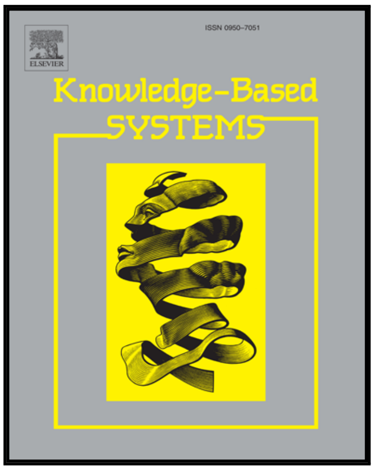
Accepted Manuscript



Tracking the Evolution of Overlapping Communities in Dynamic Social Networks

Zhixiao Wang, Zechao Li, Guan Yuan, Yunlian Sun, Xiaobin Rui, Xinguang Xiang

PII: S0950-7051(18)30257-0
DOI: 10.1016/j.knosys.2018.05.026
Reference: KNOSYS 4346

To appear in: *Knowledge-Based Systems*

Received date: 4 December 2017
Revised date: 15 April 2018
Accepted date: 19 May 2018

Please cite this article as: Zhixiao Wang, Zechao Li, Guan Yuan, Yunlian Sun, Xiaobin Rui, Xinguang Xiang, Tracking the Evolution of Overlapping Communities in Dynamic Social Networks, *Knowledge-Based Systems* (2018), doi: 10.1016/j.knosys.2018.05.026

Tracking the Evolution of Overlapping Communities in Dynamic Social Networks

Zhixiao Wang[a,b], Zechao Li[b], Guan Yuan[a], Yunlian Sun[b], Xiaobin Rui[a], Xinguang Xiang[b,*]

[a]*School of Computer Science and Technology, China University of Mining and Technology, Xuzhou Jiangsu, 221116, China*
[b]*School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing Jiangsu, 210094, China*

Abstract

Overlapping community detection, dynamic community identification and community evolution analysis are the three important problems for social network analysis. It is a challenging task to simultaneously address all these three problems with one single method, thus most traditional studies focus on only one or two of them. This paper proposes a novel Dynamic Overlapping Community Evolution Tracking (DOCET) method to solve the three problems simultaneously with one single model, i.e. topology potential field. Specifically, the proposed DOCET method first detects the initial overlapping community structure based on node location analysis in the peak-valley structure of the topology potential field; then it incrementally updates the dynamic community structure based on influence scope analysis in the topology potential field; finally it tracks community evolution events based on the variation of core nodes in the topology potential field. Experiment results on both synthetic and real-world networks show that our proposed method achieves remarkable performance over the existing state-of-the-art methods. It can both accurately partition dynamic overlapping social networks and efficiently track all kinds of community evolution events.

Keywords: Social network, Overlapping community, Community evolution,

*Corresponding author
Email address: `xgxiang@njust.edu.cn` (Xinguang Xiang)

1. Introduction

Many real-world networks exhibit a natural community structure, i.e., containing groups of vertices that have denser connections within each group and fewer connections between groups [1]. Collaboration networks, the Internet, the world-wide-web, biological networks, communication and transport networks, social networks are just some examples. So far, the definition of community is still ambiguous. Xu et al. [2] give a formula definition of community from the internal structure to the external boundary. Community structure is considered to be a significant property of social networks as it often accounts for the functionality of the system. Community identification is beneficial for understanding their internal organization principles and forecasting the behavior of social networks. For example, community detection can be used to forecast the information propagation in social networks, and to recognize functions of proteins in bioengineering networks [3].

There are many widely used disjoint community detection methods on static networks, as summarized in [4]. However, Palla et al.[5] pointed out that **overlapping** is a significant feature of the community structure. For example, a person is usually involved in several social groups such as family, friends, and colleagues. Since then, many overlapping community detection methods on static networks have been proposed, as reviewed in [6]. Local expansion [7] is a typical seed-centric approach for overlapping community detection, which locally expands or merges the selected seeds. The selected seeds can be core edges [2, 8], core nodes [9], sub-graphs [10, 11, 12, 13], or local communities [14, 15, 16]. Xu et al. [8] proposed a CFM method to find overlapping communities based on a community forest model. They defined the backbone degree to characterize the internal structure of the community, and the community expansion degree to discover the external boundary of the community. The proposed CFM method exhibits better performance in overlapping community detection, and can deal with different network structures including mesh communities and fully connected communities. Wang and Li [9] assume that a node with a considerably large degree is likely the core of a community, thus select these nodes as the core vertices of corresponding communities. Cui et al. [10] regard maximal sub-graphs extracted from the original networks as seeds and then merge them by considering the clustering coefficient of two neighboring maximal sub-graphs. Other similar methods such as [11], [12] and [13] leverage different strategies to merge the neighboring maximal sub-graphs. Li et al. [14] treat local communities as seeds, and community members will be gradually absorbed by seed communities using the absorbing degree function. Similar seed community strategies can be found in [15] and [16]. Except for local expansion, other meaningful strategies are also proposed for overlapping community detection, including neighborhood ratio matrix [17, 18] and subspace decomposition [19].

Furthermore, real-world social networks are not always static, but involve frequent changes. Take Facebook and Twitter as examples. Changes are constantly introduced by joining in or withdrawing from one or more communities [20]. Therefore, **dynamics** is also an important feature of social network communities. Dynamic community detection is a complicated problem because of rapid and unpredictable changes in social networks. Generally, there are three major kinds of dynamic community detection methods in literature: traditional static community detection [21, 22, 23], evolutionary clustering [24, 25, 26, 27, 28, 29] and incremental clustering [1, 30, 31, 32, 33, 34, 35, 36, 37, 38]. (1) Traditional static community detection. A dynamic social network can be divided into a series of consecutive snapshots, and by applying traditional static community detection methods, we can get the corresponding community structure for each snapshot. Typical methods include Clique Percolation Method (CPM) [21] and Order Statistics Local Optimization Method (OSLOM) [22]. This kind of methods ignore the relationship between consecutive snapshots and independently re-partition them at each time step, resulting in high complexity, thus they are not suitable for large-scale dynamic social networks. (2) Evolutionary clustering. Evolutionary clustering, proposed by Chakrabarti et al. [24], adds a temporal smoothness penalty to static community detection methods, and tries to trade-off between the snapshot cost (measuring how well the community structure represents the network at a time step) and the temporal cost (measuring how similar the current community structure is to the previous results) [25]. Afterwards, many evolutionary-based methods are proposed, including evolutionary spectral clustering method [27], FacetNet [28], particle-and-density based evolutionary method [29] and multi-objective evolutionary method [25]. In essence, the evolutionary clustering is an optimization problem of finding a sequence of results that minimizes the overall cost [26]. However, as pointed out in [25], as a generic algorithm, evolutionary clustering requires high execution time to generate a solution. Therefore, it is difficult to be applied to large-scale dynamic social networks too. (3) Incremental clustering. Different from above two kinds of methods, incremental clustering [30] utilizes the known community structure of the first snapshot to adjust the community ownership of incremental nodes for the rest snapshots. Modularity maximization is one major approach of incremental community detection [31], such as MIEN (Modules Identification in Evolving Networks) [32] and QCA (Quick Community Adaptation) [1]. In addition to modularity maximization, some other incremental clustering methods have been proposed, including incremental spectral clustering [33], graph-encoding-based method [34], density-based incremental method [35, 36], label-propagation-based incremental method [37] and Random Walk Sampling (RWS) method [38]. Incremental clustering is with low complexity, but incrementally re-identifying the community ownership of partial nodes may result in errors. Continuous error accumulation will lead to a deviation from the ground-truth structures [30].

The dynamic nature of social network communities may result in birth, growth, shrink, merge, split and death of communities over the entire observation period. Naturally, **evolution** becomes another fundamental characteristic of social network communities. Although evolution is closely related with dynamics, most current studies mainly focus on efficient dynamic community identification, and only limited research has been devoted to further analyzing community evolution between consecutive snapshots. Piotr Brodka et al. [39] identify the community structure of different snapshots and then further establish mappings between two consecutive snapshots to capture inherent evolutionary relations. Other methods for community evolution include AFOCS [40], iCLD [41], and CommTracker [42], etc. Most of traditional methods treat community identification and community evolution separately [43], leading to poor evolution explanations [44]. In fact, the community structure itself can provide evidence about its evolution. Sajid Yousuf Bhat and Muhammad Abulaish [45] propose the HOCTracker to analyze dynamic communities and their evolution in a unified manner over the whole observation time. However, HOCTracker uses a log-based approach to reduce community comparisons in evolutionary relation mapping. To achieve this, they have to maintain an Intermediate Evolution Log to record all intermediate transitions in the communities. Different from above works, Yu et al. [46] use a matrix factorization to model the edge structure of social networks as a function of time, and then predict the evolution of the network over time.

In summary, social network communities are characterized by overlapping, dynamics and evolution. Overlapping community detection, dynamic community identification and community evolution analysis are three important problems for social network analysis. It would be very challenging to simultaneously address all the three problems with one single method. Therefore, most of the above mentioned studies focus on only one or two of them. Different from these traditional methods, this paper proposes a novel method named Dynamic Overlapping Community Evolution Tracking (DOCET) to solve these three problems simultaneously. The joint research on the three problems considers more practical and complicated cases, covering multiple challenging properties of community structures from an integrated perspective. Our method utilizes only one single model, i.e. the topology potential field, to extract overlapping communities, incrementally update the community structure as well as track evolution events. It is a much improved work over our earlier one [7] from two aspects.

First, the current work concentrates on dynamic social networks while the earlier one focused on static social networks. Second, this work can deal with dynamic community identification and community evolution analysis, which cannot be handled by our earlier work.

The main contributions of this paper are summarized as follows:

- We propose a DOCET method that tracks the evolution of overlapping communities in dynamic social networks based on topology potential field, which solves the problems of overlapping community detection, dynamic community identification and community evolution analysis jointly with one single method.
- The proposed DOCET fully exploits the characteristics of the topology potential field, including the peak-valley structure and the influence scope, to solve the three problems. It first detects the initial overlapping community structure based on node location analysis in the peak-valley structure of the topology potential field, and then incrementally updates the overlapping community structure based on influence scope analysis in the topology potential field. Evolution events within the whole observation time are tracked based on variation of core nodes in the topology potential field.
- Experiments on both synthetic and real-world networks show that the proposed method can not only accurately partition the dynamic social networks but also efficiently track all kinds of community evolution events, outperforming the state-of-the-art methods from an overall perspective.

2. Method

This paper proposes a novel method DOCET to track the evolution of overlapping communities in dynamic social networks. Firstly, the initial overlapping community structure is partitioned based on node location analysis in the topology potential field. Secondly, the dynamic overlapping community structure is

150 incrementally updated based on influence scope analysis in the topology potential field. Finally, community evolution events in the whole observation time are tracked based on the variation of core nodes in the topology potential field.

The Table 1 lists most symbols used in the paper.

Table 1

Frequently used notations

Notation	Definition
G	Social network graph
V, E	Nodes & edges of G
G_t	A snapshot of social network at time t
V_t, E_t	Nodes & edges of G_t
v_t^i	A node v_i of the snapshot G_t
$\phi(v_t^i)$	Topology potential of node v_t^i
$m(v_t^i)$	Mass of node v_t^i
d_{ij}	Hops between node v_t^i and node v_t^j
σ	Impact factor of topology potential field
N_t^i	Neighbours of v_t^i
C_t	Community structure of G_t
c_t^k	The community centred around the core node v_t^k
$c_t^{k-internal}$	The internal nodes of community c_t^k
$c_t^{k-overlap}$	The overlapping nodes of community c_t^k
$NetCore(C_t)$	The total core nodes of C_t
$ComCore(c_t^k)$	The core node of the community c_t^k

2.1. Topology potential field

155 Topology potential field is the foundation of our proposed method. This section will provide some preliminary concepts related to it.

Nodes in a social network are not isolated, but linked by many edges. That is, there exist associations among these nodes. A novel mathematical model, topology potential field, can be used to describe these interactions and the associations among network nodes. The original purpose of the classical potential field is to describe non-contact interactions between material particles. With the development of this theory, it has become a mathematical model for describing the non-contact interactions between objects in many research areas. For example, it can be used to identify essential proteins for protein-protein interaction networks [47], or to carry out parameter selection for Support Vector Machine [48].

Definition 1 (Topology potential field). Given a social network $G(V, E)$ and a snapshot of the network $G_t(V_t, E_t)$ at time t; suppose V_t and E_t denote the node set and the edge set of $G_t(V_t, E_t)$, respectively. Each node is regarded as a field source which creates a potential field around itself. The interactions of all nodes form a topology potential field.

Definition 2 (Topology potential). Suppose there is a network $G_t(V_t, E_t)$ and its corresponding topology potential field. $V_t = \{v_t^i | i = 1, 2, ..., n\}$, $E_t = \{(v_t^i, v_t^j) | v_t^i, v_t^j \in V_t, i \neq j\}$, and n represents the total number of nodes at time t. The topology potential of any node is defined as follows[7]:

$$\phi(v_t^i) = \sum_{j=1}^{n} m(v_t^j) \times e^{-(\frac{d_{ij}}{\sigma})^2} \tag{1}$$

where $\phi(v_t^i)$ denotes the topology potential of node v_t^i, $1 \leq i \leq n$; $m(v_t^j)$ represents the mass of neighbor v_t^j, $1 \leq j \leq n$; d_{ij} indicates the hops between node v_t^i and v_t^j; σ denotes an impact factor used to control the influence scope of a node.

There are three parameters in Formula (1): $m(v_t^j)$, d_{ij} and σ. Almost all research on the topology potential field ignores the massive difference between nodes, such as [47], [48] and [49]. Similar to these research, this paper sets $m(v_t^j) = 1$. d_{ij} is the distance between node v_t^i and v_t^j. According to classical field theory, if $d_{ij} > \lfloor 3\sigma/\sqrt{2} \rfloor$, the topology potential component produced by node v_t^j on node v_t^i will become very weak and can be ignored. Thus, only one parameter needs to be tuned in Formula (1), i.e. the impact factor σ. Similar to references [47], [48] and [49], potential entropy [7] is used to select the optimal impact factor σ in this paper. Han et al. [49] have proved the existence of the optimal σ from a mathematical perspective.

Definition 3 (Influence scope). Topology potential field is a short-range field, and the influence scope of any node in the topology potential field is limited. The maximum influence scope is $\lfloor 3\sigma/\sqrt{2} \rfloor$ hops, where σ denotes the selected optimal impact factor.

2.2. Initial overlapping community detection based on node location analysis in the topology potential field

For any community detection method of dynamic social networks, the most basic step, also the most critical one is to partition the initial snapshot. It is the basis of all later deliberate strategies. In this subsection, a topology-potential-based overlapping community detection method is presented to partition the initial snapshot.

We have analysed the characteristics of the topology potential field in our earlier work [7]. As can be seen from Definition 2, the topology potential of a node is a composition of the topology potential components produced by other nodes. It is defined as the position difference of each node in the topology, i.e., the potential of each node in its position [49]. The topology potential value of each node reflects its degree to be influenced by other nodes in the network, and the potential distribution characterizes the structure of nodes in the topology space. Our earlier work [7] found that the topology potential field presents a natural peak-valley structure. Some nodes, with considerably large topology potential values, are located at relatively high positions of the field. Some nodes, with relatively small topology potential values, are located at relatively low positions.

There are three types of node positions in the topology potential field: peak, valley, and slope [7]. Peak position nodes are the local maximum potential nodes. Each local community corresponds to a local high potential area of the topology potential field [49]. The peak position node, located in the centre of the local high potential area, is the core node of the local community. The valley position node is located at a relatively low position in the topology potential field. In general, local high-potential areas are joined by valley position nodes, which are the overlapping nodes among communities. The slope position nodes are located between peak and valley position nodes. They are the internal nodes of corresponding local high-potential area. Detailed information about node position can be referred to in our earlier work [7].

Definition 4 (Core node). Suppose there is a network $G_t(V_t, E_t)$ and the corresponding topology potential field. Given a node $v_t^i \in V_t$, $\phi(v_t^i)$ represents the topology potential of v_t^i. N_t^i denotes the neighbours of v_t^i. For $\forall v_t^j \in N_t^i$, if $\phi(v_t^j) < \phi(v_t^i)$, then node v_t^i locates at the peak position of the topology potential field. Each community corresponds to a local high-potential area. The peak position node is the core node of the corresponding community.

As can be seen from Definition 4, a peak node is the local maximum-potential node. One peak node corresponds to a community, and it is the core node of the community. The topology potential field is a typical short-range field. Each nodes influence ability will quickly drop with the distance increasing, in accordance with the properties of short-range fields, and the maximum influence scope of nodes is $\lfloor 3\sigma/\sqrt{2} \rfloor$ hops. Therefore, if the distance between two peak nodes is shorter than $\lfloor 3\sigma/\sqrt{2} \rfloor$ hops, these two peak nodes interact and associate with each other. In this case, the corresponding two local high-potential areas will be regarded as one community, and the peak node with the larger topology potential value is selected as the core node of the combined community.

Definition 5 (Overlapping node). Suppose there is a network $G_t(V_t, E_t)$ and the corresponding topology potential field. Given a node $v_t^i \in V_t$, $\phi(v_t^i)$ represents the topology potential of v_t^i. N_t^i denotes the neighbours v_t^i. Node v_t^i is an overlapping node if it satisfies the following conditions: $\forall v_t^j \in N_t^i$, $\phi(v_t^j) > \phi(v_t^i)$, and the nearest two peak nodes of v_t^i belong to different communities.

Definition 5 shows that the overlapping node is located at a relatively low position in the topology potential field. In general, local high-potential areas are joined by these nodes, which are the overlapping nodes among communities [7].

Definition 6 (Internal node). Suppose there is a network $G_t(V_t, E_t)$ and the corresponding topology potential field. Given a node $v_t^i \in V_t$, $\phi(v_t^i)$ represents the topology potential of v_t^i. N_t^i denotes the neighbours v_t^i. Node v_t^i is an internal node if it meets either of the following two conditions: (1) $\exists v_t^j \in N_t^i$, $\phi(v_t^j) > \phi(v_t^i)$, and $\exists v_t^j \in N_t^i$, $\phi(v_t^j) < \phi(v_t^i)$; (2) $\forall v_t^j \in N_t^i$, $\phi(v_t^j) > \phi(v_t^i)$, and the nearest two peak nodes of v_t^i belong to the same community.

For the first condition in Definition 6, the node v_t^i locates between peak nodes and overlapping nodes, thus it is an internal node of the community to which its nearest core node belongs. The second condition handles the phenomenon of multiple peak nodes in a community. In this case, the node v_t^i locates among multiple peak nodes. Although it is a valley position, this node is still an internal node rather than an overlapping node.

The above Definitions 4, 5 and 6 reveal that the community affiliation of nodes can be determined by their positions in the inherent peak-valley structure of the topology potential field. In the following, a novel overlapping community detection method is proposed to partition the initial snapshot.

In this subsection, we use a schematic social network shown in Fig.1 (suppose it is the snapshot of one social network at time t) to illustrate the overlapping community detection based on topology potential. The optimal impact factor is 0.4721 and the topology potential of each node is shown in Table 2. Among these 13 nodes, nodes 5 and 11 satisfy the definition of core node. Therefore, there are two communities in this schematic network. Spreading out from the core nodes 5 and 11, respectively, by using Algorithm 2, we can obtain the internal and overlapping nodes of the two communities: $C_t^{5-internal} = \{1, 2, 3, 4, 6\}$, $C_t^{11-internal} = \{7, 8, 10, 12, 13\}$, $C_t^{5-overlap} = C_t^{11-overlap} = \{9\}$.

Algorithm 1: Overlapping community detection based on topology potential

---

Input: Initial snapshot $G_t(V_t, E_t)$, $|V_t| = n$, $|E_t| = m$

Output: Community detection results, including the core and internal nodes of each community, and overlapping nodes among communities

$C_t^{core} = \varnothing$; `//` C_t^{core} `denotes the core nodes of` $G_t(V_t, E_t)$`.`

for $i = 1$ *to* n **do**

  Calculate $\phi(v_t^i)$ using Formula(1);

for $i = 1$ *to* n **do**

  if v_t^i *satisfies Definition 4* **then**

    v_t^i is a core node;

    $C_t^{core} = C_t^{core} \cup \{v_t^i\}$;

for *each core node* $v_t^k \in C_t^{core}$ **do**

  Algorithm2 $(G_t(V_t, E_t), v_t^k)$;

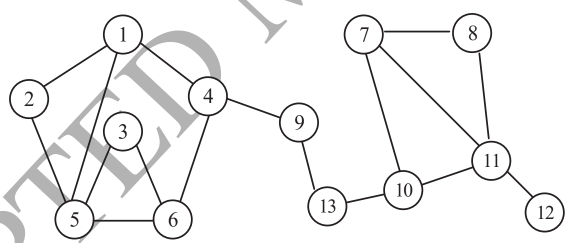


Fig. 1. The snapshot of one social network at time t

2.3. Dynamic community structure updating based on influence scope analysis in the topology potential field

We can get the initial community structure of a dynamic social network by using the above initial overlapping community detection method. In order to obtain the community structure of the rest snapshots, this subsection proposes a dynamic community updating method which utilizes the identified initial community structure to guide the community identification of the rest snapshots,

Algorithm 2: Local expansion algorithm from core node

Input: Initial snapshot $G_t(V_t, E_t)$, a core node v_t^k

Output: the community c_t^k centred around the core node v_t^k, including the internal nodes $c_t^{k-internal}$ and the overlapping nodes $c_t^{k-overlap}$ which are the boundary of community c_t^k

$C_t^k = \varnothing$, $c_t^{k-internal} = \varnothing$, $c_t^{k-overlap} = \varnothing$;
for *each node* $v_t^p \in neighbour(v_t^k)$ **do**
    if $v_t^p \notin c_t^{k-internal}$ *and* $v_t^p \notin c_t^{k-overlap}$ **then**
        Function $(v_t^p, c_t^{k-internal}, c_t^{k-overlap})$;
    $c_t^k = c_t^{k-internal} \cup c_t^{k-overlap}$;
Function $(v_t^p, C_t^{k-internal}, C_t^{k-overlap})$
    if v_t^p *satisfies Definition 5* **then**
        v_t^p is an overlapping node;
        $c_t^{k-overlap} = c_t^{k-overlap} \cup \{v_t^p\}$;
        node v_t^p is a boundary node of community c_t^k;
        local expansion from node v_t^p terminate;
        End Function;
    else if v_t^p *satisfies Definition 6* **then**
        v_t^p is an internal node;
        $c_t^{k-internal} = c_t^{k-internal} \cup \{v_t^p\}$;
        for *each node* $v_t^q \in neighbour(v_t^p)$ **do**
            Function $(v_t^q, c_t^{k-internal}, c_t^{k-overlap})$

avoiding re-computing them for each snapshots. At each snapshot, the proposed method first identifies the affected nodes of different dynamic events, and then rejudges the community affiliation of these affected nodes. Finally, the community structure of this snapshot can be get by partially updating the founded community structure of the previous snapshot.

Any dynamic event in social networks can be regarded as derived from a

Table 2

Topology potential of the schematic network nodes at time t

Node	Topology Potential
1	0.033771466
2	0.022514311
3	0.022514311
4	0.033771466
5	0.045028622
6	0.033771466
7	0.033771466
8	0.022514311
9	0.022514311
10	0.033771466
11	0.045028622
12	0.011257155
13	0.022514311

collection of four simple and basic changes: (1) a new node added, (2) an existing node removed, (3) a new edge added, and (4) an existing edge removed[20]. When these events occur, only some local parts of the social networks are affected. If we can identify the affected local parts of dynamic events, incremental community structure updating can be carried out. In topology potential field, the influence scope of a node is restricted in $\lfloor 3\sigma/\sqrt{2} \rfloor$ hops. Take the schematic network in Fig.1 as an example. The influence scope of this network is $\lfloor 3\sigma/\sqrt{2} \rfloor = \lfloor 3 * 0.4721/\sqrt{2} \rfloor = 1$ hop. If node 12 is removed from this network, only node 11, within one hop, will be affected. we can get the new community structure by only updating the community affiliation of this node.

Suppose $G_t(V_t, E_t)$ and $G_{t+1}(V_{t+1}, E_{t+1})$ are the snapshots of a dynamic social network at time t, $t+1$, respectively. The above proposed initial overlapping community detection method is used to partition the snapshot $G_t(V_t, E_t)$.

In this subsection, we will detail the process of getting the community structure at time $t+1$ from the obtained community structure C_t at time t for four kinds of basic changes.

2.3.1. New node

Consider the case when a new node v^u_{t+1} and its associated connections are introduced at time $t+1$. In this paper, we assume that the new node v^u_{t+1} has at least one adjacent edge connecting to the existing nodes. In the topology potential field, the topology potential of a node is a composition of the topology potential components produced by neighbours within the influence scope. When the new node v^u_{t+1} is introduced, only the nodes within $\lfloor 3\sigma/\sqrt{2} \rfloor$ hops will be affected, the other nodes remaining unchanged. We can efficiently update the community structure by re-identifying the community affiliation of the new node and the affected nodes within $\lfloor 3\sigma/\sqrt{2} \rfloor$ hops. If a new core node appears after the topology potential updating, Algorithm 2 is used to obtain the new community centred in this core node.

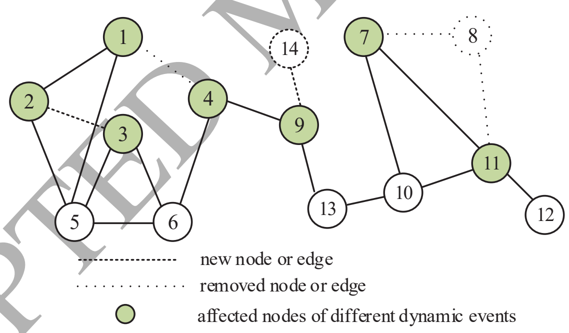


Fig. 2. The snapshot of one social network at time $t+1$

Fig.1 and Fig.2 show the two consecutive snapshots of one social network at time t and t+1, respectively. As mentioned above, the influence scope of this network is 1 hop. Node 14 is a new node at time $t+1$. With the presence of node 14, only node 9 will be affected. What we need to do is select a proper community for the new node 14, and re-determine the community affiliation for the affected node, i.e. node 9.

2.3.2. Node removal

Consider the case where an existing node v_t^u is removed at time $t+1$. At the same time, all of its edges are also deleted. When the existing node v_t^u is removed, only the nodes within $\lfloor 3\sigma/\sqrt{2} \rfloor$ hops will be affected, and the other nodes remain unchanged. We just need to re-calculate the topology potential of these affected nodes and then re-determine the community affiliation of them based on their locations in the topology potential field. If a new core node appears after node removal, Algorithm 2 is used to obtain the new community centred in this core node.

Take node 8 in Fig.1 as an example. This node is removed at time $t+1$ (see Fig.2). With the removal of node 8, only node 7 and 11 will be affected. Node 8 is not a core node, and we only need to re-identify the community affiliation of the two affected nodes. The following Algorithm 3 describe the process community structure updating for node change, including new node and node removal.

2.3.3. New edge

Then we discuss the case where a new edge $e_{t+1} = (v_{t+1}^u, v_{t+1}^w)$ connecting two existing nodes v_{t+1}^u, v_{t+1}^w is introduced at time $t+1$. There are two subcases to be considered: e_{t+1} is an intra-community link or an inter-community link. For the first subcase, the new edge e_{t+1} is totally in one community and will strengthen the inner structure of the community, and the community structure remaining unchanged. For the second subcase, the new edge e_{t+1} connects two communities, and its presence may result in some nodes leaving the current community and then joining a new one. In this situation, we re-calculate the topology potential of the affected nodes within $\lfloor 3\sigma/\sqrt{2} \rfloor$ hops and then re-determine the community affiliation of them based on their locations in the topology potential field.

For example, in Fig.2, a new edge is added between nodes 2 and 3 at time $t+1$. It is an intra-community link, which strengthens the inner structure of the left community. Therefore, the community structure remains unchanged.

Algorithm 3: Dynamic community structure updating in the case of node changes

Input: snapshots $G_t(V_t, E_t)$ and $G_{t+1}(V_{t+1}, E_{t+1})$ of one social network at time t, $t+1$, respectively; the optimal impact factor σ; the community structure C_t at time t, and the new node v_{t+1}^u (or the removed node v_t^u) with associated links

Output: the community structure C_{t+1} at time $t+1$

```
// The following steps process the new node event, the
processing procedure of the node removal event is similar as
these steps, not detailed in this paper.
```

$V_{affected} = \{v_{t+1}^u\}$; `// ` $V_{affected}$ `denotes the affected nodes of node change.`

Calculate $\phi(v_{t+1}^u)$ using formula (1);

for *any node* $v_{t+1}^k \in V_{t+1}$ **do** `// ` $k \neq u$

  if *distance*$(v_{t+1}^u, v_{t+1}^k) \leq \lfloor 3\sigma/\sqrt{2} \rfloor$ **then**

    $V_{affected} = V_{affected} \cup \{v_{t+1}^k\}$;

    Calculate $\phi(v_{t+1}^k)$ using formula (1);

for *each node* $v_{t+1}^l \in V_{affected}$ **do**

  if v_{t+1}^l *is a new core node* **then**

    Algorithm2 $(G_{t+1}(V_{t+1}, E_{t+1}), v_{t+1}^l)$;

  else `// ` v_{t+1}^l `is not a new core node`

    re-determine the community affiliation of v_{t+1}^l according to Definitions 5 and 6;

2.3.4. Edge removal

Let us consider the case when an edge $e_t = (v_t^u, v_t^w)$ at time t is removed at time $t+1$. In this paper, we assume that the edge removal will not result in the presence of an isolated node or community. The same as the case of adding a new edge, there are also two subcases: e_t is an intra-community link or an inter-community link. For the subcase of inter-community link, the removal of e_t will strengthen the current community structure, and the community structure remaining unchanged. For the subcase of intra-community link, the community structure may be changed because of this edge removal. Therefore, we need to re-calculate the topology potential of the affected nodes within influence hops and then re-determine the community affiliation of them.

For example, in Fig.2, the connection between nodes 1 and 4 is removed at time $t+1$. It is an intra-community link, thus we need to re-determine the community affiliation of affected nodes (i.e. nodes 1 and 4). The following Algorithm 4 describe the process community structure updating for edge change, including new edge and edge removal.

In the following, a schematic example (see Fig.3) is presented to show the dynamic community structure updating based on influence scope analysis. Fig.3(a) shows the snapshot $G_t(V_t, E_t)$ with community structure C_t at time t. There are some dynamic changes at time $t+1$, as shown in Fig.3(b). Table 3 lists all the changes and the corresponding affected nodes. The dynamic community structure can be efficiently updated by re-identifying the community affiliation of the affected nodes. Fig.3(c) depicts the updated community structure. Table 4 shows the topology potential of the schematic network nodes at time $t+1$. Comparing Tables 2 with 4, we can find that the topology potential values of the nodes 5, 6, 7, 8 and 10 remain the same. Table 3 reveals the reason: these five nodes are not affected by the listed dynamic changes. The topology potential of node 4 is also unchanged. The reason is that while an existing edge is removed (between nodes 1 and 4), a new edge is added (between nodes 4 and 14), resulting in the same number of neighbours.

Most traditional incremental-based dynamic community detection methods only take the new nodes (edges) or removed nodes (edges) into consideration. However, with the presence or removal of nodes (edges), their local neighbours may be affected, resulting in different community affiliation. Different from traditional methods, our proposed method identifies the affected nodes of incremental nodes (edges) based on influence scope analysis, and then re-determines

Algorithm 4: Dynamic community structure updating in the case of edge changes

Input: snapshots $G_t(V_t, E_t)$ and $G_{t+1}(V_{t+1}, E_{t+1})$ of one social network at time t, $t+1$, respectively; the optimal impact factor σ; the community structure C_t at time t, and the new edge $e_{t+1} = (v_{t+1}^u, v_{t+1}^w)$ (or the removed edge $e_t = (v_t^u, v_t^w)$)

Output: the community structure C_{t+1} at time $t+1$

```
// The following steps process the new edge event, the
processing procedure of the edge removal event is similar as
these steps, not detailed in this paper.
```

$V_{affected} = \{v_{t+1}^u, v_{t+1}^w\}$; `//` $V_{affected}$ `denotes the affected nodes of edge change.`

Calculate $\phi(v_{t+1}^u)$ and $\phi(v_{t+1}^w)$ using formula (1);

for *any node* $v_{t+1}^k \in V_{t+1}$ **do**

    if *distance*$(v_{t+1}^u, v_{t+1}^k) \le \lfloor 3\sigma/\sqrt{2} \rfloor - 1$ *or* *distance*$(v_{t+1}^w, v_{t+1}^k) \le \lfloor 3\sigma/\sqrt{2} \rfloor - 1$ **then**

        $V_{affected} = V_{affected} \cup \{v_{t+1}^k\}$;

        Calculate $\phi(v_{t+1}^k)$ using formula (1);

if e_{t+1} *is an inter-community edge* **then**

    for *each node* $v_{t+1}^l \in V_{affected}$ **do**

        if v_{t+1}^l *is a new core node* **then**

            Algorithm2 $(G_{t+1}(V_{t+1}, E_{t+1}), v_{t+1}^l)$;

        else `//` v_{t+1}^l `is not a new core node`

            re-determine the community affiliation of v_{t+1}^l according to Definitions 5 and 6;

the community affiliation of both incremental parts and affected parts. Thus, we can update the dynamic community structure more accurately, leading to lower error accumulation.

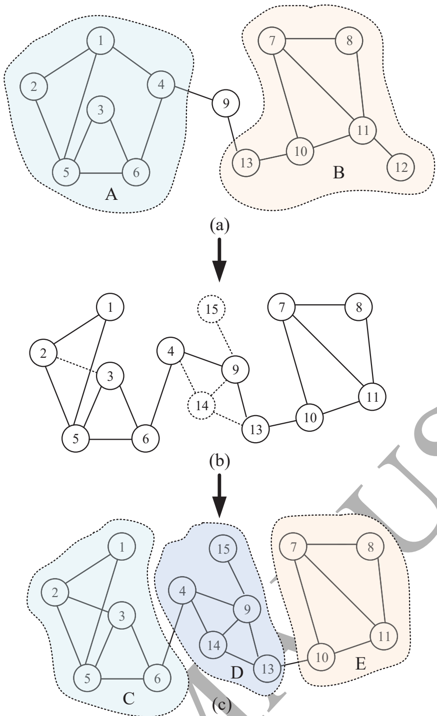


Fig. 3. A schematic example of dynamic community structure updating: (a) network $G_t(V_t, E_t)$ with community structure C_t at time t; (b) network $G_{t+1}(V_{t+1}, E_{t+1})$ with all kinds of dynamic changes; (c) updated community structure C_{t+1} based on $G_{t+1}(V_{t+1}, E_{t+1})$ and C_t

Table 3
Affected nodes of different dynamic changes

Singe Dynamic Change	Affected Nodes
Node 12 is removed	Node 11
Node 14 is added	Nodes 4,9,13,14
Node 15 is added	Node 9,15
Edge between nodes 1 and 4 is removed	Nodes 1,4
Edge between nodes 2 and 3 is added	Nodes 2,3

Table 4
Topology potential of the schematic network nodes at time t+1

Node	Topology Potential
1	0.022514311
2	0.033771466
3	0.033771466
4	0.033771466
5	0.045028622
6	0.033771466
7	0.033771466
8	0.022514311
9	0.045028622
10	0.033771466
11	0.033771466
13	0.033771466
14	0.033771466
15	0.011257155

2.4. Community evolution tracking based on the variation of core node in the topology potential field

The dynamic nature of social network communities leads to community structure evolution over the entire observation period. Dynamic communities of social networks can be characterized by a series of evolutionary events. There are six kinds of possible community evolutionary events [39], including birth, growth, shrink, merge, split and death. A birth event means a new community occurs at time $t + 1$, which does not exist at the previous time t. A death event represents that an existing community dissolves in the following consecutive snapshot. A community shrinks when some existing nodes leave the group, making its size smaller than the previous snapshot. A community grows when some new nodes join in, making its size bigger than the previous time step. A split event reveals that a single community of the current snapshot splits into two or more groups in the next time step. A merge event occurs if two or more distinct communities observed at time t merge into a single group at time $t+1$.

Wang et al. [42] pointed out that taking advantage of not all nodes but those representative and reliable core ones will be a more accurate and effective way to track community evolution. A good example is the co-authorship network, in which core nodes represent famous professors. Community evolution events can be revealed just by tracking these famous professors. In the topology potential field, each community corresponds to a local high-potential area [7]. The maximum-potential node, located at the centre of each local high-potential area, is the core node of the corresponding community. Thus, the birth, growth, shrink, merge, split and death of dynamic communities can be identified by tracking the variation of core nodes. For example, the occurrence of a new core node may lead to the birth of a new community, and the disappearance of an existing core node may result in the death of a community.

Definition 7. (Death and Merge). Given two consecutive snapshots of a dynamic social network: $G_t(V_t, E_t)$ and $G_{t+1}(V_{t+1}, E_{t+1})$, $|V_t| = n$, $|V_{t+1}| = m$. $C_t = (c_t^1, c_t^2, ..., c_t^k)$ denotes the community structure of $G_t(V_t, E_t)$, $C_{t+1} = (c_{t+1}^1, c_{t+1}^2, ..., c_{t+1}^l)$ represents the community structure of $G_{t+1}(V_{t+1}, E_{t+1})$, and k, l refer to the community number at time t and $t+1$, respectively; $NetCore(C_{t+1})$ represents the total core nodes of C_{t+1} and $ComCore(c_{t+1}^p)$ denotes the core node of the community c_{t+1}^p, $NetCore(C_{t+1}) = \bigcup_{p=1}^{l} ComCore(c_{t+1}^p)$, $1 \leq p \leq l$. Suppose there is a node $v = ComCore(c_t^q), 1 \leq q \leq k$. (1) If $v \notin V_{t+1}$, then the community c_t^q dissolves at time $t+1$; (2) if v is an internal node of the community $c_{t+1}^p, 1 \leq p \leq l$, then the community c_t^q is merged into c_{t+1}^p at time $t+1$.

Definition 8. (Birth and Split). Given two consecutive snapshots of a dynamic social network: $G_t(V_t, E_t)$ and $G_{t+1}(V_{t+1}, E_{t+1})$, $|V_t| = n$, $|V_{t+1}| = m$. $C_t = (c_t^1, c_t^2, ..., c_t^k)$ denotes the community structure of $G_t(V_t, E_t)$, $C_{t+1} = (c_{t+1}^1, c_{t+1}^2, ..., c_{t+1}^l)$ represents the community structure of $G_{t+1}(V_{t+1}, E_{t+1})$, and k, l refer to the community number at time t and $t+1$, respectively; $NetCore(C_t)$ represents the total core nodes of C_t and $ComCore(c_t^q)$ denotes the core node of the community c_t^q, $NetCore(C_t) = \bigcup_{q=1}^{k} ComCore(c_t^q), 1 \leq q \leq k$. Suppose there is a node $v = ComCore(c_{t+1}^p), 1 \leq p \leq l$. (1) If $v \notin V_t$, then the community c_{t+1}^p is born at time $t+1$; (2) if v is an internal node of the community $c_t^q, 1 \leq q \leq k$, then the community c_{t+1}^p is divided from c_t^q at time $t+1$.

Definition 9. (Shrink and Growth). Given two consecutive snapshots of a dynamic social network: $G_t(V_t, E_t)$ and $G_{t+1}(V_{t+1}, E_{t+1})$, $|V_t| = n$, $|V_{t+1}| = m$. $C_t = (c_t^1, c_t^2, ..., c_t^k)$ denotes the community structure of $G_t(V_t, E_t)$, $C_{t+1} = (c_{t+1}^1, c_{t+1}^2, ..., c_{t+1}^l)$ represents the community structure of $G_{t+1}(V_{t+1}, E_{t+1})$, and k, l refer to the community number at time t and $t+1$, respectively. Assume there are two communities $c_t^q \in C_t$ and $c_{t+1}^p \in C_{t+1}, 1 \leq q \leq k, 1 \leq p \leq l$. (1) Suppose the community c_t^q is merged into c_{t+1}^p at time $t+1$. If $\exists ComCore(c_t^r) = ComCore(c_{t+1}^p), 1 \leq r \leq\ k \ and \ r \neq q$, then the community c_t^r grows to c_{t+1}^p at time $t+1$. (2) Suppose the community c_{t+1}^p is divided from c_t^q at time $t+1$. If $\exists ComCore(c_{t+1}^s) = ComCore(c_t^q), 1 \leq s \leq l \ and \ s \neq p$, then the community c_t^q shrinks to c_{t+1}^s at time $t+1$.

In the following, three algorithms are proposed to handle the above six kinds of community evolution events.

Algorithm 5: Death and merge event identification algorithm

Input: two consecutive snapshots $G_t(V_t, E_t)$ and $G_{t+1}(V_{t+1}, E_{t+1})$, $|V_t| = n$, $|V_{t+1}| = m$; the corresponding community structure $C_t = (c_t^1, c_t^2, ..., c_t^k)$ and $C_{t+1} = (c_{t+1}^1, c_{t+1}^2, ..., c_{t+1}^l)$, $|C_t| = k$, $|C_{t+1}| = l$

Output: Identified death and merge events between the two consecutive snapshots

death_core = $\varnothing$; `// death_core represents the core nodes disappeared at time` $t+1$.

if $NetCore(C_t) - NetCore(C_{t+1}) \neq \varnothing$ **then**

  death_core = $NetCore(C_t) - NetCore(C_{t+1})$;

  for *each node* v *in death_core* **do**

    search a community c_t^q from C_t that satisfies $ComCore(c_t^q) = \{v\}$;

    if $v \notin V_{t+1}$ **then**

      community c_t^q is dead at time $t+1$;

    if $\exists c_{t+1}^p$ *satisfies* $v \in internal(c_{t+1}^p)$ **then**

      `//` v `is an internal node of the community` c_{t+1}^p.

      c_t^q is merged into c_{t+1}^p at time $t+1$;

To make sense of the above evolution events identification algorithms, let us consider the example shown in Fig.3. As shown in Fig.3(a), there are two communities at time t, and NetCore(C_t)=$\{5, 11\}$. At time $t+1$, three communities are detected, and NetCore(C_{t+1})=$\{5, 9, 11\}$. Node $9 \in$ NetCore(C_{t+1}), and node 9 is an overlapping node at time t. Therefore, the community D (the middle part of Fig.3(c)) is divided from the community A (the left part of Fig.3(a)) and B (the right part of Fig.3(a)). Meanwhile, the community A shrinks to C (the left part of Fig.3(c)) and the community B shrinks to E (the right part of Fig.3(c)).

Algorithm 6: Birth and split event identification algorithm

Input: two consecutive snapshots $G_t(V_t, E_t)$ and $G_{t+1}(V_{t+1}, E_{t+1})$, $|V_t| = n$, $|V_{t+1}| = m$; the corresponding community structure $C_t = (c_t^1, c_t^2, ..., c_t^k)$ and $C_{t+1} = (c_{t+1}^1, c_{t+1}^2, ..., c_{t+1}^l)$, $|C_t| = k$, $|C_{t+1}| = l$

Output: Identified birth and split events between the two consecutive snapshots

emerge_core $= \varnothing$; `// the new core nodes emerge at time` $t+1$.
if $NetCore(C_{t+1}) - NetCore(C_t) \neq \varnothing$ **then**
    emerge_core $= NetCore(C_{t+1}) - NetCore(C_t)$;
    for *each node* v *in emerge_core* **do**
        search a community c_{t+1}^p from C_{t+1} that satisfies $ComCore(c_{t+1}^p) = \{v\}$;
        if $v \notin V_t$ **then**
            community c_{t+1}^p occurs at time $t+1$;
        if $\exists c_t^q$ *satisfies* $v \in internal(c_t^q)$ **then**
            `//` v `is an internal node of the community` c_t^q.
            community c_{t+1}^p is divided from c_t^q at time $t+1$;

2.5. Complexity Analysis

2.5.1. Time complexity

Algorithm 1 and **Algorithm 2** detect the overlapping communities based on topology potential field. Firstly, the complexity of topology potential field construction is $O(n^2)$ [7]. Secondly, all core nodes can be identified within $O(\langle k \rangle n)$ where $O(\langle k \rangle)$ represents the average degree of network nodes. Finally, local expansion process takes $O(\langle k \rangle n)$ to visit all nodes and determine their community affiliation. Thus, the total complexity of overlapping community detection is $O(n^2) + O(\langle k \rangle n) + O(\langle k \rangle n) = O(n^2)$.

Algorithm 3 and **Algorithm 4** update the dynamic community structure

Algorithm 7: Shrink and growth event identification algorithm

Input: two consecutive snapshots $G_t(V_t, E_t)$ and $G_{t+1}(V_{t+1}, E_{t+1})$, $|V_t| = n$, $|V_{t+1}| = m$; the corresponding community structure $C_t = (c_t^1, c_t^2, ..., c_t^k)$ and $C_{t+1} = (c_{t+1}^1, c_{t+1}^2, ..., c_{t+1}^l)$, $|C_t| = k$, $|C_{t+1}| = l$; two communities $c_t^q \in C_t$ and $c_{t+1}^p \in C_{t+1}$, $1 \leq q \leq k, 1 \leq p \leq l$, which satisfy the community c_t^q is merged into c_{t+1}^p at time $t+1$ or community c_{t+1}^p is divided from c_t^q at time $t+1$

Output: Identified shrink and growth events between the two consecutive snapshots

```
if the community c_t^q is merged into c_{t+1}^p at time t+1 then
    if ∃ ComCore(c_t^r) = ComCore(c_{t+1}^p), 1 ≤ r ≤ k and r ≠ q then
        community c_t^r grows to c_{t+1}^p at time t+1;
if the community c_{t+1}^p is divided from c_t^q at time t+1 then
    if ∃ ComCore(c_{t+1}^s) = ComCore(c_t^q), 1 ≤ s ≤ l and s ≠ p then
        community c_t^q shrinks to c_{t+1}^s at time t+1;
```

based on influence scope analysis in the topology potential field. **Algorithm 3** updates community structure in the case of node changes. First, the topology potential of the effected nodes is re-calculated. This process takes $O(an)$ where a denotes the number of the affected nodes. Then, if there are no new core nodes emerging, we only need $O(a\langle k\rangle)$ to update their position and community. Otherwise, we need $O(\langle k\rangle n)$ to visit all nodes and determine their affiliation, $O(a\langle k\rangle) < O(\langle k\rangle n)$. Thus, the complexity of dynamic community structure updating in the case of node changes is $O(an) + O(\langle k\rangle n) = O((\langle k\rangle + a)n)$. **Algorithm 4** updates community structure in the case of edge changes, which is similar as **Algorithm 3** and the complexity is also $O((\langle k\rangle+a)n)$. In summary, the total complexity of dynamic community structure updating is $O((\langle k\rangle+a)n)+O((\langle k\rangle + a)n) = O((\langle k\rangle + a)n)$

Algorithm 5, **Algorithm 6** and **Algorithm 7** identify all kinds of community evolution events. **Algorithm 5** deals with the death and merge events. First, this algorithm needs to identify the core nodes disappearing at time $t+1$ with complexity $O(k+l)$, where k, l refer to the community number at time t and $t+1$, respectively. Then, for each disappearing core node, the algorithm visits all $k+l$ communities of the two consecutive snapshots to identify death and merge events, with the complexity $O(b(k+l))$, where b denotes the total number of disappearing core nodes at time $t+1$. Thus, the complexity of death and merge events identification is $O(k+l)+O(b(k+l)) = O(b(k+l))$. **Algorithm 6** identifies the birth and split events, which is similar as **Algorithm 5** and the complexity is also $O(b(k+l))$. **Algorithm 7** handles the shrink and growth events. First, the algorithm needs to get all merge and split events with the help of **Algorithm 5** and **Algorithm 6**. Then, the algorithm takes $O(kl)$ to identify the growth or shrink events between the two consecutive snapshots. In summary, the total complexity of community evolution analysis is $O(b(k+l)) + O(b(k+l)) + O(kl) = O(b(k+l)+kl)$.

To sum up, the total time complexity of the proposed DOCET method is $O(n^2)+O((\langle k\rangle+a)n)+O(b(k+l)+kl)$. Obviously, $\langle k\rangle, a, b, k$ and l are all much smaller than n, Therefore, the final time complexity of our method is $O(n^2)$.

2.5.2. Space complexity

For each snapshot, we need $O(n^2)$ to store the adjacency matrix of network nodes, and the space complexity for all snapshots is $O(zn^2)$ where z represents the snapshot number of the dynamic network. This storage space is necessary for almost all the other methods. In addition, our proposed DOCET method needs an extra $O(n)$ to store the topology potential of each network node.

3. Experimental Results

In this section, the performance of our proposed DOCET method will be evaluated from the perspective of initial overlapping community detection, dynamic community structure updating and community evolution tracking.

3.1. Datasets, baseline algorithms and evaluation metrics

Datasets. Two types of networks are used in the experiments: artificial networks and real-world social networks. Artificial networks are generated by using the LFR (Lancichinetti-Fortunato-Radichi) Benchmark generator[50], which can produce required networks with implanted communities. It is true that even with the same parameters, the LFR generator cannot produce the exactly same artificial networks in each run[7]. In order to obtain relatively accurate results, the average performance of 50 runs is taken as the final result. There are two kinds of real-world social networks: static and dynamic social networks. The static social networks are used to evaluate the performance of the initial overlapping community detection. These static networks include the Karate club network (KR), American College football network (FB), Les miserable network (LS), Dolphin social network (DP), co-authorship network CA-GrQc (CA), user network of the Pretty-Good-Privacy algorithm (PGP), community network in Enron via emails (EM), who-trust-whom online social network Epinions(EP), Gnutella peer-to-peer file sharing network (P2P), and the co-purchase network of Amazon (AM). These static networks are taken from `http://www-personal.umich.edu/~mejn/netdata/` and `http://snap.stanford.edu/data/index.html`. Two real-world dynamic social networks are used in our experiments, one is the VAST (`http://www.cs.umd.edu/hcil/VASTchallenge08`), the other is DBLP (`http://konect.uni-koblenz.de/networks/dblp_coauthor`). The VAST network consists of information about 9,834 calls among 400 members over a 10-day period in June 2006 [51]. This network is divided into ten consecutive snapshots based on the phone communication day. The DBLP network is the collaboration graph of authors of scientific papers of DBLP computer science bibliography from 1998 to 2013. It contains 18,986,618 edges with 1,314,050 nodes. An edge between two authors represents a common publication. Edges are annotated with the date of the publication. This network is also divided into ten consecutive snapshots.

Baseline algorithms. Nine state-of-the-art methods are selected as baseline algorithms, including Game [52], COPRA [53], SHRINK [54], OSLOM [22], QCA [1], MIEN [32], AFOCS [40], FacetNet [28], and HOCTracker [45]. The former four of them mainly focus on static overlapping community detection, i.e. Game [52], COPRA [53], SHRINK [54] and OSLOM [22]. Xie et al. [55] found that Game [52], COPRA [53], SHRINK [54] and OSLOM [22] show excellent performance in overlapping community detection. The Game [52] method introduces a game-theoretic framework to address the overlapping community detection problem. COPRA [53] is a label-propagation-based overlapping community detection method wherein community labels are propagated between nodes according to dynamic interaction rules. SHRINK [54] is a parameter-free hierarchical and overlapping network clustering algorithm that combines advantages of density-based clustering and modularity optimization methods.

Table 5 presents a summary of the distinguishing characteristics of our proposed DOCET method against the above nine baseline algorithms. It can be observed that DOCET aims to address all three important issues (i.e. overlapping community detection, dynamic community identification and community evolution analysis) simultaneously, with a relatively low time complexity.

In Table 5, n, m represent the node number and edge number of a social network, respectively. v of COPRA denotes the maximum number of communities to which any node can belong, and q_1 denotes the number of iterations needed for the COPRA method, which varies from dozens to 300 based on the network scale [53]. $\Delta n, \Delta m$ of QCA method refer to the changed nodes and changed edges between two snapshots, respectively. c of FacetNet denotes the number of communities, and q_2 represents the number of iterations needed for the FacetNet method, which is about 600 despite the network scale [28]. The complexity of OSLOM cannot be estimated exactly, as it depends on the specific features of different community structures [22]

Evaluation metrics. The metrics used in this paper are NMI[56], Q_{ov}^{Ni}[55] and Omega index[57]. NMI is widely used to evaluate community detection

Table 5
Characteristics and time complexity of different methods

Method	Dynamics	Overlapping	Evolution	Time Complexity
DOCET	✓	✓	✓	$O(n^2)$
Game	×	✓	×	$O(m^2)$
COPRA	×	✓	×	$O(v^3 n + q_1 vm \log(vm/n))$
SHRINK	×	✓	×	$O(m \log n)$
OSLOM	×	✓	×	$-$
QCA	✓	×	×	$O(\langle k \rangle n(\Delta m + \Delta n))$
MIEN	✓	×	×	$O(m + n \log^2 n)$
AFOCS	✓	✓	×	$O(n^2)$
FacetNet	✓	×	✓	$O(q_2 cm)$
HOCTracker	✓	✓	✓	$O(n^2)$

performance, which measures the similarity between the detected community structure and the well-known standard[56]. Modularity evaluates how good the obtained community structure is. Q_{ov}^{Ni} is an overlapping modularity based on the link belonging factor [55], and $f(x) = 60x - 30$ is adopted as the arbitrary function [55]. The Omega index is the overlapping version of the Adjusted Rand Index (ARI) [57].

3.2. The performance of initial overlapping community detection

In this subsection, our proposed method DOCET is evaluated from the perspective of initial overlapping community detection. The overlapping community detection of the initial snapshot is the most basic and critical step, and its performance is very important for the following dynamic community incremental updating. Among the nine baseline algorithms, FacetNet [28] and HOCTracker [45] mainly devoted to tracking community evolution events. Therefore, our proposed method DOCET is compared with other seven baseline methods here, including Game [52], COPRA [53], SHRINK [54], OSLOM [22], QCA [1], MIEN [32] and AFOCS [40].

3.2.1. Test on synthetic networks

Firstly, we evaluate the overlapping community detection performance of our DOCET method with varying network scales from 10,000 to 100,000. The other key parameters of the LFR generator are: intra-community strength $\mu = 0.3$, overlapping density $O_n = 10\%$, community size $= 100 \sim 2000$, and $O_m = 5$. O_m indicates the number of communities to which each overlapping node belongs [7]. The performance of DOCET is compared with the above seven methods. NMI is used to quantify the quality of the community structure identified by each method. The results are shown in Fig.4. As the number of nodes n varies from small to large, the detection performance of all methods typically decays at a moderate rate. Our proposed method DOCET shows a competitive performance to QCA, and better than the rest methods.

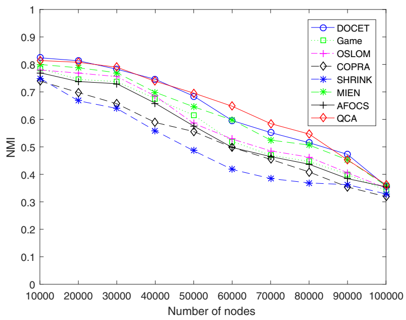


Fig. 4. NMI of different methods on LFR network with varying network scales from 10000 to 100000

Secondly, we evaluate the overlapping community detection performance of our DOCET method with the variation of membership number O_m from 2 to 8. The other key parameters of the LFR generator are: network size $n = 5000$, intra-community strength $\mu = 0.3$, overlapping density $O_n = 10\%$, and community size $= 10 \sim 50$. We compare the performance of DOCET with above seven methods. NMI and Omega are used to quantify the quality of the community structure identified by each method. The results are shown in Fig.5 and Fig.6. As the number of memberships O_m varies from small to large values, the detection performance of all methods typically decays at a moderate rate. The similar trend is found in [55]. Our proposed method DOCET shows better performance than other methods with the O_m increasing, especially when it is larger than 4.

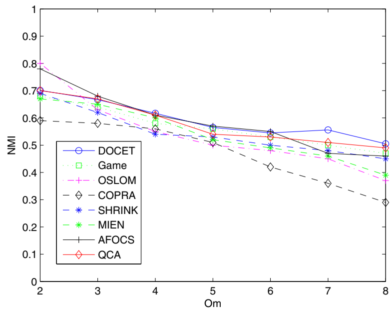


Fig. 5. NMI of different methods on LFR network with varying O_m from 2 to 8

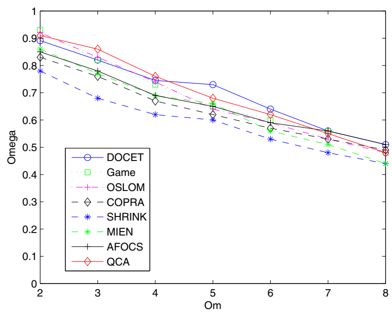


Fig. 6. Omega of different methods on LFR network with varying O_m from 2 to 8

3.2.2. Test on real-world social networks

Then, these eight methods (seven baselines plus our DOCET method) are applied to eleven real-world social networks, including KR, FB, LS, DP, CA, PGP, EM, EP, P2P, AM and DBLP. The corresponding results are shown in Fig.7. Although the proposed method DOCET does not perform equally well on different types of real-world networks, on the whole, it exhibits fairly good performance compared with the other seven algorithms. Our proposed method shows relatively poor Q_{ov}^{Ni} performance on KR, CA, and PGP. This result mainly comes from the following reasons. On one hand, the peak-valley structure of the topology potential field of these three networks is not obvious, In this case, the local maximum potential nodes of some loose and small-scale communities are in the influence scope of other local maximum potential nodes from large-scale communities. Therefore, these comparatively loose communities will be absorbed by large-scale communities around them, decreasing the identified community number [3]. On the other hand, MIEN and SHRINK methods partition the network based on the principle of modularity maximization. Therefore, they can obtain higher Q_{ov}^{Ni}. Generally, our proposed method can guarantee the accuracy of the community structure partition, and at the same time obtains competitive modularity value.

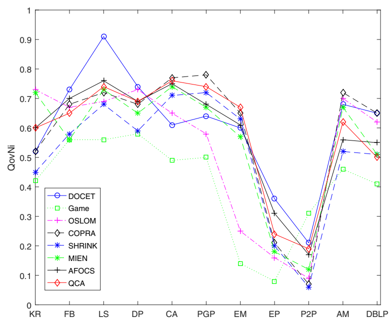


Fig. 7. Q_{ov}^{Ni} of different methods on eleven real-world networks

3.3. The performance of dynamic community structure updating

In this subsection, our proposed method DOCET is evaluated from the perspective of dynamic community structure updating. Among the nine baseline algorithms, Game [52], COPRA [53], SHRINK [54] and OSLOM [22] are not designed for dynamic community detection. Therefore, our proposed method DOCET is compared with other five methods, including QCA [1], MIEN [32], AFOCS [40], FacetNet [28] and HOCTracker [45].

3.3.1. Test on synthetic networks

A synthetic network is generated according to the method proposed by Newman et al. [58], which has also been used in FacetNet [28]. The synthetic network consists of 128 nodes that belong to 4 communities with 32 nodes in each community. Edges are placed independently and randomly between a pair of nodes. The probability that a link exists between a pair of nodes belonging to the same community is P_{in}; the probability that a link exists between a pair of nodes belonging to different communities is P_{out}. The parameter Z_{out}, which represents the mean number of edges from a node to nodes in other communities, is used to control the noise level in the synthetic networks [58]. We generate such networks for 10 consecutive time steps. In order to introduce dynamics into the network, at each time step, a certain number of (decided by a parameter named evolution rate) nodes are randomly chosen to leave their original communities and join the other three communities. The corresponding parameters of the synthetic dynamic network are listed in Table 6.

Table 6

Parameter setting of the synthetic dynamic network

Parameters	Value
network size	128
community number	4
average degree	16
overlapping density	0.1
P_{in}	0.05
P_{out}	0.16
Z_{out}	5
evolution rate	10% or 30%

Since we have the ground truth for the community membership at each time step, we can directly compare the obtained community structure with the true partition. NMI and Q_{ov}^{Ni} are adopted to evaluate the performance of different methods on the synthetic networks.

Fig.8 shows the NMI value with respect to the ground truth over the 10 time steps when evolution rate= 10%, and Fig.9 shows the corresponding Q_{ov}^{Ni}. The community structure of each time step is incrementally obtained from the result of the previous time step. However, re-identifying the community ownership of only incremental nodes may result in error, which affects the accuracy of partition results, and continuous error accumulation will lead to a deviation between the found communities and the ground-truth structure. The experiment results are consistent with the above analysis. With the time point ranging from 1 to 10, the NMI and Q_{ov}^{Ni} performance of all six methods declines. Despite all this, our proposed method DOCET still outperforms the most compared algorithms. The main reason is that our proposed method takes not only incremental nodes but also the affected neighbors of these incremental nodes into consideration, which can improve the accuracy of community structure updating.

Fig.10 shows the NMI value with respect to the ground truth over the 10 time steps when evolution rate= 30%, and Fig.11 shows the corresponding Q_{ov}^{Ni}. When evolution rate is 30%, the whole network evolves more significantly compared with 10% evolution rate, and more nodes needed to be re-identified the community affiliation. Thus, incrementally updating the community becomes difficult, and the NMI performance becomes worse than before. The performance of our proposed method declines slightly slower compared with other five method because of its low partition error accumulation.

3.3.2. Test on real-world social network

Then, these six methods are applied on the VAST and DBLP networks, two widely used real-world networks in existing literature. Due to no available ground truth of the VAST and DBLP, we can only use the Q_{ov}^{Ni} to evaluate the updating performance of different methods. The corresponding results are

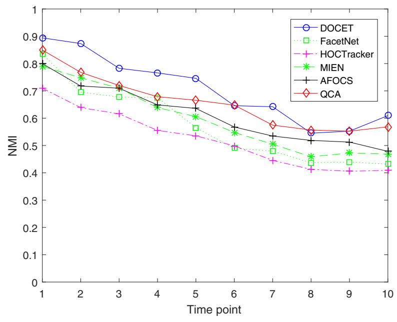


Fig. 8. NMI at different time steps of the synthetic network when evolution rate=10%

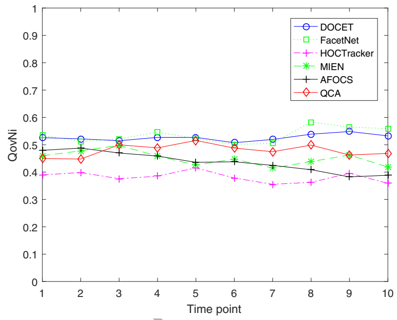


Fig. 9. Q_{ov}^{Ni} at different time steps of the synthetic network when evolution rate=10%

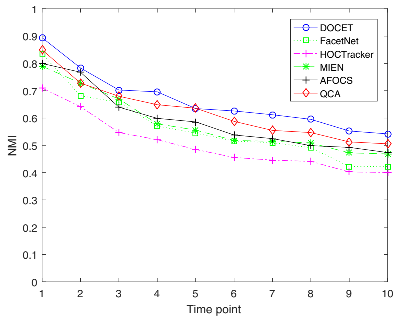


Fig. 10. NMI at different time steps of the synthetic network when evolution rate=30%

shown in Fig.12 and Fig.13. On the whole, our proposed method DOCET exhibits remarkable performance on the VAST and DBLP networks. At the 8^{th}

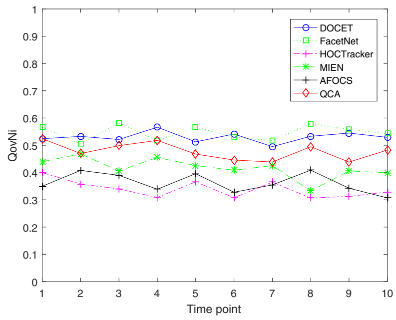


Fig. 11. Q_{ov}^{Ni} at different time steps of the synthetic network when evolution rate=30%

time step of the VAST network, the structure changes drastically [29] (and the following subsection will reveal these changes), resulting in significant performance fluctuation for all six methods. FacetNet and MIEN methods partition the network based on the principle of modularity maximization. Therefore, they can obtain higher Q_{ov}^{Ni}. Our proposed DOCET method, though not based on modularity optimization, still exhibits comparable performance.

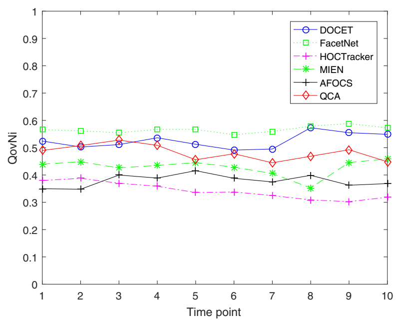


Fig. 12. Q_{ov}^{Ni} at different time steps of the VAST network

3.4. The performance of community evolution tracking

In this subsection, our proposed method DOCET is evaluated from the perspective of community evolution tracking. Among the nine baseline algorithm, FacetNet [28] and HOCTracker [45] mainly focus on dynamic community evolu-

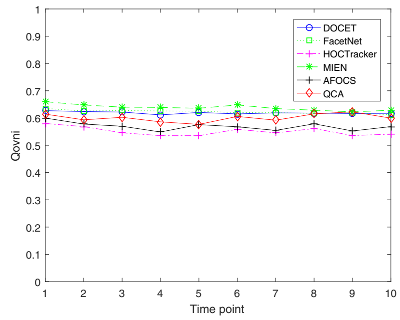


Fig. 13. Q_{ov}^{Ni} at different time steps of the DBLP network

tion tracking. Therefore, our proposed method DOCET is compared with these two methods.

3.4.1. Test on synthetic networks

In this subsection, the community number of generated synthetic networks is variable, and more evolution events are introduced to the synthetic network according to the method described in [29]. The initial synthetic network contains 400 nodes that belong to 5 communities with 80 nodes in each community. 10 consecutive time steps are generated. We randomly choose 10 nodes from each community and make those nodes form a new community, which lasts for 5 time steps, and then these nodes return to the original communities [29]. Thus, the numbers of communities at each time step are 5, 6, 7, 8, 9, 9, 8, 7, 6 and 5, respectively. The same as in [29], the average degree of each node is set as half the size of the community that the node belongs to. In order to introduce more evolution events, from the second time step, we randomly remove 20 exiting nodes from each time step and then randomly add 20 new nodes.

Fig.14 shows the Q_{ov}^{Ni} of the three methods over the 10 time steps. The FacetNet method is based on modularity optimization thus it obtains excellent Q_{ov}^{Ni} performance. Our proposed DOCET method, though not based on modularity optimization, still exhibits comparable performance. Furthermore, the former results of this section have shown that DOCET yields higher accuracy in community partition than other methods.

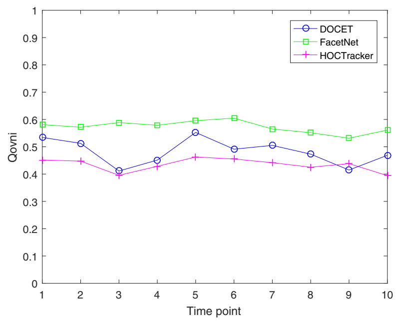


Fig. 14. The Q_{ov}^{Ni} of the three methods on synthetic networks over the 10 time steps

Fig.15 shows the community numbers detected by the three methods over the 10 time steps. Compared with other two methods, our proposed DOCET method can precisely identify the community number at most time steps.

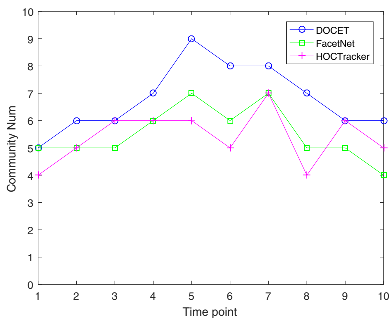


Fig. 15. The community numbers detected by the three methods on synthetic networks over the 10 time steps.

3.4.2. Test on real-world social network

In this subsection, our proposed DOCET method is applied on the VAST and DBLP networks to identify evolution events.

The first real-world network is VAST, a small scale social network. Table 7 shows the core nodes of each time step identified by our DOCET method. We have known that nodes 1, 2, 3 and 5 are the four important nodes in the first

Table 7

The core nodes of each time step identified by DOCET

Time Steps	Core Nodes
1	1,4,8,13,14,19,21,23,24,30,41,42,49,52,53,61,120,275
2	1,2,3,5,7,8,9,11,13,14,18,19,23,41,61,103,118,120,158,172,229
3	0,1,5,7,9,13,17,18,23,34,38,41,51,53,54,61,79,154,170,199
4	0,1,3,4,5,7,9,13,18,21,23,41,49,61,103,172,305
5	0,1,6,7,13,14,20,21,23,38,41,52,61,90,92,97,109,117,118,172
6	1,3,5,7,9,15,21,23,24,38,41,42,52,61,112,172,205,275,305
7	0,1,3,5,7,8,11,15,17,20,23,27,30,49,52,54,61,103,158,199
8	6,7,9,13,20,21,23,30,41,54,158,162,309
9	7,8,13,19,20,38,41,42,49,52,54,61,92,144,306,309,397
10	0,4,13,19,21,23,35,38,41,92,93,103,118,120,148,158,162,306,309,360,397

seven time steps, while nodes 306, 309, 360 and 397 are the four key nodes for the last three time steps[59]. In this subsection, we only consider the community evolution events with respect to these eight nodes, and Fig.16 shows corresponding results identified by our DOCET method. For convenience of description, we use the identifier of the core node to represent the corresponding community. At the initial time step, nodes 2, 3 and 5 directly connect with node 1. These four nodes are in the same community, and node 1 is the core node of this community. When it comes to the 2^{nd} time step, communities 2, 3 and 5 split from the community 1, resulting in the shrinkage of the community 1. At the 3^{rd} time step, communities 2 and 3 are merged into community 34. They split from the community 34 and become independent communities again at the following 4^{th} time step. Communities 2, 3 and 5 are merged into the community 1 at the 5^{th} time step because they are directly related with each other at this time step. In the following 6^{th} time step, communities 3 and 5 split from the community 1, resulting in the shrinkage of the community 1. Node 2 is an internal node of the community 5. The community structure remains unchanged at the 7^{th}

time step. At the 8^{th} time step, nodes 1, 2, 3 and 5 all disappear, and the corresponding communities are dead. At the same time step, a new community i.e. community 309 emerges. Nodes 306, 360 and 397 are all the internal nodes of community 309. When it comes to the 9^{th} time step, communities 306 and 397 split from the community 306, resulting in the shrinkage of the community 306. The community 360 splits from the community 306 in the following 10^{th} time step.

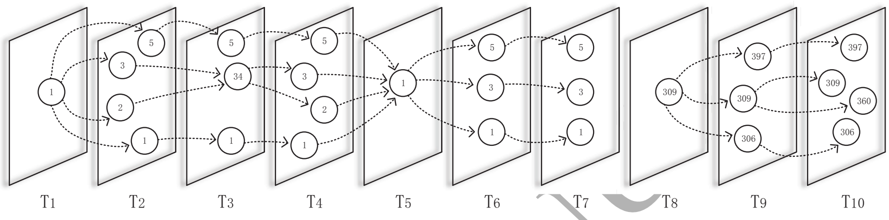


Fig. 16. The part evolution events of the VAST network identified by DOCET

The second network is DBLP, a large scale network with 1,314,050 nodes. There are many evolution events among these nodes (i.e. researchers). Fig.17 only exhibits parts of them identified by our DOCET method. For convenience of description, we use the identifier of the core node to represent the corresponding community. Fig.17 shows that some cooperation relationships are short-term, such as the community 11377 at 6^{th} time step and the community 16519 at 8^{th} time step; There are some researchers cooperating with each other for a long time, such as the community 25276, which maintains the cooperation relationship from 1^{st} time step to 10^{th} time step. Split and merge events are also identified. For example, at the 7^{rd} time step, the community 123063 is merged into the community 3350, resulting in the growth of the community 3350. At the same time step, the community 68303 split from the community 25276, resulting in the shrink of the community 25276.

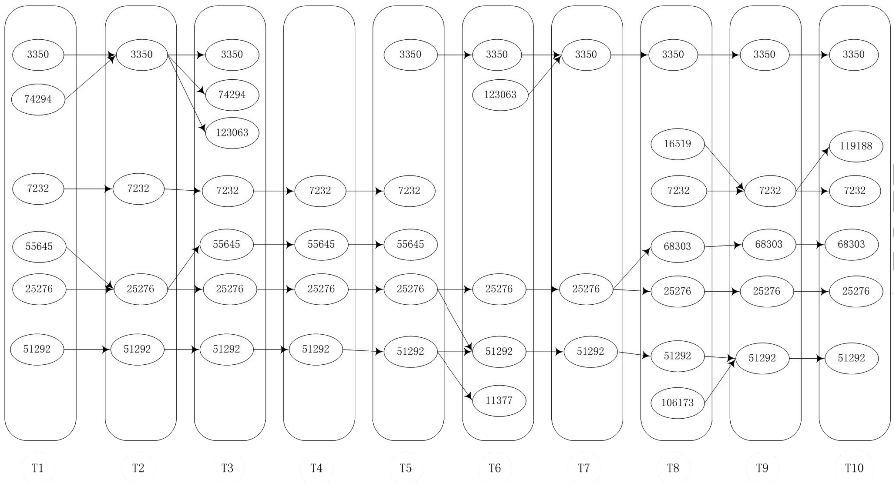


Fig. 17. The part evolution events of the DBLP network identified by DOCET

4. Conclusion

It is a challenging task to simultaneously address overlapping community detection, dynamic community identification and community evolution analysis with one single method, thus most traditional studies focus on only one or two of them. By utilizing the topology potential field, this paper provides a novel method to track the evolution of overlapping communities in dynamic social networks, which solves the above three problems with one single method. The proposed DOCET method partitions the initial overlapping community structure based on node location analysis in the peak-valley structure of the topology potential field, and then incrementally updates the overlapping community structure based on influence scope analysis in the topology potential field. Finally, evolution events within the whole observation time step are tracked based on the variation of core nodes in the topology potential field. Extensive experiments on both synthetic and real-world networks have demonstrated the effectiveness and efficiency of our DOCET method over the state-of-the-art methods from an overall perspective.

Our proposed DOCET method partitions social networks only based on the

ACCEPTED MANUSCRIPT

structure information of networks. In the future, we will take more node properties into consideration, and improve our model to make it adaptable to different network structures, such as mesh networks and fully connected networks.

The proposed method has many prospective applications. To name a few, it can be used to conduct personalized recommendation based on the identified overlapping community structure, or to predict the evolution of social networks overtime and analyze the epidemic spreading process of social networks as described in [60].

Acknowledgements

This work was supported by the Fundamental Research Funds for the Central Universities (No.2018XKQY).

References

References

[1] N. P. Nguyen, T. N. Dinh, Y. Xuan, M. T. Thai, Adaptive algorithms for detecting community structure in dynamic social networks, in: Proceedings of the 30th IEEE International Conference on Computer Communications, IEEE, 2011, pp. 2282–2290.

[2] Y. Xu, H. Xu, D. Zhang, A novel disjoint community detection algorithm for social networks based on backbone degree and expansion, Expert Systems with Applications 42 (21) (2015) 8349–8360.

[3] Z. Wang, J. Xi, Y. Xing, Z. Hu, Community number estimation for community detection in complex networks, Journal of Information Science and Engineering 33 (5) (2017) 1323–1341.

[4] G. K. Orman, V. Labatut, H. Cherifi, Comparative evaluation of community detection algorithms: a topological approach, Journal of Statistical Mechanics: Theory and Experiment 2012 (08) (2012) P08001.

[5] G. Palla, I. Derényi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (7043) (2005) 814–818.

[6] A. Amelio, C. Pizzuti, Overlapping community discovery methods: a survey, in: Social Networks: Analysis and Case Studies, Springer, 2014, pp. 105–125.

[7] Z. Wang, Z. Li, X. Ding, J. Tang, Overlapping community detection based on node location analysis, Knowledge-Based Systems 105 (2016) 225–235.

[8] Y. Xu, H. Xu, D. Zhang, Y. Zhang, Finding overlapping community from social networks based on community forest model, Knowledge-Based Systems 109 (2016) 238–255.

[9] X. Wang, J. Li, Detecting communities by the core-vertex and intimate degree in complex networks, Physica A 392 (2013) 2555–2563.

[10] Y. Cui, X. Wang, J. Li, Detecting overlapping communities in networks using the maximal sub-graph and the clustering coefficient, Physica A 405 (2014) 85–91.

[11] Y. Cui, X. Wang, J. Eustace, Detecting community structure via the maximal sub-graphs and belonging degrees in complex networks, Physica A 416 (2014) 198–207.

[12] J. Li, X. Wang, Y. Cui, Uncovering the overlapping community structure of complex networks by maximal cliques, Physica A 415 (2014) 398–406.

[13] Y. Cui, X. Wang, Detecting one-mode communities in bipartite networks by bipartite clustering triangular, Physica A 457 (2016) 307–315.

[14] J. Li, X. Wang, J. Eustace, Detecting overlapping communities by seed community in weighted complex networks, Physica A 392 (23) (2013) 6125–6134.

[15] X. Wang, X. Qin, Asymmetric intimacy and algorithm for detecting communities in bipartite networks, Physica A 462 (2016) 569–578.

[16] Y. Cui, X. Wang, Uncovering overlapping community structures by the key bi-community and intimate degree in bipartite networks, Physica A 407 (2014) 7–14.

[17] J. Eustace, X. Wang, Y. Cui, Overlapping community detection using neighborhood ratio matrix, Physica A 421 (2015) 510–521.

[18] J. Eustace, X. Wang, Y. Cui, Community detection using local neighborhood in complex networks, Physica A 436 (2015) 665–677.

[19] J. Eustace, X. Wang, J. Li, Approximating web communities using subspace decomposition, Knowledge-Based Systems 70 (2014) 118–127.

[20] N. P. Nguyen, Y. Xuan, M. T. Thai, On detection of community structure in dynamic social networks, in: Handbook of Optimization in Complex Networks, Springer, 2012, pp. 307–347.

[21] G. Palla, A.-L. Barabási, T. Vicsek, Quantifying social group evolution, Nature 446 (7136) (2007) 664–667.

[22] A. Lancichinetti, F. Radicchi, J. J. Ramasco, S. Fortunato, Finding statistically significant communities in networks, PloS One 6 (4) (2011) e18961.

[23] T. Berger-Wolf, C. Tantipathananandh, D. Kempe, Dynamic community identification, in: Link Mining: Models, Algorithms, and Applications, Springer, 2010, pp. 307–336.

[24] D. Chakrabarti, R. Kumar, A. Tomkins, Evolutionary clustering, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2006, pp. 554–560.

[25] F. Folino, C. Pizzuti, An evolutionary multiobjective approach for community discovery in dynamic networks, IEEE Transactions on Knowledge and Data Engineering 26 (8) (2014) 1838–1852.

[26] C. Guo, J. Wang, Z. Zhang, Evolutionary community structure discovery in dynamic weighted networks, Physica A 413 (2014) 565–576.

[27] Y. Chi, X. Song, D. Zhou, K. Hino, B. L. Tseng, On evolutionary spectral clustering, ACM Transactions on Knowledge Discovery from Data 3 (4) (2009) 17.

[28] Y. Lin, Y. Chi, S. Zhu, H. Sundaram, B. L. Tseng, Analyzing communities and their evolutions in dynamic social networks, ACM Transactions on Knowledge Discovery from Data 3 (2) (2009) 8.

[29] M.-S. Kim, J. Han, A particle-and-density based evolutionary clustering method for dynamic networks, Proceedings of the VLDB Endowment 2 (1) (2009) 622–633.

[30] X. Li, B. Wu, Q. Guo, X. Zeng, C. Shi, Dynamic community detection algorithm based on incremental identification, in: Proceedings of the 2015 IEEE International Conference on Data Mining Workshop, IEEE, 2015, pp. 900–907.

[31] J. Shang, L. Liu, X. Li, F. Xie, C. Wu, Targeted revision: A learning-based approach for incremental community detection in dynamic networks, Physica A 443 (2016) 70–85.

[32] T. N. Dinh, Y. Xuan, M. T. Thai, Towards social-aware routing in dynamic communication networks, in: Proceedings of the Performance Computing and Communications Conference (IPCCC), IEEE, 2009, pp. 161–168.

[33] H. Ning, W. Xu, Y. Chi, Y. Gong, T. Huang, Incremental spectral clustering with application to monitoring of evolving blog communities, in: Proceedings of the 2007 SIAM International Conference on Data Mining, SIAM, 2007, pp. 261–272.

[34] J. Sun, C. Faloutsos, S. Papadimitriou, P. S. Yu, Graphscope: parameter-free mining of large time-evolving graphs, in: Proceedings of the 13th ACM

SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2007, pp. 687–696.

[35] T. Falkowski, A. Barth, M. Spiliopoulou, Studying community dynamics with an incremental graph mining algorithm, in: Proceedings of the 14th Americas Conference on Information Systems, 2008, p. 29.

[36] H. Sun, J. Huang, X. Zhang, J. Liu, D. Wang, H. Liu, J. Zou, Q. Song, Incorder: Incremental density-based community detection in dynamic networks, Knowledge-Based Systems 72 (2014) 1–12.

[37] J. Xie, M. Chen, B. K. Szymanski, Labelrankt: Incremental community detection in dynamic networks via label propagation, in: Proceedings of the Workshop on Dynamic Networks Management and Mining, ACM, 2013, pp. 25–32.

[38] Y. Xin, Z. Xie, J. Yang, An adaptive random walk sampling method on dynamic community detection, Expert Systems with Applications 58 (2016) 10–19.

[39] P. Bródka, S. Saganowski, P. Kazienko, Ged: the method for group evolution discovery in social networks, Social Network Analysis and Mining 3 (1) (2013) 1–14.

[40] N. P. Nguyen, T. N. Dinh, S. Tokala, M. T. Thai, Overlapping communities in dynamic networks: their detection and mobile applications, in: Proceedings of the 17th Annual International Conference on Mobile Computing and Networking, ACM, 2011, pp. 85–96.

[41] R. Cazabet, F. Amblard, C. Hanachi, Detection of overlapping communities in dynamical social networks, in: Proceedings of the IEEE 2nd International Conference on Social Computing, IEEE, 2010, pp. 309–314.

[42] Y. Wang, B. Wu, N. Du, Community evolution of social network: feature, algorithm and model, arXiv preprint arXiv:0804.4356.

[43] S. Asur, S. Parthasarathy, D. Ucar, An event-based framework for characterizing the evolutionary behavior of interaction graphs, ACM Transactions on Knowledge Discovery from Data 3 (4) (2009) 16.

[44] N. Du, X. Jia, J. Gao, V. Gopalakrishnan, A. Zhang, Tracking temporal community strength in dynamic networks, IEEE Transactions on Knowledge and Data Engineering 27 (11) (2015) 3125–3137.

[45] S. Y. Bhat, M. Abulaish, Hoctracker: Tracking the evolution of hierarchical and overlapping communities in dynamic social networks, IEEE Transactions on Knowledge and Data Engineering 27 (4) (2015) 1019–1013.

[46] W. Yu, C. C. Aggarwal, W. Wang, Temporally factorized network modeling for evolutionary network analysis, in: Proceedings of the 10th ACM International Conference on Web Search and Data Mining, ACM, 2017, pp. 455–464.

[47] M. Li, Y. Lu, J. Wang, F.-X. Wu, Y. Pan, A topology potential-based method for identifying essential proteins from ppi networks, IEEE/ACM Transactions on Computational Biology and Bioinformatics 12 (2) (2015) 372–383.

[48] Y. Lin, S. Wang, L. Zhao, D. Wang, Topology potential-based parameter selecting for support vector machine, in: Proceedings of the International Conference on Advanced Data Mining and Applications, Springer, 2014, pp. 513–522.

[49] Y. Han, D. Li, T. Wang, Identifying different community members in complex networks based on topology potential, Frontiers of Computer Science in China 5 (1) (2011) 87–99.

[50] A. Lancichinetti, S. Fortunato, Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities, Physical Review E 80 (1) (2009) 016118.

[51] Y. Liu, H. Gao, X. Kang, Q. Liu, R. Wang, Z. Qin, Fast community discovery and its evolution tracking in time-evolving social networks, in: Proceedings of the 2015 IEEE International Conference on Data Mining Workshop, IEEE, 2015, pp. 13–20.

[52] W. Chen, Z. Liu, X. Sun, Y. Wang, A game-theoretic framework to identify overlapping communities in social networks, Data Mining and Knowledge Discovery 21 (2) (2010) 224–240.

[53] S. Gregory, Finding overlapping communities in networks by label propagation, New Journal of Physics 12 (10) (2010) 103018.

[54] J. Huang, H. Sun, J. Han, H. Deng, Y. Sun, Y. Liu, Shrink: a structural clustering algorithm for detecting hierarchical communities in networks, in: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, ACM, 2010, pp. 219–228.

[55] J. Xie, S. Kelley, B. K. Szymanski, Overlapping community detection in networks: The state-of-the-art and comparative study, ACM Computing Surveys 45 (4) (2013) 43.

[56] L. Danon, A. Diaz-Guilera, J. Duch, A. Arenas, Comparing community structure identification, Journal of Statistical Mechanics: Theory and Experiment 2005 (09) (2005) P09008.

[57] L. M. Collins, C. W. Dent, Omega: A general formulation of the rand index of cluster recovery suitable for non-disjoint solutions, Multivariate Behavioral Research 23 (2) (1988) 231–242.

[58] M. E. Newman, M. Girvan, Finding and evaluating community structure in networks, Physical Review E 69 (2) (2004) 026113.

[59] M. Gong, L. Zhang, J. Ma, L. Jiao, Community detection in dynamic social networks based on multiobjective immune algorithm, Journal of Computer Science and Technology 27 (3) (2012) 455–467.

[60] G. Ren, X. Wang, Epidemic spreading in time-varying community networks, Chaos 24 (2) (2014) 023116.