt{C}(v)$ is a community in which v is a single user, and $\mathtt{h}_{\mathtt{C}(v)}$ is its activation threshold.
- KS (Knapsack-like Algorithm). KS considers the activation threshold of a community as a cost to influence it. Then KS finds communities set with cost constraint k to maximize the overall benefit. This is a Knapsack problem and it is possible to obtain an optimal solution in polynomial runtime. Define R as an optimal solution to the Knapsack problem: for each community $\mathtt{C} \in R$, we selected h nodes in C and put them into the seed set S, where h is the activation threshold of C.
- IM (Influence Maximization). IM selects k nodes that maximize the influence spread. Then we estimate their expected benefit on influenced communities.



(a) Wiki-vote



(b) Pokec

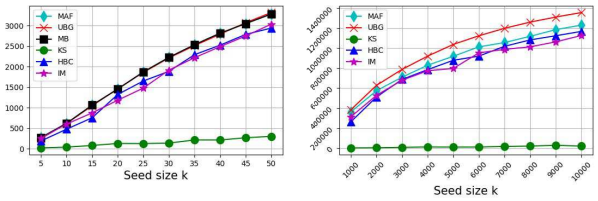
Fig. 5: Benefit of different algorithms on regular case. Y-axis is the expected benefit of influenced communities.

With the baseline algorithms, to evaluate the benefit of influenced communities, we used Dagum [19] estimation method with the same ϵ, θ we used for our proposed algorithm. Also, for simplicity, in this section UBG/MAF/MB means running the IMCAF framework with the corresponding MAXR algorithms. For each set of experiments, we ran ten times and report the average results. *Due to space limits, we only report the important observations.*

B. Performance comparison

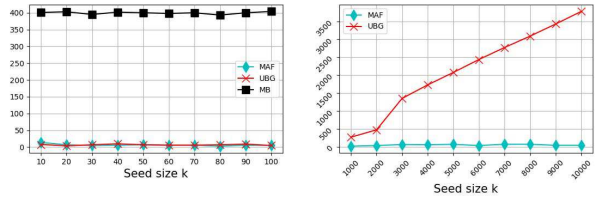
Quality of solution. First, we compare the algorithms' performance with different community formations. Also, with each community formation, we changed value of s and recorded our results in Figure 4. In this experiment, we fix $k = 10$. Figure 4a, 4b and 4d show the results of the experiments with Facebook and the DBLP dataset using the Louvain and Random community formation. Figure 4c shows the result in the case of bounded activation threshold. Generally, our algorithms always returned the best solutions regardless community formation methods. Also, the quality of all algorithms tended to decrease when s increases, which contradicts the experiment on bounded activation threshold. These experiments suggest that: there should be a relationship between s, the overall benefit, and the average activation threshold of influenced communities.

Next, we compare the algorithms' performance in regard to the change of k. Fig. 5 compares the quality of different algorithms without bounded activation threshold. Both of our proposed algorithms returned better solutions than any baseline algorithm. All algorithms performed similarly when k is small but the gap became noticeable when k increases. We could say UBG always returned the best solutions while KS was the worst one, which is explainable because KS does not consider the network topology and diffusion model of users when finding solution. Furthermore, the gap of quality between IM and our two solutions grew when k increased.



(a) Wiki-vote (b) Pokec

Fig. 6: Benefit of different algorithms on the bounded activation threshold case. Y-axis is the expected benefit of influenced communities.



(a) Espinions (b) Pokec

Fig. 7: Runtime of our algorithms. Y-axis is the average runtime in seconds

This is because when k is large, the number of activated users increases but scatters in the network. When we estimated the benefit of IM solution, we observed there was a huge amount of RIC samples that needed only one more activated node to be influenced, which occasionally happened in UBG and MAF. This further demonstrates the difference in objective between ICM and IM. The similar patterns were met in the experiments with bounded activation threshold (Fig. 6). Note that we discarded the MB's results in Fig. 6b because they exceeded the runtime limit.

Running time. We now compare the CPU runtime between our three solutions (Fig. 7) to evaluate the tradeoffs. Since the algorithms ran very fast (several seconds) on the small networks, we only show their performance on the large networks. As can be seen in Fig. 7b, MAF ran much faster than UBG. Also, the change in k did not affect much on MAF's performance, which totally contradicted to UBG. This could be explained by: MAF takes one-pass on all communities/nodes to calculate their impacts, another pass to sort them in order of this value and finally identify the k most influencing nodes. Meanwhile, the performance of the greedy algorithm in UBG was highly affected when k increases.

In case of bounded activation threshold, our three algorithms showed similar running time on the small-scale dataset. However, in the large networks shown in Fig. 7a, MAF and UBG outperformed MB by a huge margin. Furthermore, MB could not finish within the runtime limit in the experiments with the Pokec datasets. The poor performance of MB was because MB splits the IMC instance into $O(|V|)$ subproblems and solves them all to get the best solution (Alg. 4). The runtime for each subproblem is equivalent to the runtime of UBG.

Evaluation of UBG. Even UBG always returned the best results among all algorithms, UBG's approximate guarantee depends on the ratio $\frac{\mathsf{c}(S_\nu)}{\nu(S_\nu)}$ where S_ν is $1-1/e-\epsilon$ approximate solution to the submodular function $\nu(\cdot)$. Within our configuration, we observed that: the ratio has a strong relationship with k and the average activation threshold of the communities. These relations are shown in Fig. 8. We calculated the ratio by obtaining S_ν and using Monte Carlo method to estimate $\mathsf{c}(S_\nu)$ and $\nu(S_\nu)$. We observed that: the ratio increased and was closer to 1 when k grew. Also, the ratio increased when the average activation threshold of communities decreased. This could be observed by comparing the experiments in the same network with and without bounded activation threshold. For example in Pokec network, we obtained the ratio approximate to 1 with $k = 10000$ in the bounded threshold case but only get 0.4 ratio in the regular case. This is because when the activation threshold decreases, $\mathsf{c}(\cdot)$ exhibits more "submodular" behaviors and becomes closer to $\nu(\cdot)$. $\mathsf{c}(\cdot)$ would be totally submodular if the activation threshold is bounded by 1, which we already proved in subsection IV-C.

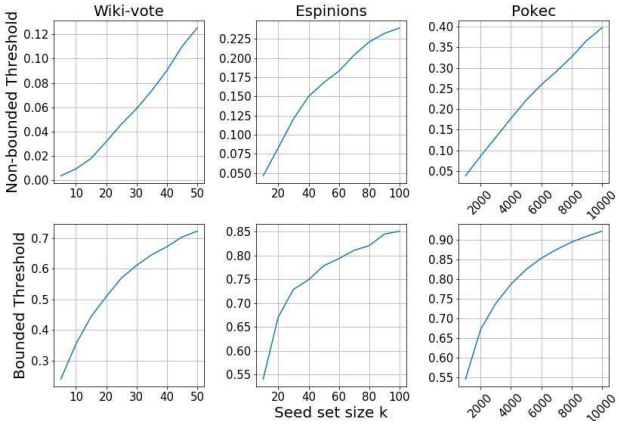


Fig. 8: Relation of UBG ratio with seed size k. Y-axis is the average value of $\frac{\mathsf{c}(S_\nu)}{\nu(S_\nu)}$.

VII. Conclusion

In this paper, we introduced a novel IMC problem, on which we consider the collaborative impact of the influence spread in social networks. Solving this problem was shown to be very challenging. We proposed a new sampling method, which could be used as a baseline for any algorithm for IMC. We next devised several approaches in a trade off between effectiveness and running time. Experimental results demonstrated the effectiveness of our proposed algorithms and confirmed with the theoretical results.

VIII. Acknowledgements

This work was supported in part by NSF EFRI-1441231, NSF CNS-1814614, NSF CNS-1443905, and DTRA HDTRA1-14-1-0055.

References

[1] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003, pp. 137–146.

[2] H. T. Nguyen, M. T. Thai, and T. N. Dinh, "Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks," in *Proceedings of the 2016 International Conference on Management of Data*. ACM.

[3] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 1029–1038.

[4] Y. Tang, X. Xiao, and Y. Shi, "Influence maximization: Near-optimal time complexity meets practical efficiency," in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*.

[5] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, "Maximizing social influence in nearly optimal time," in *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 2014.

[6] B. Wellman, J. Salaff, D. Dimitrova, L. Garton, M. Gulia, and C. Haythornthwaite, "Computer networks as social networks: Collaborative work, telework, and virtual community," *Annual review of sociology*, vol. 22, no. 1, pp. 213–238, 1996.

[7] T. Pan, S. Mishra, L. N. Nguyen, G. Lee, J. Kang, J. Seo, and M. T. Thai, "Threat from being social: Vulnerability analysis of social network coupled smart grid," *IEEE Access*, vol. 5, pp. 16 774–16 783, 2017.

[8] H. T. Nguyen, T. N. Dinh, and M. T. Thai, "Cost-aware targeted viral marketing in billion-scale networks," in *INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*.

[9] R. Impagliazzo and R. Paturi, "On the complexity of k-sat," *Journal of Computer and System Sciences*, vol. 62, no. 2, pp. 367–375, 2001.

[10] Y. Lin, W. Chen, and J. C. Lui, "Boosting information spread: An algorithmic approach," in *Data Engineering (ICDE), 2017 IEEE 33rd International Conference on*. IEEE, 2017, pp. 883–894.

[11] W. Lu, W. Chen, and L. V. Lakshmanan, "From competition to complementarity: comparative influence diffusion and maximization," *Proceedings of the VLDB Endowment*, vol. 9, no. 2, pp. 60–71, 2015.

[12] L. Ma, G. Cao, and L. Kaplan, "Graphical approach for influence maximization in social networks under generic threshold-based non-submodular model," in *Big Data (Big Data), 2017 IEEE International Conference on*. IEEE, 2017, pp. 970–975.

[13] X. Li, J. D. Smith, T. N. Dinh, and M. T. Thai, "Why approximate when you can get the exact? optimal targeted viral marketing at scale," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[14] P. Manurangsi, "Almost-polynomial ratio eth-hardness of approximating densest k-subgraph," in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 2017, pp. 954–961.

[15] "Markov's inequality," https://en.wikipedia.org/wiki/Markov%27s_inequality.

[16] Y. Tang, Y. Shi, and X. Xiao, "Influence maximization in near-linear time: A martingale approach," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 2015.

[17] K. Huang, S. Wang, G. Bevilacqua, X. Xiao, and L. V. Lakshmanan, "Revisiting the stop-and-stare algorithms for influence maximization," *Proceedings of the VLDB Endowment*, 2017.

[18] H. T. Nguyen, T. N. Dinh, and M. T. Thai, "Revisiting of revisiting the stop-and-stare algorithms for influence maximization," in *Proceedings of the 2018 International Conference on Computational Data and Social Networks*, 2018.

[19] P. Dagum, R. Karp, M. Luby, and S. Ross, "An optimal algorithm for monte carlo estimation," *SIAM Journal on computing*, 2000.

[20] J. Leskovec *et al.*, "Stanford network analysis project," *http://snap.stanford.edu*, 2010.

[21] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

[22] N. Dugué and A. Perez, "Directed louvain: maximizing modularity in directed networks," Ph.D. dissertation, Université d'Orléans, 2015.