Accepted Manuscript

Viral marketing of online game by DS decomposition in social networks

Chuangen Gao, Hai Du, Weili Wu, Hua Wang

PII: S0304-3975(19)30153-7
DOI: https://doi.org/10.1016/j.tcs.2019.03.006
Reference: TCS 11938

To appear in: *Theoretical Computer Science*

Received date: 5 January 2019
Revised date: 23 February 2019
Accepted date: 3 March 2019

Please cite this article in press as: C. Gao et al., Viral marketing of online game by DS decomposition in social networks, *Theoret. Comput. Sci.* (2019), https://doi.org/10.1016/j.tcs.2019.03.006

Viral Marketing of Online Game by DS decomposition in Social Networks

Chuangen Gao[a], Hai Du[b], Weili Wu[c], Hua Wang[a,*]

[a]*School of Computer Science and Technology, Shandong University, Jinan, China*
[b]*School of Mathematics and Information Science, Shaanxi Normal University,Xi'an, China*
[c]*Department of Computer Science, University of Texas at Dallas,Texas, USA*

Abstract

In social networks, the spread of influence has been studied extensively, but most efforts in existing literature are made on the product used by a single person. This paper attempts to address the product which is used by many persons such as the online game. When multiple people participate in one game, interaction between users is accompanied by browsing and clicking on advertisements, and operators can also earn certain advertising revenues. All these revenues are related to information interaction between people involved in one game. We use game profit to represent all of the revenues gained from players involved in one game and model the game profit maximization problem in social networks, which finds a seed set to maximize the game profit between players who are influenced to buy the game. We prove that the problem is NP-hard and the objective function is neither submodular nor supermodular. To solve it, we decompose it into the Difference between two Submodular functions (DS decomposition) and propose four heuristic algorithms. To address the complexity of computing objective function, we design a new sampling method based on reverse reachable set technology. Experiment results on real datasets show that our approaches perform well.

Keywords: social networks, game profit, influence spread, non-submodular

*Corresponding author
Email addresses: gaochuangen@gmail.com (Chuangen Gao), duhai@snnu.edu.cn (Hai Du), weiliwu@utdallas.edu (Weili Wu), wanghua@sdu.edu.cn (Hua Wang)

optimization, DS decomposition

1. Introduction

As online social networks (OSN) grow rapidly[1, 2, 3], the information diffusion and social influence get into people's daily life deeply. Therefore, the influence-driven information technology and influence-based research subjects have been studied extensively in literatures. One of subjects is the viral marketing. In fact, advertisements in online social networks have gained better results than traditional medias sometimes, such as newspapers and televisions. Among existing works in viral marketing, most of efforts in the literature are made on products used by a single person[4, 5, 6].

In this paper, we consider the product which is used by many persons, i.e., a online game with many players. When multiple players participate in an online game, interaction behaviors between game players are always accompanied by browsing on advertisements showed on the game scene, which will lead to advertising revenue. The more frequent the interaction between players, the more times an advertisement is presented and viewed which means more revenues[7]. All these benefits or revenues are related to information interaction between people involved in one game. We use game profit to represent the revenues gained from game players mentioned above.

In this paper, we study the problem of finding a seed set to maximize the game profit between players who are influenced to play the game by the information diffusion in social networks. The contributions of this paper are summarized as follows.

- We propose a new problem named game profit maximization and we analyze its modularity which is neither submodular nor supermodular and complexity which is NP-hard.
- We propose a new method for non-submodular optimization that decomposes the objective function of game profit into the difference of two submodular functions which are monotone nondecreasing.

- Based on the modular lower and upper bounds of decomposed submodular functions mentioned above, we propose four modular functions to approximate the original function and design four heuristic algorithms to solve the game profit maximization problem.
- To address the complexity of computing the value of objective function, we design a new sampling method based on reverse set technology which is highly scalable instead of using Monte Carlo simulations.
- Through real data sets, we verify the effectiveness of our proposed algorithms.

The rest of the paper is organized as follows. Sec. 2 is devoted to the related work. The game maximization problem is proposed in Sec. 3. The decomposition of objective function is provided in Sec. 4 and corresponding algorithms are provided in Sec. 5. The experiments are presented in Sec. 6.

2. Related Works

Kempe et al. [1] formulate the influence maximization problem under information diffusion IC model and the LT model and provide a greedy algorithm with an approximation ratio. And they show the influence maximization is an NP-hard problem and computing the influence spread is #P-hard. Since then, considerable works [3, 8, 9, 10, 11, 12, 13] have been devoted to extending existing models to study influence maximization and its variants. They propose random algorithms[13, 10, 11] which generate $(1 - e^{-1} - \varepsilon)$-approximation with probability $1 - \varepsilon$. For influence maximization, Bharathi et al. [14] had an interesting conjecture that the influence maximization is NP-hard even for arborescence directed into a root. This conjecture is proved by Lu et al. [15] for the IC model. For the LT model, Wang et al. [7] proved that the influence maximization is polynomial-time solvable. This is the first time to know that the IC model and the LT model may give different computational complexity for the same problem.

While most efforts pay attention to the number of nodes affected, Wang et al. [16] study the activity maximization problem which finds k seeds that maximize the sum of activity strengths among the influenced users. Their objective is similar to us. But they only consider the activity between influenced nodes which are connected by edges in social networks. Since an edge in the social network represents a friendship between two nodes, they only consider the activity between influenced node who are friends. While we consider the profit among all active nodes regardless of whether there is an edge connecting them, i.e., whether they are friends. As we know, in the online game, everybody can play a game together even if they are not friends in the social network. Thus our problem can be viewed as a significant extension of it.

From the perspective of optimization theory, both of them belong to nonlinear combinatorial optimization. In this area, the monotone submodular maximization [17] and the non-monotone submodular maximization [18, 19, 20] have been well-studied. However, the monotone non-submodular maximization gets ones' attention only recently [21, 9, 8, 16]. This paper belongs to this research area.

3. Formulation

3.1. Game profit maximization problem

In this paper, we use the directed graph to represent a social network $G = (V, E)$ and choose the IC model to describe information diffusion process. Each node has two states, active and inactive. And each directed edge (u, v) is assigned with a probability p_{uv} so that when u is active, v is activated by u with probability p_{uv}. Initially, every node is inactive. To start an information diffusion process, a set of nodes, called *seeds*, are activated. The process consists of discrete steps. In each step, each node which was newly activated at last step would try to influence its out-neighbors. An active node has only one chance to influence its out-neighbors; this rule means that an active node which

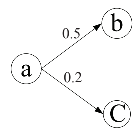


Figure 1: A toy social network

are not newly activated has no ability to influence out-neighbors. The process ends when no node is activated in current step.

In the social network, each node represents a player. Once players become active, they can play a game together, which produce some game profit for network operators. And the game profit is represented by a nonnegative function $c : V \times V \to R^+$. Note that $c(u, v) = c(v, u)$ for the unordered pair $\{u, v\}$ of node u and v. If $u = v$, then $c(u, v) = 0$ which means there is no profit for itself. For any seed set S, denote by $I(S)$ the set of all active nodes at end of the diffusion process. The expected game profit would be defined as

$$f(S) = \mathbb{E}[\sum_{\{u,v\} \subseteq I(S)} c(u, v)] \tag{1}$$

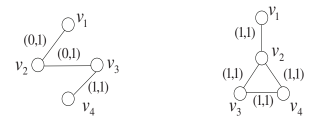
In this paper, we study the following problem.

Definition 1 (Game Profit Maximization)**.** *Given a social network $G = (V, E)$ under the IC model, a profit function $c : V \times V \to R^+$, and a positive integer k, find a set S of k seeds to maximize the expected game profit between player activated by S through influence propagation:*

$$\max f(S) \tag{2}$$

$$s.t. |S| \leq k \tag{3}$$

As an example of game profit maximization problem, we use a toy social network in Fig. 1. There are three nodes $V = \{a, b, c\}$; propagation probabilities are shown on the edges; we set $k = 1$; profit funciton $c(a, b) = 1$, $c(a, c) = 2$, $c(b, c) = 3$. Let us consider a seeding strategy $S = \{a\}$, i.e., choosing node a as seed set. Then $I(\{a\}) = \{a, b, c\}$ with probability $0.5 \times 0.2 = 0.1$, which means both b and c are activated by a; $I(\{a\}) = \{a, b\}$ with probability $0.5 \times (1 - 0.2) =$



(a) Counter example 1 (b) Counter example 2

Figure 2: Counter examples

0.4, which means b is activated by a but c is not activated by a; $I(\{a\}) = \{a, c\}$ with probability $(1-0.5) \times 0.2 = 0.1$, which means c is activated by a but b is not activated by a; $I(\{a\}) = \{a\}$ with probability $(1-0.5) \times (1-0.2) = 0.4$, which means neither b nor c is activated by a; So we have $f(\{a\}) = 0.1 \times \sum_{\{u,v\} \subseteq \{a,b,c\}} c(u,v) + 0.4 \times \sum_{\{u,v\} \subseteq \{a,b\}} c(u,v) + 0.1 \times \sum_{\{u,v\} \subseteq \{a,c\}} c(u,v) + 0.4 \times \sum_{\{u,v\} \subseteq \{a\}} c(u,v) = 0.1 \times (1+2+3) + 0.4 \times 1 + 0.1 \times 2 + 0.4 \times 0 = 1.2$.

3.2. Modularity of Objective Functions

We say that $g(\cdot)$ is submodular if it satisfies a natural "diminishing returns" property: the marginal gain from adding an element to a set X is at least as high as the marginal gain from adding the same element to a superset of X. Formally, for every $X, Y \subseteq V$ with $X \subseteq Y$ and every $e \in V \setminus Y$, it follows that

$$g(X \cup \{e\}) - g(X) \geq g(Y \cup \{e\}) - g(Y)$$

And it is monotone if $g(X) \leq g(Y)$ whenever $X \subseteq Y$.

Note that $g(\cdot)$ is supermodular if and only if $-g(\cdot)$ is submodular.

Theorem 1. *$f(S)$ is neither submodular nor supermodular under IC model.*

Proof. In Fig.2 each edge is bidirectional. As shown in Fig.2(a), the 2-tuple $(0,1)$ on edge (v_1, v_2) means propogation probability $p_{v_1 v_2} = 0$ and $c(v_1, v_2) = 1$. For vertices $\{v_i, v_j\}$ between which there is no edge set $c(v_i, v_j) = 0$, then we have $f(\{v_1\}) = 0$, $f(\{v_1, v_4\}) = 1$, $f(\{v_1, v_2\}) = 1$ and $f(\{v_1, v_2, v_4\}) = 3$. Thus, $f(\{v_1, v_4\}) - f(\{v_1\}) < f(\{v_1, v_2, v_4\}) - f(\{v_1, v_2\})$, which implies $f(\cdot)$ is not submodular. In Fig.2(b), we have $f(\{v_2\}) = f(\{v_1, v_2\}) = f(\{v_2\}) =$

4. Thus, $f(\{v_2\}) - f(\emptyset) > f(\{v_1, v_2\}) - f(\{v_1\})$ which implies $f(\cdot)$ is not supermodular. □

3.3. Hardness Result

Theorem 2. *Game profit maximization problem is NP-hard.*

Proof. We prove by reducing from the set cover problem, which is NP-complete [22]. Given a ground set $U = \{u_1, u_2, \ldots, u_n\}$ and a collection of sets $\{S_1, S_2, \ldots, S_m\}$ whose union equals the ground set, the set cover problem is to decide if there exist k sets in S so that the union equals U. Given an instance of the set cover problem, we construct a corresponding graph with $m+2n$ nodes as follows. For each set S_i we create one node p_i, and for each element u_j we create two nodes q_j and q'_j. If the S_i contains the element u_j, then we create two edges (p_i, q_j) and (p_i, q'_j). Note that each edge is live which means the probability is 1. Now we design the profit function over pairs of nodes. For the pairs $\{q_j, q'_j\}$, we set the profit function $c(q_j, q'_j) = 1$, and for the other pairs $\{u, v\}$, we set the profit function $c(u, v) = 0$. Then the set cover problem is equivalent to deciding if there is a set S of k nodes such that the profit of S equals to n. The theorem follows immediately. □

4. Decomposing Strategy

Since game profit maximization problem is not submodular, the greedy strategy can't be directly applied to our problem to get a guaranteed approximate solution. To solve this non-submodular problem, we propose a new method called decomposing strategy in which we decompose our objective function as a difference of two submodular functions. Based on this decomposition, we designed four algorithms.

Our idea is inspired by the fact that any set function can be expressed as a Difference between two Submodular functions which is called the DS decomposition[23]. However, it is open problem whether this decoposition can be found in polynomial-time for any given set function. Moreover, the DS decomposition in this paper is nontrivial.

In this paper, it is quite interesting to see that for the online game profit, such a decomposition is found explicitly. Through our constant attempts, we find that $f(S)$ can be decomposed as a difference between two submodular functions which define as follows: for seed set S, we define $u(S)$ as the profit among all activated users $I(S)$ plus the profit between the activated users $I(S)$ and the non-activated users $V \setminus I(S)$. Then we add $f(S)$ to $u(S)$ and define the $v(S)$ which is the profit between activated users $I(S)$ and all users V, and which are formulated as follows:

$$u(S) = \mathbb{E}[\sum_{\{u,v\} \subseteq I(S)} c(u,v) + \sum_{u \in I(S), v \in V \setminus I(S)} c(u,v)] \tag{4}$$

$$\begin{aligned} v(S) &= u(S) + f(S) \\ &= \mathbb{E}[\sum_{\{u,v\} \subseteq I(S)} 2 \cdot c(u,v) + \sum_{u \in I(S)} \sum_{v \in V \setminus I(S)} c(u,v)] \\ &= \mathbb{E}[\sum_{u \in I(S)} \sum_{v \in V} c(u,v)] \end{aligned} \tag{5}$$

Then we have

$$f(S) = v(S) - u(S) \tag{6}$$

Theorem 3. *$u(\cdot)$ and $v(\cdot)$ are submodular under the IC model*

Proof. Given a graph $G = (V, E)$ and an influence diffusion model, it is shown in[16]that the IC model is equivalent to the reachability in a random graph g, called live-edge graph or sample graph. In the IC model, g is generated by selecting each edge $(u, v) \in E$ independently with probability p_{uv}. The selected edges are called live and all other edges are called blocked. By claim 2.6 in [1], $u(S)$ can be written as

$$u(S) = \sum_{g \sqsubseteq G} \Pr[g] (\sum_{\{u,v\} \subseteq R_g(S)} c(u,v) + \sum_{u \in R_g(S)} \sum_{v \in V \setminus R_g(S)} c(u,v))$$

where $g \sqsubseteq G$ denotes that the sample graph g is generated from G with a probability denoted by $\Pr[g]$, $R_g(S)$ denotes the set of nodes reachable from S via live edges in g .

For the convenience of proof, denote

$$Q(S) = \sum_{\{u,v\} \subseteq R_g(S)} c(u,v) + \sum_{u \in R_g(S)} \sum_{v \in V \setminus R_g(S)} c(u,v).$$

Then $u(S)$ is rewritten as

$$u(S) = \sum_{g \sqsubseteq G} \Pr[g] \cdot Q(S).$$

Since a non-negative linear combination of submodular functions is also submodular, to prove $u(\cdot)$ is submodular, it is sufficient to prove $Q(S)$ is submodular for any sample graph g.

Let M, N be two sets such that $M \subseteq N \subseteq V$. For any node $x \in V \setminus N$, since

$$R_g(M \cup \{x\}) = R_g(M) \uplus (R_g(x) \setminus R_g(M)),$$

$$V \setminus R_g(M) = (R_g(x) \setminus R_g(M)) \uplus (V \setminus R_g(M \cup \{x\})),$$

$$R_g(N \cup \{x\}) = R_g(N) \uplus (R_g(x) \setminus R_g(N)),$$

$$V \setminus R_g(N) = (R_g(x) \setminus R_g(N)) \uplus (V \setminus R_g(N \cup \{x\})),$$

we have

$$\begin{aligned} & Q(M \cup \{x\}) - Q(M) \\ &= \sum_{\{u,v\} \subseteq R_g(x) \setminus R_g(M)} c(u,v) + \sum_{u \in R_g(x) \setminus R_g(M)} \sum_{v \in V \setminus R_g(M \cup \{x\})} c(u,v) \end{aligned} \tag{7}$$

$$\begin{aligned} & Q(N \cup \{x\}) - Q(N) \\ &= \sum_{\{u,v\} \subseteq R_g(x) \setminus R_g(N)} c(u,v) + \sum_{u \in R_g(x) \setminus R_g(N)} \sum_{v \in V \setminus R_g(N \cup \{x\})} c(u,v) \end{aligned} \tag{8}$$

Note that the disjoint union $\uplus$ means the usual union of subsets which have no element in common. Comparing all terms on the right-hand sides of 7 and 8, since $R_g(x) \setminus R_g(M) \supseteq R_g(x) \setminus R_g(N)$ and $V \setminus R_g(M \cup \{x\}) \supseteq V \setminus R_g(N \cup \{x\})$, we obtain $Q(M \cup \{x\}) - Q(M) \geq Q(N \cup \{x\}) - Q(N)$. Therefore, $Q(S)$ is submodular.

Next, we prove $v(\cdot)$ is submodular. $v(S)$ can be rewritten as

$$v(S) = \sum_{g \sqsubseteq G} \Pr[g] \cdot Q'(S),$$

in which

$$\begin{aligned} & Q'(S) \\ & = \sum_{\{u,v\} \subseteq R_g(S)} 2c(u,v) + \sum_{u \in R_g(S)} \sum_{v \in V \setminus R_g(S)} c(u,v) \\ & = \sum_{u \in R_g(S)} \sum_{v \in V} c(u,v) \end{aligned}$$

To prove $v(\cdot)$ is submodular, it is sufficient to prove $Q'(S)$ is submodular for any sample graph g. Let M, N be two sets such that $M \subseteq N \subseteq V$. For any node $x \in V \setminus N$, we have

$$Q'(M \cup \{x\}) - Q'(M) = \sum_{u \in R_g(x) \setminus R_g(M)} \sum_{v \in V} c(u,v) \tag{9}$$

$$Q'(N \cup \{x\}) - Q'(N) = \sum_{u \in R_g(x) \setminus R_g(N)} \sum_{v \in V} c(u,v) \tag{10}$$

Comparing all terms on the right-hand sides of 9 and 10, by $R_g(x) \setminus R_g(M) \supseteq R_g(x) \setminus R_g(N)$, we obtain $Q'(M \cup \{x\}) - Q'(M) \geq Q'(N \cup \{x\}) - Q'(N)$. Therefore, $Q'(S)$ is submodular. □

Theorem 4. *$u(\cdot)$ and $v(\cdot)$ are monotone nondecreasing.*

Proof. To prove $u(\cdot)$ is monotone nondecreasing, let S, T be any two seed sets with $S \subseteq T$. By $I(S) \subseteq I(T)$, disjoint union $\{(x,y) \mid x \in I(S), y \in I(S)\} \biguplus \{(x,y) \mid x \in I(S), y \in V \setminus I(S)\}$ equals $\{(x,y) \mid x \in I(S), y \in I(S)\} \biguplus \{(x,y) \mid x \in I(S), y \in I(T) \setminus I(S)\} \biguplus \{(x,y) \mid x \in I(S), y \in V \setminus I(T)\}$, it follows that the above disjointed union is a subset of $\{(x,y) \mid x \in I(T), y \in I(T)\} \biguplus \{(x,y) \mid x \in I(T), y \in V \setminus I(T)\}$. By the definition of $u(\cdot)$, we have $u(S) \leq u(T)$. Hence, $u(\cdot)$ is monotone nondecreasing. Moreover, since $f(\cdot)$ is clearly monotone nondecreasing, we obtain that $v = u + f$ is also monotone nondecreasing. $\square$

5. Algorithms

According to the decomposed submodular functions, we designed four heuristic algorithms by calculating the modular upper and lower bounds of the corresponding submodular functions. Our main idea is inspired by the framework which is used for minimization of the difference between submodular functions[24]. But our proposed algorithms are different from theirs. They address the minimization problem without constraint while we solve the maximization problem under k cardinality constraint. They only propose one modular procedure while we propose four modular functions to approximate the original function.

5.1. Preliminary

First, we introduce two modular bounds for submodular function as following.

For any submodular set function $g(\cdot)$ on V, we have the following two tight modular upper bounds that are tight at a given set X ([24]):

$$g(Y) \leq m^g_{X,1}(Y) \triangleq g(X) - \sum_{j \in X \setminus Y} g(j \mid X \setminus j) + \sum_{j \in Y \setminus X} g(j \mid \emptyset),$$

$$g(Y) \leq m^g_{X,2}(Y) \triangleq g(X) - \sum_{j \in X \setminus Y} g(j \mid V \setminus j) + \sum_{j \in V \setminus X} g(j \mid X).$$

A modular lower bound of $g(\cdot)$ is tight at a given set X can be obtained as follows ([24]). Let σ be a permutation of V and define $S^\sigma = \{\sigma(1), \sigma(2), \ldots, \sigma(i)\}$ as σ's chain containing Y, in which $S^\sigma_0 = \emptyset$ and $S^\sigma_{|X|} = X$. Define

$$h^g_{X,\sigma}(\sigma(i)) = g(S^\sigma_i) - g(S^\sigma_{i-1}).$$

Then,

$$h^g_{X,\sigma}(Y) = \sum_{v \in Y} h^g_{X,\sigma}(v)$$

is a tight lower bound of $g(Y)$, i.e., $h^g_{X,\sigma}(Y) \leq g(Y)$, $\forall Y \subseteq V$, and $h^g_{X,\sigma}(X) = g(X)$.

5.2. Procedures

According to the DS decomposition of the objective function and the modular upper bound and lower bound we design four algorithms as follows.

Algorithm 1 M-H Procedure

---

initialize $X^0 \leftarrow$ *random k seeds*; $t \leftarrow 0$;
repeat
  choose a permutation σ^t whose chain contains the set X^t;
  $X^{t+1} \leftarrow \text{argmax}_X m^v_{X^t}(X) - h^u_{X^t,\sigma^t}(X)$, s.t. $|X| = k$;
  $t \leftarrow t + 1$;
until *converged, i.e.,* $X^t = X^{t-1}$ or $t > T$;
return X^t or $\max X$;

For algorithm 1, named M-H procedure, we use the upper bound of $v(\cdot)$ minus the lower bound of $u(\cdot)$ to approximate the original problem. When the algorithm converges, that is, two adjacent iterations gain the same solution, the algorithm returns it. This M-H modular function is actually an upper bound of the original function $f(\cdot)$ and the convergence condition can not be guaranteed in polynomial time. For this reason, we set a threshold T for the number of iterations. If t is greater than T, the algorithm returns the best solution until now.

For algorithm 2, named M-M procedure, we use the upper bound of $v(\cdot)$ minus the upper bound of $u(\cdot)$ and it is modular function too. The convergence condition of the algorithm is the same as algorithm 1.

For algorithm 3, named H-H procedure, we use the lower bound of $v(\cdot)$ minus the lower bound of $u(\cdot)$ and it is modular function too. The convergence condition of the algorithm is the same as algorithm 1.

Algorithm 2 M-M Procedure

initialize $X^0 \leftarrow \textit{random } k \textit{ seeds}$; $t \leftarrow 0$;
repeat
  choose a permutation σ^t whose chain contains the set X^t;
  $X^{t+1} \leftarrow \text{argmax}_X m^v_{X^t}(X) - m^u_{X^t}(X)$, s.t. $|X| = k$;
  $t \leftarrow t + 1$;
until *converged, i.e.,* $X^t = X^{t-1}$ or $t > T$;
return X^t or $\max X$;

Algorithm 3 H-H Procedure

initialize $X^0 \leftarrow \textit{random } k \textit{ seeds}$; $t \leftarrow 0$;
repeat
  choose a permutation σ^t whose chain contains the set X^t;
  $X^{t+1} \leftarrow \text{argmax}_X h^v_{X^t,\sigma^t}(X) - h^u_{X^t,\sigma^t}(X)$, s.t. $|X| = k$;
  $t \leftarrow t + 1$;
until *converged, i.e.,* $X^t = X^{t-1}$ or $t > T$;
return X^t or $\max X$;

Algorithm 4 H-M Procedure

initialize $X^0 \leftarrow \textit{random } k \textit{ seeds}$; $t \leftarrow 0$;
repeat
  choose a permutation σ^t whose chain contains the set X^t;
  $X^{t+1} \leftarrow \text{argmax}_X h^v_{X^t,\sigma^t}(X) - m^u_{X^t}(X)$, s.t. $|X| \leq k$;
  $t \leftarrow t + 1$;
until *converged, i.e.,* $X^t = X^{t-1}$ or $t > T$;
return X^t or $\max X$;

For algorithm 4, named H-M procedure, we use the lower bound of $v(\cdot)$ minus the upper bound of $u(\cdot)$ to approximate the original problem. Note that this H-M module function is actually a lower bound of the original function. This indicates that Algorithm 4 always increases the profit value at every iteration. The convergence condition of the algorithm is the same as algorithm 1.

5.3. Analysis

Now we give the performance analysis of some algorithms and some of the results are interesting.

Algorithm 1 is actually based on modular upper bounds of the original functon f that corresponds to the iterated solution and the theoretical result is surprising which is shown as follows.

Theorem 5. *When the $M - H$ procedure (Algorithm 1) converges, then the algorithm returns the optimal solution of f. The algorithm can not converge in polynomial time if $P \neq NP$.*

Proof. Suppose the algorithm converges at $X^{t+1} = X^t$, then for $\forall X \subseteq V$ such that $|X| \leq k$, we have

$$\begin{aligned} & v(X^{t+1}) - u(X^{t+1}) \\ & = m^v_{X^t}(X^{t+1}) - h^u_{X^t}(X^{t+1}) \\ & \geq m^v_{X^t}(X) - h^u_{X^t}(X) \geq v(X) - u(X) \end{aligned}$$

where the first quation follows since it conveges at X^{t+1}, the second inequation follows since X^{t+1} is the maxima of the modular functions $m^v_{X^t}(X) - h^u_{X^t}(X)$, such that $|X| \leq k$. The third inequation follows since $m^v_{X^t}(X) - h^u_{X^t}(X)$ is the modular upper bound of f. Thus $f(X^{t+1}) \geq f(X)$ for $\forall X \subseteq V$ such that $|X| \leq k$, then X^{t+1} is an optimal solution of f.

As the algorithm converges, which means it gets the optimal solution, the time complexity of this algorithm should not be polynomial, since our problem is NP-hard as shown in Theorem 2. □

Since the algorithm can not converge in polynomial time, we set the threshold of iteration number.

Algorithm 4 is actually based on modular lower bounds of the original functon f that corresponds to the iterated solution and the theory result is shown as follows.

Theorem 6. *The procedure $H-M$(Algorithm 4) monotonically increases the value of the function f at every iteration. Under the general case, when the algorithm converged, if the function value does not decrease on checking $O(n)$ different permutations with different elements at adjacent positions and with both modular upper bounds, then the algorithm gains a local maxima of f.*

Proof. For either modular upper bound, we have

$$\begin{aligned}
&v(X^{t+1}) - u(X^{t+1})\\
&\geq h^v_{X^t}(X^{t+1}) - m^u_{X^t}(X^{t+1})\\
&\geq h^v_{X^t}(X^t) - m^u_{X^t}(X^t) = v(X^t) - u(X^t)
\end{aligned}$$

where the first inequation follows since $h^v_{X^t}(X) - m^u_{X^t}(X)$ is a modular lower bound of f, the second inequation follows since X^{t+1} is the optimal solution of the modular function $h^v_{X^t}(X) - m^u_{X^t}(X)$ such that $|X| \leq k$, the third equation follows since $h^v_{X^t}(X^t) = v(X^t)$ and $m^u_{X^t}(X^t) = u(X^t)$ both of which are exact at point X^t respectively.

The general case means that whether the objective function is monotone or not. To show the algorithm converges to a local maxima $X^t = X^{t+1}$, we should prove $f(X^t) \geq f(X^t \setminus j), \forall j \in X^t$ and $f(X^t) \geq f(X^t \cup j), \forall j \notin X^t$. Given a permutation σ, such that $S^\sigma_k = X^t$, $|X^t| = k$. To gain different $X^t \cup j$, we need to change the element σ_{k+1} in permutation σ. To gain different $X^t \setminus j$, we need to change the element σ_k in permutation σ. Thus, we need check $O(n)$ permutations. As we know $h^v_{X^t}(X) = v(X)$, such that $X = S^\sigma_i$ and $m^u_{X^t,1}(X^t \setminus j) = u(X^t) - u(j|X^t \setminus j) = u(X^t \setminus j)$ and $m^u_{X^t,2}(X^t \cup j) = u(X^t) + u(j|X^t) = u(X^t \cup j)$. When the algorithm converged, since we checked above $O(n)$ permutations with both mudular upper bounds of u, so we have $f(X^t) =$

$v(X^t) - u(X^t) = h^v_{X^t}(X^t) - m^u_{X^t,1}(X^t) \geq h^v_{X^t}(X^t \setminus j) - m^u_{X^t,1}(X^t \setminus j) = v(X^t \setminus j) - u(X^t \setminus j) = f(X^t \setminus j)$ for the modular upper bound $m^u_{X^t,1}(X^t)$, and we have $f(X^t) = v(X^t) - u(X^t) = h^v_{X^t}(X^t) - m^u_{X^t,2}(X^t) \geq h^v_{X^t}(X^t \cup j) - m^u_{X^t,2}(X^t \cup j) = v(X^t \cup j) - u(X^t \cup j) = f(X^t \cup j)$ for the modular upper bound $m^u_{X^t,2}(X^t)$. □

Note that our problem is non-decreasing monotone since the profit function is non-negtative and we have the k-cardinality constraint, thus we always have $f(X^t) \geq f(X^t \setminus j), \forall j \in X^t$. But the above algorithm applies to more general situations regardless of whether the objective function is monotone or not.

5.4. Sampling method

Our problem involves estimating $f(S)$, $u(\cdot)$ and $v(\cdot)$. To address the complexity of evaluating them as shown in following theorem 7, we design a new sampling method based on reverse reachable set (RR-set) [25, 16] which is highly scalable instead of using Monte Carlo simulations.

Theorem 7. *Give a seed S, computing the exact game profit $f(S)$, $u(S)$ and $v(S)$ is $\sharp P$-hard.*

Proof. In fact, computing $f(\cdot)$, $u(\cdot)$ and $v(\cdot)$ involves evaluating $I(S)$ for some subset S of V, and computing the value of $I(S)$ is #P-hard [26]. So the theorem follows immediately. □

To address the computing complexity, we design the sampling method based on the reverse rechable set technology which is first proposed by [25]. First we give the definiton of RR set for a weighted graph as follows.

Definition 2. *(Reverse reachable Set, RR): A random reverse reachable (RR) set $\mathcal{R}_x$ for a graph G is generated by (1) selecting a random node $x \in V$ with a probability distribution proportional, (2) generating a sample live-edge graph g randomly from G according to IC diffusion model, (3) returning $\mathcal{R}_x$ as the set of nodes that can reach x in g.*

We transform computing $f(\cdot)$, $u(\cdot)$ and $v(\cdot)$ to estimate the probability of some events as shown by the following theorem:

Theorem 8. *Given $G = (V, E, c, p)$. For each seed set $S \subseteq V$.*

$$f(S) = T \cdot \Pr_{\substack{g \sqsubseteq G \\ \{x,y\} \sqsubseteq V}} [S \cap R_x \neq \emptyset \wedge S \cap R_y \neq \emptyset]$$

$$u(S) = T \cdot \Pr_{\substack{g \sqsubseteq G \\ \{x,y\} \sqsubseteq V}} [S \cap (R_x \cup R_y) \neq \emptyset]$$

$$v(S) = 2T \cdot \Pr_{\substack{g \sqsubseteq G \\ \{x\} \sqsubseteq V}} [S \cap R_x \neq \emptyset]$$

where $g \sqsubseteq G$ means the sample graph g is generated from G with probability $\Pr[g]$. *Let $T = \sum_{\{x,y\} \subseteq V} c(x, y)$, which is the sum of profit between all users. Define $w(x) = \sum_{y \in V} c(x, y)$ as the weight of node x. Denote $\{x, y\} \sqsubseteq V$ as selecting an unordered pair of nodes x, y from V randomly with probability $\frac{c(x,y)}{T}$ and $\{x\} \sqsubseteq V$ as selecting a node x from V randomly with probability $\frac{w(x)}{2T}$.*

Proof. (1) By the definition of $f(S)$, we have

$$\begin{aligned}
& f(S) \\
=& \mathbb{E}[\sum_{\{x,y\} \subseteq I(S)} c(x, y)] \\
=& \sum_{\{x,y\} \subseteq V} \Pr[x \in I(S) \wedge y \in I(S)] c(x, y) \\
=& \sum_{\{x,y\} \subseteq V} \Pr_{g \sqsubseteq G}[x \in R_g(S) \wedge y \in R_g(S)] c(x, y) \\
=& \sum_{\{x,y\} \subseteq V} \Pr_{g \sqsubseteq G}[\exists s_1, s_2 \in S, s_1 \in R_x \wedge s_2 \in R_y] c(x, y) \\
=& T \cdot \sum_{\{x,y\} \subseteq V} \Pr_{g \sqsubseteq G}[\exists s_1, s_2 \in S, s_1 \in R_x \wedge s_2 \in R_y] \frac{c(x, y)}{T} \\
=& T \cdot \Pr_{\substack{g \sqsubseteq G \\ \{x,y\} \sqsubseteq V}} [\exists s_1, s_2 \in S, s_1 \in R_x \wedge s_2 \in R_y] \\
=& T \cdot \Pr_{\substack{g \sqsubseteq G \\ \{x,y\} \sqsubseteq V}} [S \cap R_x \neq \emptyset \wedge S \cap R_y \neq \emptyset]
\end{aligned}$$

(2) By the definition of $u(S)$,

$$
\begin{aligned}
& u(S) \\
=& \mathbb{E}[\sum_{\{x,y\}\subseteq I(S)} c(x,y) + \sum_{x\in I(S), y\in V\setminus I(S)} c(x,y)] \\
=& \sum_{\{x,y\}\subseteq V} \Pr[x \in I(S) \vee y \in I(S)] c(x,y) \\
=& \sum_{\{x,y\}\subseteq V} \Pr_{g\sqsubseteq G}[x \in R_g(S) \vee y \in R_g(S)] c(x,y) \\
=& \sum_{\{x,y\}\subseteq V} \Pr_{g\sqsubseteq G}[\exists s_1, s_2 \in S, s_1 \in R_x \vee s_2 \in R_y] c(x,y) \\
=& T \cdot \sum_{\{x,y\}\subseteq V} \Pr_{g\sqsubseteq G}[\exists s_1, s_2 \in S, s_1 \in R_x \vee s_2 \in R_y] \frac{c(x,y)}{T} \\
=& T \cdot \Pr_{\substack{g\sqsubseteq G \\ \{x,y\}\subseteq V}} [\exists s_1, s_2 \in S, s_1 \in R_x \vee s_2 \in R_y] \\
=& T \cdot \Pr_{\substack{g\sqsubseteq G \\ \{x,y\}\subseteq V}} [S \cap R_x \neq \emptyset \vee S \cap R_y \neq \emptyset] \\
=& T \cdot \Pr_{\substack{g\sqsubseteq G \\ \{x,y\}\subseteq V}} [S \cap (R_x \cup R_y) \neq \emptyset].
\end{aligned}
$$

(3) By the definition of $v(S)$,

$$
\begin{aligned}
v(S) &= u(S) + f(S) = \mathbb{E}[\sum_{x\in I(S)} \sum_{y\in V} c(x,y)] \\
&= \mathbb{E}[\sum_{x\in I(S)} w(x)]
\end{aligned}
$$

Thus, $v(\cdot)$ is a weighted variation of the influence spread. By the proof of the Lemma 2 in [27], the claim holds. □

To restrict the estimate error of the sampling, we use the following lemma in [28].

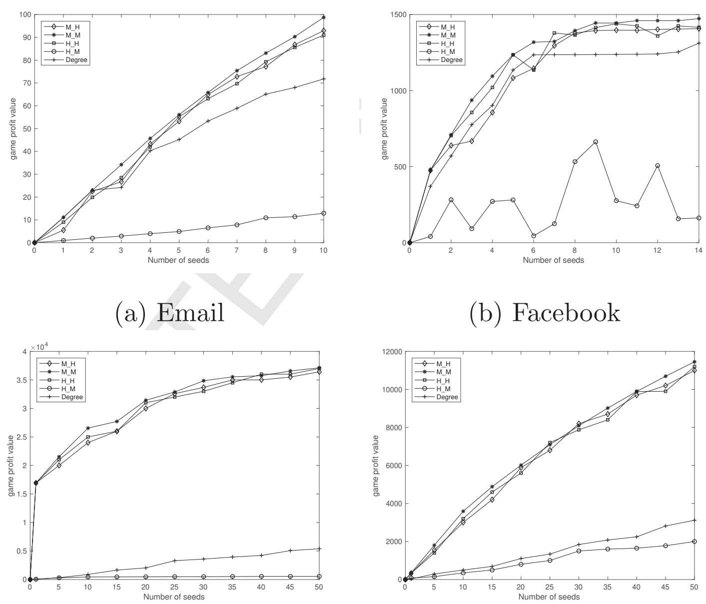
Lemma 1. *((ϵ, δ)-approximation) Let $Z_1, Z_2, \ldots$ be independently and identically distributed samples according to Z in the interval $[0, 1]$ with mean μ_Z. Let $Cov(Z) = \sum_{i=1}^{N} Z_i$ and $\hat{\mu}_Z = \frac{1}{N} Cov(Z)$. Let $\Upsilon = 4(e-2)\ln(2/\delta)/\epsilon^2$ and $\Upsilon_1 = 1 + (1+\epsilon)\Upsilon$. If N is the number of samples at which $Cov(Z) \geq \Upsilon_1$, then $\Pr[|\hat{\mu}_Z - \mu_Z| \leq \epsilon\mu_Z] > 1 - \delta$ and $\mathbb{E}(N) \leq \Upsilon_1/\mu_Z$.*

6. Performance Evaluation

6.1. Settings

We use four social networks in our experiments. All datasets are publicly available. Email, DBLP can be obtained from SNAP website, while Facebook and Douban can be obtained from KONECT website. The propagation probability for IC model is set to $\frac{1}{degree(v)}$ as widely used in other literature[13, 26], and the profit between nodes is proportional to propagation probability on corresponding edges. For comparison, we use the algorithm of HighDegree[12] as a baseline which selects k nodes with highest degrees.

We implement our algorithms and the baseline in Python and the experiments run on a workstation with an Intel Xeon 4.0GHz CPU and 64GB memory.



(a) Email (b) Facebook

(c) Douban (d) DBLP

Figure 3: The relationship between profit and seed set size

6.2. Effectiveness and Analysis

The results of the game profit computed by our proposed algorithms on four data sets are shown in Fig. 3 respectively. As the number of selected seeds increases, the performance of Algorithm 1-3 is always superior to the baseline algorithm. This is because the baseline algorithm only considers the number of active nodes and ignores the network structure. And the gap between them is getting bigger and bigger which means that the quality of the solution of our algorithms is improved much more. Our proposed three algorithms performs quite close to each other. On both small-scale and large-scale networks, they perform quite well and demonstrate good scalability.

The $h_X^v - m_X^u$ procedure (Algorithm 4) should increase the game profit value at every iteration. However, the experiment results shows that, in many cases, the algorithm executes only one iteration and returns the initial input. Its performance is always worse than the baseline algorithm, since its performance is closely related to the selection of initial solution. In the following, we try to explain the experiment results of algorithm 4 under some assumption.

First, by $v(X^t)-u(X^t) = h_{X^t}^v(X^t)-m_{X^t}^u(X^t) \leq h_{X^t}^v(X^{t+1})-m_{X^t}^u(X^{t+1}) \leq v(X^{t+1}) - u(X^{t+1})$, we see that $f(X^t) \leq f(X^{t+1})$. Next we analysis the reason why the algorithm does not loop. By the submodularity of v, $v(X \cup \{y\}) \leq v(X) + v(\{y\}), \forall X \subseteq V, \forall y \in V$. However, based on experiment, we find that for most points in V, the inequality $v(X \cup \{y\}) \leq v(X) + \frac{1}{2}v(\{y\})$ also holds. Then, under the assumption that $v(X \cup \{y\}) \leq v(X) + \frac{1}{2}v(\{y\})$ holds for any $X \subseteq V, y \in V$. Also by the definitions of $v(\cdot), u(\cdot)$, we have $v(X) \leq 2u(X), \forall X \subseteq V$. Thus, we obtain $v(X \cup \{y\}) \leq v(X) + u(\{y\})$, *i.e.*, $v(y|X) \leq u(\{y\})$. For the given input $X_0, \mid X_0 \mid = k$, randomly choose a permutation σ whose chain contains X_0. $\forall x_i \in X, \forall x_j \in V \setminus X$, we obtain $(h_{X_0}^v - m_{X_0}^u)(x_i) - (h_{X_0}^v - m_{X_0}^u)(x_j) = v(S_i^\sigma) - v(S_{i-1}^\sigma) - u(i \mid X \setminus i) - (v(S_j^\sigma) - v(S_{j-1}^\sigma)) + u(\{x_j\})$. By the submodularity of v, $v(S_i^\sigma) - v(S_{i-1}^\sigma) - u(i \mid X \setminus i) \geq v(i \mid X \setminus i) - u(i \mid X \setminus i) = f(i \mid X \setminus i) \geq 0$(the last inequality holds since f is monotone nondecreasing). By our above assumption, $v(S_j^\sigma) - v(S_{j-1}^\sigma) - u(\{x_j\}) \leq \frac{1}{2}u(\{x_j\}) - u(\{x_j\}) < 0$. Then, the inequality $(h_{X_0}^v - m_{X_0}^u)(x_i) > (h_{X_0}^v - m_{X_0}^u)(x_j)$ holds, which implies $X_0 = \underset{X \subseteq V, |X|=k}{\operatorname{argmax}}\ h_{X_0}^v(X) - m_{X_0}^u(X)$. Thus, by the stop condition, the algorithm will return X_0 at the end of the first iteration.

7. Conclusion

In this paper, we propose game profit maximization problem under IC model which is neither submodular nor supermodular. To address the problem we decompose it into the difference of two submodular functions, and propose four heuristic algorithms according to the lower bound and upper bounds of the decomposed submodular functions. Our experimental results verify the effectiveness of our methods.

Acknowledgment

This work is partly supported by the National Natural Science Foundation of China (NSFC No. 61672323), the Fundamental Research Funds of Shandong University (Grant No. 2017JC043), the Key Research and Development Program of Shandong Province (2017GGX10122, 2017GGX10142) and the Fundamental Research Funds for the Central Universities (Grant No. GK201703005).

References

[1] D. Kempe, J. Kleinberg, E. 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, KDD '03, ACM, New York, NY, USA, 2003, pp. 137–146. doi:10.1145/956750.956769.
URL http://doi.acm.org/10.1145/956750.956769

[2] W. Chen, C. Wang, 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, KDD '10, ACM, New York, NY, USA, 2010, pp. 1029–1038. doi:10.1145/1835804.1835934.
URL http://doi.acm.org/10.1145/1835804.1835934

[3] W. Chen, Y. Yuan, L. Zhang, Scalable influence maximization in social networks under the linear threshold model, in: 2010 IEEE International Conference on Data Mining, 2010, pp. 88–97. doi:10.1109/ICDM.2010.118.

[4] W. Xu, Z. Lu, W. Wu, Z. Chen, A novel approach to online social influence maximization, Social Network Analysis and Mining 4 (1) (2014) 153.

[5] W. Chen, Y. Wang, S. Yang, Efficient influence maximization in social networks, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09, ACM, New York, NY, USA, 2009, pp. 199–208. doi:10.1145/1557019.1557047.
URL http://doi.acm.org/10.1145/1557019.1557047

[6] W. Lu, L. V. Lakshmanan, Profit maximization over social networks, in: 2012 IEEE 12th International Conference on Data Mining, IEEE, 2012, pp. 479–488.

[7] A. Wang, W. Wu, L. Cui, On bharathi–kempe–salek conjecture for influence maximization on arborescence, Journal of Combinatorial Optimization 31 (4) (2016) 1678–1684.

[8] W. Chen, T. Lin, Z. Tan, M. Zhao, X. Zhou, Robust influence maximization, in: Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, ACM, New York, NY, USA, 2016, pp. 795–804. doi:10.1145/2939672.2939745.
URL http://doi.acm.org/10.1145/2939672.2939745

[9] W. Lu, W. Chen, L. V. S. Lakshmanan, From competition to complementarity: Comparative influence diffusion and maximization, Proc. VLDB Endow. 9 (2) (2015) 60–71. doi:10.14778/2850578.2850581.
URL http://dx.doi.org/10.14778/2850578.2850581

[10] Y. Tang, Y. Shi, X. Xiao, Influence maximization in near-linear time: A martingale approach, in: Proceedings of the 2015 ACM SIGMOD Inter-

national Conference on Management of Data, SIGMOD '15, ACM, New York, NY, USA, 2015, pp. 1539–1554. doi:10.1145/2723372.2723734.
URL http://doi.acm.org/10.1145/2723372.2723734

[11] H. T. Nguyen, M. T. Thai, 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, SIGMOD '16, ACM, New York, NY, USA, 2016, pp. 695–710. doi:10.1145/2882903.2915207.
URL http://doi.acm.org/10.1145/2882903.2915207

[12] J. Tang, X. Tang, J. Yuan, Towards profit maximization for online social network providers, CoRR abs/1712.08963. arXiv:1712.08963.
URL http://arxiv.org/abs/1712.08963

[13] Y. Tang, X. Xiao, Y. Shi, Influence maximization: Near-optimal time complexity meets practical efficiency, in: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, SIGMOD '14, ACM, New York, NY, USA, 2014, pp. 75–86. doi:10.1145/2588555.2593670.
URL http://doi.acm.org/10.1145/2588555.2593670

[14] S. Bharathi, D. Kempe, M. Salek, Competitive influence maximization in social networks, in: International Workshop on Web and Internet Economics, Springer, 2007, pp. 306–311.

[15] Z. Lu, Z. Zhang, W. Wu, Solution of bharathi–kempe–salek conjecture for influence maximization on arborescence, Journal of Combinatorial Optimization 33 (2) (2017) 803–808.

[16] Z. Wang, Y. Yang, J. Pei, L. Chu, E. Chen, Activity maximization by effective information diffusion in social networks, IEEE Transactions on Knowledge and Data Engineering 29 (11) (2017) 2374–2387. doi:10.1109/TKDE.2017.2740284.

[17] G. L. Nemhauser, L. A. Wolsey, Best algorithms for approximating the maximum of a submodular set function, Mathematics of Operations Research 3 (3) (1978) 177–188. arXiv:https://doi.org/10.1287/moor.3.3.177, doi:10.1287/moor.3.3.177.
URL https://doi.org/10.1287/moor.3.3.177

[18] U. Feige, V. S. Mirrokni, J. Vondrák, Maximizing non-monotone submodular functions, SIAM Journal on Computing 40 (4) (2011) 1133–1153. arXiv:https://doi.org/10.1137/090779346, doi:10.1137/090779346.
URL https://doi.org/10.1137/090779346

[19] J. Lee, V. S. Mirrokni, V. Nagarajan, M. Sviridenko, Non-monotone submodular maximization under matroid and knapsack constraints, in: Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing, STOC '09, ACM, New York, NY, USA, 2009, pp. 323–332. doi:10.1145/1536414.1536459.
URL http://doi.acm.org/10.1145/1536414.1536459

[20] M. Feldman, J. Naor, R. Schwartz, A unified continuous greedy algorithm for submodular maximization, in: 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, 2011, pp. 570–579. doi:10.1109/FOCS.2011.46.

[21] U. Feige, R. Izsak, Welfare maximization and the supermodular degree, in: Proceedings of the 4th Conference on Innovations in Theoretical Computer Science, ITCS '13, ACM, New York, NY, USA, 2013, pp. 247–256. doi:10.1145/2422436.2422466.
URL http://doi.acm.org/10.1145/2422436.2422466

[22] N. Alon, B. Awerbuch, Y. Azar, The online set cover problem, in: Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing, STOC '03, ACM, New York, NY, USA, 2003, pp. 100–105. doi:10.1145/780542.780558.
URL http://doi.acm.org/10.1145/780542.780558

[23] M. Narasimhan, J. Bilmes, A submodular-supermodular procedure with applications to discriminative structure learning, in: Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence, UAI'05, AUAI Press, Arlington, Virginia, United States, 2005, pp. 404–412.
URL http://dl.acm.org/citation.cfm?id=3020336.3020387

[24] R. Iyer, J. Bilmes, Algorithms for approximate minimization of the difference between submodular functions, with applications, in: Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence, UAI'12, AUAI Press, Arlington, Virginia, United States, 2012, pp. 407–417.
URL http://dl.acm.org/citation.cfm?id=3020652.3020697

[25] C. Borgs, M. Brautbar, J. Chayes, B. Lucier, Maximizing social influence in nearly optimal time, in: Proceedings of the Twenty-fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '14, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2014, pp. 946–957.
URL http://dl.acm.org/citation.cfm?id=2634074.2634144

[26] W. Chen, C. Wang, 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, KDD '10, ACM, New York, NY, USA, 2010, pp. 1029–1038. doi:10.1145/1835804.1835934.
URL http://doi.acm.org/10.1145/1835804.1835934

[27] H. T. Nguyen, T. N. Dinh, M. T. Thai, Cost-aware targeted viral marketing in billion-scale networks, in: IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications, 2016, pp. 1–9. doi:10.1109/INFOCOM.2016.7524377.

[28] P. Dagum, R. Karp, M. Luby, S. Ross, An optimal algorithm for monte carlo estimation, SIAM Journal on Computing 29 (5) (2000) 1484–1496. arXiv:https://doi.org/10.1137/S0097539797315306, doi:10.

1137/S0097539797315306.

URL https://doi.org/10.1137/S0097539797315306