Computational Social Networks
a SpringerOpen Journal

RESEARCH Open Access

Influence-based community partition for social networks

Zaixin Lu[1*], Yuqing Zhu[2], Wei Li[1,3], Weili Wu[2] and Xiuzhen Cheng[4]

*Correspondence: luz@tsu.edu
[1] NSF Center for Research on Complex Networks, Texas Southern University, 3100 Cleburne Street, Houston, TX 77004, USA
Full list of author information is available at the end of the article

Abstract

Background/Purpose: Community partition is of great importance in sociology, biology and computer science. Due to the exponentially increasing amount of social network applications, a fast and accurate method is necessary for community partition in social networks. In view of this, we investigate the social community partition problem from the perspective of influence propagation, which is one of the most important features of social communication.

Methods: We formulate social community partition as a combinatorial optimization problem that aims at partitioning a social network into K disjoint communities such that the sum of influence propagation within each community is maximized. When $K = 2$ we develop an optimal algorithm that has a provable performance guarantee for a class of influence propagation models. For general K, we prove that it is $\mathcal{NP}$-hard to find a maximum partition for social networks in the well-known linear threshold and independent cascade models. To get near-optimal solutions, we develop a greedy algorithm based on the optimal algorithm. We also develop a heuristic algorithm with a low computational complexity for large social networks.

Results: To evaluate the practical efficiency of our algorithms, we do a simulation study based on real world scenarios. The experiments are conducted on three real-world social networks, and the experimental results show that more accurate partitions according to influence propagation can be obtained using our algorithms rather than using some classic community partition algorithms.

Conclusions: In this study, we investigate the community partition problem in social networks. It is formulated as an optimization problem and investigated both theoretically and practically. The results can be applied to find communities in social networks and are also useful for the influence propagation problem in social networks.

Keywords: Influence propagation; Community partition; $\mathcal{NP}$-hard

Background

Motivation

Social network is an interdisciplinary research area which has attracted a lot of attention in recent years. One important problem in social networks is community partition that provides the insight of the relationships and attributes of the users that a social network comprises. Generally, a social network can be modeled as a graph in which the nodes represent the users and the edges represent the relationships among the users. The objective of community partition is to cluster the users into groups according to their graph topology [1-8]. Another important problem in social networks is influence propagation. It is one of the most important features of social communication and plays a significant role in a variety of affairs such as diffusion of medical innovations and popularization of new technologies. For example, the influence maximization problem, with the objective of finding a small set of users in a social network as seeds to trigger a large influence propagation, has wide applications in viral marketing [9-13].

Due to the nondeterminacy of human behaviors, the influence propagation is mostly studied in probabilistic models such as the Linear Threshold (LT) model and Independent Cascade (IC) model [14-16], that is, the behaviors and decisions of users are uncertain and depend on the behaviors of others. For example, a user's adoption of a new product may have impacts on their friends, whose adoptions may further influence others. Therefore, probabilistic models are more suitable than deterministic models for simulating an influence propagation in social networks. Unfortunately, one important issue however is that the expected influence propagation through the entire social network is hard to estimate for most probabilistic models such as LT and IC [15,16]. Therefore, many works (e.g., [15-17]) construct a local area for each user and use the local influence propagation instead of the global one. But in some large social networks, there may be millions of users so that it is impossible to construct local areas for all the users.

There are also many works studying community-based algorithms for influence maximization, assuming that influence propagates rarely across different communities. However, based on our observation, there are few works done on community partition aiming specially at influence propagation in social networks. The performance of community-based algorithms cannot be guaranteed unless there exists an accurate influence-based community partition. In this paper, we investigate the problem inherent in the question that how to partition a social network into disjoint communities in terms of influence propagation. We believe this study is useful for the influence maximization problem and possibly activates further research and potential applications of community in social networks.

Related work

Community partition is of great importance not only for social networks but also for areas such as computer networks and biology networks. There are lots of works done on community partition in general networks (e.g., [6,8,18,19]), and much effort has been devoted to formalizing the intuition that a community is a set of nodes having more connections with each other while fewer connections with the remainder of the network. The first investigation for community partition were done by Weiss et al. [20]. For subsequent approaches, there are mainly four categories: hierarchy-based methods [1,2], spectrum-based methods [3,4], density-based methods [5] and modularity-based methods [6-8,21-29]. Particularly, Newman's notion of modularity [6,8], which considers the internal connectivity with reference to a randomized model, has been a very popular measure for community partition in general networks. In spite of the excellent performance on many real-world networks, this family of approaches usually has 'resolution limit' problems, i.e., modularity-based methods favor larger communities and fail to discover communities of small sizes [25,30]. Therefore some works investigate new methods for detecting communities, such as the self-reference methods and the comparative methods [18]. In addition, in [19], Hu et al. proposed an algorithm from the node's point of view to incorporate nodes into a community with the largest attractive force. In [31], Zhang et al. proposed an algorithm from the aspect of combinatorial optimization to partition nodes into disjoint parts. There are also many works which view communities from different perspectives. To learn more about the large body of works in community partition, please refer to [29,32-37].

Besides community partition, influence propagation is also an important issue in social networks. Domingos and Richardson in [13] and [12] first proposed general descriptive models for influence propagation in social networks. In [14], Kempe et al. formulated the influence propagation as an optimization problem, namely, influence maximization. They proved that the greedy algorithm has a provable performance guarantee for the LT and IC models. However, how to evaluate the expected influence propagation for selecting the nodes with the maximum marginal gain was left as an open problem, and the greedy algorithm in [14] was implemented by Monte Carlo (MC) simulation. After that many researchers started to investigate how to compute the influence propagation efficiently and a large volume of methods (e.g., [15,16,38]) have been proposed for the LT and IC models. Meanwhile, there are also many works investigating new influence propagation models (e.g., [39,40]) to approach the real-world scenarios.

Due to the nature of the communities, applying the research of community partition into influence propagation is promising. In [17], Wang et al. proposed a community-based greedy algorithm for mining the most influential nodes. In [41], Li et al. further proposed an algorithm for influence maximization in online social networks. They assume that each node's influence propagation is limited to the community it resides and thus they evaluate the influence propagation within each community to improve the computational efficiency. There are also many works for influence propagation or other social network applications taking the advantage of community structures (please see e.g., [42-45] for recent works).

Our contribution

Although there are a lot of works done on general community partition, based on our observation, there are few works done on community partition for influence propagation. In view of this, we investigate how to partition a social network into communities according to influence propagation. Our main contributions are as follows:

1. We formally define the influence-based community partition problem as a combinatorial optimization problem with the objective of partitioning a social network into K disjoint communities such that the sum of influence propagation within each community is maximized. We call the problem Maximum K-Community Partition (MKCP). The motivation is to keep as much influence propagation as possible after the partition and reduce the estimation errors caused using local influence propagation increased of the global one.
2. When $K = 2$, i.e., partition a social network into two disjoint parts, we develop an optimal algorithm for a class of influence propagation models. For general K, we prove there exists no polynomial time algorithm unless $\mathcal{P} = \mathcal{NP}$ for MKCP in the well-known LT and IC models, and a greedy algorithm based on the two partition algorithm is exhibited. We also develop a fast heuristic algorithm with a low computational complexity in case that the social network is very large.

3. We conduct simulation on real-world social networks to demonstrate the practical efficiency of the proposed algorithms. The influence propagation is based on the well-known LT and IC models, and the experimental results show that significantly better partitions can be obtained using our algorithms rather than using some community partition methods that are not specialized for influence propagation.

Paper organization

The rest of this paper is organized as follows. In 'Problem description' section, we give the background information, including the notation and problem definition. In 'Methods' section, we present our algorithms as well as the theoretical analysis of both the proposed algorithms and the MKCP problem. In 'Results and discussion' section, we show the simulation results on some real-world social networks. In 'Conclusions' section, we conclude the paper.

Problem description

In this study, we formulate a social network as a simple directed graph without self-loops, where nodes represent users and edges represent relationships among the users. We first introduce some notations and then present the MKCP problem based on the notations.

1. For a social network G, we denote by $V = \{1, 2, \ldots, n\}$ the set of nodes and $E = \{(i,j)\}$ the set of directed edges. A directed edge (i,j) denotes that there exists a chance of influence propagation between nodes i and j where i is the sender and j is the receiver. For each node $i \in V$, we denote by $p(i)$ $(0 \leq p(i) \leq 1)$ the probability that node i would produce an influence propagation or would share an idea with others through the social network. For example, in the Twitter social network, $p(i)$ should be related to the number of tweets i posts periodically. For each edge $(i,j) \in E$, we denote by $w(i,j)$ the influential degree from node i to node j, which depends on their closeness and the probability $p(i)$ for node i.
2. Let K denote the number of communities. We denote by $c_i \in \{1, 2, \ldots, K\}$ the community identifier of node i. We denote by $C_k = \{i | c_i = k\}$ the set of nodes with community identifier k $(1 \leq k \leq K)$. For each pair of nodes i and j in the same set C_k, we denote by $p_{C_k}(i,j)$ $(0 \leq p_{C_k}(i,j) \leq 1)$ the probability that node j receives the influence from node i through propagation within community C_k.
3. For a community C_k and a node $i \in C_k$, we denote by $\sigma_{C_k}(i)$ the influence propagation of node i within community C_k, i.e., $\sigma_{C_k}(i) = \sum_{j \in (C_k \setminus i)} p_{C_k}(i,j)$. For any nonempty subset $D \subseteq C_k$, we denote by $\sigma_{C_k}(D)$, the sum of influence propagation within community C_k for every node in D, i.e., $\sigma_{C_k}(D) = \sum_{i \in D} \sigma_{C_k}(i)$. For simplicity, we let $\sigma(X)$ denote $\sigma_X(X)$ for community X and in the rest of this paper we call $\sigma(\cdot)$ the influence propagation function for community '$\cdot$'.

The probability that node j receives the influence from node i not only depends on the influential degree $w(i,j)$ but also depends on the network topology and the influence propagation model. For example, in the LT model, the sum of influence node j receives can be formulated as $\sum_{i \in N_{\text{active}}(j)} w(i,j)$ where $N_{\text{active}}(j)$ denotes the set of active nodes around j and $\sum_{i \in N_{\text{active}}(j)} w(i,j) \leq 1$. The influence propagation runs in discrete steps. At any time t, a node $j \in V$ becomes active when $\sum_{i \in N_{\text{active}}(j)} w(i,j) \geq \lambda(j)$ where $\lambda(j)$ is a threshold selected uniformly at random between 0 and 1. Therefore in the LT model, for any community C_k, $p_{C_k}(i,j)$ is the probability that j is eventually active when i is initially active. As an example shown in Figure 1, the numbers on the edges and nodes denote the influential degrees and random thresholds. Assume that all the nodes are in the same community and node u is a seed, then all the white nodes (including node y) can be activated by node u, because they can either be activated by u or by paths from u. All the black nodes (p, q and w) cannot be activated by node u, even though q is a direct outgoing neighbor of u. Therefore in the LT model, $p_{C_k}(i,j)$ not only depends on the influential degree $w(i,j)$. We next present the definitions of K-valid disjoint partition (K-VDP) and the MKCP problem.

Definition 1. (K-*VDP*). Given a graph $G(V,E)$ as a social network, a K*-valid disjoint partition* $\mathcal{P}$ is a collection of K sets $\{C_1,\ C_2,\ \dots,\ C_K\}$ satisfying: (1) $\bigcup_{k=1}^{K}(C_k) = V$ and (2) $\forall i \neq j$, $C_i \cap C_j = \emptyset$.

Let K be an integer no less than 2. According to Definition 1, a K-VDP is a partition of V into K nonempty subsets such that each node is in exact one subset. We denote the influence propagation function for a K-VDP $\{C_1,\ C_2,\ \dots,\ C_K\}$ by $f(C_1,\ C_2,\ \dots,\ C_K) = \sum_{k=1}^{K}\sigma(C_k)$ and we want to maximize $f(C_1,\ C_2,\ \dots,\ C_K)$. The formal definition of MKCP is given in Definition 2.

Definition 2. (*MKCP*). Given a graph G as a social network, an influence propagation model $\mathcal{I}$ (such as IC or LT) and an integer $K \geq 2$, Maximum K-Community Partition (MKCP) is the problem of finding a partition $\mathcal{P} = \{C_1,\ C_2, \dots,\ C_K\}$ of K subsets of nodes,

$$\begin{aligned} \text{maximize} \quad & f(C_1,\ C_2, \dots,\ C_K) = \sum_{k=1}^{K}\sigma(C_k) \qquad (1) \\ \text{subject to} \quad & \{C_1,\ C_2, \dots,\ C_K\} \text{ is a } K\text{-VDP for } G. \end{aligned}$$

Consider the node set V as a single community, we have

$$f(\{V\}) = \sum_{i\in V}\ \sum_{j\in V\setminus\{i\}} p_V(i,j).$$

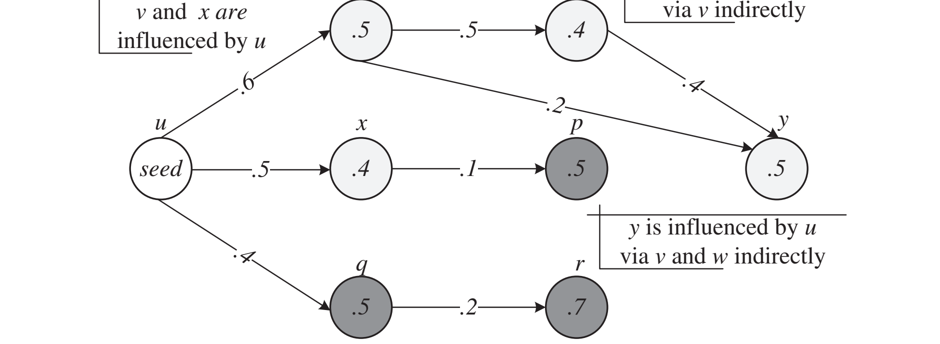
v and x are influenced by u
v
.5
w
.4
w is influenced by u via v indirectly
.6
.5
.2
.4
u
seed
.5
x
.4
.1
p
.5
y
.5
.4
y is influenced by u via v and w indirectly
q
.5
.2
r
.7

Figure 1 An illustration of influence propagation.

It is clear that when partitioning the social network into two or more communities, some pairs (i, j) will be separated and thus both $p_V(i, j)$ and $p_V(j, i)$ have to be removed in the sum of influence propagation. In addition, even though nodes i and j are partitioned into the same community X, $p_X(i, j)$ may be less than $p_V(i, j)$, and $p_X(j, i)$ may be less than $p_V(j, i)$ because X is a subset of V. Therefore, the influential propagation between any pair of nodes i and j is different for different community partitions no matter they are in the same community or not.

Methods

Optimal algorithm for M2CP

In this subsection, we present an optimal algorithm to M2CP for a class of influence propagation models. The algorithm is based on the Min Cut algorithm proposed in [46]. Before giving the formal algorithm and its theoretical analysis, we briefly discuss the difference between the Min Cut problem and the M2CP problem. A min cut of a graph G is a set of edges with the least number of elements (un-weighted case) or the least sum of weights (weighted case) that partitions G into two parts. On this basis, for M2CP, one may want to find a cut to minimize the influence propagation leaking out between the two parts. However, maximizing the sum of influence propagation within each community is not equivalent to minimizing the influence propagation crossing different communities. Figure 2 shows an example. There are eight nodes which are partitioned into two communities $C_1 = \{1, 2, 3, 4\}$ and $C_2 = \{5, 6, 7, 8\}$. Assume the gray-directed arcs are the possible influence propagation. Consider nodes 7, 5, and 1, respectively. It is clear that the influence received by nodes 7 and 5 will decrease after the partition because node 3 cannot influence node 7 and it cannot influence node 5 via node 7 indirectly. The influence received by node 1 also decreases because of the following: (1) node 5 cannot influence node 1, (2) node 7 cannot influence node 1 indirectly, and (3) node 3 cannot influence node 1 through the path ($3 \rightarrow 7 \rightarrow 5 \rightarrow 1$). The first two kinds of influence propagation are between nodes in different communities, but the last one is between nodes in the same community. Therefore, maximizing the sum of influence propagation within each community is not just minimizing the influence propagation crossing different communities.

Given a social network as well as an influence propagation model, our algorithm iteratively finds $n-1$ partitions and selects the one with the maximum value as the final output.

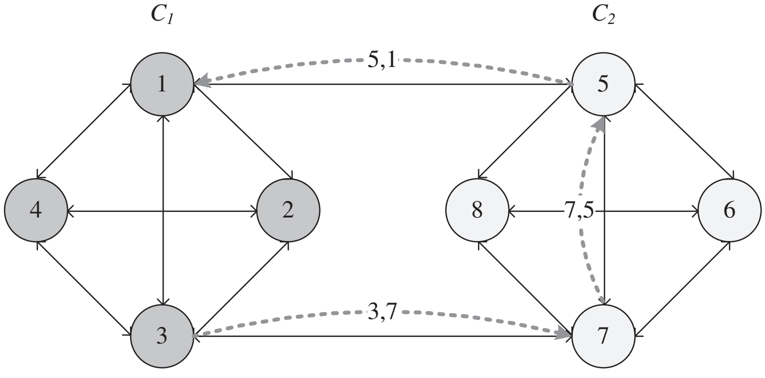

Figure 2 An example of M2CP.

In the beginning, we consider each node i as a single set and let $\mathcal{V} = \{S_1, S_2, \ldots, S_n\}$ as the collection of all the sets where $S_i = \{i\}$. Select an arbitrary set $S_i \in \mathcal{V}$ and let $\mathcal{A} = \{S_i\}$. We then add the remainder sets one by one iteratively into $\mathcal{A}$. Each time a set S_j with the maximum value of $\varsigma(\mathcal{A}, S_j)$ is added, where $\varsigma(\mathcal{A}, S_j) = \sigma(\mathcal{A} \cup S_j) - \sigma(S_j)$. When there are only one set S_l left, $\{\nu(\mathcal{A}), \nu(\mathcal{V} \backslash \mathcal{A})\}$ are considered as the first partition where $\nu(\mathcal{X})$ is defined as the set of nodes in $\mathcal{X}$. In addition, the last two sets not in $\mathcal{A}$, say S_r and S_l, are merged as a single set $(S_r \cup S_l)$ for computing the next partition. The algorithm terminates when there are only one set in $\mathcal{V}$. The pseudo-code is given in Algorithm 1.

Algorithm 1 Algorithm for M2CP (AM2CP)

```
Input:  Given a graph $G$ as a social network and an influence propagation model $\mathcal{I}$.
Output:  a 2-VDP for $G$.
1:  construct a collection $\mathcal{V}$ of $n$ sets: $S_1, S_2, \ldots, S_n$, each of which contains a single node
    in graph $G$;
2:  while $|\mathcal{V}| > 1$ do
3:      let $\mathcal{A} = \{\{i\}\}$ where $\{i\}$ is an arbitrary set in $\mathcal{V}$;
4:      while $|\mathcal{V}| - |\mathcal{A}| > 1$ do
5:          let $S_j \leftarrow \mathrm{argmax}_{S_z \in \mathcal{V} \backslash \mathcal{A}}(\varsigma(\mathcal{A}, S_z))$;
6:          add $S_j$ into $A$;
7:      end while
8:      let $\mathcal{P} \leftarrow (\nu(\mathcal{A}), \nu(\mathcal{V} \backslash \mathcal{A}))$.
9:      let $\mathcal{P}_{\max}$ store the partition with the maximum objective value$f(\nu(\mathcal{A}), \nu(\mathcal{V} \backslash \mathcal{A}))$;
10:     let $S_{r,l}$ to be the union of last two sets $S_r$ and $S_l$ in $(\mathcal{V} \backslash \mathcal{A})$;
11:     delete $S_r$ and $S_l$ from $\mathcal{V}$ and add $S_{r,l}$ into $\mathcal{V}$;
12: end while
13: return $\mathcal{P}_{\max}$;
```

The computational complexity of AM2CP (Algorithm 1) depends on the time complexity of computing $\sigma(\cdot)$, which further depends on the time complexity of computing the influence propagation $p_{C_k}(i,j)$ for community C_k and all the pairs (i,j) of nodes in it. In [15], Chen et al. prove that it is $\#\mathcal{P}$-hard to compute the exact influence propagation in LT and IC models. Therefore, in this work, $p_{C_k}(i,j)$ is estimated by MC simulation. Assume we have a simulator to estimate $\sigma(\cdot)$ in τ time. Following Algorithm 1, we run steps (3 to 11) $n-1$ times for the $n-1$ partitions. For each partition, we add all the sets greedily into $\mathcal{A}$ that calls the function $\sigma(\cdot)$ $\mathcal{O}(n^2)$ times. Therefore, the overall running time of AM2CP is $\mathcal{O}(n^3\tau)$.

We next show that AM2CP is an optimal solution for M2CP when the community influence propagation function $\sigma(\cdot)$ is super-modular. Let S be a finite set. A function $f : 2^S \rightarrow R$ is super-modular if for any $B \subset A \subset S$ and $u \notin A$,

$$\sigma(A \cup \{u\}) - \sigma(A) \geq \sigma(B \cup \{u\}) - \sigma(B), \tag{2}$$

or equivalently for any $B, A \subset S$,

$$\sigma(A \cup B) + \sigma(A \cap B) \geq \sigma(A) + \sigma(B). \tag{3}$$

Theorem 1. *If the influence propagation function $\sigma(\cdot)$ is super-modular, AM2CP is an optimal solution for M2CP.*

Proof. Based on AM2CP, each time we find a partition $\mathcal{P} = (v(\mathcal{A}), v(\mathcal{V}\backslash\mathcal{A}))$ that separates the last two sets S_r and S_l, and we merge the two sets for the next round. To show Theorem 1, it is sufficient to show that $\mathcal{P}$ has the maximum objective function value $\sigma(v(\mathcal{A})) + \sigma(v(\mathcal{V}\backslash\mathcal{A}))$ among all the partitions separating S_r and S_l, where $v(\mathcal{X})$ is the set of nodes in $\mathcal{X}$. We prove it by induction.

Without loss of generality, we assume the sets added into $\mathcal{A}$ are in the order: $S_{i_1}, S_{i_2}, \ldots, S_{i_{|\mathcal{V}|}}$ for round i and let $\mathcal{A}_{i_j}$ denote the collection of the first j sets added into $\mathcal{A}$ in round i. Then for any $\mathcal{S} \subseteq \mathcal{A}_{i_1}$ and S_{i_j} with $j > 2$, we have $\sigma(v(\mathcal{A}_{i_2})) + \sigma(S_{i_j}) \geq \sigma(v(\mathcal{A}_{i_2}\backslash\mathcal{S})) + \sigma(S_{i_j} \cup v(\mathcal{S}))$ because $v(\mathcal{S})$ is either S_{i_1} or $\emptyset$. Assume $\sigma(v(\mathcal{A}_{i_{k'}})) + \sigma(S_{i_j}) \geq \sigma(v(\mathcal{A}_{i_{k'}}\backslash\mathcal{S})) + \sigma(S_{i_j} \cup v(\mathcal{S}))$ for any $2 \leq k' < k$, $\mathcal{S} \subseteq \mathcal{A}_{i_{k'-1}}$ and S_{i_j} with $j > k'$. We next show that $\sigma(v(\mathcal{A}_{i_k})) + \sigma(S_{i_j}) \geq \sigma(v(\mathcal{A}_{i_k}\backslash\mathcal{S})) + \sigma(S_{i_j} \cup v(\mathcal{S}))$ for any $\mathcal{S} \subseteq \mathcal{A}_{i_{k-1}}$ and S_{i_j} with $j > k$.

Consider the following two cases: (1) $S_{i_{k-1}} \in \mathcal{S}$ and (2) $S_{i_{k-1}} \notin \mathcal{S}$. When $S_{i_{k-1}} \notin \mathcal{S}$, we have $\sigma(v(\mathcal{A}_{i_{k-2}})) + \sigma(S_{i_j}) \geq \sigma(v(\mathcal{A}_{i_{k-2}}\backslash\mathcal{S})) + \sigma(S_{i_j} \cup v(\mathcal{S}))$ due to the assumption. Therefore, $\sigma(v(\mathcal{A}_{i_k})) + \sigma(S_{i_j}) \geq \sigma(v(\mathcal{A}_{i_k}\backslash\mathcal{S})) + \sigma(S_{i_j} \cup v(\mathcal{S}))$ because (1) $v(\mathcal{A}_{i_k}) = v(\mathcal{A}_{i_k}\backslash\mathcal{S}) \cup v(\mathcal{A}_{i_{k-2}})$, (2) $v(\mathcal{A}_{i_{k-2}}\backslash\mathcal{S}) = v(\mathcal{A}_{i_k}\backslash\mathcal{S}) \cap v(\mathcal{A}_{i_{k-2}})$ and (3) $\sigma(\cdot)$ is super-modular.

When $S_{i_{k-1}} \in \mathcal{S}$, we have $\sigma(v(\mathcal{A}_{i_{k-1}})) + \sigma(S_{i_k}) \geq \sigma(v(\mathcal{S})) + \sigma(S_{i_k} \cup v(\mathcal{A}_{i_{k-1}}\backslash\mathcal{S}))$ due to the assumption in which $\sigma(v(\mathcal{S})) = \sigma(v(\mathcal{A}_{i_{k-1}})\backslash v(\mathcal{A}_{i_{k-1}}\backslash\mathcal{S}))$. Since $\sigma(\cdot)$ is super-modular, we have $\sigma(v(\mathcal{A}_{i_{k-1}}) \cup S_{i_j}) - \sigma(v(\mathcal{A}_{i_{k-1}})) \geq \sigma(v(\mathcal{S}) \cup S_{i_j}) - \sigma(v(\mathcal{S}))$. In sum, we have $\sigma(v(\mathcal{A}_{i_k}\backslash\mathcal{S})) + \sigma(S_{i_j} \cup v(\mathcal{S})) \leq \sigma(v(\mathcal{A}_{i_{k-1}}) \cup S_{i_j}) + \sigma(S_{i_k})$. In addition we have $\sigma(v(\mathcal{A}_{i_{k-1}}) \cup S_{i_j}) + \sigma(S_{i_k}) \leq \sigma(v(\mathcal{A}_{i_k})) + \sigma(S_{i_j})$ because in AM2CP, $S_{i_k} = \text{argmax}_{S_z \in \mathcal{V}\backslash\mathcal{A}_{i_{k-1}}}(\sigma(\mathcal{A}_{i_{k-1}} \cup S_z) - \sigma(S_z))$. Therefore in both cases, we have $\sigma(v(\mathcal{A}_{i_k})) + \sigma(S_{i_j}) \geq \sigma(v(\mathcal{A}_{i_k}\backslash\mathcal{S})) + \sigma(S_{i_j} \cup v(\mathcal{S}))$. By induction, we have $\sigma(v(\mathcal{A}_{i_{|\mathcal{V}|-1}})) + \sigma(S_{i_{|\mathcal{V}|}}) > \sigma(v(\mathcal{A}_{i_{|\mathcal{V}|-1}}\backslash\mathcal{S})) + \sigma(S_{i_{|\mathcal{V}|}} \cup v(\mathcal{S}))$ for any $\mathcal{S} \subseteq \mathcal{A}_{i_{|\mathcal{V}|-2}}$. Therefore, the partition $\mathcal{P}$ of each round i in AM2CP has the maximum objective function value among all the partitions separating the last two sets. Each time we compare $\mathcal{P}$ with $\mathcal{P}_{\max}$ and merge the last two sets. Therefore $\mathcal{P}_{\max}$ is an optimal partition for the M2CP problem when the influence propagation function $\sigma(\cdot)$ is super-modular. □

Since AM2CP is an optimal solution if $\sigma(\cdot)$ is super-modular, we are interested in the influence propagation models in which the influence propagation function $\sigma(\cdot)$ is super-modular. Note that $\sigma(\cdot)$, in this paper, is different from the influence function defined in [14]. In this paper $\sigma(X)$ is the sum of influence propagation within X for every node in X, i.e., $\sigma(X) = \sum_{i \in X} \sigma_X(i)$. In [14] $\sigma(X)$ is the influence propagation of seed set X in the entire social network. We show the following lemma.

Lemma 1. *When the influence propagation model is LT, for any two communities: $B \subset A$, and a node $u \notin A$, we have $\sigma(A \cup \{u\}) - \sigma(A) \geq \sigma(B \cup \{u\}) - \sigma(B)$.*

Proof. The influence propagation in the LT model, as shown in [14], can be simulated as a random process by flipping coins. Assume we have flipped all the coins in advance, then an edge is declared to be 'live' if the coin flip indicated an influence will be propagated successfully and it is declared blocked otherwise. A node j is influenced by a seed i if and only if there is a path of live edges from i to j. According to this principle, any simple path from i to j has a certain probability to be a live path. In [15], Chen et al. prove that for any node i, the influence propagation of i is equal to $\sum_{\mathrm{sp} \in \mathrm{SP}(i)} w(\mathrm{sp})$ where $\mathrm{SP}(i)$ is the set of all the simple paths starting from i and $w(\mathrm{sp})$ is the probability that sp is a live path. Therefore, for a community X and a node $i \in X$, $\sigma_X(i) = \sum_{\mathrm{sp} \in \mathrm{SP}_X(i)} w(\mathrm{sp})$ where $\mathrm{SP}_X(i)$ is the set of simple paths starting from i in community X, and $\sigma(X) = \sum_{i \in X} \sigma_X(i)$ is the sum of probabilities for all the simple paths in X. Since for any two communities, $B \subset A$, the set of simple paths in B is a subset of the set of simple paths in A, we have $\sigma(A) \geq \sigma(B)$. Similarly, we have $\sigma(A \cup \{u\}) - \sigma(A) \geq \sigma(B \cup \{u\}) - \sigma(B)$ because $\sigma(A \cup \{u\}) - \sigma(A)$ is the sum of probabilities of simple paths visit u exactly once in community $(A \cup \{u\})$, and $\sigma(B \cup \{u\}) - \sigma(B)$ is the sum of probabilities of simple paths visit u exactly once in community $(B \cup \{u\})$ which is a subset of the former. Therefore, the influence propagation function $\sigma(\cdot)$ in the LT model is super-modular. □

Theorem 2. *AM2CP is an optimal solution for M2CP in the LT model.*

Proof. The theorem follows directly from Theorem 1 and Lemma 1. □

By Lemma 1, we show that $\sigma(\cdot)$ is super-modular in the LT model. We next show that $\sigma(\cdot)$ in the IC model, however, is not super-modular. The description of IC model can be found in detail in [14]. Here we just give a counterexample. As an example shown in Figure 3, the weights are as follows: $w(1,2) = w(1,3) = w(1,4) = 1$ and $w(2,5) = w(3,5) = w(4,5) = 0.5$. According to the edges in Figure 3, nodes 2, 3, and 4 cannot influence each other and nodes 2, 3, 4, and 5 cannot influence node 1. Let community $A = \{1,2,3,5\}$ and community $B = \{1,2,5\}$. So B is a subset of A. By direct computing, we have $\sigma(A \cup \{4\}) - \sigma(A) = 5.375 - 3.75 = 1.625$ and $\sigma(B \cup \{4\}) - \sigma(B) = 3.75 - 2 = 1.75$. Therefore, $\sigma(A \cup \{4\}) - \sigma(A) < \sigma(B \cup \{4\}) - \sigma(B)$ which implies $\sigma(\cdot)$ is not super-modular in the IC model.

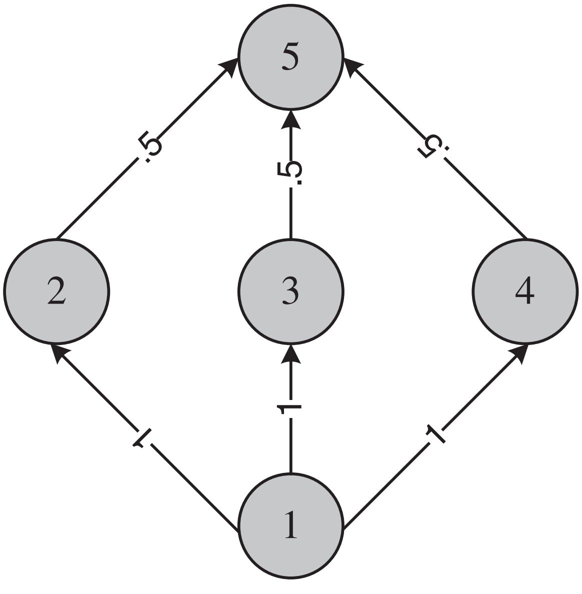

Figure 3 An example of the IC model.

Hardness

In this subsection, we study the hardness of MKCP. We show that the MKCP problem, with arbitrary K, is $\mathcal{NP}$-hard in the LT or IC model.

Theorem 3. *The MKCP problem is $\mathcal{NP}$-hard in the LT model for general K.*

Proof. To prove Theorem 3, we do a polynomial time reduction from the Minimum K-Cut problem. The input of Minimum K-Cut is a simple graph $G(V, E)$ without directions and an integer M. The objective is to find a set of at most M edges which when deleted, separate the graph into exactly K nonempty components. It is well known that the Minimum K-Cut problem is $\mathcal{NP}$-hard for general K.

Given a graph $G(V, E)$ for the Minimum K-Cut problem, we construct a social network $G'(V', E')$ as follows: (1) For each node $i \in V$, create a node i' in V'. (2) For each edge $(i, j) \in E$, create two edges (i', j') and (j', i') in E'. (3) Let Δ denote the maximum degree in G and n denote the number of nodes in G. Assign weight $w(i', j') = \frac{1}{(n\Delta)^2}$ for all the edges $(i', j') \in E'$.

It is clear that the reduction can be done in polynomial time. We next show that there is a K-Cut with M edges if and only if there is a K-VDP $\mathcal{P}$ with $f(\mathcal{P}) \geq \frac{2(|E|-M)}{(n\Delta)^2}$. Assume there is a K-Cut with M edges, then graph G can be partitioned into K communities with $|E| - M$ edges within the K communities. Consider the same partition in G'. The one-hop influence propagation is $\frac{2(|E|-M)}{(n\Delta)^2}$. Therefore, we have a K-VDP $\mathcal{P}$ with $f(\mathcal{P}) > \frac{2(|E|-M)}{(n\Delta)^2}$ for G'. Conversely, assume there is a K-VDP $\mathcal{P}$ for G' with $f(\mathcal{P}) \geq \frac{2(|E|-M)}{(n\Delta)^2}$. It has been shown in [16] that for any nodes $\forall i, j, l \in V$, the probability of influence propagation from i to j via node l is equal to $w(i, l)w(l, j)$ in the LT model. Therefore, a single two-hop influence propagation is $\frac{1}{(n\Delta)^4}$. The number of two-hop simple paths for any node $i' \in V'$ is no more than Δ^2. Therefore, the sum of two-hop influence propagation for every node in V' is no more than $\frac{n\Delta^2}{(n\Delta)^4} = \frac{1}{n^3\Delta^2}$. By direct computing, we have the sum of $(r+1)$-hop influence propagation is less than the sum of r-hop influence propagation for any node i. Since the length of simple paths is no more than n, we have the sum of multi-hop influence propagation for every node in V' is less than $\frac{1}{(n\Delta)^2}$. This implies that $f(\mathcal{P}) \geq \frac{2(|E|-M)}{(n\Delta)^2}$ if and only if the one-hop influence propagation is no less than $\frac{2(|E|-M)}{(n\Delta)^2}$. Therefore, the same partition in G is a K-Cut with at most M edges. In sum, we prove Theorem 3. □

Theorem 4. *The MKCP problem is $\mathcal{NP}$-hard in the IC model for general K.*

Proof. To prove Theorem 4, we can do the same reduction as the one in the proof of Theorem 3, i.e., assign uniform weight $\frac{1}{(n\Delta)^2}$ on all the edges. It can be shown by induction that the sum of $(r+1)$-hop influence propagation a node i received is less than the sum of r-hop influence propagation it received for any node $i \in V'$ in the IC model. Therefore, by a similar argument, we have the sum of multi-hop influence propagation received for every node $i \in V'$ is less than the edge weight. Therefore, there exists a K-Cut with M edges if and only if there is a K-VDP $\mathcal{P}$ with $f(\mathcal{P}) \geq \frac{2(|E|-M)}{(n\Delta)^2}$. □

The proofs of Theorems 3 and 4 are nothing but assign specific weights to make the multi-hop influence propagation negligible. It is intuitive that the general MKCP problem is even harder when multi-hop influence propagation is not negligible.

Heuristic algorithm for M*K*CP

In this subsection, we present two heuristic algorithms for M*K*CP. As mentioned in 'Related work' section in the literature, there are mainly four categories of methods for community partition: hierarchy-based methods, spectrum-based methods, density-based methods, and modularity-based methods. In our point of view, spectrum-based methods, density-based methods, and modularity-based methods are not suitable for M*K*CP. In spectrum-based methods, communities are partitioned by studying the adjacency matrix which cannot reflect the information of influence propagation. In density-based methods, communities are defined as areas of higher density than the remainder of the data set. Therefore, this category of methods requires the location knowledge of nodes which cannot be formulated in our M*K*CP problem. In modularity-based methods, the objective of community partition is only to maximize the global modularity score. Therefore, all the three categories of methods cannot be applied for M*K*CP and we focus on hierarchy-based methods.

Generally speaking, hierarchical community partition is a method to build a hierarchy of communities. There are two strategies for hierarchical partition. One is *split* and the other is *merge*. Split is a top down approach, i.e., all the nodes start within one community, and splits are performed on one of the communities recursively. Conversely, merge is a bottom up approach, i.e., each node starts in a distinct community, and pairs of communities are merged recursively as a new community. For typical hierarchical community partition problems, $n-1$ splits (or respectively merges) have to be done to build a hierarchy where n is the number of nodes. But for the M*K*CP problem, we need only $K-1$ splits or $n-K$ merges respectively to obtain a K-VDP. We will determine the splits and merges in a greedy manner. The Split algorithm runs by calling AM2CP recursively, and each time it partitions a community X into two communities X_1 and X_2 with the minimum value of $\sigma(X) - (\sigma(X_1) + \sigma(X_2))$. The pseudo-code is given in Algorithm 2. The Merge algorithm runs by randomly selecting a community X each time and finding another community Y to maximize the value of $\sigma(X \cup Y) - (\sigma(X) + \sigma(Y))$. The pseudo-code is given in Algorithm 3.

```
Algorithm 2 Split algorithm for MKCP (SAMKCP)
Input: Given a graph $G$ as a social network, an influence propagation model $\mathcal{I}$ and an
    integer $K$.
Output: a $K$-VDP for $G$.
let $\mathcal{P} \leftarrow \{V\}$ ($P$ holds the current communities);
while $|\mathcal{P}| < K$ do
    let $C_{z_1}$ and $C_{z_2} \leftarrow \text{argmin}_{C_z \in \mathcal{P}}(\sigma(C_z) - (\sigma(C_{z_1}) + \sigma(C_{z_2})))$ subject to $C_{z_1} \cap C_{z_2} = \emptyset$
        and $C_{z_1} \cup C_{z_2} = C_z$;
    put $C_{z_1}$ and $C_{z_2}$ into $\mathcal{P}$ and delete $C_z$ from $\mathcal{P}$;
end while
return $\mathcal{P}$;
```

In the general case, the running time of a split with an exhaustive search requires exponential time. However, when $\sigma(\cdot)$ is super-modular, we can apply AM2CP to determine C_{z_1} and C_{z_2} for each C_z which requires only $\mathcal{O}(|C_z|^3\tau)$ time. Now let us consider the computational complexity of SAMKCP (Algorithm 2). To avoid duplicate computations, we can keep the optimal partition for each community in $\mathcal{P}$ and apply AM2CP on both C_{z_1} and C_{z_2} at step 4 to obtain their optimal partitions. Then the overall running time of SAMKCP is $\mathcal{O}(Kn^3\tau)$ when $\sigma(\cdot)$ is super-modular.

Algorithm 3 Merge algorithm for MKCP (MAMKCP)

```
Input: Given a graph $G$ as a social network, an influence propagation model $\mathcal{I}$ and an
    integer $K$.
Output: a $K$-VDP for $G$.
let $\mathcal{P} \leftarrow \{C_1, C_2, \ldots, C_n\}$ where each $C_i = \{i\}$ contains a single node in $G$;
while $|\mathcal{P}| > K$ do
    select a community $C_i \in \mathcal{P}$ randomly;
    let $C_j \leftarrow \text{argmax}_{C_j \in \mathcal{P} \setminus C_i}(\sigma(C_i \cup C_j) - \sigma(C_i) - \sigma(C_j))$;
    let $C_{i,j} \leftarrow C_i \cup C_j$;
    put $C_{i,j}$ into $\mathcal{P}$ and delete $C_i$ and $C_j$ from $\mathcal{P}$;
end while
return $\mathcal{P}$;
```

In step 4 of MAMKCP (Algorithm 3), in order to maximize the marginal gain, we have to compute $\sigma(C_i \cup C_j)$ for all the communities $C_j \in \mathcal{P}$, thus, MAMKCP requires $\mathcal{O}(n^2\tau)$ time to obtain a K-VDP when n is large and K is small. The computational complexity of SAMKCP is even higher. Therefore, they may be not suitable for large social networks. To improve the running time performance, here we provide an alternative merge strategy for implementing MAMKCP. Instead of merging the communities with the maximum marginal gain, in step 4 we estimate the influence propagation of C_i through the entire graph, i.e., $\sigma_V(C_i)$, and then compute the average influence received by C_j from C_i, which is defined as $\frac{\sum_{l \in C_i} \sum_{r \in C_j} p_V(l,r)}{|C_j|}$, for all the communities $C_j \neq C_i$. This can be done by simply accumulating $p_V(l,r)$ for each community C_j when we computing $\sigma_V(C_i)$. Finally, we merge C_i with a community with the highest average received influence. In such a way, a merge can be done in $O(\tau)$ time. The overall running time of MAMKCP is only $\mathcal{O}(n\tau)$.

According to the complexity analysis, MAMKCP is better than SAMKCP in terms of the running time performance. For some large social networks, we can apply the simplified version of MAMKCP which requires only linear time. In terms of the partition quality, intuitively, SAMKCP is better than MAMKCP because it considers the global optimization (top-down approach) each time and MAMKCP considers the local optimization (bottom-up approach). We will demonstrate their performance through simulation in the next section.

Results and discussion

In this section, we carry out experiments over real-world social networks. The influence propagation is based on the well-known LT and IC models, and we run MC simulation to estimate the influential propagation function $\sigma(\cdot)$. We begin by describing the algorithms, data sets, and experimental settings in 'Algorithm,' 'Data set,' and 'Experiment setting' sections, respectively, and then discuss the experimental results in 'Experiment result' section.

Algorithm

In addition to the proposed algorithms, (SAMKCP, Algorithm 2) and (MAMKCP, Algorithm 3), we also implement two classic community partition algorithms for comparison purposes. One is a Modularity-based Algorithm (MODUA) proposed in [47] and the other is a Spectrum-based Algorithm (SPECA) proposed in [48]. Given a graph G, MODUA finds communities by optimizing the modularity score locally and it terminates until a maximal modularity score is obtained. Therefore, MODUA cannot partition G into a given number K of communities. While SPECA is flexible for the number K of communities, it partitions a graph iteratively into K communities by minimizing the general cut each time according to the adjacent matrix. To the best of our knowledge, we do not find any algorithm which is designed for disjoint community partition with the objective of maximizing the influence propagation within each community. In addition, we do not find any density-based algorithm that can be applied to our MKCP problem.

Data set

We conduct simulation on three real-world social networks as follow: (1) NetHEPT: taken from the co-authorship network in 'High Energy Physics (Theory)' section (from 1991 to 2003) of arXiv (http://arXiv.org). The nodes in NetHEPT denote the authors, and the edges represent the co-authorship. HetHEPT has 15,229 nodes and 31,376 edges. (2) NetEmail: taken from the email interchange network in University of Rovira i Virgili (Tarragona). The nodes in NetEmail denote the members in the university, and the edges represent email interchanges among the members (the data set is available at http://deim.urv.cat/~alephsys/data.php). NetEmail has 1,133 nodes and 10,902 edges. (3) NetCLUB: taken from the relationship network in Zachary's Karate club network, which is described by Wayne Zachary in [49]. NetCLUB has 34 nodes and 78 edges.

Experiment setting

In this study, we assume that the influential degree from nodes i to j depends on the closeness of their relationship and the probability $p(i)$ for node i where $p(i)$, as defined in Problem description' section, is the probability that node i would produce an influence propagation or would share knowledge with others. We apply the method proposed in [14] to estimate the closeness $c(i,j)$ between i and j. Let $\deg_{\text{in}}(j)$ denote the in-degree of node j, then $c(i,j) = e(i,j)/\deg_{\text{in}}(j)$, where $e(i,j)$ denotes the number of edges from i to j. Due to the lack of ground truth, we independently assign uniform random 0.1%, 1%, and 10% to sharing probabilities $p(i)$ for all the nodes i. Then we assume $\forall(i,j) \in E$, i has a chance of $w(i,j) = \frac{p(i)e(i,j)}{\deg_{\text{in}}(j)}$ to influence j.

Experiment result

We first evaluate the performance of our algorithms on NetCLUB. In algorithm SAMKCP or MAMKCP, $\sigma(\cdot)$ is computed by running MC simulation 1,000 times and get the average. Although AM2CP is not an optimal solution in the IC model, we still apply it in the splits in the simulation of IC model to improve the computational efficiency. Since MODUA is not flexible for the number of communities, we first apply MODUA to get a partition of NetCLUB and then apply our algorithms and SPECA to partition NetCLUB into the same number of communities. Figures 4 and 5 show the experimental results for the LT and IC models respectively. NetCLUB is partitioned into four communities. In

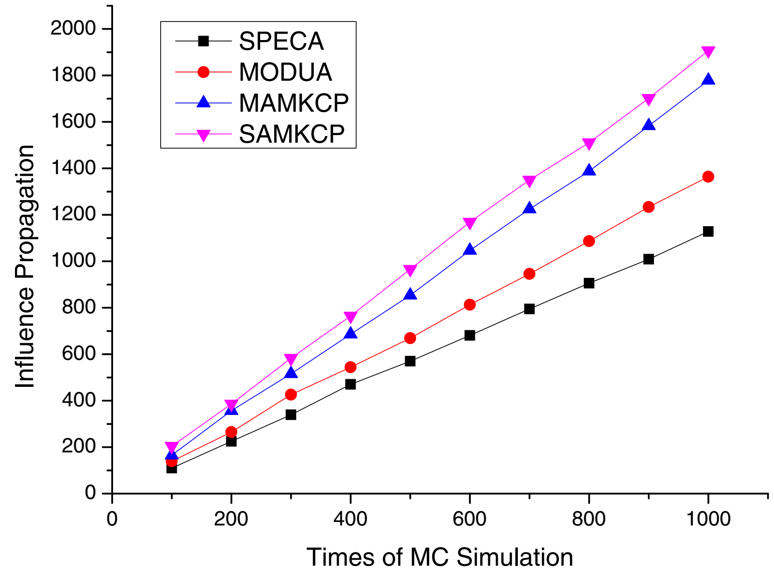

Figure 4 Experimental results on NetCLUB in LT model.

terms of influence propagation, both SAM*K*CP and MAM*K*CP are better than MODUA and SPECA. SAM*K*CP outperforms MODUA and SPECA by about 40% and 70% respectively. In addition, from Figures 4 and 5, we can see the influence propagation of each partition is increasing gradually and linearly when the times of simulation increase, which reflects the reliability of experimental results.

In the second experiment, we compare MAM*K*CP with MODUA and SPECA on NetEmail. SAM*K*CP is removed due to its high computational complexity. Figures 6 and 7 show the experimental results. The network is partitioned into 88 communities. MAM*K*CP has the maximum sum of influence propagation. The performance of SPECA

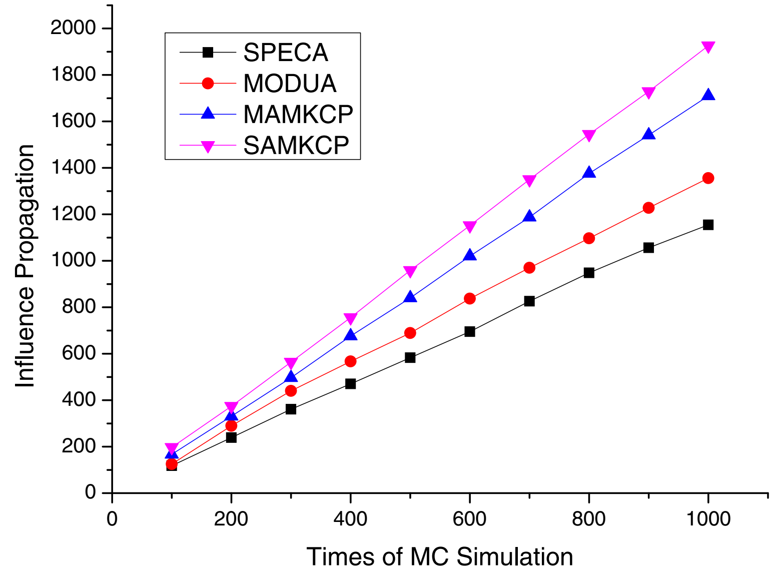

Figure 5 Experimental results on NetCLUB in IC model.

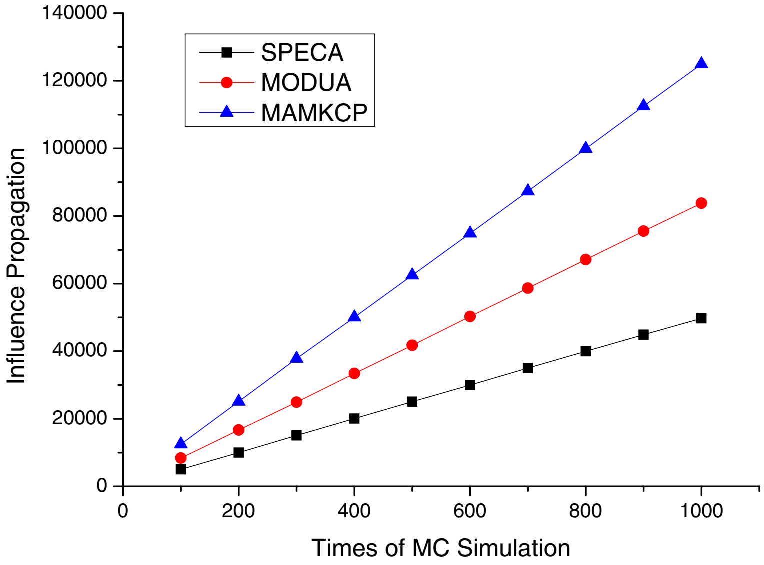

Figure 6 Experimental results on NetEmail in LT model.

is poor compared with MAM*K*CP and MODUA. The influence propagation within the partition of SPECA is about two times less than that of MAM*K*CP and about one time less than that of MODUA.

In the last experiment, we compare MAM*K*CP with MODUA and SPECA on NetHEPT. Since this network has 15,229 nodes and 31,376 edges, we use the simplified version of MAM*K*CP. Figures 8 and 9 show the experimental results. The network is partitioned into 1,820 communities. MAM*K*CP is still better than MODUA and SPECA, but the gap between MAM*K*CP and MODUA in this experiment is less than that in the second experiment. This agrees with our intuition in that simplified MAM*K*CP has a

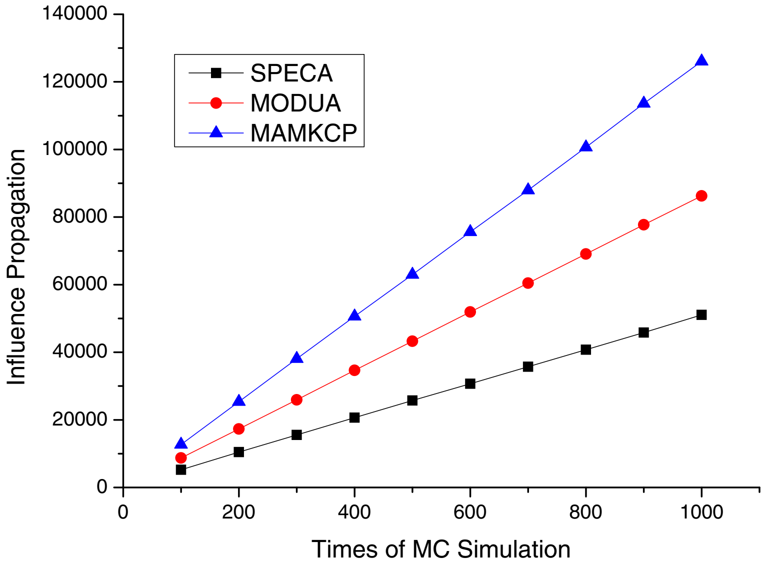

Figure 7 Experimental results on NetEmail in IC model.

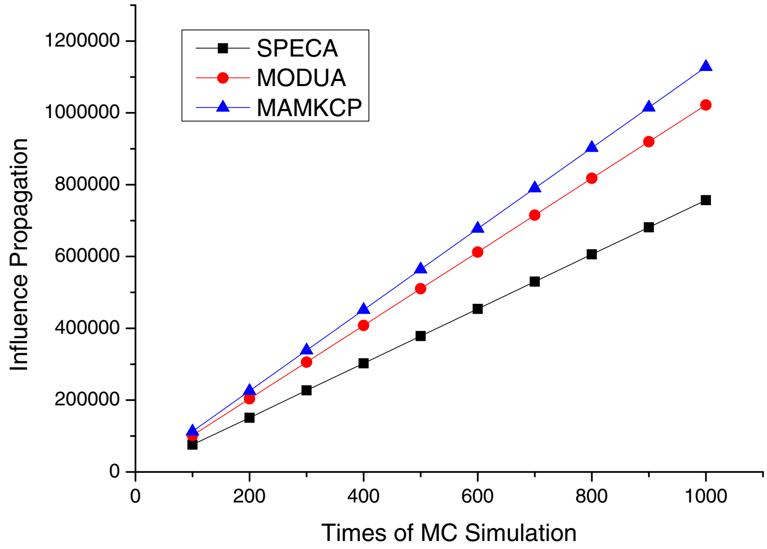

Figure 8 Experimental results on NetHEPT in LT model.

lower computational complexity but also has some loss in performance. According to the three experimental results, we can conclude that the proposed algorithms are better than modularity-based and spectrum-based methods for finding communities in terms of influence propagation.

Conclusions

Community partition and influence propagation are important problems in social networks. In this paper, we investigate the Maximum K-Community Partition (MKCP) problem to maximize the sum of influence propagation within each community. We analyze the problem both theoretically and practically. Especially we show that the M2CP

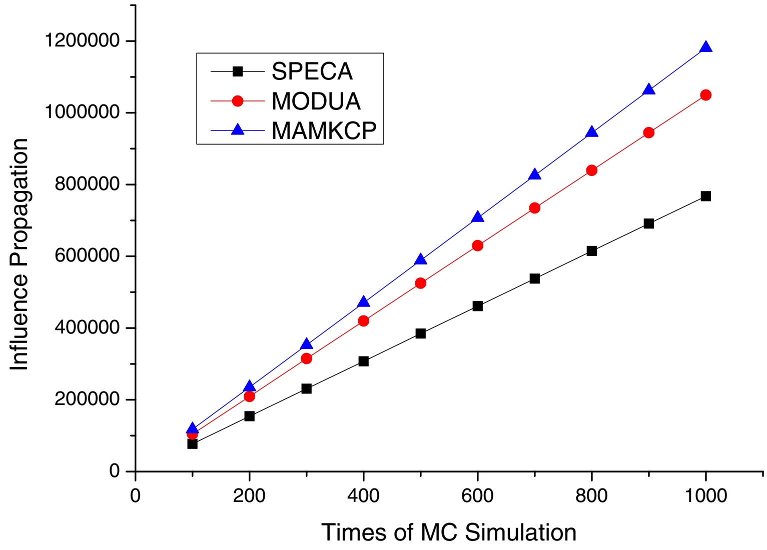

Figure 9 Experimental results on NetHEPT in IC model.

problem can be solved efficiently for a class of influence propagation models. In addition, we prove that the MKCP problem is $\mathcal{NP}$-hard in the well-known LT and IC models for general K. We also develop two heuristic algorithms and demonstrate their efficiency through simulation on real-world social networks.

We believe this study is useful for the influence propagation problems. In future research, we plan to extend our work to the influence maximization problem to select the most influential nodes based on influence-based communities. Furthermore, we will study potential applications of influence-based communities in social networks.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
ZL and YZ formulated the problem and did the algorithm design and implementation. WL, WW, and XC contributed to the theoretical part of algorithm design and organized this research. All authors read and approved the final manuscript.

Acknowledgements
This research work is supported in part by National Science Foundation of USA under grants NSF 1137732 and NSF 1241626.

Author details
[1]NSF Center for Research on Complex Networks, Texas Southern University, 3100 Cleburne Street, Houston, TX 77004, USA. [2]Department of Computer Science, University of Texas at Dallas, 800 W. Campbell Road, Richardson, TX 75080, USA. [3]Department of Computer Science, Texas Southern University, 3100 Cleburne Street, Houston, TX 77004, USA. [4]Department of Computer Science, George Washington University, 2121 Eye Street NW, Washington DC 20052, USA.

Received: 31 December 2013 Accepted: 7 May 2014
Published online: 15 October 2014

References

1. Bollobas, B: Modern Graph Theory. Springer Verlag, New York (1998)
2. Girvan, M, Newman, MEJ: Community structure in social and biological networks. Proc. Natl. Acad. Sci. **99**(12), 7821–7826 (2002)
3. Luxburg, U: A tutorial on spectral clustering. Stat. Comput. **17**, 395–416 (2007)
4. Kannan, R, Vempala, S, Vetta, A: On clusterings: good, bad and spectral. J. ACM. **51**(3), 497–515 (2004)
5. Mancoridis, S, Mitchell, BS, Rorres, C: Using automatic clustering to produce high-level system organizations of source code. In: *Proceedings of the 6th International Workshop on Program Comprehension,* Ischia, Italy, 24–26 June 1998, pp. 45–53, (1998)
6. Newman, M, Girvan, M: Finding and evaluating community structure in networks. Phys. Rev. E. **69**, 026113 (2004)
7. White, S, Smyth, P: A spectral clustering approach to finding communities in graphs. In: *SDM'05: Proceedings of the 5th SIAM International Conference on Data Mining*, pp. 76–84, (2005)
8. Newman, M: Modularity and community structure in networks. Proc. Natl. Acad. Sci. USA. **103**(23), 8577–8582 (2006)
9. Brown, J, Reinegen, P: Social ties and word-of-mouth referral behavior. J. Consum. Res. **14**, 350–362 (1987)
10. Goldenberg, J, Libai, B, Muller, E: Using complex systems analysis to advance marketing theory development: modeling heterogeneity effects on new product growth through stochastic cellular automata. Acad. Market. Sci. Rev. **9**(3), 1–18 (2001)
11. Goldenberg, J, Libai, B, Muller, E: Talk of the network: a complex systems look at the underlying process of word-of-mouth. Market. Lett. **12**, 211–223 (2001)
12. Richardson, M, Domingos, V: Mining knowledge-sharing sites for viral marketing, Edmonton, Alberta, Canada, 23–26 July 2002, pp. 61–70. KDD (2002)
13. Domingos, P, Richardson, M: Mining the network value of customers, San Francisco, CA, USA, 26–29 August 2001, pp. 57–66. KDD (2001)
14. Kempe, D, Kleinberg, JM, Tardos, E: Maximizing the spread of influence through a social network. In: *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 137–146. ACM, New York, (2003)
15. Chen, W, Yuan, Zhang, L: Scalable influence maximization in social networks under the linear threshold model. In: *Proceedings of the 10th IEEE International Conference on Data Mining,* Sydney, Australia, 14–17 December 2010, pp. 88–97, (2010)
16. Chen, W, Wang, C, Wang, Y: 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*, pp. 1029–1038. ACM, New York, (2010)
17. Wang, Y, Cong, G, Song, G, Xie, K: Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'10)*, pp. 1039–1048. ACM, New York, (2010)

18. Radicchi, F, Castellano, C, Cecconi, F, Loreto, V, Parisi, D: Defining and identifying communities in networks. Proc. Natl. Acad. Sci. USA. **101**(9), 2658–2663 (2004)
19. Hu, Y, Chen, H, Zhang, P, Zhang, P, Li, M, Di, Z, Fan, Y: Comparative definition of community and corresponding identifying algorithm. Phys. Rev. E. **78**, 026121 (2008)
20. Weiss, RS, Jacobson, E: A method for the analysis of the structure of complex organizations. Am. Sociol. Rev. **20**(6), 661–668 (1955)
21. Boettcher, S, Percus, AG: Extremal optimization for graph partitioning. Phys. Rev. E. **64**, 026114 (2001)
22. Clauset, A, Newman, MEJ, Moore, C: Finding community structure in very large networks. Phys. Rev. E. **70**(6), 066111 (2004)
23. Newman, MEJ: Fast algorithm for detecting community structure in networks. Phys. Rev. E. **69**, 066133 (2004)
24. Wakita, K, Tsurumi, T: Finding community structure in mega-scale social networks. In: *Proceedings of the 16th International Conference on World Wide Web, WWW'07*, pp. 1275–1276. ACM, New York, (2007)
25. Guimera, R, Pardo, MS, Amaral: LAN: modularity from fluctuations in random graphs and complex networks. Phys. Rev. E. **70**(2), 025101 (2004)
26. Massen, CP, Doye, JPK: Identifying communities within energy landscapes. **71**, 046101 (2005)
27. Duch, J, Arenas, A: Community detection in complex networks using extremal optimization. Phys. Rev. E. **72**(2), 027104 (2005)
28. Holland, JH: Adaptation in Natural and Artificial Systems. MIT, Cambridge (1992)
29. Pizzuti, C: Community detection in social networks with genetic algorithms. In: *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, GECCO'08*, pp. 1137–1138. ACM, New York, (2008)
30. Fortunato, S, Barthelemy, M: Resolution limit in community detection. Proc. Natl. Acad. Sci. USA. **104**(1), 36–41 (2007)
31. Zhang, X, Li, Z, Wang, R, Wang, Y: A combinatorial model and algorithm for globally searching community structure in complex networks. J. Combin. Optim. **23**(4), 425–442 (2010)
32. Fortunato, S: Community detection in graphs. Phys. Rep. **486**, 75–174 (2010)
33. Gaertler, M: Clustering. In: *Brandes, U, Erlebach, T (eds.) Network Analysis: Methodological Foundations*, pp. 178–215. Springer (2005)
34. Lancichinetti, A, Fortunato, S: Community detection algorithms: a comparative analysis. Phys. Rev. E. **80**, 056117 (2009)
35. Schaeffer, S: Graph clustering. Comput. Sci. Rev. **1**(1), 27–64 (2007)
36. Andersen, R, Chung, F, Lang, K: Local graph partitioning using PageRank vectors. In: *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science,* Berkeley, CA, USA, 21–24 October 2006, pp. 475–486 (2006)
37. Leicht, EA, Newman, MEJ: Community structure in directed networks. Phys. Rev. Lett. **100**(11), 118703 (2008)
38. 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,* San Jose, CA, USA, 12–15 August 2007, pp. 420–429, (2007)
39. Kimura, M, Saito, K: Tractable models for information diffusion in social networks, pp. 259–271, PKDD, (2006)
40. Kimura, M, Saito, K, Motoda, H: Efficient estimation of influence functions for SIS model on social networks. In: *Proceedings of the 21st International Joint Conference on Artificial Intelligence,* Pasadena, CA, USA, 11–17 July 2009, pp. 2046–2051, (2009)
41. Li, H, Bhowmick, S, Sun, A: CINEMA: conformity-aware greedy algorithm for influence maximization in online social networks, pp. 323–334. EDBT, (2013)
42. Galstyan, A, Musoyan, V, Cohen, P: Maximizing influence propagation in networks with community structure. Phys. Rev. E. **79**(5), 056102 (2009)
43. Nguyen, NP, Yan, G, Thai, MT, Eidenbenz, S: Containment of misinformation spread in online social networks. WebSci, pp. 213–222 (2012)
44. Dinh, TN, Xuan, Y, Thai, MT: Towards social-aware routing in dynamic communication networks. IPCCC, pp. 161–168 (2009)
45. Belak, V, Lam, S, Hayes, C: Targeting online communities to maximise information diffusion. In: *Proceedings of the WWW Workshop on Mining Social Networks Dynamics*, pp. 1153–1160. Lyon, France, (2012)
46. Stoer, M, Wagner, F: A simple min-cut algorithm. J. ACM. **44**(4), 585–591 (1997)
47. Blondel, V, Guillaume, J, Lambiotte, R, Lefebvre, E: Fast unfolding of communities in large networks. J. Stat. Mech. Theor. Exp (2008)
48. Dhillon, I, Guan, Y, Kulis, B: A fast kernel-based multilevel algorithm for graph clustering. In: *Proceedings of The 11th ACM SIGKDD,* Chicago, Illinois, USA, 21-24 August 2005, pp. 629–634, (2005)
49. Zachary, W: An information flow model for conflict and fission in small groups. J. Anthrop. Res. **33**, 452–73 (1977)

doi:10.1186/s40649-014-0001-4
Cite this article as: Lu *et al.*: **Influence-based community partition for social networks.** *Computational Social Networks* 2014 **1**:1.