Top-K Influential Nodes in Social Networks: A Game Perspective

Yu Zhang
Dept. of Computer Science
Peking University
Beijing, China
yutazh@pku.edu.cn

Yan Zhang
Dept. of Machine Intelligence
Peking University
Beijing, China
zhy@cis.pku.edu.cn

ABSTRACT

Influence maximization, the fundamental of viral marketing, aims to find top-K seed nodes maximizing influence spread under certain spreading models. In this paper, we study influence maximization from the perspective of game theory. We propose a Coordination Game (CG) model, in which every individuals make their decisions based on the benefit of coordination with their network neighbors, to study information propagation. Our model serves as the generalization of some existing models, such as Majority Vote model and Linear Threshold model. Under the generalized model, we study the hardness of influence maximization and the approximation guarantee of the greedy algorithm. We also combine several strategies to accelerate the algorithm. Experimental results show that after the acceleration, the accuracy of our algorithm is on par with the original greedy algorithm and better than other heuristics, while it is three orders of magnitude faster than the original greedy method.

CCS Concepts

•Information systems → Data mining; •Theory of computation → Design and analysis of algorithms;

Keywords

influence maximization; coordination game model; social networks; viral marketing

1. INTRODUCTION

Social networks play an important role in information diffusion. They give us motivation to use a small subset of influential individuals in a social network to activate a large number of people. Kempe et al. [9] build a theoretical framework of influence maximization, aiming to find top-K influential nodes under certain spreading models. They discuss two popular models - Independent Cascade (IC) model and Linear Threshold (LT) model and propose a greedy algorithm with $(1 - 1/e - \epsilon)$-approximation rate.

SIGIR '17 August 7–11, 2017, Tokyo, Japan

DOI: 10.475/123_4

In most spreading models, each node has two states: active and inactive. Equivalently saying, it has two choices. We can model information diffusion as the process of individual decision-making. As individuals make their decisions based on choices of their neighbors, a particular pattern of behavior can begin to spread across the links of the network. Easley and Kleinberg [8] divide the cause of information propagation into two categories: information effects and direct-benefit effects. Obviously, IC model and LT model belong to the former one, while we focus on the latter one. In our Coordination Game (CG) model, every individuals make their decisions based on the benefit of coordination with their network neighbors.

Influence maximization under CG model is useful in viral marketing. Let us recall the example in [9]. A company would like to market a new product, hoping it will be adopted by a large fraction of the network. The company can initially target a few influential nodes by giving them free samples of the product. Then other nodes will probably switch to using the new product and give up the old product because of the following two reasons: (1) They have higher evaluation of the new product than the old one. (2) They have to coordinate with their neighbors because using different products brings a lot of inconvenience and reduces their benefits. (e.g., people using different operating systems may have some problems with compatibility when working together, and users from different kinds of social media platforms cannot communicate with each other timely.) Our model describes these two reasons precisely.

In this paper, we study how to find Top-K influential nodes under CG model. Firstly, we propose our model which serves as the generalization of some well-known spreading models, such as Majority Vote model [3] and Linear Threshold model [9]. We then prove some generalized theoretical results under CG model, including the NP-hardness of the optimization problem itself and the #P-hardness of computing the objective function. Then we try to find a good approximation algorithm for the problem. We embed our CG model into the scenario of general diffusion process defined by Mossel and Roch [12]. We prove that the objective function is monotone and submodular if and only if the *cumulative distribution function* (CDF) of people's threshold is concave, in which case the greedy algorithm can return a $(1 - 1/e - \epsilon)$ approximate solution.

As a traditional method, Kempe et al. [9] use 10,000 times of Monte Carlo simulations to approximate the objective function every time, but it takes us too much time on large-scale networks. To accelerate our algorithm, we use two efficient heuristics - LazyForward [10] and StaticGreedy [7] to reduce the times of simulations. Experimental results

u \ v	A	B
A	p_{uA}, p_{vA}	0,0
B	0,0	p_{uB}, p_{vB}

Figure 1: Payoff matrix of the coordination game.

show that our `Greedy` and `Greedy++` algorithms can activate more nodes than other heuristics. Moreover, `Greedy++` runs faster than `Greedy` by three orders of magnitude.

Related Work. Kempe et al. [9] first build an algorithmic framework of influence maximization by transforming it into a discrete optimization problem. The approximation rate of their algorithm is guaranteed by the monotonicity and submodularity of the global objective function. Mossel and Roch [12] prove that the submodularity of the "local" influence function indicates the submodularity of the "global" objective function.

Morris [11] is the first to propose a coordination game model in contagion. Easley and Kleinberg use a whole chapter to give more extended discussions about the model in their textbook [8].

A lot of efforts have been made on efficient computing methods of the objective function. Some methods aim to reduce the number of trials that need Monte Carlo simulations, such as `CELF` [10]. Other researchers focus on how to calculate the influence spread efficiently. For instance, Chen et al. [4, 5] use arborescences or DAGs to represent the original graph. Cheng et al. propose a `StaticGreedy` strategy [7] and a self-consistent ranking method [6].

2. MODEL

In a social network $G = (V, E)$, we study a situation in which each node has a choice between two behaviors, labeled A and B. If nodes u and v are linked by an edge, then there is an incentive for them to have their behaviors match. We use a game model to describe this situation. There is a coordination game on each edge $(u, v) \in E$, in which players u and v both have two strategies A and B. The payoffs are defined as follows:

- if u and v both adopt strategy A, they will get payoffs $p_{uA} > 0$ and $p_{vA} > 0$ respectively;
- if they both adopt strategy B, they will get payoffs $p_{uB} > 0$ and $p_{vB} > 0$ respectively;
- if they adopt different strategies, they each get a payoff of 0.

The payoff matrix is shown in Figure 1.

We define the total payoff of player u as the sum of the payoffs it gets from all coordination games with its neighbors $N(u) = \{v|(u, v) \in E\}$. If u can get a higher total payoff when it adopt A than that when it adopt B, it will choose strategy A. Otherwise it will choose strategy B.

According to the actual situation, we have the following assumptions for the payoffs: (1) All the p_{uA} and p_{uB} $(u \in V)$ may not be equal to each other because each person in the social network values behaviors A and B differently. (2) p_{uA} and p_{uB} $(u \in V)$ can either be constants or independent and identically distributed random variables because the cascading behaviors in networks are always considered to have determinate principles with some stochastic factors.

Suppose u knows all the choices of its neighbors: there are x_B nodes adopting B and $x_A = deg(u) - x_B$ nodes adopting A. Obviously, u will adopt B if and only if

$$p_{uB} x_B \geq p_{uA} x_A = p_{uA}(deg(u) - x_B), \tag{1}$$

or

$$x_B \geq \frac{p_{uA}}{p_{uA} + p_{uB}} deg(u) = \delta_u deg(u), \quad \delta_u \in [0, 1]. \tag{2}$$

Influence Maximization Problem. Suppose now the market is dominated by A (i.e., all of the nodes in the network choose A). Given a constant k, we want to find a seed set $S_0 \subseteq V$, $|S_0| \leq k$. Initially we let each node in S_0 adopt B (and they will never change their choices again). Time then runs forward in unit steps. In each step, each node decides whether to switch from strategy A to strategy B according to the payoff-maximization principle. We can regard the evolution of nodes' choices as a spreading process of B in the network. The spread of behavior B will finally stop in at most $n = |V|$ steps.

We define $S_i = |\{u \in V | u \text{ adopts } B \text{ in step } i\}|$, $i = 1, 2, ..., n$. Our objective function is (the expectation of) the nodes affected by B at last, or

$$\sigma(S_0) = \mathbb{E}_{\{p_{uA}, p_{uB} | u \in V\}}[|S_n|] = \mathbb{E}_{\{\delta_u | u \in V\}}[|S_n|]. \tag{3}$$

Our purpose is to maximize $\sigma(S_0)$ subject to $|S_0| \leq k$.

Our model can be regarded as the generalization of the following two well-known spreading models.

Majority Vote Model. Suppose all the p_{uA} $(u \in V)$ are constants and are equal to each other. So are all the p_{uB} $(u \in V)$. Equivalently, let

$$p_A = p_{uA}, \quad p_B = p_{uB}, \quad \delta = \delta_u = \frac{p_A}{p_A + p_B}, \quad \forall u \in V. \tag{4}$$

δ is a constant threshold same to every nodes. When $p_A = p_B$, or $\delta = \frac{1}{2}$, the spreading model is called Majority Vote model, which is extensively studied in [3].

Linear Threshold Model. If we set $p_{uA} = 1$ and let p_{uB} follow a continuous power-law distribution, i.e., the probabilistic density function (PDF) of p_{uB} is

$$f_B(x) = \frac{\alpha}{(x+1)^\gamma}, \ x \geq 0, \ \gamma > 1, \ \alpha = \frac{1}{\int_0^\infty \frac{1}{(x+1)^\gamma} \mathrm{d}x} = \gamma - 1, \tag{5}$$

then $\forall 0 \leq x \leq 1$,

$$\begin{aligned} \Pr[\delta_u \leq x] &= \Pr[\frac{1}{1 + p_{uB}} \leq x] = \Pr[p_{uB} \geq 1/x - 1] \\ &= \int_{1/x-1}^{+\infty} f_B(t) \mathrm{d}t = -(t+1)^{-\gamma+1} \Big|_{1/x-1}^{+\infty} = x^{\gamma-1}. \end{aligned} \tag{6}$$

If $\gamma = 2$, we will have $\delta_u \sim U[0, 1]$. This is the famous Linear Threshold model where the weight on each edge adjacent to node u is $1/deg(u)$ (i.e., $b_{vu} = \frac{1}{deg(u)}, \forall u, v \in V$).

Hardness. Under CG model, we have the following hardness result.[1]

THEOREM 1. *(1) Influence maximization under CG model is NP-hard. (2) Computing the objective function under CG model is #P-hard.*

The hardness result serves as a generalization of the NP-hardness of Influence Maximization under Majority Vote

[1]We omit the proof here. For more details about the proof (or the following algorithms and experiments), please refer to the full version: `https://github.com/yuzhimanhua/Influence-Maximization/`.

model [3] and LT model [9], and the #P-hardness of computing the objective function under LT model [5] as well.

3. ALGORITHMS

Submodularity. To find a greedy algorithm with approximation guarantee, the submodularity of the objective function is necessary.

We first recall the general diffusion process defined by Mossel and Roch in [12].

Suppose each node v in the social network $G = (V, E)$ has a threshold $\theta_v \sim U[0, 1]$ *i.i.d* and a "local" spreading function $f_v : 2^V \to [0, 1]$. Initially there is a seed set $S_0 \subseteq V$. After step $t \geq 1$,

$$S_t = S_{t-1} \cup \{v | v \in V - S_{t-1} \ \wedge \ f_v(S_{t-1}) \geq \theta_v\}. \tag{7}$$

The spreading process will stop in at most $n = |V|$ steps. So the objective function is $\sigma(S_0) = \mathbb{E}_{\{\theta_u | u \in V\}}[|S_n|]$.

We can embed our model into the scenario of the general diffusion process.

Let F_δ be the cumulative distribution function of δ_u. Since $\delta_u \in [0, 1]$, we have $F_\delta(0) = 0$ and $F_\delta(1) = 1$. $\forall v$ and S, let

$$\theta_v = F_\delta(\delta_v) \ \text{ and } \ f_v(S) = F_\delta(\frac{|S \cap N(v)|}{deg(v)}). \tag{8}$$

Suppose F_δ is continuous and strictly monotone increasing in $[0, 1]$, then F_δ^{-1} exists, and $\forall x \in [0, 1]$,

$$\Pr[F_\delta(\delta_v) \leq x] = \Pr[\delta_v \leq F_\delta^{-1}(x)] = F_\delta(F_\delta^{-1}(x)) = x. \tag{9}$$

So $F_\delta(\delta_v) \sim U[0, 1]$. Therefore

$$\begin{aligned} &f_v(S) \geq \theta_v \iff F_\delta(\frac{|S \cap N(v)|}{deg(v)}) \geq \theta_v \iff \\ &|S \cap N(v)| \geq F_\delta^{-1}(\theta_v) deg(v) \iff |S \cap N(v)| \geq \delta_v deg(v). \end{aligned} \tag{10}$$

LEMMA 1. *Suppose F_δ is continuous and strictly monotone increasing in $[0, 1]$, f_v is monotone and submodular for any node v (in any graph) **iff** F_δ is concave in $[0, 1]$.*

It is not difficult for us to understand Lemma 1 intuitively because submodularity can be considered as a kind of concavity. F_δ being concave in $[0, 1]$ means that the distribution of people's threshold has a negative skewness, or they tend to have a higher evaluation on new products than old ones. This assumption is reasonable in some cases (e.g., the mobile phone market). F_δ being continuous and strictly monotone increasing in $[0, 1]$ is a technical assumption instead of an essential one. We define these two assumptions as the *concave threshold property*.

For the general diffusion process, Mossel and Roch [12] have proved that $\sigma(S_0)$ is monotone and submodular if and only if f_v is monotone and submodular for any $v \in V$. Using this result and Lemma 1, we can get Theorem 2 immediately.

THEOREM 2. *$\sigma(S_0)$ is monotone and submodular **iff** F_δ satisfies the concave threshold property.*

Theorem 2 provides a strong tool to judge the objective function's submodularity under certain spreading models. For example, under Majority Vote model, $\sigma(S_0)$ is not submodular because $F_\delta(x) = \mathbb{I}(x \geq \delta)$ is not concave in $[0, 1]$, where $\mathbb{I}()$ is the indicator function. The counterexample is also easy to find according to the "non-concave point" $x = \delta$. In contrast, under Linear Threshold model, $\sigma(S_0)$ is submodular because $F_\delta(x) = x$ is concave in $[0, 1]$.

Up till now, we have proved the monotonicity and submodularity of the objective function under CG model with some necessary assumptions. Using the result in [9], the greedy algorithm given in Algorithm 1 (`Greedy`) returns a $(1 - 1/e - \epsilon)$-approximate solution. The algorithm simply selects seed nodes one by one, and each time it always selects the node that provides the largest marginal gain of the objective function.

Speeding-Up Algorithm. Due to the hardness of computing $\sigma(S_0)$, we use two efficient heuristics - LazyForward [10] and StaticGreedy [7] to accelerate our algorithm.

We maintain a priority queue. When finding the next node, we go through the nodes in decreasing order of their marginal gain. If the marginal gain of the top node has not been updated, we recompute it, and insert it into the priority queue again. The correctness of this lazy procedure can be guaranteed due to the submodularity of the objective function.

Instead of conducting a huge number of Monte Carlo simulations each time, we generate a rather small number of snapshots G_i $(i = 1, 2, ..., R')$ at the very beginning. In all the iterations, we run the simulation on these snapshots and use the average number of influenced nodes $\frac{1}{R'} \sum_{i=1}^{R'} \sigma_{G_i}(S_0)$ to estimate the objective function $\sigma(S_0)$.

We name the accelerated algorithm as `Greedy++`.

Algorithm 1 `Greedy`(k, σ)

```
initialize S_0 = ∅
for i = 1 to k do
    select u = argmax_{v∈V−S_0}(σ(S_0 ∪ {v}) − σ(S_0))
    S_0 = S_0 ∪ {u}
end for
output S_0
```

4. EXPERIMENTS

To test the effectiveness and efficiency of our `Greedy` and `Greedy++` algorithms, we conduct experiments on three real-world networks and compare our algorithms with other existing heuristics.

Datasets. The three real-world datasets include two collaboration networks `NetHEPT` [1] and `NetPHY` [1], and one online social network `Epinions` [2]. We summarize the statistical information of the these datasets in Table 1.

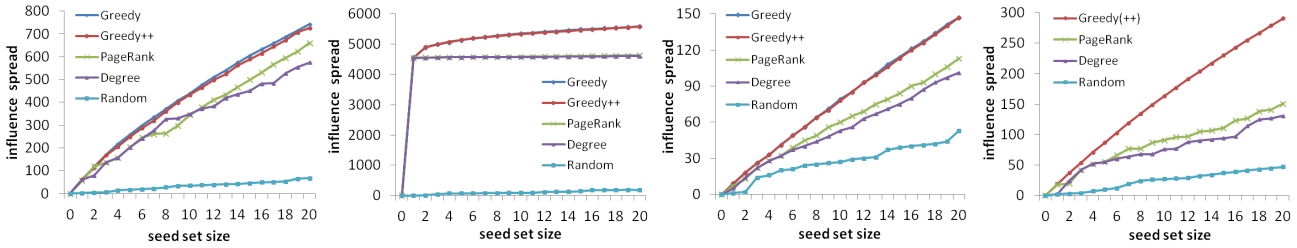
Datasets	$\|V\|$	$\|E\|$	Type
`NetHEPT`	15,233	58,991	Undirected
`NetPHY`	37,154	231,584	Undirected
`Epinions`	75,879	508,837	Directed

Table 1: Statistical information of three datasets.

Algorithms. A total of five algorithms are tested. Besides of `Greedy` and `Greedy++` proposed in this paper, we use other three heuristic algorithms as benchmark methods.

(1) `PageRank` chooses nodes with the largest PageRank value. Since influential nodes are considered to have a large number of out-links, while nodes with high PageRank value are considered to have lots of in-links, we first change the direction of all edges in the graph and then run PageRank algorithm. We use $\alpha = 0.9$ as the random jump parameter. (2) `Degree` chooses nodes with the largest out-degree. (3) `Random` chooses nodes at random.

There are a lot of other efficient algorithms to solve the influence maximization problem under IC model or LT model, such as `PMIA` [4], `LDAG` [5], `IMRank` [6], and `IMM` [13]. But they cannot be applied in CG model directly, and we will not put them into the comparison.



(a) Linear Threshold (b) Concave Threshold (c) Convex Threshold (d) Majority Vote

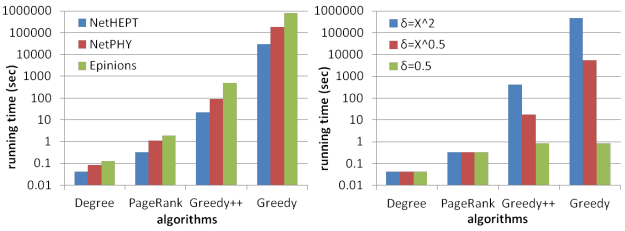
Figure 2: Influence spread of various algorithms in NetHEPT, with different distribution of δ_u. ($X \sim U[0,1]$.) (a) $\delta_u = X$ (submodular). (b) $\delta_u = X^2$ (submodular). (c) $\delta_u = \sqrt{X}$ (nonsubmodular). (d) $\delta_u = 0.5$ (nonsubmodular).

Effectiveness. We first compare the effectiveness of Greedy and Greedy++ with other algorithms by showing influence spread (i.e., $|S_n|$) of the obtained seed set.

In our CG model, distribution of δ_u can be various. We run influence maximization algorithms under four different spreading models in NetHEPT. In the four models, we set δ_u to be X, X^2, $\sqrt{X}$ and 0.5 respectively, where $X \sim U[0,1]$. Therefore the distribution function $F_\delta(x)$ is x, $\sqrt{x}$, x^2 and $\mathbb{I}(x \geq 0.5)$ corresponding to the four cases.

Figure 2 shows our experimental results. In Figure 2, Greedy++ consistently matches the performance of Greedy and significantly outperforms other heuristic algorithms in all cases. According to Theorem 2, the first two cases are submodular cases, while the other two are not. Therefore the approximation rates of Greedy and Greedy++ are not guaranteed in the third and the fourth cases. But our experimental results indicate that they still perform well in these cases. Besides, all the curves of Greedy and Greedy++ are concave no matter in submodular or nonsubmodular cases. In two larger graphs NetPHY and Epinions, we get similar experimental results.

Efficiency. We now test the running time of these algorithms. Figure 3 shows our experimental results.



(a) (b)

Figure 3: (a) Running time of various algorithms on three datasets. ($F_\delta(x) = x$.) (b) Running time of various algorithms in NetHEPT, with different distribution of δ_u. ($X \sim U[0,1]$.)

As we expected, Greedy++ runs consistently faster than Greedy, with more than three orders of magnitude speedup. For example, in the linear threshold case, it takes Greedy more than 9 days to get the top-20 influential nodes in Epinions while Greedy++ only requires 8 minutes.

In other stochastic cases, our conclusion is still true. Note that Greedy++ will spend more time if δ_u is small, or the influence spread tends to be wide. In Majority Vote model, the efficiency of the greedy algorithm dramatically rises because the estimation of influence spread becomes easy.

5. CONCLUSIONS

In this paper, we have discussed how to find top-K influential nodes in social networks under a game theoretic model. We show the hardness of the optimization problem itself, as well as the hardness of calculating the objective function. We prove the approximation guarantee of the greedy algorithm under necessary assumptions. We also accelerate our algorithm with the combination of LazyForward and StaticGreedy. Our experimental results demonstrate that Greedy++ matches Greedy in the spreading effect while significantly reducing running time, and it outperforms other heuristic algorithms such as MaxDegree and PageRank.

Acknowledgements. This work is supported by 973 Program under Grant No.2014CB340405, NSFC under Grant No.61532001 and No.61370054, and MOE-RCOE under Grant No.2016ZD201.

6. REFERENCES

[1] `http://research.microsoft.com/en-us/people/weic/graphdata.zip`.
[2] `http://snap.stanford.edu/data`.
[3] N. Chen. On the approximability of influence in social networks. In *SODA'09*, pages 1029–1037. SIAM, 2009.
[4] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD'10*, pages 1029–1038. ACM, 2010.
[5] W. Chen, Y. Yuan, and L. Zhang. Scalable influence maximization in social networks under the linear threshold model. In *ICDM'10*, pages 88–97. IEEE, 2010.
[6] S. Cheng, H. Shen, J. Huang, W. Chen, and X. Cheng. Imrank: Influence maximization via finding self-consistent ranking. In *SIGIR'14*, pages 475–484. ACM, 2014.
[7] S. Cheng, H. Shen, J. Huang, G. Zhang, and X. Cheng. Staticgreedy: solving the scalability-accuracy dilemma in influence maximization. In *CIKM'13*, pages 509–518. ACM, 2013.
[8] D. Easley and J. Kleinberg. *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press, 2010.
[9] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD'03*, pages 137–146. ACM, 2003.
[10] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *KDD'07*, pages 420–429. ACM, 2007.
[11] S. Morris. Contagion. *The Review of Economic Studies*, 67:57–78, 2000.
[12] E. Mossel and S. Roch. Submodularity of influence in social networks: From local to global. *SIAM Journal on Computing*, 39(6):2176–2188, 2010.
[13] Y. Tang, Y. Shi, and X. Xiao. Influence maximization in near-linear time: a martingale approach. In *SIGMOD'15*, pages 1539–1554. ACM, 2015.