Applied Intelligence (2020) 50:354–364
https://doi.org/10.1007/s10489-019-01529-x

Community-based influence maximization in attributed networks

Huimin Huang[1] · **Hong Shen**[1,2] · **Zaiqiao Meng**[1]

Published online: 24 July 2019

Abstract

Influence Maximization, aiming at selecting a small set of seed users in a social network to maximize the spread of influence, has attracted considerable attention recently. Most existing influence maximization algorithms focus on pure networks, while in many real-world social networks, nodes are often associated with a rich set of attributes or features, aka attributed networks. Moreover, most of existing influence maximization methods suffer from the problems of high computational cost and no performance guarantee, as these methods heavily depend on analysis and exploitation of network structure. In this paper, we propose a new algorithm to solve community-based influence maximization problem in attributed networks, which consists of three steps: community detection, candidate community generation and seed node selection. Specifically, we first propose the candidate community generation process, which utilizes information of community structure as well as node attribute to narrow down possible community candidates. We then propose a model to predict influence strength between nodes in attributed network, which takes advantage of topology structure similarity and attribute similarity between nodes in addition to social interaction strength, thus improve the prediction accuracy comparing to the existing methods significantly. Finally, we select seed nodes by proposing the computation method of influence set, through which the marginal influence gain of nodes can be calculated directly, avoiding tens of thousands of Monte Carlo simulations and ultimately making the algorithm more efficient. Experiments on four real social network datasets demonstrate that our proposed algorithm outperforms state-of-the-art influence maximization algorithms in both influence spread and running time.

Keywords Attributed networks · Influence maximization · Influence strength · Community detection

1 Introduction

Influence maximization, as a technique to help in solving social issues (e.g. preventing terrorist attacks, anticipating natural hazards) and optimizing business performance (e.g. optimizing social marketing campaigns), has attracted considerable attention recently. It aims at solving the problem of selecting a fixed size set of seed nodes in a network to maximize the influence spread, according to a specially designed influence diffusion model.

Motivated by the idea of viral marketing, Domingos and Richardson proposed the problem of influence maximization in 2001 [1, 2]. It attracts extensive interests bacause of its wide application. To solve influence maximization problem, researchers have proposed various influence diffusion models, which can be mainly classified into two categories: Linear Threshold (LT) models and Independent Cascade (IC) models [3]. In LT model, each node is activated jointly by its all neighbors, with a thresholds ranging from 0 to 1 assigned by stochastic, while in IC model, diffusion process of nodes is simulated by a stochastic process [3]. Most existing algorithms [3–11, 16] either consider a greedy strategy by applying the monotonicity and submodularity of the objective function or take advantage of topology structure of network to increase the performance of algorithms.

Recently, the community-based influence maximization algorithms [12–15, 22, 25] are widely studied. The efficiency of these algorithms outperforms the general greedy algorithms largely, as they assume the independence between communities and suits to run parallelized. However, the existing community-based influence maximization algorithms demonstrate a number of major drawbacks: (a) They fail to model influence strength of attributes between nodes in attributed network, while attribute is essential

✉ Huimin Huang
huanghm45@gmail.com

[1] School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China

[2] School of Computer Science, University of Adelaide, Adelaide, Australia

indicator to describe influence strength of nodes; (b) They have no effective methods to reduce the number of candidate seeds, that results in the huge search space for selecting seed nodes when facing the extremely large networks in realistic settings, (c) They suffer from the problems of high computational cost caused by tens of thousands of Monte Carlo simulations used to evaluate the influence of internal nodes within communities.

To address these issues, we propose a new algorithm to solve **C**ommunity-based **I**nfluence **M**aximization problem in **A**ttributed networks, called **CIMA**, which consists of three main steps: community detection, candidate community generation and seed node selection. CIMA performs pruning insignificant communities by a process of candidate community generation, and jointly models topology structure similarity and attribute similarity between nodes as well as social interaction strength to predict influence strength between nodes.

As community structure tends to be detected through connection closeness between nodes, there are more chances to trigger activations of nodes if we place a seed in a large community. Communities with different sizes should not be treated the same when generating candidate communities. Hence, the size of community is a factor we should consider when assigning seed nodes to communities. Besides, previous research on social networks [18, 24] has revealed that the more similar the attributes between nodes, the closer relationship they have. Thus, in addition to topology structure, node attribute similarity ensures community structure can be reflected well. Therefore, we decide candidate communities by two factors: size of the community and attribute similarity between nodes within the community.

As much research on psychology and marketing has revealed that similar attributes of individuals can cause similar behaviors [21], when modeling influence strength in attributed network, attribute similarity between nodes should be taken into consideration in addition to social interaction strength, topology structure similarity between nodes.

To sum up, our major contributions in this paper are:

(i) We propose a new algorithm to solve community-based influence maximization problem in attributed networks which consists of three main steps: community detection, candidate community generation and seed node selection. The process of candidate community generation prunes the insignificant communities and increases the efficiency of the algorithm largely.

(ii) We propose a model to predict influence strength between nodes in attributed network, which takes advantage of topology structure similarity and attribute similarity between nodes in addition to social interaction strength, thus improves the prediction accuracy comparing to the existing methods significantly.

(iii) We propose the computation method of influence set when selecting seed nodes, through which influence marginal gain of a node on seed node set can be calculated directly, avoiding tens of thousands of Monte Carlo simulations and ultimately making the algorithm more efficient.

(iv) We conduct comprehensive experiments on the real-world social networks datasets, and the results validate the effectiveness of our algorithm compared with state-of-the-art influence maximization algorithms.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 provides some notations and terminology. Section 4 details our CIMA Algorithm. Section 5 reports our experiment analysis. Section 6 concludes the paper.

2 Related work

The problem of influence maximization was first formulated into a discrete optimization problem by Kempe et al. in [3]. They prove the optimization problem of influence maximization is NP-hard and provide a (1-1/e)-approximation greedy algorithm with proving objective function is monotonous and submodular under Independent Cascade model and Linear Threshold model. Since then, influence maximization has been studied extensively. Some greedy-based algorithms in [3–7, 11] are proposed, which apply submodularity and monotony of IC model or LT model, and use Monte Carlo simulations to approximately evaluate the influence propagation, that makes them obtain better accuracy of the influence spread evaluation. However, in large-scale social networks, the efficiency of the algorithms will be reduced greatly because of large number of Monte Carlo simulations.

Some algorithms in [5, 8–10, 16] emphasize analyzing and exploiting topology structure of network. Generally, they are much more efficient than greedy algorithms, but they have no theoretical guarantee.

Community-based influence maximization algorithm are proposed in recent years. They are widely studied in [12–15, 22, 25]. The efficiency of these algorithms outperforms the general greedy algorithms largely, as they assume the independence between communities and suit to run parallelized. However, many of them depend on the specific diffusion models, that makes them uncertain to have better performance under classical Independent Cascaded model and Linear Threshold model or not.

To the best of our knowledge, there is no community-based algorithm that models influence strength between nodes, taking into account attribute similarity and topology

structure similarity between nodes in addition to social interaction strength. In most cases, they tend to consider the social interaction strength only.

3 Preliminaries

In this section, we briefly review our notations and terminology.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W}, A)$ be an attributed network, where $\mathcal{V}$ is the set of network nodes, $\mathcal{E}$ is the set of directed network edges, $\mathcal{W}$ is the set of real positive weights recording each-edge association and A is a matrix recording node-attribute associations. Let $R^{n\times n}$ be a weighted adjacent matrix of the network with $r_{u,v}$ being the elements of matrix R, such that:

$$r_{u,v} = \begin{cases} w_{uv} \in \mathcal{W} & \text{if there is a edge from } u \text{ to } v \\ 0 & others, \end{cases} \tag{1}$$

where n is the number of nodes, w_{uv} is the weight of edge from node u to node v.

Let $A \in \{0, 1\}^{n\times F}$ be a binary-valued attribute matrix with $a_{u,f}$ being the elements of matrix A, such that:

$$a_{u,f} = \begin{cases} 1 & \text{attribute } f \text{ associates with node } u \\ 0 & others \end{cases} \tag{2}$$

where F is the number of attributes.

The community structure $C = \{c_1, c_2, ..., c_p\}$ is the division of the graph $\mathcal{G}$, and we only consider the non-overlapping community structure where each node in graph $\mathcal{G}$ is only assigned to a certain community.

Given an information diffusion model I, the spread $\Gamma(\mathcal{S})$ of a set of seed nodes $\mathcal{S}$ is defined as the total number of nodes that are active at the end of the information diffusion process, including both the newly activated nodes and the initially active set $\mathcal{S}$.

Definition 1 (Influence maximization). Given a social network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a diffusion model and the corresponding parameters (e.g. IC model and activation probabilities between pairs of nodes), influence maximization is to find a subset of $\mathcal{V}$, i.e., $\mathcal{S}^*$, such that

$$\mathcal{S}^* = \arg\max_{\mathcal{S}} \sigma(\mathcal{S}), \tag{3}$$

where $\mathcal{S}$ denotes the set of seed nodes and $\sigma(\mathcal{S}) = \mathbb{E}[\Gamma(\mathcal{S})]$ represents the expected spread of $\mathcal{S}$.

In this paper, we use the IC model as diffusion model since it is the one being most extensively and deeply studied.

4 Methodology

To address the above problem we propose CIMA, a new algorithm to solve **C**ommunity-based **I**nfluence **M**aximization problem in **A**ttributed networks based on theory of influence diffusion as well as research of psychology and marketing. Our CIMA composes of three steps: community detection, candidate community generation and seed node selection. In Step 1, we perform community detection,using a non-overlapping community detection algorithm, i.e., Louvain algorithm [17] because of its excellent performance (verified in paper [23]). In Step 2, we decide the candidate communities, pruning the insignificant communities and ultimately making the algorithm more efficient largely. In Step 3, we utilize a model to predict influence strength between nodes, and take advantage of our computation method of influence set to mine top-k influential nodes. The workflow of our CIMA algorithm is as Fig. 1 shows. We will detail our method in the following subsections.

4.1 Community detection

We directly use the Louvain Algorithm [17] to execute community detection, which uses greedy strategies in a local manner to optimize modularity, an index to measure the quality of community structure. In the following, we briefly introduce Louvain Algorithm.

$$\begin{aligned} \Delta Q = &\left[\frac{\sum_{in} + 2k_{u,in}}{2m} - (\frac{\sum_{tot} + k_u}{2m})^2\right] \\ &- \left[\frac{\sum_{in}}{2m} - (\frac{\sum_{tot}}{2m})^2 - (\frac{k_u}{2m})^2\right] \end{aligned} \tag{4}$$

The pseudocode of Louvain is as Algorithm 1 shows. It composes of two procedures [17]. In Algorithm 1, 1:, 2:, 3: are included in Procedure 1, and 4:, 5: are included in Procedure 2. In Procedure 1, Louvain assigns a different community to each node in network, then for each node u, computes the increment of modularity ΔQ caused by removing u from its community and by placing u in the

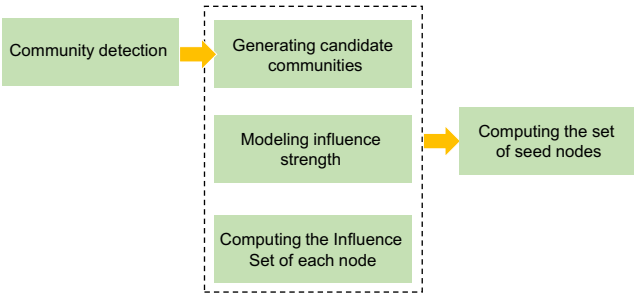

Fig. 1 The workflow of our CIMA algorithm

community of u's neighbor v. Increment of modularity ΔQ is as (4) shows, where $\sum_{in}$ denotes the sum of the weights of the links inside community c_j, $\sum_{tot}$ denotes sum of weights of the links incident to nodes in community c_j, k_u denotes sum of weights of the links incident to node u, $k_{u,in}$ denotes sum of weights of the links from u to nodes in community c_j and m denotes sum of weights of all the links in the network. Node u will be assigned to the community for which the increment modularity ΔQ is maximum. This process is repeated for all nodes until no improvement can be achieved any more, and then Procedure 1 is completed.

Algorithm 1 Pseudocode of Louvain.

1: Look each node in the network as a separate community.
2: For each node u, it is sequentially attempted to assign node u to the community in which each of its neighbor nodes is located, calculate the gain of modularity ΔQ before and after the allocation, and record the neighbor node with the largest ΔQ. If $\max\Delta Q > 0$, then assign node u to the community where the neighbor node with the largest ΔQ is located, otherwise it remains unchanged.
3: Repeat 2: until the community of all nodes no longer changes.
4: Look all nodes in the same community as a new node. To do so, the weights of the links between the new nodes are given by the sum of the weight of the links between nodes in the corresponding two communities.
5: Repeat 1:,2:,3:,4: until there are no more changes and a maximum of modularity of the entire network is attained.

In Procedure 2, the algorithm builds new topology structure of network by treating the communities found during Procedure 1 as new nodes, and computes the weights of the links between the new nodes, which are given by the sum of the weight of the links between nodes in the corresponding two communities. After Procedure 2 is completed, re-perform Procedure 1 with the new topology structure of network and weights. The iterations are carried on until no more changes and a maximum of modularity is attained.

4.2 Generating candidate community

We have discussed in Section 1 that seeds selected from a large community could activate more nodes than seeds selected from a small community, thus we introduce the significant community. Significant community is the community that has the size more than the average number of nodes a seed may influence instead of an arbitrary number. Average attribute similarity is the average of attribute similarities of all node within a community. As discussed in Section 1 that candidate communities are decided by two factors, i.e., size of the community and attribute similarity between nodes within the community, we determine significant communities as candidate communities and in addition, we add the insignificant communities into candidate community set such that average attribute similarities of nodes within these communities are more than a threshold α.

Definition 2 (Significant Community). Given an attributed network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W}, A)$, the size of seed node set k, and the community structure $C = \{c_1, c_2, ..., c_p\}$ derived from Louvain Algorithm, significant community set can be defined as follows:

$$C_s = \left\{ c_j \in C | n_j \geq \frac{\sum_{i=1}^{p} n_i}{k} \right\}, \tag{5}$$

where c_j and n_j respectively denote the jth community and the number of nodes in community c_j.

It mean that Significant community must have the size larger than the average number of nodes a seed may influence. As community structure tends to be detected through connection closeness between nodes, there are more chances to trigger activations of nodes if we place a seed in a large community. Thus we consider the size of community when generating candidate communities.

Definition 3 (Average Attribute Similarity). Given an attributed network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W}, A)$ and the community structure $C = \{c_1, c_2, ..., c_p\}$ derived from Louvain algorithm, average attribute similarity within community c_j is defined as follows:

$$AS_{c_j} = \frac{\sum_{u,v \in c_j} a_u \cdot a_v}{n_j}, \tag{6}$$

where $a_u \in \{0, 1\}^F$ and $a_v \in \{0, 1\}^F$ denote the attribute vectors of node u and v respectively.

Previous research on social networks [18, 24] has revealed that The more similar attributes between nodes, the closer relationship they have. Node attribute similarity ensures community structure can be reflected well. Therefore, we consider adding the insignificant communities into candidate community set such that average attribute similarities of nodes within these communities are more than a threshold α.

4.3 Modeling influence strength

Different from pure networks, attributed network has more complex characters. More precisely, node attributes provide

rich and complementary sources of information that should be used for revealing, understanding influence strength in attributed network.

In this paper, for attributed network, we model influence strength between pairs of nodes with three factors: social interaction strength, topology structure similarity and attribute similarity between nodes within the community.

Modeling social interaction strength Social interaction strength can be measured by the number of interactions between two nodes in attributed network and can normalized as follows:

$$L_{uv} = \frac{I_{uv} - I_{min}}{I_{max} - I_{min}}, \tag{7}$$

where L_{uv} denotes normalized social interaction strength from node u to node v, I_{uv} denotes the number of interactions from node u to node v, and I_{max}, I_{min} are the maximal and minimal value of the number of interactions in the attributed network respectively.

Modeling topology structure similarity between nodes Previous studies on social network has revealed that opinions and behaviors of nodes are influenced by network topology structure [19]. In modeling Influence Strength between nodes, to preserve the effect of network topology structure, the first intuition is that edges of pairwise vertices must be preserved. Assume node u and node v are in the same community, and let r_u and r_v be the n-dimension weighted edge vectors of u and v respectively. We use the logistic model with edge vectors r_u and r_v as input to compute the topology structure similarity between node u and node v:

$$R_{uv} = \sigma(r_u^T \cdot r_v), \tag{8}$$

where r_u^T denotes the transposed vector of r_u and $\sigma(\cdot)$ is a logistic function defined as $\sigma(x) = (1 + e^{-x})^{-1}$.

Modeling attribute similarity between nodes The idea of preserving edge information could be extended to model attribute similarity when modeling influence strength in attributed network as psychology and marketing theory has suggested that similar attributes of individuals are causes of similar behaviors [21]. Assume node u and node v are in the same community, and let a_u and a_v be the F-dimension attribute vectors of u and v respectively. Similar to modeling topology structure similarity, we implement the same procedure on node attribute affinity, using a logistic function with attribute vectors of two nodes as input:

$$W_{uv} = \sigma(a_u^T \cdot a_v), \tag{9}$$

where a_u^T denotes the transposed vector of a_u.

Thus, combining (7), (8) and (9), we compute influence strength X_{uv} from node u to node v with a logistic function as:

$$X_{uv} = \left\{1 + e^{6-4(L_{uv}+R_{uv}+W_{uv})}\right\}^{-1} \tag{10}$$

The expression is in the form of $6 - 4(L_{uv} + R_{uv} + W_{uv})$ such that $6 - 4(L_{uv} + R_{uv} + W_{uv})$ is within [-6,6], which makes the logistic function $\left\{1 + e^{6-4(L_{uv}+R_{uv}+W_{uv})}\right\}^{-1}$ is between 0 and 1 approximately. That effectively model the value of influence strength between pairs of nodes. Thus we determine the expression of 6-4(Luv+Ruv+Wuv) in (10).

4.4 Computation method of influence set

For any inactive node v in the network, the computation of probability that v is activated by seed node set $\mathcal{S}$, i.e., $P_v(\mathcal{S})$, can be divided into two cases: (1) when $v \in \mathcal{S}$, $P_v(S) = 1$; (2) when $u \notin \mathcal{S}$, we propose a method of influence set to calculate the influence spread directly.

Definition 4 (Influence Path)[16]. An influence path from u to v, denoted by $P_{u,v}\langle u = u_1, u_2, ..., u_n = v\rangle$ is a non-cyclic sequence of users where adjacent users are connected by edges in $\mathcal{E}$. The activation probability of this influence path is

$$pp(P_{u,v}) = \prod_{i=1}^{n-1} pp(u_i, u_{i+1}), \tag{11}$$

where $pp(u_i, u_{i+1})$ denotes the activation probability of influence path $P_{u_i,u_{i+1}}\langle u_i, u_{i+1}\rangle$.

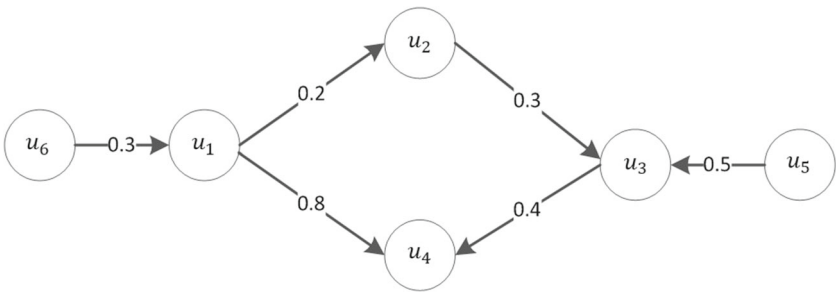

Fig. 2 An example of IC model

As shown in Fig. 2, the influence paths starting from node u_1 include $PA_1 = \langle u_1, u_2, u_3, u_4\rangle$, $PA_2 = \langle u_1, u_4\rangle$. Let $PS(u, v)$ represents the set of influence paths from node u to node v. Since the influence diffusion process in each influence path is independent, the probability that node u activates node v can be expressed as:

$$ap(v|u) = 1 - \prod_{\forall PA \in PS(u,v)} (1 - pp(PA)). \tag{12}$$

According to the above formula, the set of influence paths from u_1 to u_4 in Fig. 2 is $PS(u_1, u_4) = \{PA_1, PA_2\}$, therefore, $ap(u_4|u_1) = 1-(1-pp(PA_1))(1-pp(PA_2)) = 1 - (1 - (0.2 \times 0.3 \times 0.4))(1 - 0.8) = 0.8048$.

Definition 5 (Influence Set). Given an active node $u \in V$, the influence set is defined as a set of dual tuples

$$IS(u) = \{(v, ap(v|u))|v \in \mathcal{V}, ap(v|u) \in [0, 1]\}, \tag{13}$$

the first tuple of which represents the node v which is activated by node u and the second tuple represents the probability that u activates v.

Through this method of set of dual tuples, influence marginal gain of a node W.R.T seed node set can be calculated directly, so we propose to utilize the computation method of influence set.

In IC model, the activation probability of an active node to itself is 1, then the influence set of any active node contains at least one dual tuple. For example, the influence set of an active node u at least contains a tuple $(u, 1)$, namely $IS(u) = \{(u, 1)\}$. The calculation result of influence set of each node in Fig. 1 is shown in Table 1. According to the influence set of the active nodes, we can directly calculate the influence set of the seed nodes set $\mathcal{S}$, i.e., $IS(\mathcal{S}) = \sum_{u\in\mathcal{S}} IS(u)$. From this, we can calculate the influence spread of the seed node set $\mathcal{S}$:

$$f(\mathcal{S}, \mathcal{V}) = \sum_{\forall v \in V} ap(v|\mathcal{S}) = \sum_{\forall v \in V} (1 - \prod_{\forall u \in \mathcal{S}} (1 - ap(v|u))) \tag{14}$$

Table 1 The influence set of nodes in Fig. 1

User	The influence set
u_1	$\{(u_1, 1), (u_2, 0.2), (u_3, 0.06), (u_4, 0.8048)\}$
u_2	$\{(u_2, 1), (u_3, 0.3), (u_4, 0.12)\}$
u_3	$\{(u_3, 1), (u_4, 0.4)\}$
u_4	$\{(u_4, 1)\}$
u_5	$\{(u_3, 0.5), (u_4, 0.2), (u_5, 1)\}$
u_6	$\{(u_1, 0.3), (u_2, 0.06), (u_3, 0.018), (u_4, 0.245), (u_6, 1)\}$

4.5 CIMA algorithm

Algorithm 2 CIMA $(\mathcal{G}, k, X_{uv})$.

Input: $\mathcal{G}, k, X_{uv}$
Output: Seed nodes set $\mathcal{S}$.

```
S ← ∅; H ← ∅;
for each u ∈ V do
    IS(u) ← ∅;
    u.mg = 0;
end for
C ← Louvain(G)        //perform community detection and obtain community structure C;
generating candidate community C'
for each u ∈ V do
    IS(u) = ObtInfSet(u, θ);
end for
for each u ∈ V do
    u.mg = f(u, V);
    u.flag = 0;
    add u to H by u.mg in descending order;
end for
while |S| < k do
    u = H[top];
    if u.flag = |S| then
        S = S ∪ {u};
        H = H \ {u};
    else
        u.mg = σ(S ∪ u) − σ(S);
        u.flag = |S|;
        Resort H by u.mg in descending order;
    end if
end while
return S;
```

We propose to employ Louvain algorithm to execute community detection for its superior performance. Then we use a process of candidate community generation to obtain candidate communities. To avoid tens of thousands of Monte Carlo simulations suffered by the traditional influence maximization greedy algorithms when computing influence spread of nodes, we propose the influence set of nodes to calculate the influence spread directly. In the part of influence seed node selection, the algorithm selects the node with the rule of maximum influence marginal gain for each round. We use a table $\mathcal{H}\langle u, u.mg, u.flag\rangle$, each record of which corresponds to a node in the network and is sorted by $u.mg$ in descending order, where $u.flag$ denotes iteration

times to add u into $\mathcal{H}$ and $u.mg$ is the influence marginal gain of node u w.r.t. seed node set $\mathcal{S}$ in the current iteration, i.e., $u.mg = \sigma(\mathcal{S} \cup u) - \sigma(S)$. We need to choose the node making the greatest $u.mg$ (greatest influence marginal gain) as the seed node in the current iteration. We optimize our algorithm using the thought of CELF Algorithm, which avoids the re-computation of $u.mg$ for each node u in repeated iterations.

As detailed in Algorithm 2, the input of the CIMA algorithm is the attributed network $\mathcal{G}$, the number of seed nodes k and the influence strengths between pairs of nodes X_{uv}, and the output is seed node set $\mathcal{S}$ with size of k. The algorithm first does some initiation (lines 1-5). In Line 6, the algorithm employ Louvain algorithm to perform community detection and obtain community structure C. In Line 7, the algorithm generating candidate community C'. From Line 8 to Line 10, the algorithm invokes Algorithm 3 to compute the influence set of each node. From Line 11 to Line 26, the algorithm utilizes submodularity property to find the current seed with the greatest influence marginal gain by multiple iterations. In the first iteration, the influence spread of each node u is calculated and regarded as the marginal gain of u, then the record of u is added to $\mathcal{H}$ in descending order w.r.t. the value of $u.mg$ (lines 11-15). Then, in following each iteration, for the first node u in $\mathcal{H}$, the algorithm examines whether $u.mg$ is last computed in the current iteration with a counter of iteration times, $u.flag$. If yes, because of the submodularity of function $\sigma(\cdot)$, u is the node with the greatest influence marginal gain, and is selected as the seed in current iteration (lines 16-20). Otherwise, the influence marginal gain $u.mg$ will be recomputed, the counter of iteration times, $u.flag$ will be updated, and $\mathcal{H}$ with the new $u.mg$ added into it, will be resorted (lines 21-26). When $|\mathcal{S}| = k$, the algorithm ends and returns the set of influential seed nodes $\mathcal{S}$.

Algorithm 3 describes how influence set of each node is obtained. The input of this algorithm is node u and the threshold θ, through which we remove insignificant influence paths and reduce search space of seed nodes effectively: if $pp(u, v) < \theta$, v is assumed not to be activated by u. The output is the influence set of node u, i.e. $IS(u)$, composed of a set of dual tuples. In the course of obtaining the influence path of node u, a queue $\mathcal{Q}$ is used to traverse and store the influence path of node u. The threshold θ can guarantee influence paths are not too long, and is an important indicator to balance the efficiency and accuracy of algorithm execution. Target node v and the probability of v being activated by u are stored as a dual tuple in the influence set of node u.

Algorithm 3 ObtInfSet (u, θ).

Input: u, θ
Output: the influence set $IS(u)$.
1: $\mathcal{Q} \leftarrow \varnothing$; $\mathcal{IS}(u) \leftarrow \varnothing$;
2: **for** $v \in N^{out}(u)$ **do**
3: $PS(u, v) \leftarrow \varnothing$;
4: $\mathcal{Q}.push(P_{u,v})$;
5: **end for**
6: **while** $\mathcal{Q} \neq \varnothing$ **do**
7: $PA \leftarrow \mathcal{Q}.front()$;
8: $PS(u, v) \leftarrow PS(u, v) \cup PA$;
9: **for each** $w \in N^{out}(v)$ **do**
10: **if** $w \notin PA$ and $pp(PA) \cdot pp(v, w) \geq \theta$ **then**
11: $\mathcal{Q}.push(PA \cup \{w\})$;
12: **end if**
13: **end for**
14: **end while**
15: **for** v in $PS(u, v)$ **do**
16: $ap(v|u) = 1 - \prod_{\forall PA \in PS(u,v)}(1 - pp(PA))$;
17: $IS(u) \leftarrow IS(u) \cup (v, ap(v|u))$;
18: **end for**
19: **return** $IS(u)$;

As detailed in Algorithm 3, the algorithm first initiates the queue $\mathcal{Q}$ and the set $\mathcal{IS}(u)$ (line 1). Then initiates u's path set $PS(u, v)$ and push all paths from u to v, i.e., $P_{u,v}$, into the queue $\mathcal{Q}$ (lines 2-5). From Line 6 to Line 14, the algorithm obtain all the influence paths from node u to target node v ($v \in \mathcal{V}$) using Breadth-First Search and a queue $\mathcal{Q}$. Then, the algorithm proceeds to compute probabilities that u activates v when there is a path from u to v, and updates influence set $\mathcal{IS}(u)$ (lines 15-18). And finally, the algorithm returns u's influence set $\mathcal{IS}(u)$.

4.6 Complexity analysis

In this section, we analyze the time complexity of our CIMA Algorithm. Our CIMA Algorithm mainly includes five parts: community detection, candidate community generation, modeling influence strength,influence set acquisition and seed node selection. Firstly, the time complexity of community detection is $O(m)$, where $m = |\mathcal{E}|$.

Secondly, the time complexity of candidate community generation is $O(n + n_j^2 F)$, where n_j is the total number of nodes in the largest community.

Thirdly, the time complexity of modeling influence strength is $O(n^3 + n^2 F)$.

Table 2 Experiment datasets

dataset	NetPHY	NetHEPT	Amazon	DBLP
# of nodes	37K	15K	335K	317K
# of edges	174K	31K	926K	1M
max community size	88	660	1763	14121
average community size	4.62	6.73	27.2	27.85
min community size	1	1	1	1
communities	8034	2262	12326	119952
max degree	178	64	290	355
min degree	9.38	4.12	4.34	6.81

Fourthly, computing the influence set of all nodes includes acquiring all influence paths and obtaining the influence set of all nodes, the time complexity of which are $O(n+m)$ and $O(nd)$ respectively, where d denotes the average degree of influence paths under the threshold θ. Thus, the time complexity of computing the influence set of all nodes is $O(n+m+nd)$.

Fifthly, in the process to find the current seed with the greatest influence marginal gain by multiple iterations, it takes $O(nd_{max}+n\log n)$ time in the first iteration, and takes $O(k^2 d_{max})$ time to find the current seed with the greatest influence marginal gain in the other iterations, where k denotes the seed node set size and d_{max} represents the maximum degree of nodes. Thus, the time complexity of the seed node selection is $O(nd_{max}+n\log n+k^2 d_{max})$.

In summary, for the number of Monte Carlo simulations, i.e., r, the time complexity of the CIMA Algorithm is $O(m+n_j^2 F+n^3+n^2 F+n+m+nd+r(nd_{max}+n\log n+k^2 d_{max}))$. It seems that the time complexity of our algorithm is not superior, but in fact, in the practical execution, most of time is expended on Monte Carlo simulations (verified by previous work [20]). Whereas, our algorithm can reduce thousands of Monte Carlo simulations through the process of candidate community generation and the computation method of influence set. Thus, it can still obtain superior experimental performance (You can see it in the following section).

5 Evaluation

5.1 Data set

The datasets are described in Table 2. Two medium-sized datasets (NetHEPT, NetPHY) are downloaded from the website[1], provided by Chen et al in [5]. And the larger two datasets (Amazon, DBLP) are downloaded from the SNAP website[2], maintained by Jure Leskovec.

The dataset NetPHY is provided by an electronic printing arXiv platform[3] which is a collaborative relationship between scholars in the field of physics.

The dataset NetHEPT is also provided by the arXiv platform, which includes collaborations between scholars in the field of high energy physics.

The dataset Amazon is available on Stanford University's SNAP[4] and the dataset is derived from the search data of the shopping site Amazon[5].

The data set DBLP is provided by the online scientific research platform DBLP[6]in the field of computer science, which provides a collaborative relationship between scholars.

5.2 Baselines

We evaluate the performance of our CIMA compared with several state-of-art baselines, including a greedy-based influence maximization algorithm IMM [10]) and three community-base influence maximization CIM [13], CINEMA [14], CoFIM [22].

IMM (Influence Maximization via Martingales) is a martingale-based influence maximization algorithm proposed by Tang et al. [10], which is an extension of the TIM+ algorithm [9]. By using a classical statistical tool, martingale, IMM provides much higher efficiency in practice because it requires generating a much smaller number of RR sets than TIM+ algorithm. In our experiment, the parameter ε of IMM algorithm is set as the best value in the original paper, i.e. $\varepsilon = 0.1$.

CIM (Community-based Influence Maximization) proposed in [13], it is an algorithm that uses the heat diffusion

[1] http://research.microsoft.com/enus/people/weic/graphdata.zip

[2] http://snap.stanford.edu/data/

[3] http://www.arxiv.org/

[4] http://snap.stanford.edu/data/

[5] http://www.amazon.com/

[6] http://dblp.uni-trier.de/

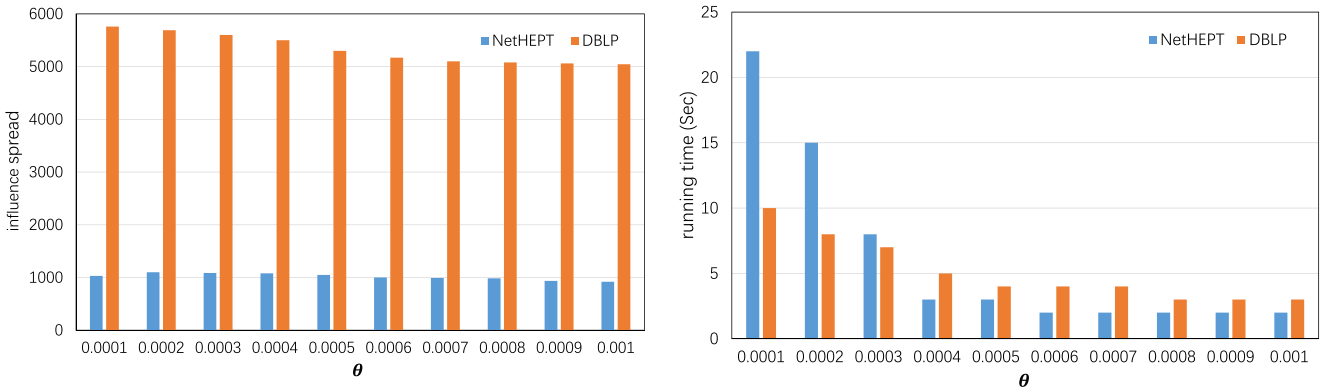

Fig. 3 Experiments on our algorithm when varying influence path threshold θ

model to reduce the overhead incurred in computing influence spreads, utilizes the clustering phenomenon among nodes in a community to avoid the computation of overlapped influence spreads among nodes in the same community.

CINEMA (Conformity-aware INfluEnce MAximization) proposed in [14], it is an algorithm to leverage on interplay between influence and conformity in obtaining the influence probabilities of nodes.

CoFIM (Community-based Framework for Influence Maximization) proposed in [22], it is an algorithm that proposes an influence propagation process, which is constituted of two phases: (i) seeds expansion; and (ii) intra-community propagation, and approximates the influence spread within communities with Bernoulli distribution and the formula $1-(1-x)^n \simeq nx$.

5.3 Experiment results

For evaluating the performance of influence maximization algorithms, we set influence spread and running time as evaluation metrics, which are two extensively adopted metrics in related work. We evaluate the performance of the algorithms from both accuracy and efficiency. In addition, we analyze the impact of influence path threshold θ to achieve the optimal performance of the algorithm.

First, we test the optimal value of the influence path threshold θ in CIMA Algorithm, which guarantees the effect of the node on the influence path. By varying the threshold θ within the range [0.0001, 0.001], we execute experiments on the Dataset DBLP and NetHEPT, which are with the largest number of edges and with the smallest number of edges respectively, as the number of edges is the very important parameter to characterize the influence diffusion. Here we set the seed nodes set size k=50. The impact of influence path threshold θ on Dataset DBLP and NetHEPT is shown in Fig. 3. The similar results can be obtained on Dataset NetPHY and Amazon.

From Fig. 3a , it can be seen that the influence spread of the CIMA seed node set under different thresholds does not change much, that is, the threshold θ within the range of [0.0001, 0.001] has no significant impact on the influence spread of influential seed nodes. As can be seen from the Fig. 3b, when $\theta < 0.0004$, the algorithm runs for a long time. When the threshold reaches 0.0004, the running time of the algorithm is relatively stable. From the above analysis, we can see that when threshold $\theta = 0.0004$, the performance of CIMA Algorithm is best relatively. Therefore, in the following comparison experiments, the influence path threshold θ of CIMA Algorithm is set to 0.0004.

Second, we compare influence spread of our CIMA and the baselines with the seed node set size k varying from 1 to 50. The results are shown in Fig. 4. From Fig. 4, we can see that for the four datasets, the influence spread of CIMA is significantly higher than other algorithms. The CIM algorithm shows the worst performance on all the networks. The difference of performance between our CIMA and the baselines indicates that taking advantage of three factors including social interaction strength, topology structure similarity and attribute similarity between nodes can significantly improve the accuracy of prediction. We also can observe that our algorithm shows its robustness across different values of k.

Furthermore, we compare running time of our CIMA and the baselines with the seed node set size k fixed as 50. And the experimental results are shown in Fig. 5. Figure 5a presents the running time of the five algorithms on the two smaller datasets (NetHEPT and NetPHY). Among the five algorithms, our CIMA and the CoFIM have the highest time efficiency, and the IMM and CINEMA have lower time efficiency. However, due to the smaller data

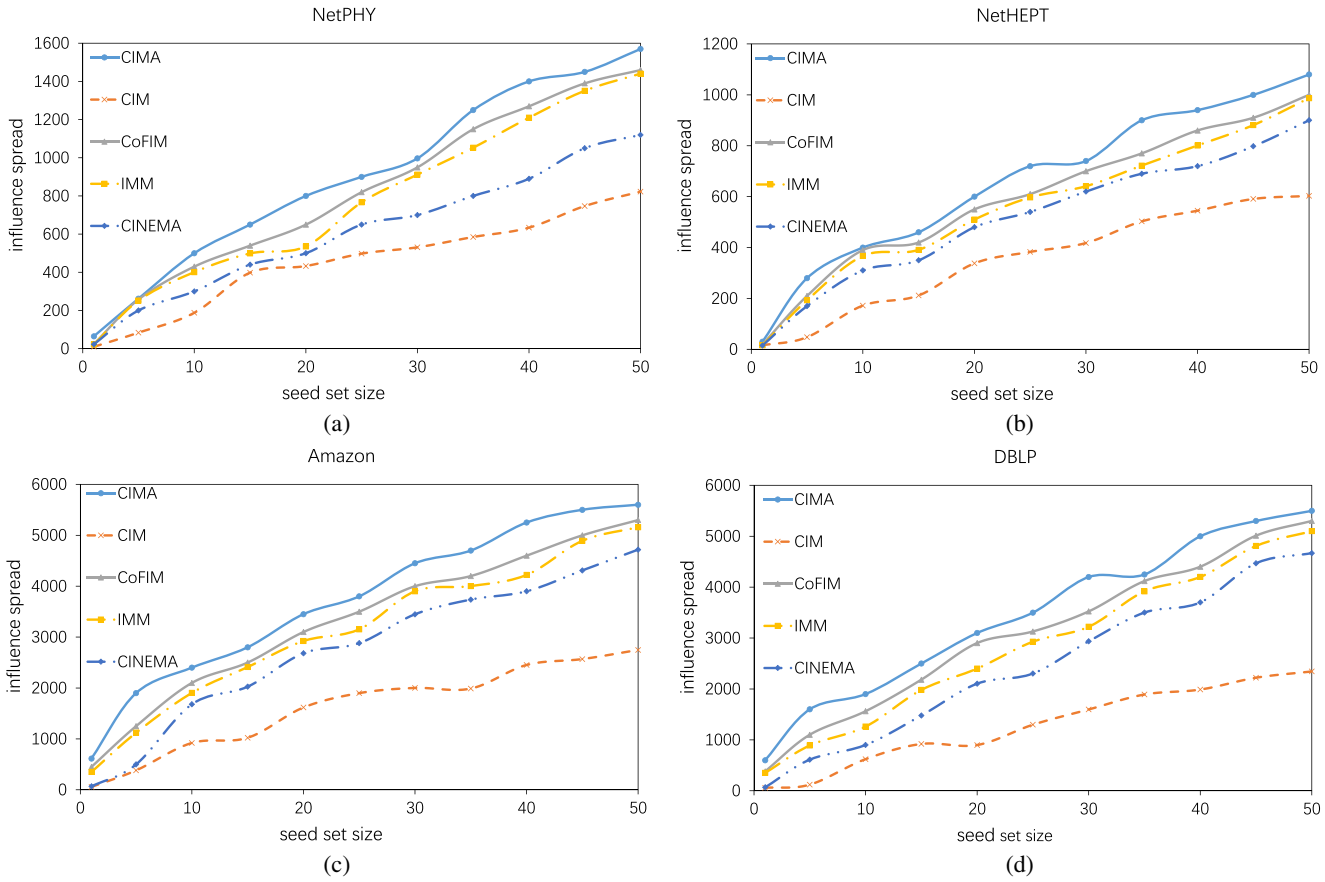

Fig. 4 Comparison experiments of influence spread

set, the difference between the fastest running time and the slowest running time is smaller (3.3 times for NetPHY and 2 times for NetHEPT). Figure 5b presents the running time of the five algorithms on the two larger datasets (Amazon and DBLP). From Fig. 5b, it can be seen that running time of CIMA is close to that of CoFIM, which are with high time efficiency. This is due to that our CIMA is equipped with a process of candidate community generation as well as a efficient computation method of influence set to reduce the search spaces of seed selection and to avoid tens of thousands of Monte Carlo simulations respectively. Algorithm IMM and CINEMA are less time efficient on the

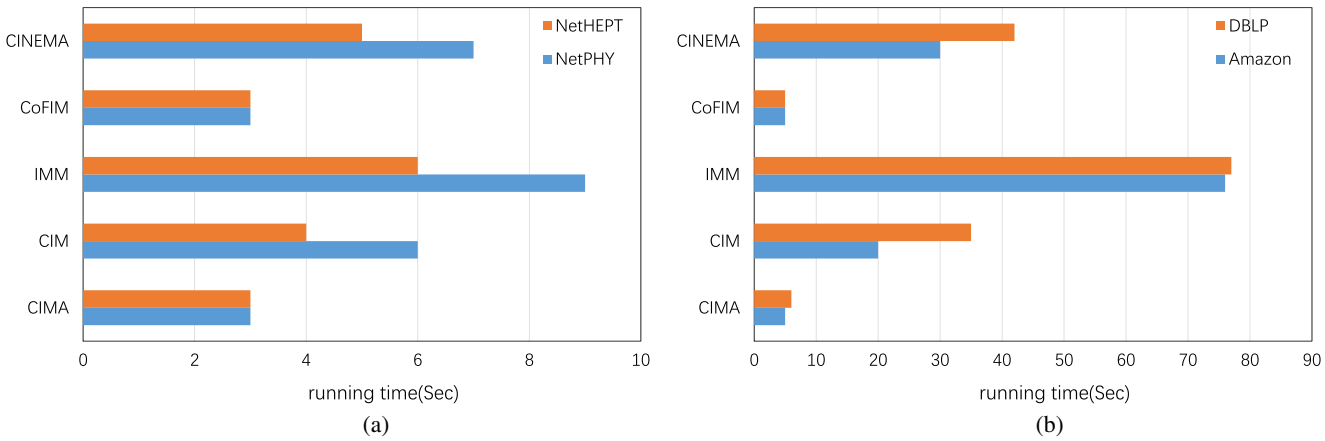

Fig. 5 Comparison experiments of running time

datasets Amazon and DBLP. Specially, the running time of IMM is about 15 times that of our CIMA on dataset Amazon (Fig. 5).

6 Conclusions

We have studied the problem of community-based influence maximization in attributed networks. To tackle the problem, we have proposed a new algorithm CIMA which consists of three main steps: community detection, candidate community generation and seed node selection. To improve the prediction accuracy of influence spread, we propose a model to predict influence strength between nodes in attributed network. To reduce the search spaces of seed selection, we propose a process of candidate community generation. To avoid tens of thousands of Monte Carlo simulations and ultimately make the algorithm more efficient, we propose a computation method of influence set. Experimental results on four publicly available datasets demonstrate the effectiveness and efficiency of the proposed algorithm. As to future work, we plan to extend our CIMA algorithm under Linear Threshold Model. In addition, we plan to explore influence maximization problem w.r.t. dynamic evolution of attributed networks.

Acknowledgments This work is supported by National Key R & D Program of China Project #2017YFB0203201 and Australian Research Council Discovery Project DP150104871.

References

1. Domingos P, Richardson M (2001) Mining the network value of customers. In: Proc SIGKDD, San Francisco, pp 57–66
2. Richardson M, Domingos P (2002) Mining knowledge-sharing sites for viral marketing. In: Proc SIGKDD, Edmonton, Alberta, pp 61–70
3. Kempe D, Kleinberg J, Tardos E (2003) Maximizing the spread of influence through a social network. In: Proc SIGKDD, Washington, pp 137–146
4. Leskovec J et al (2007) Cost-effective outbreak detection in networks. In: Proc SIGKDD, San Jose, pp 420-429
5. Chen W, Wang Y, Yang S (2009) Efficient influence maximization in social networks. In: Proc SIGKDD, Paris, pp 199–208
6. Cheng S, Shen H, Huang J (2013) Staticgreedy: solving the scalability-accuracy dilemma in influence maximization. In: Proc CIKM, San Francisco, pp 509–518
7. Goyal A, Lu W, Lakshmanan LV (2011) Celf++: optimizing the greedy algorithm for influence maximization in social networks. In: Proc WWW, Hyderabad, India, pp 47–48
8. Galhotra S, Arora A, Roy S (2016) Holistic influence maximization: Combining scalability and efficiency with opinion-aware models. In Proc SIGMOD, San Francisco, pp 743–758
9. Tang Y, Xiao X, Shi Y (2014) Influence maximization: near-optimal time complexity meets practical efficiency. In: Proc SIGMOD, Snowbird, pp 75–86
10. Tang Y, Shi Y, Xiao X (2015) Influence maximization in near-linear time: a martingale approach. In: Proc SIGMOD, Melbourne, pp 1539-1554
11. Luo ZL, Cai WD, Li YJ, Peng D (2012) A pagerank-based heuristic algorithm for influence maximization in the social networks. In: Proc RPDEIT, pp 485–490
12. Cao T, Wu X, Wang S, Hu X (2010) Oasnet: an optimal allocation approach to influence maximization in modular social networks. In: Proc SAC, Sierre, Switzerland, pp 1088–1094
13. Chen YC, Zhu WY, Lee WC, Lee SY (2014) Cim:, communitybased influence maximization in social networks. IEEE Trans Intell Syst Technol 5(2):25:1–25:31. https://doi.org/10.1145/2532549
14. Li H, Bhowmick SS, Sun A, Cu J (2015) Conformity-aware influence maximization in online social networks. VLDB J 24:117–141. https://doi.org/10.1007/s00778-014-0366-x
15. Wang Y, Cong G, Song G, Xie K (2010) Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In: Proc SIGKDD, Washington, pp 1039–1048
16. Chen W, Wang C, Wang Y (2010) Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: Proc SIGKDD, pp 1029–1038
17. Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. J Stat Mech Theory Exp 2008(10):P10008
18. Meng Z, Shen H (2018) Dissimilarity-constrained node attribute coverage diversification for novelty-enhanced top-k search in large attributed networks. Knowl-Based Syst 150:85–94
19. Tang J, Wu S, Sun J (2013) Confluence: conformity influence in large social networks. In: Proceddings of SIGKDD, pp 347–355
20. Chen W, Lakshmanan LVS, Castillo C (2013) Information and influence propagation in social networks California: Morgan & Claypool publishers
21. McPherson M, Smith-Lovin L, Cook JM (2001) Birds of a feather: homophily in social networks. Ann Rev Sociol 274:15–444
22. Shang J, Zhou S, Li X, Liu L, Wu H (2017) CoFIM: A community-based framework for influence maximization on large-scale networks. Knowl-Based Syst 117:88–100
23. Shang J, Liu L, Li X et al (2016) Targetd revision: a learning-basd approach for incremental community detection in dynamic networks. Physica A:, Statistical Mechanics and its Applications 443:70–85
24. Bi J, Zhang C (2018) An empirical comparison on state-of-the-art multi-class imbalance learning algorithms and a new diversified ensemble learning scheme. Knowl-Based Syst 158:81–93
25. Zhang X, Zhu J, Wang Q, Zhao H (2013) Identifying influential nodes in complex networks with community structure. Knowl-Based Syst 42:74–84