Maximizing Influence on Social Networks with Conjugate Learning Automata

Chong Di, Fangqi Li, Kaiyue Qi, Shenghong Li
School of Electronic Information and
Electrical Engineering,
Shanghai Jiao Tong University
Shanghai, China
dichong95@sjtu.edu.cn, solour_lfq@sjtu.edu.cn

***Abstract*—The problem of maximizing the spread of influence by selecting a subset of participants in a social network as source, known as influence maximization, is a fruitful topic with straightforward application value. Greedy algorithms that select the optimal node one by one lays the foundation of follow-up researches, and plentiful studies have been taken to improve the efficiency of greedy-based algorithms. However, the greedy methods can easily fall into adversary pitfalls, and corresponding improvements have been few. In this paper, a conjugate learning automata based method, utilizing the ability of cooperation in learning automata games, is proposed to obtain better-than-greedy propagation range. Comprehensive simulations in both synthetic and real-world datasets verify that the proposed method can attain better propagation range in some scenarios and is equally competitive respecting time consumption.**

***Index Terms*—influence maximization, learning automata, social network**

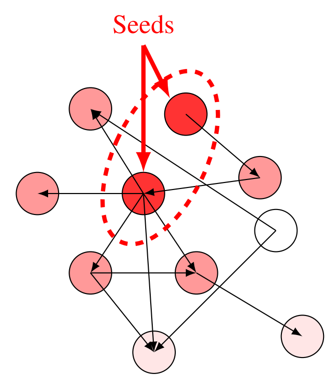


Fig. 1. The propagation of information in a social network

I. INTRODUCTION

Online social network (OSN), the republic of cyber citizens, has been attracting attention of people in the way the physical world has been doing to our ancestors. As a graph structure whose vertices (nodes) and edges denote participants and the connections within respectively, an OSN reflects the collective behavior of a human community in sharing knowledge, coming up with a decision, etc.

Among the infinite variety of aspects to be explored in OSN, Information Maximization (IM) has drawn rising attention [1]. IM aims at selecting a subset of participants in an OSN as source or "seeds" which then propagate their information to other participants through connections represented by edges. Figure 1 illustrates an example of information propagation, the black arrows represent the directed connections, and the intensity of color denotes the propagation order. The "seeds" are besieged by the dashed ellipse. For a given graphical structure $\mathcal{G} = (V, E)$ of an OSN and the maximal number of source vertices, K, an IM algorithm should find a subset $S \subset V, |S| = K$ to maximize the propagation range, namely the number of vertices that receive information [1] [2].

IM is the prototype of many practical tasks as viral marketing [3], pollution detection [4], rumor supervision [5], etc. These seemingly different missions share the same spirit: identifying an optimal subset of participants in order to maximize the number of affected nodes. Therefore IM has aroused interest from diversified disciplines: social science, commercial advertisement, cyber security and algorithmic studies.

Approaches to solve the problem of IM are split into three genres: the greedy approaches, the heuristic approaches and a mixture of them. A *greedy approach* identifies the set of "seeds" by including locally optimal vertex one at a time. By locality, we mean that it takes the selected vertices as given and looks for a new vertex that maximizes the marginal increment of propagation range by simulation. It then includes the new node into the set of "seeds". A *heuristic approach* takes advantage of intrinsic features within the graphical representation, e.g., the in-degree, out-degree, centrality, clique structure, etc. It then uses some specific rules to filter incapable vertices and selects an optimal subset of "seeds". Though heuristic approaches prevail in efficiency by evading tremendous Monte-Carlo procedures, they lack provable accuracy and objectiveness.

As a combinatorial optimization task, IM is essentially an NP-hard problem [2], not to mention that the optimization has to be conducted in a stochastic environment. The ratio between the outcome of greedy approaches and the optimal one is lower bounded by $(1 - \frac{1}{e}) \approx 0.63$ [2], leaving plentiful space for adversary constructions that could drastically hinder the efficacy of greedy approaches. Consider the setting in Figure 2, where the propagation ranges of four optional vertices (v_1, v_2, v_3, v_4) are $(s_1 + s_2, s_1 + s_3, s_2 + s_4, s_5)$ with $s_1 = s_2 > 2 \cdot s_3 > s_5 > s_3 = s_4$. If $K = 2$, a greedy method will select v_1 and v_4 with coverage $s_1 + s_2 + s_5$, but the optimal choice is apparently v_2 and v_3 with coverage $s_1 + s_2 + s_3 + s_4$. In adversary cases like Figure 2, conjugate optimization can outperform greedy ones by selecting multiple vertices simultaneously instead of selecting the one that maximizes the marginal interest greedily. Meanwhile, the design philosophy of selecting the locally optimal vertex implies that the propagation is crucially promoted by a series of centroids that are particularly powerful in propagation. This assumption, however, might be fainting since decentralization of social networks has become popular [6] [7]. Therefore seeking an efficiency conjugate Monte-Carlo method is of practical value and is an interesting end to be investigated in itself. By conjugate Monte-Carlo, we mean to resort to simulation rather than heuristic features and attempt to acquire an optimal collection of vertices simultaneously. However, this line of reasoning has been suppressed by the formidable cost in time consumption.

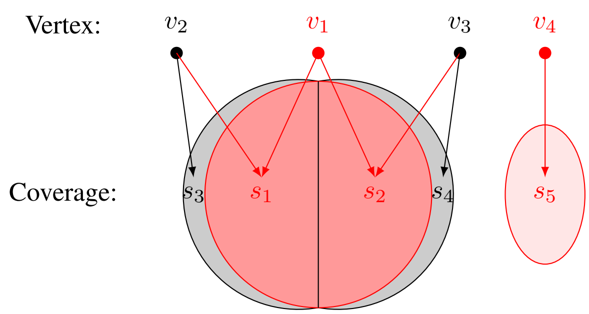


Fig. 2. A coined toy pitfall that deceives a greedy IM method. $|Seeds| = 2$.

It has been reported that learning automaton (LA) can cooperate to achieve combinatorial optimization tasks with acceptable time complexity [8]. As an adaptive unit, LA can learn the most appropriate action/decision in a stochastic environment. When a collection of LAs are trained jointly, they can possibly avoid the pitfall designed to trap greedy methods and obtain a better performance while preserving efficiency. Thus it is intuitively feasible to maximize influence on social networks with conjugate trained LAs.

The contribution of this paper is three-folded:

1) We propose to train a collection of LA conjugately to solve the problem of influence maximization, which is capable of obtaining better-than-greedy results in propagation range.
2) The proposed conjugate LA (CLA) is shown to be a proper generalization of greedy approaches. Since it can degenerate to greedy methods gracefully by altering hyper-parameters.
3) Extensive experiments are conducted on synthetic datasets and real-world datasets. It is observed that pitfalls designed to confuse greedy approaches fail to confuse CLA, and CLA is competitive in efficiency against other state-of-art methods.

This paper proceeds as follows: Section II formalizes the problem of IM and reviews related literature. Section III presents the proposed method, conjugate learning automata, in detail. Section IV is devoted to simulations on comprehensive datasets, comparisons between the proposed model and state-of-the-art approaches and consequent discussions. Section V concludes the paper.

II. Related Works

A. Influence Maximization

The last decades have witnessed the development of IM from a tentative marketing proposal to a fruitful academic topic. The OSN is represented by a weighted directed graph $\mathcal{G} = (V, E)$, where the weight of an edge $v_{i,j} \in V$ denotes the possibility that the i-th participant sends its information to the j-th. The propagation function $\sigma : \mathcal{P}(V) \rightarrow \mathbb{R}^+$ can be simulated from $\mathcal{G}$. Given a collection of "seeds" $S \subset V$, $\sigma(S)$ is the average range of propagation, i.e., the cardinality of influenced vertices. An IM algorithm aims to find an optimal subset S_{IM} of V such:

$$S_{\text{IM}} = \arg \max_{|S|=K, S \subset V} \{\sigma(S)\}. \tag{1}$$

The difficulty of IM is embodied in two aspects: *i).*Selecting an optimal collection in IM has been proved to be NP-hard [2]. So IM algorithms have to attain a balanced compromise between accuracy and efficiency. *ii).*The propagation function $\sigma(\cdot)$ is stochastic, leaving the analytic optimization w.r.t S intractable, types of methods that deal with this problem are as follows:

1) *Greedy methods* usually utilize the *submodularity* of $\sigma(\cdot)$ to accelerate the simulation process. CELF [4] and CELF++ [9] have alleviated time complexity of naive Monte-Carlo greedy methods dramatically. And the influence range obtained by a greedy method is proved to be no less than 63% compared with the optimal solution [2]. But adversary constructions such as Figure. 2 still haunt greedy approaches and leave space for improvement.
2) *Heuristic methods* explore features from $\mathcal{G}$ rather than taking $\sigma(\cdot)$ as a black box and improve the efficiency by narrowing the candidate set of "seeds". Methods as Degree [2] take advantage of degree information to derive results. Methods as PageRank [10] show that models from other disciplines can be used in IM heuristically by analogy as well. Some late studies [11] [12] aim at identifying the cluster structure in the OSN to help initialization. However, the outcome of a heuristic method can be arbitrarily bad.
3) *Hybrid methods* use heuristic features and greedy methods simultaneously such as [13] and [14].

B. Utilizing Learning Automata in IM

The optimization target $\sigma(\cdot)$ is a stochastic function, so is the environment for learning automata. Therefore it is intuitively feasible that LA can find the optimal subset. A single LA is an appropriate optimizer for single-valued function [15], thus it is plausible to use LA in the greedy methods of

IM to maximize the marginal increment of influenced range, and it turns out that LA can be even faster than traditional greedy methods [16]. Additionally, it has been proposed that a collection of LA trained conjugately, i.e., simultaneously, is capable of addressing combinatorial optimization tasks [17]. Therefore it is possible to apply conjugate LA in the problem of IM to attain better-than-greedy results while preserving some degree of efficiency.

III. Proposed Method

A. Learning Automaton

LA is an important reinforcement learning method, which can adaptively explore the optimal action that maximizes the reward among all possible choices by interacting with a stochastic environment. An LA with its environment is formalized as a triplet $< A, B, \mathbf{D} >$, where $A = \{\alpha_1, \alpha_2, \cdots\}$ is the set of optional actions, $B = \{\beta_1, \beta_2, \cdots\}$ is the set of possible feedback from the environment, and $\mathbf{D}$ is the reward matrix of the environment following

$$\Pr\{\beta_q|\alpha_r\} = d_{r,q}, \beta_q \in B, \alpha_r \in A. \tag{2}$$

For most LA schemes, the training is equivalent to tuning the normalized action probability vector $\mathbf{P} = [P_1, P_2, \cdots]$ to maximize the expected reward $\sum_{r,q} d_{r,q} \cdot P_r \cdot \beta_q$. The training process consists of a number of iterations, at the t-th iteration, an LA selects the action $\alpha(t)$ according to $\mathbf{P}(t)$

$$\Pr\{\alpha(t) = \alpha_r\} = P_r(t). \tag{3}$$

The environment receives $\alpha(t)$ and returns the feedback $\beta(t)$ satisfying (2). The LA receives $\beta(t)$ and updates $\mathbf{P}(t)$ into $\mathbf{P}(t+1)$ according to some specific strategy. The LA gets converged and terminates training when $\max_r \{P_r\} > \mathcal{T}$, where $\mathcal{T}$ is a predefined threshold.

B. Conjugate Learning Automata

The conjugate learning automata (CLA) originates from learning automata games where multiple independent LAs co-operate or compete with each other to obtain the Nash equilibrium [18]. Since it seeks a global equilibrium rather than a greedy one, it can possibly handle the coined pitfall that confuses greedy methods in the problem of IM and enlarge the influence range.

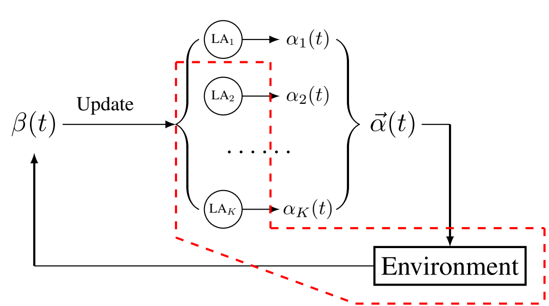


Fig. 3. CLA with K LA. The external environment of LA_1 is besieged by the dashed red polygon.

As shown in Figure 3, a CLA comprises multiple LAs. At the t-th iteration, each learning automaton LA_k selects an action $\alpha_k(t)$ according to its probability vector $\mathbf{P}^k(t)$. The CLA subsequently combines the actions of its components to form a vector $\vec{\alpha}(t)$ and sends it to the environment. The stochastic environment then reacts in $\beta(t)$ with respect to $\vec{\alpha}(t)$. Upon receiving the feedback, each LA independently updates its internal states/probability vector. Literature [18]- [20] have theoretically proved the convergence of CLA. However, when the Nash equilibrium in the environment is not unique, CLA only converges to one of them, and global optimality is intractable.

C. Conjugate Learning Automata in Influence Maximization

To map the problem of IM into CLA, we define the action set, the feedback and the external environment as follows:

- *CLA structure*. If K members of OSN are to be selected as seeds, then we have K LA operate conjugately.
- *Action*. The action set A_k for the k-th LA is defined as all possible vertices that can be chosen as one of the "seeds". The associated action vector corresponds to the chosen set of "seeds".
- *Feedback*. The information spread range is the top concern in IM. For the given "seeds", the feedback is the number of influenced nodes in the social network, i.e., $\beta(t) = \sigma(\vec{\alpha}(t))$, which is obtained by a propagation simulation in OSN.
- *External Environment*. The external environment for each LA includes not only the network but also other automata, hence is non-stationary. For any LA, the reaction pattern of its environment, which consists of the network structure and other LAs, varies in time (the probability vector of other LAs are changing at the same time). Thus the convergence theorem for LA in stationary environments fails.

The training procedure of CLA is modified as in Algorithm 1 to ensure convergence. During iterations, only one LA can update its internal states, while other LAs keep their probability vector fixed. In this way each LA is trained in a stationary environment, hence convergence is secured. Meanwhile, we prohibit an LA to converge to its optimal action while leaving other LAs totally untrained, because this is tantamount to greedy solution and yields no more interest. So the temporary halt condition is $\max_r \{P_r\} > \delta$, where δ is the temporary threshold smaller than $\mathcal{T}$. Thence each LA learns some rough knowledge about the optimal distribution of seeds while the flexibility for cooperation is saved. The philosophy behind is an analogy of adiabatic process, with δ as the counterpart of present temperature and $\mathcal{T}$ the destinated temperature. To ensure cooperation, the adiabatic assumption is exerted so δ increases so slowly that the system is always in equilibrium.

Having altered the structure of CLA, the specification of individual LA is left to be addressed. This aspect is as vital as the CLA training scheme since an efficient LA algorithm

Algorithm 1 CLA for IM

1: **Input** Temporary threshold: $\delta = \delta_0 \in (0, 1)$.
2: **Input** Iterative increment: $\Delta\delta \in (0, 1)$.
3: **Input** Convergence threshold: $\mathcal{T} \in (0, 1)$.
4: **Initialize** Convergence flag: $\mathcal{I} = 1$.
5: **Initialize** Flag vector: $\mathbf{I} = \mathbf{1}_K$, an all-one vector.
6: **repeat**
7:   **for** $k = 1$ to K **do**
8:     **if** $\mathbf{I}_k = 1$ **then**
9:       $\dot{k} = k$.
10:       Break.
11:     **end if**
12:   **end for**
13:   //Choose the $\dot{k}$-th LA to update.
14:   **repeat**
15:     Each LA independently selects an action, the actions compose the associated action vector of CLA, i.e.,the trial set of "seeds". The efficacy of this selection of seeds is evaluated by a propagation simulation.
16:     Only the $\dot{k}$-th LA updates its probability vector $\mathbf{P}^{\dot{k}}$.
17:   **until** $\max_r\{P_r^{\dot{k}}\} \geq \delta$.
18:   $\mathbf{I}(\dot{k}) = 0$.
19:   **if** $\sum_k \mathbf{I}_k = 0$ **then**
20:     //All LAs have acquire certain information. Lifting the threshold promotes cooperation.
21:     **if** $\delta < \mathcal{T}$ **then**
22:       $\delta = \min\{\delta + \Delta\delta, \mathcal{T}\}$.
23:       Re-initialize $\mathbf{I} = \mathbf{1}_K$.
24:     **else**
25:       Convergence Flag $\mathcal{I} = 0$.
26:     **end if**
27:   **end if**
28: **until** Convergence Flag $\mathcal{I} = 0$.

can reduce the required interactions with the environment and improve the speed of finding the optimal set of "seeds".

However, most established LA algorithms operate in environments where the feedback is binary, so they have to be modified since the feedback in IM is multi-valued. We propose to extend the traditionary Discrete Generate Pursuit Algorithm (DGPA) [21] as Algorithm 2. DGPA prevails in concise structure, low computation cost and analytic optimality. And DGPA can be readily extended to multi-valued feedback environments.

To summary, CLA in IM is the Algorithm 1 with lines 15-16 specified by Algorithm 2. The trade-off between locality and globality is realized by δ.

- When $\delta_0 \ll \mathcal{T}$, Algorithm 1 reduces to an ordinary CLA where all LA are trained almost independently, but stable convergence is sacrificed.
- When $\delta_0 \to \mathcal{T}$, our proposal degenerates gracefully to the greedy method. Since the fluctuations from under-trained LA are approximately white noise and eliminate each other, CLA obtains the greedy option one at a time.

Algorithm 2 The extended version of DGPA, eDGPA

Input Temporary threshold: $\delta \in (0, 1)$.
Input Resolution parameter: $\mathcal{N} \in \mathbb{N}^*$.
Initialize : Update Step Size: $\Delta = \frac{1}{\mathcal{N}R}$, where R is total number of actions.
Initialize : Action probability vector: $\mathbf{P}(0) = \frac{1}{R}\mathbf{1}_R$.
Initialize : Action selection times vector: $\mathbf{Z} = \mathbf{1}_R$.
Initialize : Rewards Estimation Vector: $\mathbf{E} = \mathbf{1}_R$.
repeat
  At the t-th iteration, select action $\alpha(t) = \alpha_r$ by (3).
  Receive the feedback $\beta(t)$ from the environment and update the states as follows:
  $E_r = [Z_r \cdot E_r + \beta(t)] \, / \, [Z_r + 1]$,
  $Z_r + +$.
  $W_r = | \{s : \alpha_s \in A, E_s > E_r\} |$.
  Update the action probability vector $\mathbf{P}(t) \to \mathbf{P}(t+1)$ as follows:
  $P_s = \min\{P_s + \frac{\Delta}{W_r}, 1\}, \forall \alpha_s \in A : E_s > E_r$,
  $p_s = \max\{P_s - \frac{\Delta}{R - W_r}, 0\}, \forall \alpha_s \in A : E_s < E_r$,
  $P_r = 1 - \sum_{s \neq r, \alpha_s \in A} P_s$.
  $t + +$.
until $\max_r \{P_r\} \geq \delta$.

The step $\Delta\delta$, by which δ gradually increases to $\mathcal{T}$, is the resolution of the simulation of the adiabatic process. It naturally reflects the trade-off between exploration (when $\Delta\delta$ is small) and exploitation (when $\Delta\delta$ is large), an innate duality in reinforcement learning.

IV. Experimental and Discussion

A. Experimental Settings

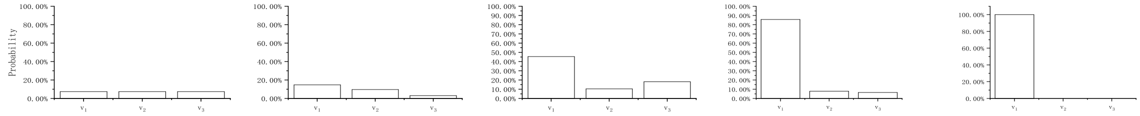
All experiments are conducted under the weighted cascade (WC) diffusion model [2], where a node v in social networks is influenced by edge $(u, v) \in E$ with probability $\frac{1}{\text{in-degree}(v)}$. For comparison, CELF, the representative greedy algorithm with high efficiency, is selected as the baseline algorithm. The heuristic methods have generally smaller spread and are saved from comparison [16].

For the extended DGPA, the resolution parameter $\mathcal{N}$ is set as the number of "seeds", K. CLA in all experimental are parameterized by $(\delta_0, \Delta\delta, \mathcal{T}) = (\frac{1}{K}, \frac{1}{2K}, 0.999)$. Besides, for greedy-based algorithms, i.e., the naive greedy algorithm and CELA, the number of Monte-Carlo simulations is uniformly set to 10000, which is conventional in literature.

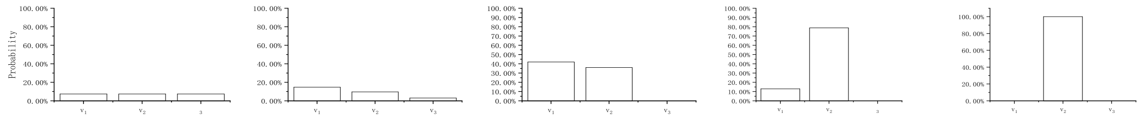
B. Synthetic Datasets

Firstly, consider the toy example in Figure 2 with 17 vertices altogether, specifically $s_1 = s_2 = 5, s_3 = s_4 = 2, s_5 = 3$ and $K = 2$. We compare CLA with the naive greedy method using eDGPA ($\delta = 0.999$) to select the local optimal vertex greedily.

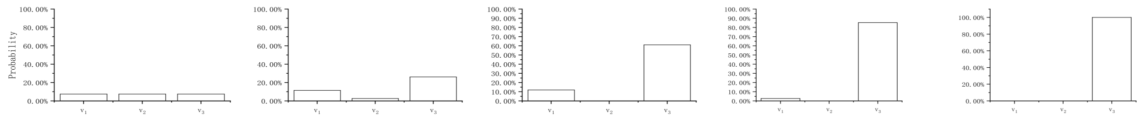
The greedy approach will select v_1 at first anyway, as shown in Figure 4(a). While the CLA with two LA can identify the optimal options $\{v_2, v_3\}$ correctly, as illustrated in Figure 4(b)



(a) Probability vector of the eDGPA in greedy method to select the first node, after the 0_{th}, 25_{th}, 50_{th}, 75_{th} and 100_{th} iteration.



(b) Probability vector of the first LA in CLA, after the 0_{th}, 25_{th}, 50_{th}, 75_{th} and 100_{th} iteration.



(c) Probability vector of the second LA in CLA, after the 0_{th}, 25_{th}, 50_{th}, 75_{th} and 100_{th} iteration.

Fig. 4. The change of probability vectors through training for greedy LA and CLA in Figure.2. In all subfigures the vector varies during iterations from left to right. The horizontal axis denotes the nodes while the vertical axis denotes the probability.

and 4(c). The components for v_4-v_{15} remain trivial and are truncated from the demonstration.

Secondly, a large synthetic network following the WC model with 7451 nodes is coined to verify the outperformance of CLA to greedy methods. Consider a network shown by Figure 5 in which $\mathcal{M}_p$ is a subnetwork with an average propagation range of M with respect to the information input from v_p where $p = 1, 2, \cdots, P$.

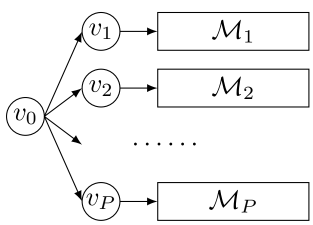
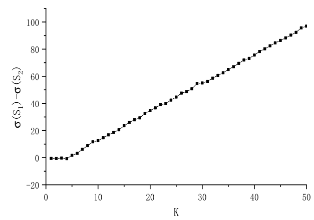
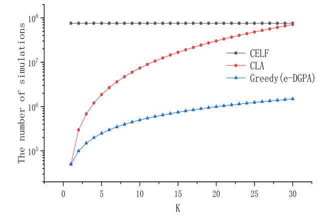


Fig. 5. A large-scale adversary network against greedy IM methods.

Assuming that v_0 can send its information to only one successor. By setting $P = 50$, $M = 100$ and $K \leq P$. The optimal solution is to choose K nodes directly from $\{v_p\}_{p=1}^{P}$ with the propagation range $\sigma(S_1) = MK + K$, while the greedy methods will choose v_0 and $(K-1)$ vertices from $\{v_p\}_{p=1}^{P}$, with the expected propagation range $\sigma(S_2) = \frac{(K-1)(MK+K+1-M)+(P-K+1)(MK+K+1)}{P}$. The comparison between the spread range between CLA $\sigma(S_1)$ and greedy methods $\sigma(S_2)$ and the difference in time consumption are shown in Figure 6, where CLA, the naive greedy method using eDGPA ($\delta = 0.999$) and CELF are compared.



(a) The increment of influence range by using CLA instead of greedy methods with eDGPA.



(b) The number of required simulations in networks of CLA, CELF and the greedy method with eDGPA.

Fig. 6. Comparisons between CLA and greedy methods in the network of Figure 5.

C. Real-World Social Networks

Three real-world datasets including Arxiv GrQc, HEP-TH and HEP-PH [1], which have been widely used to evaluate the performances of IM algorithms, are adopted to verify the effectiveness of the proposed CLA. Table I shows the statistics of the three datasets.

TABLE I
STATISTICS OF THE REAL-WORLD DATASETS

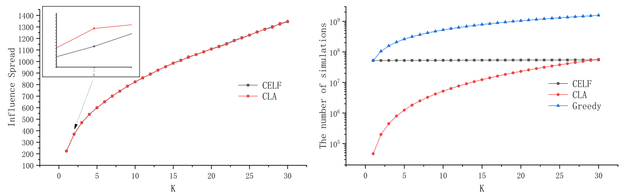
Datasets	GrQc	HEP-TH	HEP-PH
Nodes	5242	9877	12008
Edges	14496	25998	118521

We compare the spread ranges of CELF and CLA (the spread range of the naive greedy is almost the same as CELF), and the computing efficiency between the naive greedy methods, CELF and CLA. The result is illustrated by Figure 7.

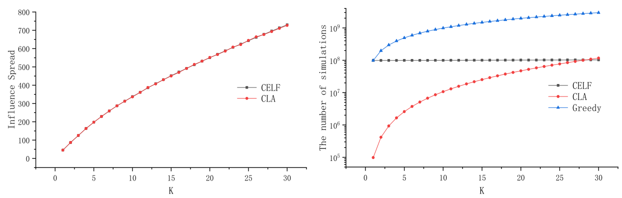
It can be observed from Figure 7 that:

1) There exist cases where CLA outperforms greedy methods in influence spread range by choosing different source nodes (GrQc with $K = 2$, etc.). This indicates that CLA is capable of outperforming greedy methods by avoiding potential pitfalls.
2) In most settings, the influence ranges of greedy methods and CLA are the same. This result implies that adversary

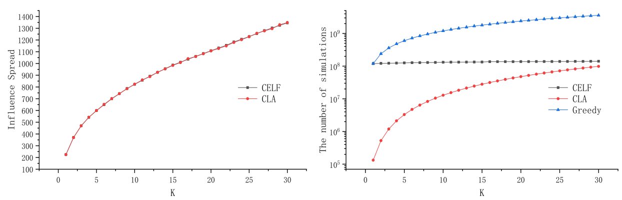
[1] http://www.arXiv.org



(a) The GrQc dataset.



(b) The HEP-TH dataset.



(c) The HEP-PH dataset.

Fig. 7. The comparisons between the naive greedy method, CELF and CLA in influence range and efficiency.

construction is scanty in at least studied datasets. Otherwise there would be an explicit difference in influence range as Figure 6. This result also helps to validate the accuracy of greedy methods on these datasets.

3) Respecting the efficiency, or the times of simulations, CLA outperforms the naive greedy method and CELF. This fact indicates that apart from achieving similar influence range, CLA prevails in efficiency and is an adorable method to take to solve the problem of IM.

V. Conclusions

On addressing IM, there have been plentiful studies, admitting the greedy paradigm, aimed at improving the computation efficiency. In this paper we attempt to obtain better-than-greedy propagation range using CLA. An iterative learning strategy with an extension of traditional DGPA is proposed to implement the conjugate optimization. Comprehensive simulations in both synthetic and real-world datasets verify that CLA can attain better propagation range in some scenarios and is equally competitive respecting time consumption.

Acknowledgment

References

[1] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2001, pp. 5766.

[2] D. Kempe, J. Kleinberg, and 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.* ACM, 2003, pp. 137146.

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

[4] Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., & Glance, N. "Cost-effective outbreak detection in networks" in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2007, pp.420-429.

[5] Kostka, Jan, Yvonne Anne Oswald, and Roger Wattenhofer. "Word of mouth: Rumor dissemination in social networks." *International colloquium on structural information and communication complexity.* Springer, Berlin, Heidelberg, pp.185-196, 2008.

[6] Cutillo, Leucio Antonio, Refik Molva, and Thorsten Strufe. "Privacy preserving social networking through decentralization." in *Proceedings of the Sixth International Conference on Wireless On-Demand Network Systems and Services.* IEEE, 2009, pp.145-152.

[7] Datta, A., Buchegger, S., Vu, L. H., Strufe, T., & Rzadca, K. "Decentralized online social networks." *Handbook of Social Network Technologies and Applications.* Springer, Boston, MA, pp.349-378, 2010.

[8] Torkestani, Javad Akbari, and Mohammad Reza Meybodi. "A cellular learning automata-based algorithm for solving the vertex coloring problem." *Expert Systems with Applications* vol.38, no.8, pp.9237-9247, 2010.

[9] Goyal, Amit, Wei Lu, and Laks VS Lakshmanan. "Celf++: optimizing the greedy algorithm for influence maximization in social networks." in *Proceedings of the 20th international conference companion on World wide web.* ACM, 2011, pp.47-48.

[10] Brin, Sergey, and Lawrence Page. "The anatomy of a large-scale hypertextual web search engine." *Computer networks and ISDN systems* vol.30, no.1-7, pp.107-117,1998.

[11] Qian, Fulan, et al. "A Novel Approach for Influence Maximization Based on Clonal Selection Theory in Social Networks." *2018 IEEE International Conference on Data Mining Workshops (ICDMW).* IEEE, 2018, pp.430-437.

[12] Gong, M., Song, C., Duan, C., Ma, L., & Shen, B. An efficient memetic algorithm for influence maximization in social networks. *IEEE Computational Intelligence Magazine*, vol.11, no.3, pp.22-33, 2016.

[13] Cui, Laizhong, et al. "DDSE: A novel evolutionary algorithm based on degree-descending search strategy for influence maximization in social networks." *Journal of Network and Computer Applications* vol.103, 119-130, 2018.

[14] Ko, Yun-Yong, Kyung-Jae Cho, and Sang-Wook Kim. "Efficient and effective influence maximization in social networks: a hybrid-approach." *Information Sciences* vol.465, pp.144-161, 2018.

[15] Beigy, Hamid, and M. R. Meybodi. "A new continuous action-set learning automaton for function optimization." *Journal of the Franklin Institute* vol.343, no.1, pp.27-47, 2006.

[16] Ge, Hao, et al. "Learning automata based approach for influence maximization problem on social networks." *IEEE Second International Conference on Data Science in Cyberspace (DSC).* IEEE, 2017, pp.108-117.

[17] Vafashoar, Reza, and Mohammad Reza Meybodi. "Reinforcement learning in learning automata and cellular learning automata via multiple reinforcement signals." *Knowledge-Based Systems* vol.169, pp.1-27, 2019.

[18] Fu, King-Sun, and Timothy J. Li. "Formulation of learning automata and automata games." *Information Sciences* vol.1, no.3, pp.237-256, 1969.

[19] Lakshmivarahan, S., and Kumpati S. Narendra. "Learning algorithms for two-person zero-sum stochastic games with incomplete information: A unified approach." *SIAM Journal on Control and Optimization* vol.20, no.4, pp.541-552, 1982.

[20] Tilak, Omkar, Ryan Martin, and Snehasis Mukhopadhyay. "Decentralized indirect methods for learning automata games." *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* vol.41, no.5, pp.1213-1223, 2011.

[21] Agache, Mariana, and B. John Oommen. "Generalized pursuit learning schemes: new families of continuous and discretized learning automata." *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* vol.32, no.6, pp.738-749, 2002.