Non-Submodularity and Approximability: Influence Maximization in Online Social Networks

Huanyang Zheng*, Ning Wang†, and Jie Wu‡
*Google, Menlo Park, USA
†Rowan University, Glassboro, USA
‡Center for Networked Computing, Temple University, Philadelphia, USA
Email: huanyang.zheng@gmail.com, wangn@rowan.edu, and jiewu@temple.edu

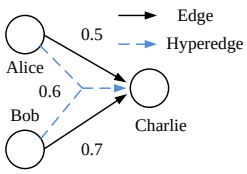
***Abstract*—Motivated by many Online Social Network (OSN) applications such as viral marketing, the Social Influence Maximization Problem (SIMP) has received tremendous attention. SIMP aims to select k initially-influenced seed users to maximize the number of eventually-influenced users. Under the independent cascade model, the SIMP has been proved to be NP-hard, monotone, and submodular. Therefore, a naive greedy algorithm that maximizes the marginal gain obtains an approximation ratio of $1-e^{-1}$. This paper extends the SIMP by considering the crowd influence which is combined group influence in additional to individual influence among a given crowd. Our problem is proved to be NP-hard and monotone, but not submodular. It is proved to be inapproximable within a ratio of $|V|^{\epsilon-1}$ for any $\epsilon > 0$. However, since user connections in OSNs are not random, approximations can be obtained by leveraging the structural properties of OSNs. We prove that the supmodular degree, denoted as Δ, of most OSNs has the following property $\lim_{|V|\to\infty} \frac{\Delta}{O(|V|)} = 0$, i.e., $\Delta \in o(|V|)$ for most OSNs. The supermodularity, denoted by Δ, is used to measure to what degree our problem violates the submodularity. Two approximation algorithms have been applied with ratios of $\frac{1}{\Delta+2}$ and $1-e^{-1/(\Delta+1)}$, respectively. Experiments demonstrate the efficiency and effectiveness of our algorithms.**

***Index Terms*—Social influence maximization; independent cascade; non-submodularity; approximability; supermodularity.**

I. Introduction

Online Social Networks (OSNs) mainly focus on building social relationships among users who share interests, activities, backgrounds, stories, and real-life connections. OSNs are very valuable tools used by numerous people to extend their daily contacts. Existing OSNs such as Facebook, Twitter, and VK are three of the top ten most-visited websites in the world. As of January 2014, 74% of online adults use OSNs.

Motivated by many OSN applications such as viral marketing [1] and personalized recommendation [2], social influence propagations have received tremendous attention in the last decade, especially for the Social Influence Maximization Problem (SIMP). Given an influence propagation model such as the independent cascade model, the SIMP selects k initially-influenced seed users to maximize the number of eventually-influenced users [3]. In the literature, the influence propagation model is generally submodular [4]. Therefore, a simple greedy algorithm can obtain an approximation ratio of $1-\frac{1}{e}$ to the optimal algorithm. However, few results [5] are provided when the influence propagation model even slightly violates the submodularity. In contrast, this paper studies a non-submodular SIMP under the independent cascade model in hypergraphs.



Influenced?		Probability to propagate the influence to Charlie
Alice	Bob	
Yes	No	0.5
No	Yes	0.7
Yes	Yes	1-(1-0.5)(1-0.6)(1-0.7) = 0.94

Fig. 1. Social influences through edges and hyperedges.

Our key justification for the non-submodularity comes from the crowd psychology [6]: One reason is that people are afraid of doing anything new. Attempting new things always requires courage, since we do not know what will happen in the future. Following the crowd gives people a cushion of comfort to make mistakes. Another reason is because of criticism. Once we do something atypical, we may be criticised heavily by friends, family and parents for not doing what everyone else is doing. After we fail, we would start doing what the crowd does. As a result, the crowd psychology reveals that the crowd influence is different from the combined independent influences of people in the crowd. That is, in additional to individual influence among a given crowd, there is a combined group influence called crowd influence. This phenomenon yields non-submodularity in social influence propagations, which are modeled through hypergraphs.

Fig. 1 provides a more specific example. Directed edges represent the influence from Alice or Bob to Charlie. The influences Charlie receives from Alice and Bob are independent of each other. According to crowd psychology, if both Alice and Bob are influenced, there should exist a crowd influence in addition to Alice's and Bob's influences. Fig. 1 shows how a combined influence on Charlie is calculated using both individual influence and the crowd influence of Alice and Bob. A *hyperedge* (of a hypergraph) is used to depict such a crowd influence. Note that influences through hyperedges are not submodular since seed user selections in the SIMP are no longer diminishing returns. Consequently, solving the SIMP in hypergraphs poses unique challenges. The first challenge is to deal with non-submodularity. The problem hardness and approximability both need to be explored. New algorithms are needed, since a simple greedy algorithm can no longer guarantee an approximation ratio. Another challenge is scalability. Since hyperedges change the scalability of the SIMP, it is difficult to reduce their complexities.

This paper studies the SIMP under the independent cascade model in hypergraphs. The problem is proven to be NP-hard, and cannot be approximated within a ratio of $|V|^{\epsilon-1}$ for any $\epsilon > 0$. $|V|$ is the number of nodes in the hypergraph, meaning that no algorithm can do better than a random seed user selection in terms of the asymptotic approximation ratio. However, since user connections in OSNs are not random, approximation algorithms are proposed by leveraging certain structural properties of OSNs. A concept called supermodularity (denoted by Δ) is used. Δ measures to what degree our problem violates the submodularity, and Δ is expected to be bounded in OSNs. We prove that the supmodular degree, denoted as Δ, of most OSNs has the following property $\lim_{|V|\to\infty} \frac{\Delta}{O(|V|)} = 0$, i.e., $\Delta \in o(|V|)$ for most OSNs. The supermodularity, denoted by Δ, is used to measure to what degree our problem violates the submodularity. Two approximation algorithms are applied with ratios of $\frac{1}{\Delta+2}$ and $1 - e^{-\frac{1}{\Delta+1}}$, respectively.

Our main contributions are summarized as follows:

- Motivated by the crowd psychology, we study the SIMP in hypergraphs. Our problem is proven to be NP-hard, monotone, non-submodular, and inapproximable.
- A concept called supermodularity is used to measure to what degree hyperedges violate the submodularity. The supermodularity is expected to be bounded in OSNs.
- Two approximation algorithms are applied with ratios of $\frac{1}{\Delta+2}$ and $1 - e^{-\frac{1}{\Delta+1}}$, respectively. Here, Δ is the supermodularity. Algorithms are scalable for large OSNs.
- Real data-driven experiments are conducted to evaluate the proposed solutions. The results are shown from different perspectives to provide insightful conclusions.

II. Related Works

Motivated by applications such as viral marketing, personalized recommendations, and online gaming [1, 2, 7], researches on the social influence propagation have received tremendous attention in the last decade, especially for the SIMP. The original SIMP was proposed by Kempe et al. [3] with two influence propagation models of independent cascade and linear threshold. The SIMP aims to select k initially-influenced seed users to maximize the number of eventually-influenced users. Under the independent cascade and linear threshold models, the SIMP has been proven to be NP-hard, monotone, and submodular. Consequently, a simple greedy algorithm, which iteratively maximizes the marginal gain, obtains an approximation ratio of $1 - \frac{1}{e}$ to the optimal algorithm. A fruitful literature for the SIMP [8–11] has been developed. However, almost all variations of the SIMP are submodular or unbounded. For example, Chen et al. [12] considered a variation of the SIMP with both positive and negative influence propagations. Their model maintains submodularity for maximizing the spread of positive influences. Unbounded variations of the SIMP are studied, usually through a data mining approach. For example, Goyal et al. [13] used available traces to learn how influence propagates in OSNs. Based on the learned model, the expected influence spread can be estimated to solve the SIMP. Tang et al. considered the seed cost in SIMP problem [14].

Unfortunately, few results [5] are provided when the influence propagation model even slightly violates the submodularity. Hung et al. [5] studied a variation of the SIMP with multiple items. Their problem is NP-hard and non-submodular, and thus, only heuristic algorithms are provided. This is because the problem of non-submodular function maximization [15] has not been perfectly solved in the literature [16]. Although the problem of supermodular function maximization can be optimally solved by the minimum-norm-point algorithm [17], non-submodular functions are not the same. The latest approach is based on the curvature [18], which assumes that the marginal gain of the non-submodular function varies within a given curvature. This paper can be viewed as a curvature-based approach that is specially designed for the SIMP in OSNs.

Recent studies in network science show that many networks exhibit special structures. This paper relates to the structural properties of OSNs. Recent research show that user connections in OSNs are not truly random [19]. The degree distribution in OSNs is acknowledged to follow the power-law distribution [19]: a majority of users are inactive with a small number of connections, while a minority of users are active with a large number of connections. Based on [20], OSNs usually have small diameters (about 6), high clustering coefficients (larger than 0.1), and community structures. These structural properties can be incorporated into algorithmic designs.

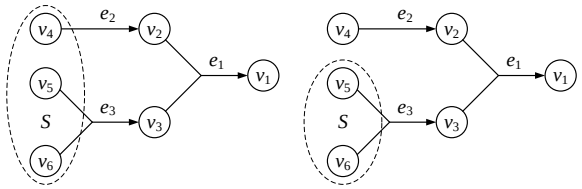
III. Model and Formulation

A. Model and Notations

Our scenario is based on a directed hypergraph $G = (V, E)$, where $V = \{v\}$ is a set of nodes (i.e., users in an OSN), and $E = \{e\}$ is a set of directed hyperedges. Hyperedges represent influence propagation directions, including personal and crowd influences. $|\cdot|$ denotes the set cardinality: $|V|$ and $|E|$ are the numbers of nodes and hyperedges, respectively. For a hyperedge e, let H_e and T_e denote its head and tail sets of nodes (i.e., e connects nodes in H_e to nodes in T_e). Hyperedges are a generalization of normal edges. As a special case, when $|H_e| = |T_e| = 1$, e becomes a normal edge. Let w_e denote the weight of e, representing the influence propagation probability ($0 \leq w_e \leq 1$). Given an OSN, the hypergraph G can be generated using Hung's approach [5]: while nodes are just users in the OSN, hyperedges and their weights can be learned based on a statistical inference-based framework. Therefore, this paper assumes that G is known a priori.

B. Independent Cascade in Hypergraphs

The independent cascade is a classic model [3] that simulates influence propagations in OSNs. Since the independent cascade is designed for normal graphs, a simple extension is made for hypergraphs. Let us start with a set, S, of nodes. All nodes in S are initially active and are also called seed users [3]. In contrast, all other nodes are initially inactive. Independent cascade unfolds in discrete steps according to the following randomized process. Given a hyperedge e, when all



(a) v_1 is reachable from S. (b) v_1 is not reachable from S.

Fig. 2. An example of the reachability.

nodes in H_e first becomes all active in step t, attempts are made to activate each inactive node in T_e. Each activation attempt is independent of all others, and it succeeds with a probability w_e. Here, w_e represents the influence propagation probability. If an inactive node has received multiple activation attempts, these activation attempts can be sequenced in an arbitrary order. If an inactive node is successfully activated, it will become active in step $t+1$. In addition, whether or not an activation attempt succeeds, it will have no further impacts in subsequent steps. The above process iterates step by step and terminates when no more activations are possible.

Active nodes in the independent cascade model represent influenced nodes. All seed nodes in S are initially-influenced, while all active nodes at the end of the process are eventually-influenced. Note that a hyperedge e could only propagate the influence when all nodes in H_e first become all active. If H_e includes an inactive node, e cannot propagate the influence. This is because hyperedges represent crowd influences. We use $\sigma(S)$ to denote the expected number of eventually-influenced nodes. $\sigma(S)$ is also called the influence spread of S.

C. Problem Formulation

Our objective is to select k initially-influenced seed users to maximize the number of eventually-influenced users:

$$\text{maximize} \quad \sigma(S) \tag{1}$$

$$\text{s.t.} \quad |S| \leq k \tag{2}$$

k is a pre-defined constant to bound the size of the seed set S. Given S, $\sigma(S)$ is determined by the independent cascade model. Our problem is almost the same as the classic SIMP in [3], except that our problem uses a hypergraph rather than a normal graph. However, this difference, despite how small it seems, leads to unique challenges. This is because the SIMP has become non-submodular instead of submodular.

For presentation simplicity, four concepts (i.e., hyperdegree, cycle, path, and reachability) are defined:

***Definition** 1:* Let d_v denote the hyperdegree of the node v. d_v is the number of hyperedges that include v in their heads or tails, i.e., $d_v = |\{e \,|\, v \in H_e \text{ or } v \in T_e\}|$.

***Definition** 2:* For a directed hypergraph $G = (V, E)$, a cycle in G is defined as a sequence $(v_0, e_0, v_1, e_1, ..., e_{n-1}, v_0)$ of alternating nodes and hyperedges where (i) e_i is distinct for $\forall i \in \{0, ..., n-1\}$, (ii) $v_i \in H_{e_i}$ for $\forall i \in \{0, ..., n-1\}$, and (iii) $v_{(i+1)\%n} \in T_{e_i}$ for $\forall i \in \{0, ..., n-1\}$.

***Definition** 3:* For a directed hypergraph $G = (V, E)$, a path from v_0 to v_n in G is a sequence $(v_0, e_0, v_1, e_1, ..., e_{n-1}, v_n)$ of alternating nodes and hyperedges where (i) e_i is distinct for $\forall i \in \{0, ..., n-1\}$, (ii) $v_i \in H_{e_i}$ for $\forall i \in \{0, ..., n-1\}$, and (iii) $v_{i+1} \in T_{e_i}$ for $\forall i \in \{0, ..., n-1\}$.

***Definition** 4:* For a directed hypergraph $G = (V, E)$, a node v is said to be reachable from a node set S if (i) there exists a path from a node in S to v and (ii) each head node of each hyperedge of the above path is reachable from S.

Note that the reachability is recursively defined. An example is shown in Fig. 2. In Fig. 2(a), v_1 is reachable from S since v_2 and v_3 are also reachable from S. In contrast, in Fig. 2(b), v_1 is not reachable from S since v_2 is not reachable from S.

IV. Analysis

A. NP-hard and Monotone

We start with the problem hardness:

***Theorem** 1:* The SIMP in a hypergraph is NP-hard.

This is because the classic SIMP in a normal graph [3] is NP-hard by reduction from the set cover problem, which is NP-complete. Meanwhile, every instance in the classic SIMP is a special case of our problem (a graph is a special case of a hypergraph). Therefore, the SIMP in a hypergraph is NP-hard.

***Theorem** 2:* Given a hypergraph G, $\sigma(S)$ is monotone with respect to S, meaning that $\sigma(S') \leq \sigma(S)$ for $\forall S' \subseteq S$.

Proof: We prove through formulating an equivalent view of the independent cascade. Let us convert the hypergraph G to another hypergraph G'. G and G' have the same nodes. Each hyperedge, e, in G is mapped to a set, $\{e'\}$, of hyperedges in G': $\{e' \,|\, w_e = w_{e'}, H_{e'} = H_e, |T_{e'}| = 1, T_{e'} \subseteq T_e\}$. Such a conversion decomposes hyperedges in G by separating their tail nodes. By definition, the independent cascades in G and G' should be exactly the same.

In G', let us consider a hyperedge e' whose head nodes first become all active. Note that e' has exactly one tail node by definition. Now e' tries to activate its tail node, succeeding with probability $w_{e'}$. We can view the outcome of this random event as being determined by flipping a coin of bias $w_{e'}$. From the view of the independent cascade, it does not matter whether the coin was flipped at the moment that e' tries to activate its tail node, or whether it was flipped at the very beginning of the whole process and is only being revealed now. Continuing this reasoning, we can assume that for each hyperedge $e' \in G'$, a coin of bias $w_{e'}$ is flipped at the very beginning of the process (independently of the coins for all other hyperedges), and the result is stored so that it can be checked later when e' tries to activate its tail node.

The remainder is similar to [3]. With all the coins flipped in advance, the independent cascade in G' can be equivalently viewed as follows. In G', the hyperedges, for which the coin flip indicated an activation will be successful, are declared to be live; the remaining hyperedges are declared to be blocked. If we fix the outcomes of the coin flips and then initially activate a seed set S, it is clear how to determine the full set of active nodes at the end of the independent cascade: a node v ends up active if and only if it is reachable (see Definition 4) from S via only live hyperedges. In each possible set of outcomes (in terms of all coin flip outcomes on the hyperedges), if a node v is reachable from S', it must be

Algorithm 1 Naive Greedy (NG)

Input: a hypergraph, G, and a constant, k.
Output: a set of seed nodes, S, initiated $\emptyset$.

1: **while** $|S| < k$ **do**
2: Find $v = \arg\max_{v \in V} \sigma(S \cup \{v\}) - \sigma(S)$.
3: Update $S = S \cup \{v\}$.

reachable from S, since $S' \subseteq S$. Therefore, $\sigma(S') \leq \sigma(S)$ holds for $\forall S' \subseteq S$, and the proof completes. ■

B. Non-Submodular and Inapproximability

Theorem 3: Given a hypergraph G, $\sigma(S)$ is not submodular with respect to S, meaning that $\sigma(S \cup \{v\}) - \sigma(S) > \sigma(S' \cup \{v\}) - \sigma(S')$ for $\exists\, v \in V, S' \subset S, S \subseteq V$.

Proof: We prove by a counterexample in Fig. 2(a). We focus on three nodes of v_1, v_2, v_3, and one hyperedge of e_1 with $w_1 = 1$. Let us set $S' = \emptyset$, $S = \{v_2\}$, and $v = v_3$. We have $\sigma(S \cup \{v\}) = 3$, since v_1 will be influenced by v_2 and v_3. Meanwhile, we have $\sigma(S) = 1$ and $\sigma(S' \cup \{v\}) = 1$, since not all head nodes of e_1 are active, and thus, e_1 cannot activate v_1. In addition, we have $\sigma(S') = 0$ since $S' = \emptyset$. As a result, $\sigma(S \cup \{v\}) - \sigma(S) = 2 > \sigma(S' \cup \{v\}) - \sigma(S') = 1$. ■

The submodularity for the influence propagation in a normal graph is not preserved in a hypergraph. The intuition is that the SIMP is no longer diminishing return due to the crowd influence, which is in addition to the influence of each person in the crowd. This phenomenon yields non-submodularity. A simple variation of Hung's first theorem in [5] can lead to the following approximability result:

Theorem 4: For the SIMP in a general hypergraph, unless $\mathbb{P} = \mathbb{NP}$, no algorithm can guarantee an approximation ratio of $|V|^{\epsilon-1}$ for any $\epsilon > 0$.

Theorem 4 validates that the SIMP in a general hypergraph is not approximable, i.e., any given algorithm must perform poorly under certain hypergraphs. Therefore, Theorem 4 poses a unique challenge to solve the SIMP in OSNs.

C. Naive Greedy

Since Theorem 4 shows the inapproximability of the SIMP in a general hypergraph, this subsection focuses on a naive greedy algorithm, as shown in Algorithm 1. Starting with an empty seed set (line 1), it iteratively add a node that maximizes the marginal gain of $\sigma(S_i)$, until k nodes are selected (lines 2 to 5). Since $\sigma(\cdot)$ is no longer submodular, such a greedy algorithm cannot guarantee an approximation ratio of $1-1/e$. Note that Algorithm 1 involves the sub-problem of computing $\sigma(S)$ for a given S. This sub-problem is NP-hard [21] and has been extensively studied in the literature. This paper does not explore this sub-problem, and uses the Monte Carlo simulation [21] to compute $\sigma(S)$ for a given S in G.

V. Algorithms

The previous section has proven that the SIMP in a general hypergraph is NP-hard, monotone, non-submodular, and not approximable within a ratio of $|V|^{\epsilon-1}$ for any $\epsilon > 0$. However, OSNs are not general hypergraphs. Recent studies in network science validate that user connections in OSNs are not truly random [19]. Consequently, approximation algorithms become possible by leveraging certain structural properties of OSNs.

We have proved that the SIMP in a general hypergraph is NP-hard, monotone, non-submodular, and not approximable within a ratio of $|V|^{\epsilon-1}$ for any $\epsilon > 0$. However, OSNs are not general hypergraphs, and thus, approximation algorithms are possible by leveraging certain structural properties of OSNs.

A. Supermodularity

To enable approximation algorithms by revealing structural properties of OSNs, two concepts in [15] are used:

Definition 5: Given a monotone objective function $\sigma(\cdot)$, the modularity set of a node v is $M_v = \{v' \,|\, \sigma(S \cup \{v, v'\}) - \sigma(S \cup \{v'\}) > \sigma(S \cup \{v\}) - \sigma(S)$ for $\exists\, v \in V, v' \in V, S \subseteq V\}$, which includes all nodes that might increase the marginal gain of v.

Definition 6: The supermodularity, Δ, is the maximum cardinality among all modularity sets, i.e., $\Delta = \max_v |M_v|$.

For a node v, only nodes in M_v might increase the marginal gain of v for the objective function $\sigma(\cdot)$. In contrast, nodes that are not in M_v never increase the marginal gain of v. If v is locally submodular for $\sigma(\cdot)$, then $M_v = \emptyset$. Consequently, the supermodularity Δ measures the degree to which $\sigma(\cdot)$ violates the submodularity. $\sigma(\cdot)$ gets closer to the submodularity for a smaller Δ, and is submodular when $\Delta = 0$.

For a general hypergraph, Δ is not bounded, and can be as large as $O(|V|)$. This is the reason that SIMP in a general hypergraph is not approximable.

B. OSNs as Scale-Free Hypergraphs

Recent studies in network science show that OSNs are scale-free networks [19], meaning that the degree distribution in an OSN follows the power-law distribution [22]. Let p_d denote the fraction of nodes with a hyperdegree d. The power-law means that $p_d = (\gamma - 1)d^{-\gamma}$, in which γ ranges from 2 to 4 in OSNs [19]. Let $\overline{w} = \frac{1}{|E|}\sum_e w_e$ denote the average weight of the hyperedges. We proves that the supmodular degree, denoted as Δ, of most OSNs has the following property $\lim_{|V|\to\infty} \frac{\Delta}{O(|V|)} = 0$, i.e., $\Delta \in o(|V|)$ for most OSNs.

Theorem 5: In scale-free OSNs with γ and $\overline{w}$, it is expected to have $\Delta \in o(|V|)$ when $4 + 6\overline{w}\frac{\gamma-1}{\gamma-2} \leq 3(\frac{\gamma-1}{\overline{w}^{\gamma+1}})^2$.

Proof: Similar to the proof of Theorem 2, we again form an equivalent view of the independent cascade to compute Δ. For each hyperedge e, coins are flipped based on its weight, w_e. Let C_v be the maximum Weakly Connected Component (WCC) containing v via live hyperedges in G after the coin flip. Here, two nodes are weakly connected if there exists a path connecting them (see Definition 3), when each hyperedge is regarded as bi-directional. We claim that $M_v \subseteq C_v$. This is because nodes outside C_v cannot increase the marginal gain of v. Consequently, we can conclude that Δ is upper-bounded by the size of the maximum WCC via live hyperedges in G.

The following part of the proof uses Molloy's Theorem 1 in [23] to derive the size of the maximum WCC through live hyperedges in G. All prerequisites of this theorem are satisfied.

In addition, Molloy's Theorem 1 was developed under general graphs, but it can be applied to hypergraphs through separating each hyperedge into a set of normal edges. Let q_d denote the fraction of nodes with a live-hyperdegree d (live-hyperdegree is the hyperdegree that only counts live hyperedges). We have:

$$q_d = \frac{1}{\overline{w}} p_{d/\overline{w}} = \frac{\gamma - 1}{\overline{w}} \Big(\frac{d}{\overline{w}}\Big)^{-\gamma} \tag{3}$$

Let $\overline{d} = \overline{w}\frac{\gamma-1}{\gamma-2}$ be the average live-hyperdegree. We define:

$$\begin{aligned}
\chi(\alpha) &= \overline{d} - 2\alpha - \sum_{d=1}^{\infty} d q_d \Big(1 - \frac{2\alpha}{\overline{d}}\Big)^{\frac{d}{2}} \\
&\approx \overline{d} - 2\alpha - \int_{d=1}^{\infty} d q_d \Big(1 - \frac{2\alpha}{\overline{d}}\Big)^{\frac{d}{2}} \mathrm{d}d - \psi \\
&= \overline{d} - 2\alpha - \frac{\gamma - 1}{\overline{w}^{\gamma+1}} \Big[\frac{2d^{1-\gamma}(1 - \frac{2\alpha}{\overline{d}})^{\frac{d}{2}+1}}{d+2}\Big]\Big|_{d=1}^{\infty} - \psi \\
&= \overline{d}(1 - \frac{2\alpha}{\overline{d}}) + \frac{\gamma - 1}{\overline{w}^{\gamma+1}} \times \frac{2}{3}(1 - \frac{2\alpha}{\overline{d}})^{\frac{3}{2}} - \psi
\end{aligned} \tag{4}$$

Eq. 4 is derived under $0 \le 2\alpha \le \overline{d}$. E q. 4 does not consider the case of $d < 1$ (no impact on the graph connectivity). Here, ψ comes from Euler-Maclaurin formula for integral boundaries:

$$\psi = \frac{1}{2} \times [\frac{\gamma - 1}{\overline{w}^{\gamma+1}}(1 - \frac{2\alpha}{\overline{d}})^{\frac{1}{2}} + 0] = \frac{1}{2}\frac{\gamma - 1}{\overline{w}^{\gamma+1}}(1 - \frac{2\alpha}{\overline{d}})^{\frac{1}{2}} \tag{5}$$

Let α_D be the smallest positive root for $\chi(\alpha) = 0$. We have the following result for α_D and definition for ϵ_D:

$$(1 - \frac{2\alpha_D}{\overline{d}})^{\frac{1}{2}} = \frac{\sqrt{12(\frac{\gamma-1}{\overline{w}^{\gamma+1}})^2 + 9\overline{d}^2} - 3\overline{d}}{4\frac{\gamma-1}{\overline{w}^{\gamma+1}}} \tag{6}$$

$$\begin{aligned}
\epsilon_D &= 1 - \sum_{d} q_d (1 - \frac{2\alpha_D}{\overline{d}})^{\frac{d}{2}} \le 1 - q_1 (1 - \frac{2\alpha_D}{\overline{d}})^{\frac{1}{2}} \\
&= \frac{(4 + 3\overline{d}) - \sqrt{12(\frac{\gamma-1}{\overline{w}^{\gamma+1}})^2 + 9\overline{d}^2}}{4}
\end{aligned} \tag{7}$$

We can have $\epsilon_D \le 0$ when $4 + 6\overline{w}\frac{\gamma-1}{\gamma-2} \le 3(\frac{\gamma-1}{\overline{w}^{\gamma+1}})^2$. Molloy's Theorem 1 in [23] proved that the size of the maximum WCC via live hyperedges in G (the size of a giant component in a random graph) is $\epsilon_D|V| + o(|V|)$, or just $o(|V|)$ when $\epsilon_D \le 0$. Since Δ is upper-bounded by the size of the maximum WCC via live hyperedges in G, the proof completes. ■

The insight of Theorem 5 is that the influence propagation from a node decays quickly with respect to $\overline{w}$. The influence of a node becomes limited when $\overline{w}$ is small (we have $\Delta = 0$ when $\overline{w} = 0$). Note that $4 + 6\overline{w}\frac{\gamma-1}{\gamma-2} \le 3(\frac{\gamma-1}{\overline{w}^{\gamma+1}})^2$ is satisfied for most OSNs that have $\overline{w} < 0.7$ and $\gamma > 2.1$ [21]. Therefore, the SIMP in OSNs is approximable. In addition, γ has a big impact on Δ. When γ is smaller, the hyperdegree distribution is closer to "uniform," and Δ is larger. On the other hand, when γ is larger, fewer nodes have large hyperdegrees and Δ becomes smaller. A smaller Δ can represent a smaller gap from the submodularity. The following subsections will use the existing supermodularity techniques [15, 24] to approximate the SIMP with a bounded Δ in OSNs. We simplify Feldman's algorithm and proof [15] to a special case for the SIMP.

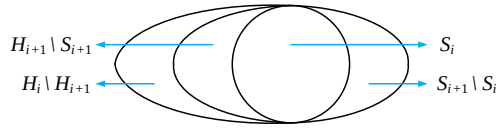


Fig. 3. Relationship between S_i and H_i. We have $S_{i+1} = (S_{i+1} \setminus S_i) \cup S_i$, $H_{i+1} = (H_{i+1} \setminus S_{i+1}) \cup S_{i+1}$, and $H_i = (H_i \setminus H_{i+1}) \cup (H_{i+1} \setminus S_{i+1}) \cup S_i$.

C. Improved Greedy

By leveraging the structural properties of OSNs, Δ is shown to be bounded in OSNs (Theorem 5). Consequently, approximation algorithms become possible. The key idea is that, when the node v is selected as a seed node, nodes in M_v should be further considered, since they can improve v's influence propagations. This observation can improve Algorithm 1.

Consequently, Algorithm 2 is proposed as another greedy algorithm. In line 1, it initializes $i = 0$ and $S_0 = \emptyset$. Lines 2 to 5 describe greedy iterations. While Algorithm 1 iteratively selects one seed node, Algorithm 2 iteratively selects a set of seed nodes, in order to mitigate the negative impact resulting from the non-submodularity. In line 3, once v is selected as a seed node, partial nodes in M_v (denoted as M'_v) are jointly selected as seed nodes. The greedy criterion is that v and M'_v can maximize the marginal gain of the current seed set, i.e., maximize $\sigma(S_i \cup \{v\} \cup M'_v) - \sigma(S_i)$. The constraint is that $|S_i \cup \{v\} \cup M'_v| \le k$, i.e., at most k seeds are selected. Lines 4 and 5 update the seed set S_i and the index i. The greedy iteration terminates once k seed nodes are selected.

Computing $\sigma(S)$ for a given S is considered to take $O(|E|)$ in the Monte Carlo simulation [21]. Algorithm 2 has at most k greedy iterations, and each iteration it exhausts v and M_v in line 3. Consequently, the time complexity of Algorithm 2 is $O(2^{\Delta} k|V||E|)$. We claim that Algorithm 2 is bounded:

***Theorem** 6:* Algorithm 2 has an approximation ratio of $\frac{1}{\Delta+2}$ to the optimal algorithm.

Proof: Let S^* denote the optimal set of seed nodes, in terms of maximizing $\sigma(\cdot)$. An auxiliary parameter, H_i, is used. With $H_0 = S^*$, H_i is recursively defined as an arbitrary subset of $H_{i-1} \cup S_i$, under the constraint that $S_i \subseteq H_i$ and $|H_i| = k$. Intuitively, H_i consists of S_i and a part of S^*. The relationship between S_i and H_i is shown in Fig. 3. When i becomes larger, nodes from S_i are added to H_i, and nodes in S^* are removed from H_i, maintaining $|H_i| = k$. By definition, we have:

$$|H_{i+1}| = |H_i| - |H_i \setminus H_{i+1}| + |S_{i+1} \setminus S_i| \tag{8}$$

Since $|H_{i+1}| = |H_i| = k$, we have:

$$|H_i \setminus H_{i+1}| = |S_{i+1} \setminus S_i| \le |\{v\} \cup M'_v| \le 1 + \Delta \tag{9}$$

This is because $M'_v \subseteq M_v$ and $\Delta = \max_v |M_v|$. Eq. 9 means that, in each greedy iteration, at most $1 + \Delta$ nodes in S^* are ignored by Algorithm 2.

We claim that the marginal gain in each greedy iteration of Algorithm 2 has a lower bound with respect to $\sigma(H_i)$:

$$\sigma(H_i) - \sigma(H_{i+1}) \le (\Delta+1) \times [\sigma(S_i \cup \{v\} \cup M'_v) - \sigma(S_i)] \tag{10}$$

Algorithm 2 Improved Greedy (IG)

Input: a hypergraph, G, and a constant, k.
Output: a set of seed nodes, S, initiated $\emptyset$.

1: **while** $|S| < k$ **do**
2: Find $\arg\max_{v \in V, M'_v \subseteq M_v} \sigma(S \cup \{v\} \cup M'_v) - \sigma(S)$ s.t. $|S \cup \{v\} \cup M'_v| \leq k$.
3: Update $S = S \cup \{v\} \cup M'_v$.

To prove Eq. 10, let us order nodes of $H_i \setminus H_{i+1}$ in an arbitrary order (say $v_1, v_2, ..., v_l$), and let $H_i^j = H_i \setminus \{v_1, v_2, ..., v_j\}$ for $1 \leq j \leq l$ ($H_i^0 = H_i$ and $H_i^l \subseteq H_{i+1}$). For each j, we have:

$$\begin{aligned}
&\sigma(S_i \cup \{v_j\} \cup (M_{v_j} \cap H_i^j)) - \sigma(S_i) \\
\geq &\sigma(S_i \cup \{v_j\} \cup (M_{v_j} \cap H_i^j)) - \sigma(S_i \cup (M_{v_j} \cap H_i^j)) \\
\geq &\sigma(S_i \cup \{v_j\} \cup H_i^j) - \sigma(S_i \cup H_i^j) \qquad (11)
\end{aligned}$$

The first inequality is from the monotonicity in Theorem 2, since $S_i \subseteq S_i \cup (M_{v_j} \cap H_i^j)$ and $\sigma(S_i) \leq \sigma(S_i \cup (M_{v_j} \cap H_i^j))$. The second inequality is from the definition of the modularity set, because only nodes in M_{v_j} can increase the marginal gain of v_j. Hence, nodes in $H_i^j \setminus M_{v_j}$ might decrease the marginal gain of v_j. By accumulating Eq. 11 among j, we obtain:

$$\begin{aligned}
&\sum_{j=1}^{l} [\sigma(S_i \cup \{v_j\} \cup (M_{v_j} \cap H_i^j)) - \sigma(S_i)] \\
\geq &\sum_{j=1}^{l} [\sigma(S_i \cup \{v_j\} \cup H_i^j) - \sigma(S_i \cup H_i^j)] \\
= &\sigma(S_i \cup H_i^0) - \sigma(S_i \cup H_i^l) \geq \sigma(H_i) - \sigma(H_{i+1}) \qquad (12)
\end{aligned}$$

The first inequality is from Eq. 11. The equality results from the definition of H_i^j, since $\{v_j\} \cup H_i^j = H_i^{j-1}$. The last inequality is because $\sigma(S_i \cup H_i^0) = \sigma(H_i)$ and $\sigma(S_i \cup H_i^l) \leq \sigma(H_{i+1})$. We have $\sigma(S_i \cup H_i^0) = \sigma(H_i)$, since $H_i^0 = H_i$ and $S_i \subseteq H_i$. We have $\sigma(S_i \cup H_i^l) \leq \sigma(H_{i+1})$ by the monotonicity, since $S_i \subseteq S_{i+1} \subseteq H_{i+1}$ and $H_i^l \subseteq H_{i+1}$. We have:

$$\begin{aligned}
&(\Delta + 1) \times [\sigma(S_i \cup \{v\} \cup M'_v) - \sigma(S_i)] \\
\geq &\sum_{j=1}^{l} [\sigma(S_i \cup \{v\} \cup M'_v) - \sigma(S_i)] \\
\geq &\sum_{j=1}^{l} [\sigma(S_i \cup \{v_j\} \cup (M_{v_j} \cap H_i^j)) - \sigma(S_i)] \\
\geq &\sigma(H_i) - \sigma(H_{i+1}) \qquad (13)
\end{aligned}$$

The first inequality results from Eq. 9, in which $1 \leq j \leq l = |H_i \setminus H_{i+1}| \leq \Delta + 1$. The second inequality comes from line 3 in Algorithm 2, which always selects the maximum marginal gain in each greedy iteration. The third inequality comes from Eq. 12. Therefore, Eq. 10 is valid.

Since the marginal gain in each greedy iteration of Algorithm 2 has a lower bound, we can accumulate Eq. 10 among all greedy iterations (note that $S_{i+1} = S_i \cup \{v\} \cup M'_v$):

$$\sigma(H_0) - \sigma(H_i) \leq (\Delta + 1) \times [\sigma(S_i) - \sigma(S_0)] \qquad (14)$$

Since $H_0 = S^*$, $H_i = S_i = S$ when Algorithm 2 terminates, and $S_0 = \emptyset$, we have $\sigma(S^*) \leq (\Delta + 2) \times \sigma(S)$. ■

The key insight of Theorem 6 is that Algorithm 2 ignores at most $1+\Delta$ nodes in the optimal set of seed nodes, resulting in a bounded marginal gain for each greedy iteration. Theorem 6 does not violate the inapproximability in Theorem 4, since Δ can be as large as $\Theta(|V|)$ in a general hypergraph. However, Algorithm 2 still has a critical drawback. Although Theorem 5 validates that $\Delta \in o(|V|)$ in OSNs, Δ may still be numerically large, in terms of the time complexity and the approximation ratio. As a result, Algorithm 2 may perform poorly in an OSN with a small γ, especially when γ gets closer to 2.

Algorithm 3 Capped Greedy (CG)

Input: a hypergraph, G, and a constant, k.
Output: a set of seed nodes, S, initiated $\emptyset$.

1: **for** each $v' \in V$ **do**
2: **for** each Δ' from 1 to Δ **do**
3: **for** each $S' \subseteq \{v'\} \cup M_{v'}$ s.t. $|S'| \leq \min\{k, \Delta'\}$ **do**
4: **while** $|S'| < k$ **do**
5: Find $\arg\max_{v \in V, M'_v \subseteq M_v} \sigma(S' \cup \{v\} \cup M'_v) - \sigma(S')$ s.t. $|S' \cup \{v\} \cup M'_v| \leq k$ and $M'_v \leq \Delta'$.
6: Update $S' = S' \cup \{v\} \cup M'_v$.
7: **if** $\sigma(S') > \sigma(S)$ **then**
8: Update $S = S'$.

D. Capped Greedy

Since Δ has a critical impact on Algorithm 2, we need to further identify its role in the algorithm design. The key idea is that, although $|M_v|$ could be large, not all nodes in M_v have huge impacts on the marginal gain of v for $\sigma(\cdot)$. Intuitively, only v's neighbors, who share hyperedges with large weights, are important in M_v. Moreover, the optimal set of seed nodes are not able to include all nodes in M_v if $|M_v| > k$. Therefore, capping the number of selected seed nodes in M_v might lead to a better performance, since low impact nodes in M_v can be replaced by high impact nodes outside M_v. To find out the best cap, we can simply exhaust all possible caps.

As a result, Algorithm 3 is proposed as an extension of Algorithm 2. In line 1, it initializes $S = \emptyset$. Line 2 includes a loop statement to exhaust all possible scenarios, in terms of the combination of each node $v' \in V$, each Δ' from 1 to Δ, and each set $S_0 \subseteq M_{v'}$ constrained by $|S_0| = k \mod (\Delta'+1)$. Instead of Δ, Δ' is used as the cap. Lines 3 to 7 are basically the same as Algorithm 2, except for the cap. This part embeds Algorithm 2 to search the set of seed nodes in each possible scenario. The cap is added at the end of line 5 ($M'_v \leq \Delta'$), while Algorithm 2 uses $M'_v \leq \Delta$ by default ($\Delta = \max_v |M_v|$ by definition). Lines 8 and 9 record the best set of seed nodes searched among all possible scenarios (specified by line 2).

The total number of all possible scenarios is $O(|V| \cdot \Delta \cdot 2^{\Delta})$. The time complexity of Algorithm 2 is $O(2^{\Delta} k |V||E|)$. Hence, the time complexity of Algorithm 3 is $O(4^{\Delta} \Delta k |V|^2 |E|)$. But Algorithm 3 has a better bound than Algorithm 2:

***Theorem** 7:* Algorithm 3 has an approximation ratio of $1 - e^{-\frac{1}{\Delta+1}}$ to the optimal algorithm.

Proof: Let S^* denote the optimal set of seed nodes, in terms of maximizing $\sigma(\cdot)$. Since Algorithm 3 exhausts all possible v', Δ', and S_0 in line 2, there must exist a scenario in which $v' = \arg\max_{v'} |M_{v'} \cap S^*|$, $\Delta' = |M_{v'} \cap S^*|$, $S_0 \subseteq M_{v'} \cap S^*$, and $|S_0| = k \mod (\Delta' + 1)$. All the following proof is based

on the above scenario, although Algorithm 3 picks the best effort among all scenarios (lines 8 and 9).

In the above scenario, we claim that $\sigma(S_i)$ in each greedy iteration of Algorithm 2 has a lower bound to $\sigma(S^*)$:

$$\sigma(S_i) \geq (1-\frac{1}{k'})^i \times \sigma(S_0) + [1-(1-\frac{1}{k'})^i] \times \sigma(S^*) \quad (15)$$

Here, k' is defined as $k - [k \bmod (\Delta'+1)]$. In other words, k' is the largest multiple of $\Delta'+1$ constrained by $k' \leq k$. Eq. 15 is proved by induction. It is trivial that Eq. 15 holds when $i=0$, since $(1-\frac{1}{k'})^0 = 1$. Assume that Eq. 15 holds for i, and we prove that Eq. 15 holds for $i+1$. Since $S_0 \subseteq (M_{v'} \cap S^*) \subseteq S^*$ and $|S_0| = k \bmod (\Delta'+1)$, we have:

$$|S^* \setminus S_0| = |S^*| - |S^0| = k - [k \bmod (\Delta'+1)] = k' \quad (16)$$

Similarly, let us order nodes of $|S^* \setminus S_0|$ in an arbitrary order (say $v_1, v_2, ..., v_{k'}$), and let $S^*_j = \{v_1, v_2, ..., v_j\}$ for $1 \leq j \leq j'$ ($S^*_0 = \emptyset$). Similar to Eq. 11, for each j, we have:

$$\begin{aligned}
&\sigma(S_i \cup \{v_j\} \cup (M_{v_j} \cap S^*)) - \sigma(S_i) \\
\geq &\sigma(S_i \cup \{v_j\} \cup (M_{v_j} \cap S^*_{j-1})) - \sigma(S_i) \\
\geq &\sigma(S_i \cup \{v_j\} \cup (M_{v_j} \cap S^*_{j-1})) - \sigma(S_i \cup (M_{v_j} \cap S^*_{j-1}))) \\
\geq &\sigma(S_i \cup \{v_j\} \cup S^*_{j-1}) - (S_i \cup S^*_{j-1}) \quad (17)
\end{aligned}$$

The first and second inequalities are from the monotonicity in Theorem 2, since $S^*_{j-1} \subseteq S^*$ and $S_i \subseteq S_i \cup (M_{v_j} \cap S^*_{j-1}))$. So $\sigma(S_i \cup \{v_j\} \cup (M_{v_j} \cap S^*)) \geq \sigma(S_i \cup \{v_j\} \cup (M_{v_j} \cap S^*_{j-1}))$ and $\sigma(S_i) \leq \sigma(S_i \cup (M_{v_j} \cap S^*_{j-1})))$. The third inequality is from the definition of the modularity set, since only nodes in M_{v_j} can increase the marginal gain of v_j (other nodes might decrease the marginal gain of v_j). By accumulating Eq. 17 among j, we have the following inequality:

$$\begin{aligned}
&\sum_{j=1}^{k'} [\sigma(S_i \cup \{v_j\} \cup (M_{v_j} \cap S^*)) - \sigma(S_i)] \\
\geq &\sum_{j=1}^{k'} [\sigma(S_i \cup \{v_j\} \cup S^*_{j-1}) - (S_i \cup S^*_{j-1})] \\
= &\sigma(S_i \cup S^*) - \sigma(S_i) \geq \sigma(S^*) - \sigma(S_i) \quad (18)
\end{aligned}$$

The first inequality is from Eq. 17. The equality results from the definition of S^*_j, since $\{v_j\} \cup S^*_{j-1} = S^*_j$. Note that, since $S^*_{k'} \subseteq S^* \setminus S_0$ and $S_0 \subseteq S_i$, we have $S_i \cup S^*_{k'} = S_i \cup S^*$. We have $S_i \cup S^*_0 = S_i$ since $S^*_0 = \emptyset$. The last inequality is from the monotonicity, since $S^* \subseteq S_i \cup S^*$. We have:

$$\begin{aligned}
&\sigma(S_{i+1}) - \sigma(S_i) = \sigma(S_i \cup \{v\} \cup M'_v) - \sigma(S_i) \\
= &\frac{1}{k'} \sum_{j=1}^{k'} [\sigma(S_i \cup \{v\} \cup M'_v) - \sigma(S_i)] \\
\geq &\frac{1}{k'} \sum_{j=1}^{k'} [\sigma(S_i \cup \{v_j\} \cup (M_{v_j} \cap S^*)) - \sigma(S_i)] \\
\geq &\frac{1}{k'} [\sigma(S^*) - \sigma(S_i)] \quad (19)
\end{aligned}$$

The first inequality is because line 5 in Algorithm 3 always selects the maximum marginal gain in each greedy iteration. $M_{v_j} \cap S^*$ is also constrained by $|(M_{v_j} \cap S^*)| \leq \Delta'$, since the scenario sets $v' = \arg\max_{v'} |M_{v'} \cap S^*|$ and $\Delta' = |M_{v'} \cap S^*|$. The second inequality is from Eq. 18. We rewrite Eq. 19 as:

$$\begin{aligned}
&\sigma(S_{i+1}) \geq \frac{1}{k'} [\sigma(S^*) - \sigma(S_i)] + \sigma(S_i) \\
= &\frac{1}{k'} \times \sigma(S^*) + (1-\frac{1}{k'}) \times \sigma(S_i) \\
\geq &(1-\frac{1}{k'})^{i+1} \times \sigma(S_0) + [1-(1-\frac{1}{k'})^{i+1}] \times \sigma(S^*) \quad (20)
\end{aligned}$$

TABLE I
DATASET STATISTICS.

	Forum	Board	Citation
Number of nodes	899	355	16,726
Number of hyperedges	67,332	2,684	92,462
Maximum hyperdegree	8,577	107	351
Power-law exponent γ	2.36	3.50	3.36

The first equality is from Eq. 19 and the last equality is from the induction hypothesis (substituting $\sigma(S_i)$ in Eq. 15). As a result, Eq. 15 is proved by induction.

Since each greedy iteration of Algorithm 3 selects at most $\Delta'+1$ seed nodes, Algorithm 3 has at least $\lfloor k/(\Delta'+1) \rfloor = k'/(\Delta'+1)$ greedy iterations. If we use $i = k'/(\Delta'+1)$ for Eq. 15, the proof completes:

$$\begin{aligned}
&\sigma(S) \geq \sigma(S_{k'/(\Delta'+1)}) \\
\geq &(1-\frac{1}{k'})^{\frac{k'}{\Delta'+1}} \times \sigma(S_0) + [1-(1-\frac{1}{k'})^{\frac{k'}{\Delta'+1}}] \times \sigma(S^*) \\
\geq &[1-(1-\frac{1}{k'})^{\frac{k'}{\Delta'+1}}] \times \sigma(S^*) \\
\geq &(1-e^{-\frac{1}{\Delta'+1}}) \times \sigma(S^*) \geq (1-e^{-\frac{1}{\Delta+1}}) \times \sigma(S^*) \quad (21)
\end{aligned}$$

The approximation ratio is $1-e^{-\frac{1}{\Delta'+1}} \geq 1-e^{-\frac{1}{\Delta+1}}$. ■

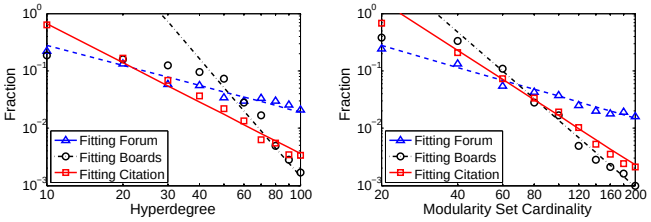
E. Time Complexity Reduction

Although Algorithm 3 has a better bound than Algorithm 2, its time complexity is much larger. However, we claim that the time complexity of Algorithm 3 can be reduced for OSNs. This is mainly because we do not need to exhaust all possible scenarios for practical usage. Rather than exhausting Δ' from 1 to Δ, we can simply stop at a small constant. For example, we only exhaust Δ' from 1 to 3. This is because only v's neighbors who share hyperedges with large weights are important in M_v (people are not likely to have many close friends). Similarly, we do not need to exhaust each node v for the initialization of S_0. Instead, we can focus on the largest-hyperdegree nodes (e.g., only the top 100 nodes). This is because the optimal set of seed nodes is not likely to exclude the largest-hyperdegree nodes. Using this approach, the time complexity of Algorithm 3 can be reduced to $O(k|V||E|)$, which is asymptotically the same as Algorithm 1. Our experiments demonstrate that this approach only slightly hurts the performance of Algorithm 3.

VI. EXPERIMENTS

A. Dataset Information and Statistics

Our experiments are based on three datasets (Forum, Board, and Citation) from Tore Opsahl [25]. Forum records user activities in a forum with different topics. Board records directors belonging to the boards of some companies. Citation records collaborations among paper authors. The dataset statistics are shown in Table I. The distributions of node hyperdegree (i.e., d_v) and modularity set cardinality (i.e., $|M_v|$) are shown in Fig. 4. For the above three datasets, flags (triangles, circles, and squares) represent the real distributions by statistics, and lines (dotted, dashed, and solid) are the fitting curves. Fig. 4 is plotted in a log-log manner, and the y-axis shows the fraction of nodes corresponding to the x-axis. Fig. 4(a) validates the



(a) Distribution of d_v among v. (b) Distribution of $|M_v|$ among v.

Fig. 4. Distribution of d_v and $|M_v|$ in three datasets.

TABLE II
RUNNING TIME STATISTICS.

	WH	NG	HG	IG	CG-2	CG-3	CG
Forum	2s	31m	2d	45m	75m	155m	22h
Board	1s	7m	21h	15m	31m	58m	5h
Citation	18s	85m	28d	114m	169m	423m	5d

power-law distribution. It can be seen that the fraction of nodes with hyperdegree d is proportional to $d^{-\gamma}$ in each of these three dataset. The distribution of modularity set cardinality also follows power-law, as shown in Fig. 4(b).

B. Comparison Algorithm and Performance

Algorithms 1, 2, and 3 are denoted as NG, IG, and CG, respectively. Comparison algorithms are:

- Weighted Hyperdegree (WH). It ranks all nodes by their hyperdegrees and selects the top k nodes as seed nodes.
- Hyperedge-aware Greedy (HG) from Hung et al. [5]. It iteratively selects the set of nodes that maximizes the ratio of marginal gain to set cardinality. The greedy iteration terminates when k nodes are selected. HG is not bounded.

All evaluation results are shown in Fig. 5. Figs. 5(a), 5(b), and 5(c) correspond to the Forum, Board, and Citation datasets, respectively. A larger result represents a better performance, since seed nodes could eventually influence more nodes on expectation. Interestingly, Fig. 5 shows that not all algorithms follow the principal of diminishing return. The marginal gain of one seed node might not scale down with respect to the number of existing seed nodes in S. This is because our SIMP in hypergraphs is not submodular, i.e., $\sigma(S)/|S|$ might be larger than $\sigma(S')/|S'|$ for $S' \subset S$. Among all the algorithms, CG achieves the best performances in all these three datasets, while WH has the worst performances. This is simply because CG considers the impact of crowd influences while WH does not. Compared to other algorithms, CG has at least 10%, 5%, and 15% more eventually-influenced users in Forum, Board, and Citation, respectively (for $k = 8$). Compared with NG, the CG achieves 20% more eventually-influenced users in all three datasets. As for NG, HG, and IG their performance has different order in different datasets.

HG has the second best performance, although it does not outperform IG in Citation. This is because NG and IG are essentially special cases of HG. HG reduces to NG by only selecting one node in each greedy iteration, and it reduces to IG by ignoring the set cardinality in each greedy iteration. CG, HG, IG, and NG become identical when only one seed node is selected (i.e., $k = 1$). HG loses to CG, since CG has a better granularity control through its cap to capture the crowd influence. Moreover, HG is unbounded and has a larger time complexity than CG. Finally, we find that the network density may not significantly change the algorithm performance. Both Board and Citation are sparse, but their algorithm performance gaps are not similar. Forum and Citation have different densities, but their algorithm performance gaps are similar.

C. Running Time and Complexity Reduction

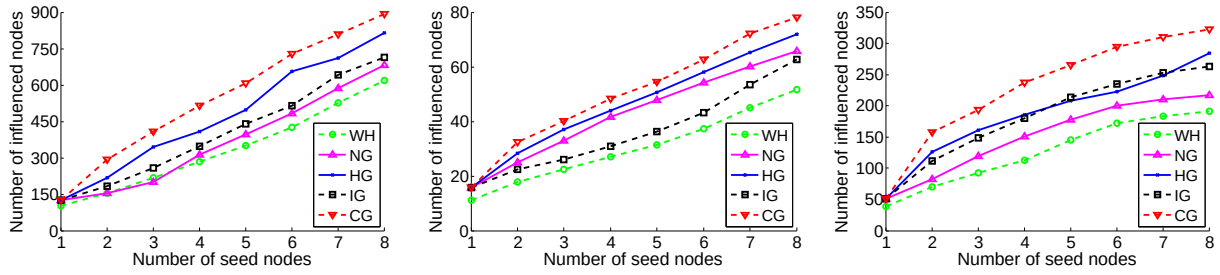
This subsection evaluates the running time of the proposed algorithms. Codes are implemented in Matlab and are executed on Dell Inspiron i15RN-3647BK laptop with a 2.5GHz Intel Core i5 2450M processor. We further introduce two variations of CG by using different maximum cap sizes. The first one is CG-2, using 2 as its maximum cap size (Δ' ranges from 1 to 2). In each greedy iteration, CG-2 selects at most 2 nodes into seed nodes. The second one is CG-3, using 3 as its maximum cap size. We evaluate the impact of the cap size, in terms of both performance and running time. k is set to be 8.

We start with the running time, as shown in the above table (units are seconds, minutes, hours, and days). WH is fastest, since it is linear and does not evaluate $\sigma(S)$ for a given S. As a trade-off, it has the worst performance. Both NG and IG take minutes. IG runs slower than NG, since IG exhausts a set of nodes during each greedy iteration. However, IG is not exponentially slower than NG, since IG has fewer greedy iterations. The performance gap between NG and IG is limited in these three datasets. HG and CG have the longest running times to obtain the best performances. The running times of both HG and CG grow quickly with respect to the dataset size. However, if we cap Δ' at 2 or 3, the running time of CG can be significantly reduced. CG-2 and CG-3 have asymptotically the same time complexity as NG. Therefore, the comparison between CG and NG is fair.

Fig. 6 shows the impact of the cap size for CG in these three datasets. CG-2, CG-3, and CG have close performances (CG-2 $\leq$ CG-3 $\leq$ CG). in all three datasets, the performance different between CG-2 and CG-3 is less than 10%. This is because CG is not likely to select a large set of nodes in each iteration (people are not likely to have many close friends). CG-3 has almost the same performance as CG, especially when seed nodes are few. If we jointly consider the running time aspect, capping Δ' to 3 in CG is a practical strategy for large-scale OSNs.

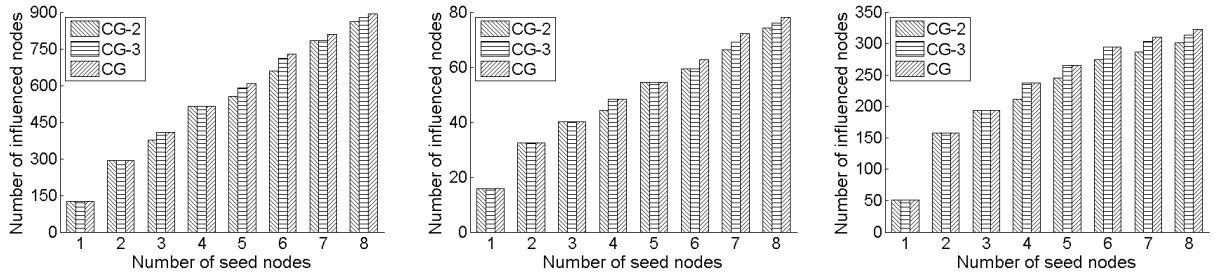
VII. CONCLUSION

Motivated by the impact of the crowd influence, this paper studies the Social Influence Maximization Problem (SIMP) in Online Social Networks (OSNs).The proposed problem turns out to be NP-hard, monotone, non-submodular, and inapproximable within a ratio of $|V|^{\epsilon-1}$ for any $\epsilon > 0$ in a general hypergraph. However, since user connections in OSNs are not random, approximations could be obtained by leveraging the structural properties of OSNs. The supermodularity, Δ, can measure to what degree our problem violates the submodularity. We prove that the supmodular degree, denoted as Δ, of most OSNs has the following property $\lim_{|V|\to\infty} \frac{\Delta}{O(|V|)} = 0$, i.e., $\Delta \in o(|V|)$ for most OSNs. Based on the property of



(a) Forum dataset. (b) Board dataset. (c) Citation dataset.

Fig. 5. Algorithm performance for SIMP in hypergraphs.



(a) Forum dataset. (b) Board dataset. (c) Citation dataset.

Fig. 6. The impact of cap size for CG.

OSNs, two approximation algorithms are applied with ratios of $\frac{1}{\Delta+2}$ and $1-e^{-1/(\Delta+1)}$, respectively. Experiments demonstrate the efficiency and effectiveness of our algorithms, compared with the traditional naive greedy algorithm.

VIII. Acknowledgement

This research was supported in part by NSF grants CNS 1824440, CNS 1828363, CNS 1757533, CNS 1629746, CNS-1651947, CNS 1564128.

References

[1] H. Nguyen and R. Zheng, "On budgeted influence maximization in social networks," *IEEE Journal on Selected Areas in Communications*, 2013.

[2] X. Yang, H. Steck, and Y. Liu, "Circle-based recommendation in online social networks," in *ACM SIGKDD*, 2012.

[3] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *ACM SIGKDD*, 2003.

[4] A. Guille, H. Hacid, C. Favre, and D. A. Zighed, "Information diffusion in online social networks: A survey," *ACM SIGMOD Record*, 2013.

[5] H.-J. Hung et al., "When social influence meets item inference," in *ACM KDD*, 2016.

[6] M. Edelson, T. Sharot, R. J. Dolan, and Y. Dudai, "Following the crowd: brain substrates of long-term memory conformity," *Science*, 2011.

[7] N. Wang and J. Wu, "Latency minimization through optimal user matchmaking in multi-party online applications," in *IEEE WoWMoM*, 2018.

[8] H. Zhang, D. T. Nguyen, H. Zhang, and M. T. Thai, "Least cost influence maximization across multiple social networks," *IEEE/ACM ToN*, 2016.

[9] J. L. Z. Cai, M. Yan, and Y. Li, "Using crowdsourced data in location-based social networks to explore influence maximization," in *IEEE INFOCOM*, 2016.

[10] G. Tong, W. Wu, S. Tang, and D.-Z. Du, "Adaptive influence maximization in dynamic social networks," *IEEE/ACM ToN*, 2016.

[11] S. Galhotra, A. Arora, S. Virinchi, and S. Roy, "Asim: A scalable algorithm for influence maximization under the independent cascade model," in *ACM WWW*, 2015.

[12] W. Chen et al., "Influence maximization in social networks when negative opinions may emerge and propagate," in *SDM*, 2011.

[13] A. Goyal, F. Bonchi, and L. V. Lakshmanan, "A data-based approach to social influence maximization," *VLDB Endowment*, 2011.

[14] J. Tang, X. Tang, and J. Yuan, "Profit maximization for viral marketing in online social networks: Algorithms and analysis," *IEEE TKDE*, 2017.

[15] M. Feldman and R. Izsak, "Constrained monotone function maximization and the supermodular degree," in *ACM-SIAM SODA*, 2014.

[16] S. Dughmi, "Algorithmic information structure design: a survey," *ACM SIGecom Exchanges*, 2017.

[17] S. Fujishige and S. Isotani, "A submodular function minimization algorithm based on the minimum-norm base," *PJO*, 2011.

[18] M. Sviridenko, J. Vondrák, and J. Ward, "Optimal approximation for submodular and supermodular optimization with bounded curvature," in *ACM-SIAM SODA*, 2015.

[19] A. Mislove et al., "Measurement and analysis of online social networks," in *ACM IMC*, 2007.

[20] R. Kumar, J. Novak, and A. Tomkins, "Structure and evolution of online social networks," in *Link mining: models, algorithms, and applications*, 2010.

[21] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *ACM SIGKDD*, 2010.

[22] T. Gradowski and A. Krawiecki, "Majority-vote model on scale-free hypergraphs," *Acta Physica Polonica A*, 2015.

[23] M. Molloy and B. Reed, "The size of the giant component of a random graph with a given degree sequence," *Combinatorics, probability and computing*, vol. 7, no. 3, pp. 295–305, 1998.

[24] U. Feige and R. Izsak, "Welfare maximization and the supermodular degree," in *ACM ITCS*, 2013, pp. 247–256.

[25] https://toreopsahl.com/datasets/#newman2001.