Received: 4 March 2019 | Revised: 1 April 2019 | Accepted: 7 April 2019
DOI: 10.1002/cpe.5314

SPECIAL ISSUE PAPER

WILEY

Dynamic community detection method based on an improved evolutionary matrix

Ling Wu[1,2] | Qishan Zhang[1] | Kun Guo[2] | Erbao Chen[2] | Chaoyang Xu[3]

[1]School of Economics and Management, Fuzhou University, Fuzhou, China
[2]College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China
[3]School of Information Engineering, Putian University, Putian, China

Correspondence
Kun Guo, College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350002, China.
Email: gukn123@163.com

Funding information
National Natural Science Foundation of China, Grant/Award Number: 61300104, 61300103, and 61672158; Fujian Province High School Science Fund for Distinguished Young Scholars, Grant/Award Number: JA12016; Program for New Century Excellent Talents in Fujian Province University, Grant/Award Number: JA13021; Fujian Natural Science Funds for Distinguished Young Scholar, Grant/Award Number: 2014J06017 and 2015J06014; Major Production and Research Project of Fujian Scientific and Technical Department; Technology Innovation Platform Project of Fujian Province, Grant/Award Number: 2009J1007 and 2014H2005; Fujian Collaborative Innovation Center for Big Data Applications in Governments; Natural Science Foundation of Fujian Province, Grant/Award Number: 2013J01230 and 2014J01232; Industry-Academy Cooperation Project, Grant/Award Number: 2014H6014 and 2017H6008; Haixi Government Big Data Application Cooperative Innovation Center; Natural Science Foundation of Fujian Province of China, Grant/Award Number: 2018J01791 and 2017J01752

Summary

Most of networks in real world obviously present dynamic characteristics over time, and the community structure of adjacent snapshots has a certain degree of instability and temporal smoothing. Traditional Temporal Trade-off algorithms consider that communities found at time *t* depend both on past evolutions. Because this kind of algorithms are based on the hypothesis of short-term smoothness, they can barely find abnormal evolution and group emergence in time. In this paper, a Dynamic Community Detection method based on an improved Evolutionary Matrix (DCDEM) is proposed, and the improved evolutionary matrix combines the community structure detected at the previous time with current network structure to track the evolution. Firstly, the evolutionary matrix transforms original unweighted network into weighted network by incorporating community structure detected at the previous time with current network topology. Secondly, the Overlapping Community Detection based on Edge Density Clustering with New edge Similarity (OCDEDC_NS) algorithm is applied to the evolutionary matrix in order to get edge communities. Thirdly, some small communities are merged to optimize the community structure. Finally, the edge communities are restored to the node overlapping communities. Experiments on both synthetic and real-world networks demonstrate that the proposed algorithm can detect evolutionary community structure in dynamic networks effectively.

KEYWORDS
dynamic community detection, evolutionary matrix, link community structure

1 | INTRODUCTION

Complex networks abstract the real-world representation of complex systems such as the World Wide Web, social networks, and transportation networks. Community detection has become an effective way to reveal the relationship between structure and function of networks. It divides a network into group of nodes, where nodes are densely connected inside but sparsely connected outside. However, most of networks in real world obviously present dynamic characteristics over time. Detecting hidden communities and tracking their evolution process have become important issues and draw lots of attention.[1]

https://doi.org/10.1002/cpe.5314

At present, the studies of dynamic community detection, which are often based on static community detection method or clustering method,[2-7] can be divided into three categories by spatio-temporal approaches.[8] (1) **Instant-optimal Communities Discovery (Instant-optimal CD)**: the Instant-optimal CD considers that communities existing at time *t* only depend on the current state of the network at time *t*. The method first detects clusters at each time step and then matches them across different time-steps[9-12]; (2) **Cross-Time Communities Discovery (Cross-Time CD)**: the Cross-Time CD considers that communities found at time *t* depend both on past and future evolutions[13,14]; (3) **Temporal Trade-off Communities Discovery (Temporal Trade-off CD)**: the Temporal Trade-off CD considers that communities defined at time *t* not only depend on the topology of the network at that time but also on the past evolutions of the topology, past partitions found, or both. Temporal Trade-off ones are incrementally temporally smoothed.[15-25]

The network is consisted of nodes and links which are among nodes. The nodes and links in network are studied by researchers and taken as research objects. Researches of dynamic community detection can be divided into node-based algorithm and link-based algorithm by research object. Most of the existing literature is node-based algorithm.[16,20,26] More recently, some scholars begin to focus on link community discovery.[3,27,28] Link community discovery regard communities as partitions of links rather than nodes and then mapping the link communities to node communities gathering nodes incident to all edges within each link community.

Inspired by the Temporal Trade-off CD and link community discovery mentioned above, this paper proposes a novel evolutionary link community structure discovery algorithm based on evolutionary matrix. First of all, the evolutionary matrix is calculated by combining community structure detected at the previous time with current network structure. Secondly, the Overlapping Community Detection based on Edge Density Clustering with New edge Similarity (OCDEDC_NS) algorithm is applied to the evolutionary matrix in order to get edge communities. In addition, some current communities are merged. Finally, the edge communities are mapped to the node communities. The contributions of DCDEM can be summarized as below:

(1) The historic community structure information is considered in calculating evolutionary matrix, so DCDEM can detect evolutionary community structure in dynamic networks effectively. In addition, abnormal evolution can be found.
(2) DCDEM is based on link community detection and applies OCDEDC_NS in community partition process. Overlapping communities in dynamic network can be detected naturally.
(3) Novel merging strategy that merging some smaller communities can avoid community fragmentation and optimize the final community structure.

The rest of this paper is organized as follows. In Section 2, related work is introduced. A framework for Evolutionary Clustering is proposed in Section 3. In Section 4, fundamental concepts are defined and the proposed algorithm DCDEM is described. In Section 5, experiments of DCDEM algorithm on synthetic and real-world networks are displayed. The conclusions of experimental results and future work are demonstrated in Section 6.

2 | RELATED WORK

It can be seen from above introduction that there are mainly three categories for dynamic community detection (ie, Instant-optimal CD, Cross-Time CD, and Temporal Trade-off CD).

The Instant-optimal CD first detects clusters at each time step and then matches them across different time-steps. Aynaud et al[29] identified Two Stage Approach in the literature. Hopcroft et al[11] proposed an agglomerative clustering method based on cosine similarity of nodes at different snapshots. Palla et al[12] applied CPM(Clique Percolation Method) in community detection for each time. Instant-optimal CD considers that communities existing at time *t* only depend on the current state of the network at *t*. Approaches falling in this class are non-temporally smoothed. Temporal networks include three parts: history networks , current networks and future networks. The affection of history and future evolution are not considered. Thus, it is a very common weak point that historic community structure information is not taken into account.

The Cross-Time CD takes past and future evolutions information into account. Sarkar et al[14] proposed a method based on latent space where it is easier to establish connections between close nodes than between distant nodes. Such approaches are not able to handle real-time community detection because the whole computation on all history steps is needed at each new step of modification of the network.

The Temporal Trade-off CD not only depends on the topology of the network at time *t* but also on the past evolutions of the topology, past partitions found, or both. Chakrabarti et al[16] first put forward evolutionary clustering problems and models in 2006, and some literature works employ the concept of evolutionary clustering.[17-19,21] Some researchers proposed methods based on incremental strategy to detect community structure for large scale networks, such as IDCM[24] and GredMod.[20] In the general case, it is not possible to easily parallelize the community detection, as each step needs the communities found at the previous ones as input, and because of the hypothesis of short-term smoothness, this kind of algorithm can barely find abnormal evolution and group emergence in time.

In the category of link community discovery, Ahn et al[27] proposed a novel link community detection method by utilizing link structure information. One weakness of this method is that it needs to get the whole network structure to construct hierarchy structure of links and assumes there are no unconnected nodes in the network. In order to handle border edges and isolated edges, Guo et al[3] proposed the Overlapping

Community Detection based on Edge Density Clustering (OCDEDC) algorithm to effectively handle border edges and isolated edges. However, OCDEDC can only detect static community structure.

Inspired by evolutionary clustering and link community detection mentioned above, an evolutionary link community detection method is put forward in this paper.

3 | EVOLUTIONARY CLUSTERING FRAMEWORK

Evolutionary Clustering are methods that detect communities at time t based on the topology of the graph at t and on previously found community structures. Evolutionary clustering framework[16] proposes two implicit conditions for dynamic networks: the community structure of the network at any time should be represented by the current data as much as possible. Secondly, the community structure of adjacent networks will not undergo dramatic changes.

Chakrabarti et al[16] proposed the framework for generic data clustering and applied it to two well-known clustering methods, k-means and agglomerative hierarchical clustering, to deal with evolving data. Thus, they introduced the cost function for generic data objects. A specialized version of this function in the context of dynamic networks has been first introduced by Chi et al[17] and adopted also by Lin et al[19] and Kim and Han.[18] Chi et al[17] defined the cost function as follows:

$$\cos t = \alpha \cdot \mathrm{SC} + (1 - \alpha) \cdot \mathrm{TC} \tag{1}$$

where SC means snapshot cost, TC means temporal cost and α is an input parameter used by the user to emphasize one of the two objectives. When $\alpha = 1$ the approach returns the clustering without temporal smoothing. When $\alpha = 0$, however, the same clustering of the previous time step is produced, i.e. $\mathrm{CR}_t = \mathrm{CR}_{t-1}$. Thus a value between 0 and 1 is used to control the preference degree of each sub-cost.

4 | APPROACH

Historical community structure has significant impact on the current network structure based on the evolutionary clustering framework. The proposed approach can analyze the communities of each node in the social network, and each node can belong to one or more communities. The communities of a node can be expressed as the attributes of the node, and the similarity between each two nodes can be measured by the Jaccard factor in accordance with the attributes of nodes. If Jaccard similarity between two nodes is higher, the relationship between the two nodes is more significant.

Approach is explained in two parts: basic concepts and dynamic community detection algorithm based on evolutionary matrix. In Section 4.1, the basic concepts of the algorithm are explained. In Section 4.2, the design idea, implementation and complexity analysis of DCDEM algorithm are illustrated respectively.

4.1 | Basic concepts

In this section, the basic concepts of dynamic community detection are explained.

Definition 1 (Dynamic Network Representation).
The dynamic network sequence with T time slices is defined as $G = \{G^1, G^2, \ldots, G^T\}$, in which $G^t = (V^t, E^t)$ represents the network at time t, $V^t = \{V_1^t, V_2^t, \ldots, V_{N^t}^t\}$ is the vertex set of the network at time t, $E^t = \{e_{ij}^t \mid i, j \in (1, N^t)\}$ is the edge set at time t. The adjacency matrix $W^t = [w_{ij}^t]$ is used to represent the connection of nodes at time t.

Definition 2 (Evolutionary Matrix).
The evolutionary matrix $U^t = [u_{ij}^t]$ at time t in the dynamic network G is defined by

$$u_{ij}^t = \begin{cases} w_{ij}^t & \text{if } t = 1; \\ (1-\beta) w_{ij}^t + \beta \gamma_{ij}^t & \text{if } t > 1, \end{cases} \tag{2}$$

where w_{ij}^t denote the connection relationship between node i and node j, and $w_{ij}^t = 1$ when node i and node j is connected, $w_{ij}^t = 0$ otherwise. Also, $\beta \in [0, 1]$ represents weight factor.

The community structure subordination γ_{ij}^{t-1} between node i and node j at $t - 1$ time in the dynamic network G is defined as follows.

Definition 3 (Community Structure Subordination).

$$\gamma_{ij}^{t-1} = \begin{cases} 0 & \text{if } i,j \notin V^{t-1}; \\ \frac{|G^{t-1}(i) \cap G^{t-1}(j)|}{|G^{t-1}(i) \cup G^{t-1}(j)|} & \text{otherwise,} \end{cases} \tag{3}$$

where $G^{t-1}(i)$ represents a set of the community of node i at $t-1$ time in the dynamic network G.

After introducing the evolutionary matrix, the edges between nodes in the network contain weights so that the original weightless network should be transformed into a weighted network. In the literature,[3] the original edge similarity formula in OCDEDC algorithm does not consider weights. However, the modified edge similarity formula is needed and is given by the following definition.

Definition 4 (Edge Similarity).
Let e_{ij}^t be the edge between node i and node j at time t. The edge similarity $sim(e_{ik}^t, e_{kj}^t)$ related to $U^t[u_{ij}^t]$ is defined by

$$sim(e_{ik}^t, e_{kj}^t) = \frac{\sum_{z \in N(i) \cap N(j)} \left(u_{iz}^t + u_{jz}^t\right)}{\sum_{z \in N(i)} u_{iz}^t + \sum_{z \in N(j)} u_{jz}^t}, \tag{4}$$

in the dynamic network G, and where $N(i)$ represents the neighbor node of the node i. Note that the edge similarity is zero unless e_{ik}^t and e_{kj}^t contain a common node.

Definition 5 (Edge Community Density).
Assume that $Set_{LC}^t(M) = \{LC_1^t, LC_2^t, \dots, LC_M^t\}$ is an M edge partition at time t in the network G^t. The density $D_{LC_i^t}$ of the LC_i^t and the total edge community density D^t are defined as follows, respectively:

$$D_{LC_i^t} = \frac{e_{LC_i^t} - \left(n_{LC_i^t} - 1\right)}{n_{LC_i^t}\left(n_{LC_i^t} - 1\right)/2 - \left(n_{LC_i^t} - 1\right)}, \tag{5}$$

$$D^t = \sum_{i=1}^{M} D_{LC_i^t}, \tag{6}$$

in the dynamic network G, and where $N(i)$ represents the neighbor node of the node i. Note that the edge similarity is zero unless e_{ik}^t and e_{kj}^t contain a common node.

Definition 6 (Edge Community Density Increment).
Let LC_p and LC_q be two edge communities at time t in the network G, the edge community density increment which is merged by ΔD_{pq}^t communities LC_p and LC_q is defined as follows:

$$\Delta D_{pq}^t = D_{LC_p \cup LC_q} - \left(D_{LC_p} + D_{LC_q}\right). \tag{7}$$

Definition 7 (Abnormal Evolution).
Given any dynamic network sequence $G = \{G^1, G^2, \dots, G^T\}$, if there exists a time slice $G^t = (V^t, E^t)$ that differs significantly from its neighbor time slices $G^{t-\delta}, \dots, G^{t+\delta}$, =1, 2, 3, …, G is called a dynamic network sequence with abnormal evolution.

4.2 | Dynamic Community Detection algorithm based on Evolutionary Matrix (DCDEM)

4.2.1 | Algorithm design idea

The basic idea of DCDEM algorithm is to combine the community structure of the previous network with current network topological structure into the evolutionary matrix of the dynamic community. Recall the evolutionary matrix appeared in the Definition 2. The edge similarity proposed by Definition 4 is used to replace the corresponding edge similarity in OCDEDC algorithm, such that it can be applied directly to the weighted network. Furthermore, in order to optimize the obtained edge community, the merging operation is taken into DCDEM algorithm.

4.2.2 | Algorithm implementation

The implementation of DCDEM algorithm is shown in Table 1, and the implementation of merge() function is shown in Table 2. Explanation in detail is as bellow.

(1) DCDEM Algorithm
In order to obtain the initial community structure which is stored in variable preCommStr, DCDEM applies OCDEDC in the initial network G^1 at Step 1 of Table 1. While there still is next time network G^t, DCDEM calculates U^t according to Equation (2) based on the current network

TABLE 1 DCDEM Algorithm

```
DCDEM Algorithm
Input: network sequence $\{G^1, G^2, \dots, G^T\}$ , the parameter $u, \varepsilon, \beta, \theta$
Output: communities sequence $\{C^1, C^2, \dots, C^T\}$
    1: $preCommStr = OCDEDC(G^1, \varepsilon, u)$  //obtain initial community structure
    2: $t = 2$
    3: while $t <= T$ do
    4:        calculate $U^t$ according to Equation (2)
    5:     update $G^t$ according to $U^t$
    6:     $S_{LC}^t$ = OCDEDC_NS($G^t, u, \varepsilon$)
    7:        $S_{LC'}^t$ = merge $(S_{LC}^t)$ // merge communities
    8:        $C^t$ = transform $(S_{LC}^t)$
    9:     output $C^t$ // output community structure at $t = i$ moment
    10:          $preCommStr = C^t$
    11:      $t = t + 1$
    12: end while
```

TABLE 2 merge() Function

```
Function: merge($Set_{LC}^t, \theta$)
Input: edge communities $Set_{LC}^t$, the parameter $\theta$ of edge communities $Set_{LC}^t$
Output: edge communities $Set_{LC}'^t$ after operation of community merging
    $S_1, S_2 \leftarrow \emptyset$
    sorted($Set_{LC}^t$) // all communities of $Set_{LC}^t$ are sorted in descending order
                          according to the number of nodes in the community
    $z$ = int ($\theta \cdot size(Set_{LC}^t)$)   // calculating $z$
    $S_1, S_2$ = partition($Set_{LC}^t, z$)   // dividing all communities of $Set_{LC}^t$ into
                                  two subsets $S_1$ and $S_2$ by $z$
    FOR $LC_p^t \in S_2$ DO
      $max = -1, k = 0$
      FOR $q, LC_q^t$ in emulate($S_1$) DO
        $\Delta D_{pq}^t$ = calcDensity($S_2, S_1$)  // calculates the community density
                                        increment $\Delta D_{pq}^t$ for edge community
                                        $LC_p^t$ of $S_2$ after merging with the
                                        edge communities $LC_q^t$ of $S_1$
        IF $\Delta D_{pq}^t > max$ THEN
          $max = \Delta D_{pq}^t$
          $k = q$
        END IF
      END FOR
      IF $max > 0$ THEN
        $S_1[k] = S_1[k] \cup S_2[p]$
        $S2$.remove($LC_p^t$)
      END IF
    END FOR
    $Set_{LC}^t = S_1 \cup S_2$
    END
```

G^t and previous community structure preCommStr at Step 4. The element u_{ij} in the evolution matrix U^t is taken as the weight between the nodes i and j in the current network G^t so that the original unweighted network G^t is transformed into the weighted network at Step 5. Because the OCDEDC algorithm is fit for unweighted network, the edge similarity equation in the OCDEDC algorithm is replaced with the Equation (4), and the OCDEDC algorithm with the New edge Similarity is named as OCDEDC_NS. The OCDEDC_NS algorithm is applied to process the updated network G^t and the corresponding edge community S_{LC}^t is obtained at Step 6. The merge() function merges some small communities at Step 7. The implementation of the merging function will state as below. Finally, the transform() function transform the edge communities into node communities at Step 8.

(2) Merge function

In order to effectively reduce community debris, a new merge strategy is proposed. Assume that $Set^t_{LC} = \{LC^t_1, LC^t_2, \dots, LC^t_M\}$ is an M edge partition at time t in the network G^t. First of all, all communities of Set^t_{LC} are sorted in descending order according to the number of nodes in the community. Then, communities of Set^t_{LC} are partitioned into two subsets S_1 and S_2 according to number z. The number of nodes of communities in S_1 must be larger than those in S_2 and lager than z, and S_2 is considered as a candidate community subset in which elements in S_2 would be merged later. Each candidate community in S_2 will be merged with some community in S_1 in order to get the largest increment of community density with Equation (7). For each candidate community in S_2, if the largest increment of community density is negative, there is nothing to do. Conversely, the candidate community in S_2 will be merged with some community in S_1 to gain the largest increment of community density, and the implement of merge() function is shown in Table 2 as follows.

At Step 2 in Table 2, a function sorted() has been inserted which sort all communities of Set^t_{LC} in descending order according to the number of nodes in the community. At Step 4, a function partition() is assigned which divides all communities of Set^t_{LC} into two subsets S_1 and S_2 according to z. The double FOR loops (Step 5 to 18) calculates the community density increment ΔD^t_{pq} for each edge community LC^t_p of S_2 after merging with the edge communities LC^t_q of S_1 and then preserves the maximum density increment max and the corresponding edge community index number k of S_1. If max greater than zero, LC^t_p and LC^t_k will be merged because the merged community density can be increased. Conversely, the two community LC^t_p and LC^t_k will not be merged.

4.2.3 | Algorithm complexity analysis

The complexity analysis of DCDEM algorithm includes the time complexity and the spatial complexity. In order to understand the following complexity analysis, the notation of network in Definition 1 should be recalled.

(1) The time complexity analysis

In the algorithm DCDEM, the time complexity of calculating the evolutionary matrix U^t is $O(m^t)$, the time complexity of updating the evolutionary matrix U^t is $O(m^t)$ and the time complexity of algorithm OCDEDC_NS is $O(k^t m^t)$. It is assumed that the total number of edge communities is p before merging, the time complexity of function merge() is $O(p^2)$. Consequently, the time complexity of algorithm DCDEM in G^t is $O(m^t + m^t + k^t m^t + p^2)$. Considering in real-world complex networks, p is far less than m, and k is also far less than m, so the time complexity of algorithm DCDEM in G^t can be simplified as $O(m^t)$. Thus, the time complexity of algorithm DCDEM in the whole dynamic network G is $O(Tm^t)$, where T is a constant, so the DEDCM is near linear complexity. It is confirmed by the Figure 9.

(2) The spatial complexity analysis

In the algorithm DCDEM in G^t, in order to compute the evolutionary matrix U^t, the $t - 1$ time community structure and the current time network G^t are needed. The spatial complexity of storing community structure at the last moment is $O(n^{t-1})$, the spatial complexity of storing community structure at the current moment is $O(m^t)$ and the spatial complexity of storing the evolutionary matrix U^t is $O(m^t)$. Totally, the spatial complexity of calculating evolutionary matrix is $O(2m^t + n^{t-1})$. The spatial complexity of algorithm OCDEDC_NS is $O(m^t)$. In summary, the spatial complexity of algorithm DCDEM is $O(m^t)$.

5 | EXPERIMENTS

In order to verify the performance of DCDEM algorithm, experiments were carried out on different scale artificial networks and real-world networks. Experimental environment: Win 7 operating system, Intel (R) Core (TM) i5-3210M 2.50 GHz CPU, 2GB memory. The DCDEM algorithm code is based on Python 3.6.

Experiments section consists of three subsections: datasets description, baselines for comparison, description for metrics and result and discussion.

5.1 | Datasets description

The datasets used in experiments include two artificial networks and two real networks, consequently, the description of artificial networks and real networks is following.

5.1.1 | Artificial networks

In this paper, dynamic network generation tool,[26] which is improved from the static network generation LFR and can generate artificial networks of different evolutionary types, is used to generate artificial dynamic network N_1 and N_2. The setting of the generating parameters in artificial dynamic networks N_1 and N_2 is presented in Table 3.

Notations of parameters in Table 3 are shown as below. N is the number of nodes, k is the degree of nodes, $maxk$ is the maximum degree of nodes, mu is the complexity of the network, on is the number of overlapping nodes, om is the number of communities overlapping nodes

TABLE 3 The generating parameters of artificial networks

The Label of Networks	Parameter
N_1	$N = 1000, k = 15, maxk = 20, mu. = 0.2, p = 0.1$ $on = 300, om = 2, t = 7$
N_2	N = 1000, k = 15, maxk = 20, mu. = 0.2, on = 300, om = 2, $t = 7, expand = 3, contract = 2, r = 0.2$

belongs to, t is the number of time slices generated by dynamic networks, in particular, parameter p in N_1 is the probability of nodes change community ownership in adjacent time slices. The parameter *expand* in N_2 indicates the number of community which grows in each time slice, *contract* indicates the number of community which contracts in each time slice, and parameter r indicates the rate of community growth or community contraction. Therefore, the dynamic network N_1 contains seven moments, and compared with the previous time moment, the node always separates from the existing community with a probability of $p = 0.1$, the dynamic network N_2 contains seven moments, and compared with the previous time moment, the network always contains three growing communities and two contracting communities at each time, and the rate of growing and contracting is $r = 0.2$.

5.1.2 | Real networks

The DCDEM algorithm was tested on Enron mail dataset and as733 dataset. The introduction and analysis of datasets is in following.

(1) Enron mail dataset[30]
Enron mail dataset records the email communication data of Enron employees. The Enron mail dataset contains 619446 valid mail and 158 account information. William Cohen of Carnegie Mellon University made the dataset available online for interested researchers. This paper intercepts the data from January 1, 2000 to December 31, 2001, and takes each three months as a snapshot of the network at one time. Hence eight snapshots of the network are obtained.

(2) The as733 dataset[31]
The as733 dataset records 733 daily communication logs between servers from November 1997 to January 2000. The as733 dataset contains 1036474 nodes and 24313233 edges. The edges in the network always change with time. This paper selects communication log of 40 days from March 1, 1998 to April 10, 1998. Hence, forty snapshots of the network are obtained.

5.1.3 | The analysis of datasets

Edge number is changing with time passing in Enron mail dataset and as733 dataset. The changing of edge in the two real-world network is visualized in Figure 1 and Figure 2.

Figure 1 shows that the added edges of the network are always larger than the reduced edges at any time and the change number of the network's edge is relatively flat before time 5, but the change of the number of edge is more violent after time 5 because Enron company was involved in the scandal during this period, there was a large increase in mail communication and a large fluctuation in the network structure during this period. Such violent change in the network is called abnormal evolution.

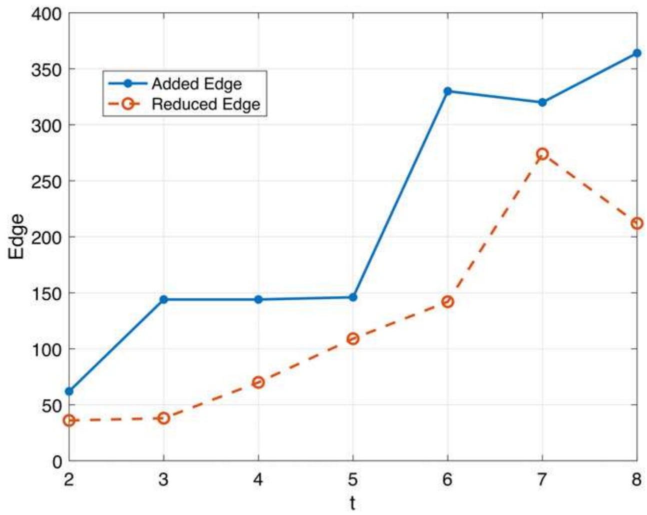

FIGURE 1 Changes in the number of edges in Enron mail dataset between adjacent times

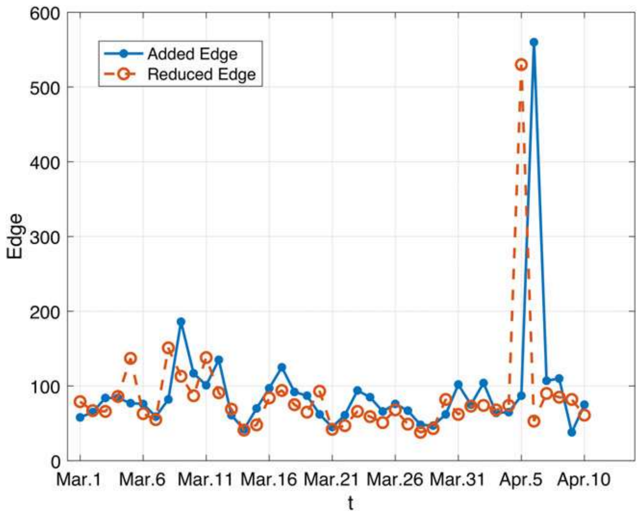

FIGURE 2 Changes in the number of edges in as733 dataset between adjacent times

Figure 2 shows that the curve is relatively flat except for the point corresponding to April 5th. There was a sudden spike change in the network. Such dramatic change in the network is called abnormal evolution.

5.2 | Baselines for comparison

IDCM,[24] GredMod,[20] and OCDEDC[3] are taken as baselines for comparison in this paper.

5.3 | Description for metrics

NMI and EQ are adopted as evaluation metrics in the experiment, the equations of evaluation metrics present as follows.

(1) NMI[32] (Normalized Mutual Information)

NMI is evaluation metrics for experimental results of artificial networks. It can reflect the difference between the community structure identified by the algorithm and the real community. The NMI equation is as below,

$$\mathrm{NMI}=\frac{-2\sum_{i=1}^{C_T}\sum_{j=1}^{C_A}N_{ij}\log\left(\frac{N_{ij}N}{N_{i\cdot}N_{\cdot j}}\right)}{\sum_{i=1}^{C_T}N_{i\cdot}\log\left(\frac{N_{i\cdot}}{N}\right)+\sum_{j=1}^{C_A}N_{\cdot j}\log\left(\frac{N_{\cdot j}}{N}\right)}, \tag{8}$$

where C_T represent the number of the real community, C_A represent the number of the community detected by the algorithm, $N_{i\cdot}$ represents the sum of line i, and $N_{\cdot j}$ represents the sum of column j. NMI $\in$ [0, 1], the larger the NMI value, the better the result of community partition is. When the NMI = 1, it indicates that T and A are the same division of the network.

(2) EQ (Extension Modularity)

EQ is evaluation metrics for experimental results of real networks. Shen's overlap modularity EQ[33] is used here as follows:

$$\mathrm{EQ}=\frac{1}{2m}\sum_{i}\sum_{v,w\in C_i}\frac{1}{o_v o_w}\left[A_{vw}-\frac{k_v k_w}{2m}\right], \tag{9}$$

where o_v and o_w denote the number of communities to which node v and w belongs, and k_v, k_w are the degree of node v and w, respectively, and m is the number of total nodes in network. A is nodes adjacency matrix, if v, w have edge, then $A_{vw} = 1$, else $A_{vw} = 0$.

5.4 | Result and Discussion

The experiments deployed on three parts: parameter experiment, precision experiment and running time experiment. The parameter experiment results are shown in Figures 3 and 4. The precision experimental results are shown in Figures 5, 6, 7 and 8. The running time experiment results are shown in Figure 9.

5.4.1 | Parameter experiment

The evolutionary matrix takes into account the community structure of the network at the previous moment and the network topology at the current moment. In Equation (2), the value of parameter β measures the different weight of community structure of the previous time and network

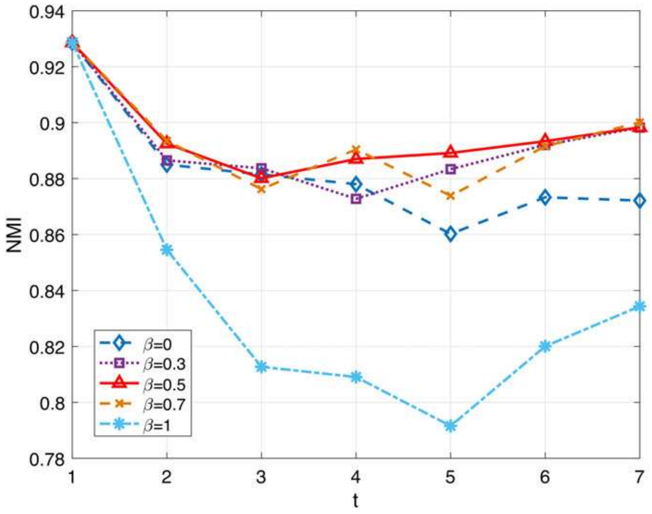

FIGURE 3 Parameter experiment of DCDEM algorithm on N_1 network

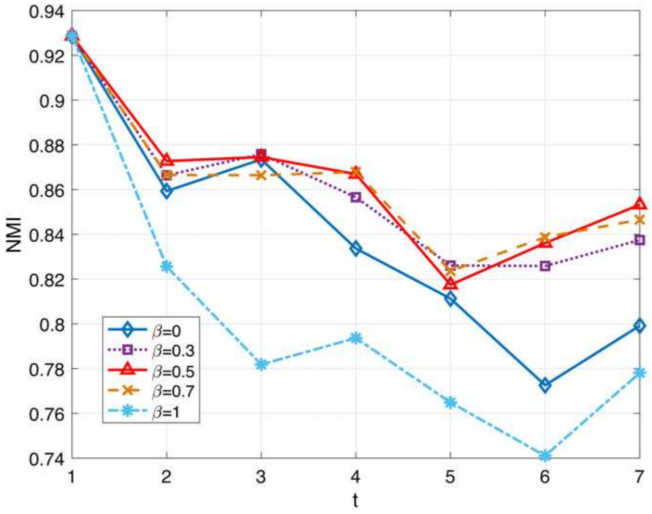

FIGURE 4 Parameter experiment of DCDEM algorithm on N_2 network

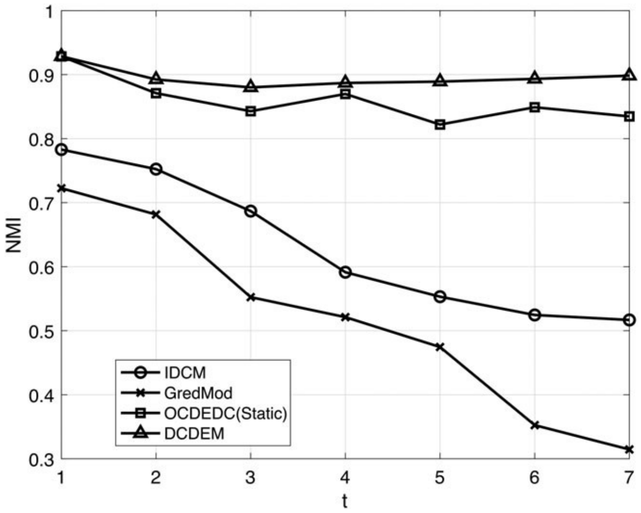

FIGURE 5 The NMI experiment of comparison algorithms on artificial dataset N_1

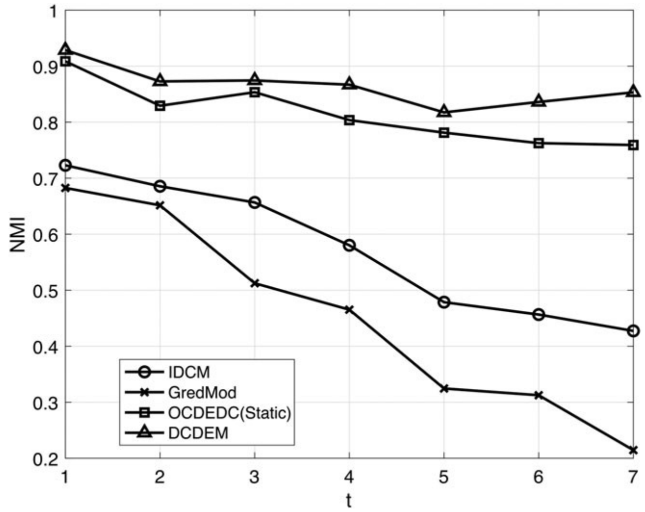

FIGURE 6 The NMI experiment of comparison algorithms on artificial dataset N_2

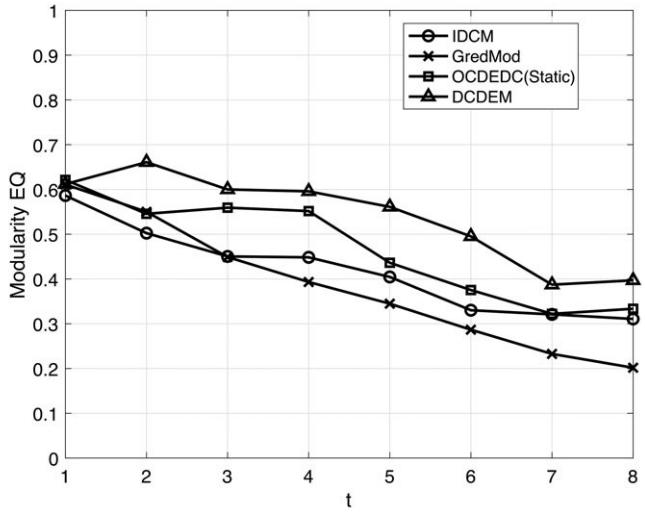

FIGURE 7 The EQ experiment of comparison algorithms on Enron mail dataset

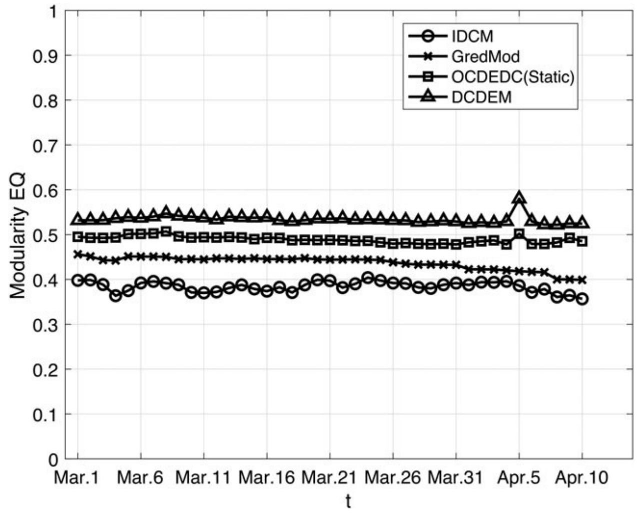

FIGURE 8 The EQ experiment of comparison algorithms on as733 dataset

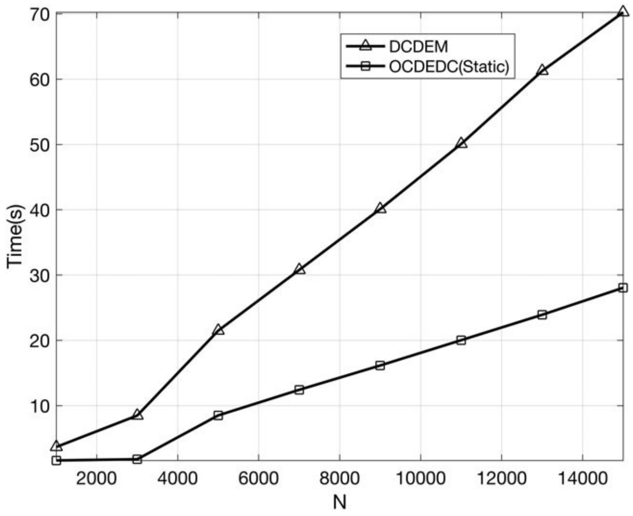

FIGURE 9 The running time of the DEDCM algorithm on N_3

topological structure of the current time. In the evolutionary matrix, the larger the value of β, the larger the proportion of the community structure of the previous time. In order to verify the performance of the evolutionary matrix and find the best range of parameter β, analyzing the NMI variation of DCDEM algorithms with different parameters β on artificial dynamic networks N_1 and N_2, the result is shown in Figure 3 and Figure 4.

In Figure 3 and Figure 4, algorithm has the lowest NMI value and the largest curve fluctuation at all times when β equals to 1 because β equals to 1 means that the algorithm only considers the community structure of the previous moment. When β equals to 0, the NMI curves in N_1 network are relatively stable and have high NMI values at all times in Figure 3, But the NMI curves in N_2 network fluctuates greatly in Figure 4 because the structure of the network N_2 changes greatly between adjacent moment, but the algorithm only take the network topological information of the current time into account and affect the stability of the algorithm. In summary, when the parameter β equals to 0.5, the accuracy of DCDEM algorithm is the highest. Therefore, in the subsequent experiments, parameter β equals to 0.5.

5.4.2 | Precision experiment

In this section, parameter setting in algorithm is shown in Table 4. Subsequently, the NMI experiment of comparison algorithms on artificial dataset N_1 and N_2 is showed in Figure 5 and Figure 6. Finally, the EQ experiments of comparison algorithms on real dataset Enron mail and as733 are showed in Figure 7 and Figure 8.

In Figure 5 and Figure 6, the DCDEM algorithm has the highest NMI values on the two artificial dynamic networks N_1 and N_2, and the NMI curve is stable. Thus, it confirms that the evolutionary matrix which incorporates the community structure of the previous moment and the current network topology structure is effective. The IDCM algorithm and the GredMod algorithm are not ideal at all times on the N_1 and N_2 networks, and the NMI curve of GredMod algorithm is always below the NMI curve of IDCM algorithm. Because the IDCM algorithm and the GredMod algorithm are both incremental dynamic community discovery algorithms, the error accumulation of the algorithm at each time will continue to increases over time. In particular, the GredMod algorithm does not consider the disappearance of edge at the next moment in the incremental strategy, but the artificial network N_1 and N_2 have both edge addition and edge disappearance at each time, so the GredMod algorithm has the lowest NMI value. The NMI value of DCDEM algorithm is always higher than that of static OCDEDC algorithm. There are two main reasons why DCDEM algorithm is better than static OCDEDC algorithm. Firstly, the evolutionary matrix calculation method, which combines the previous and current network topology structures, is helpful to detect a reasonable and highly modular community structure in dynamic networks. Secondly, merging smaller communities with larger ones will help to cut down the total number of communities while reducing the number of small communities, thus improving the modularity of community structure.

As can be seen from Figure 7, the DCDEM algorithm has the highest EQ values in the real dynamic networks Enron mail, and from time 5 to time 7, the EQ value of the DCDEM algorithm dropped significantly because Enron was involved in the scandal during this period, and a large increase in mail communication brought a large fluctuation in the network structure. The magnitude of the edge change also reflects this

TABLE 4 The parameter setting in algorithm

Algorithm	Parameter
IDCM	$Minpts = 5 \sim 9, \varepsilon = 0.2 \sim 0.4.$
GredMod	/
OCDEDC	$u = 3 \sim 10, \varepsilon = 0.3 \sim 0.5, \theta = 0.05, \beta = 0.5$
DCDEM	$\varepsilon = 0.2 \sim 0.4, u = 2 \sim 10$

event in Figure 1. The EQ value of the static OCDEDC algorithm is always lower than the DCDEM algorithm at each moment because the new computation of evolutionary matrix and the new community merging strategy, proposed in the DCDEM algorithm, are effective and can gain higher accuracy in real dynamic networks. The IDCM algorithm and the GredMod algorithm are in a significant downward trend. Because the IDCM algorithm and the GredMod algorithm are both incremental dynamic community discovery algorithms, the error accumulation of the algorithm at each moment will continue to increases over time.

As can be seen from Figure 8, the DCDEM algorithm has the highest EQ values in the real dynamic networks as733, and the EQ curve is stable, except for the point corresponding to April 5th, a small fluctuation but went to stable. Recalled with the magnitude of the edge change in Figure 2, there are also small fluctuations of increased edges curve and reduced edges curve around April 5th, then went to stable. Thus, the EQ curve of the method can accordingly variate to the fluctuation of the network in Figure 2, especially some dramatic change. Therefore, the DCDEM algorithm and the OCDEDC algorithm can better respond to this change, but the IDCM algorithm and the GredMod algorithm cannot reflect this situation. The DCDEM algorithm has the best accuracy, the performance of OCDEDC algorithm is lower than DCDEM algorithm, and the IDCM algorithm has the worst accuracy.

5.4.3 | Running time experiment

To verify the operational efficiency of the DCDEM algorithm, the network generation tool is used to generate an artificial dynamic network N_3. The network N_3 has 7 moments, and the number of nodes N is between 1000 and 15000, the increment number of nodes at each moment is 2000, and the average degree k is 15, therefore, compared with the number of edges at the previous moment, the increment number of edge at current moment is approximately 30,000.

Figure 9 shows the running time of the DCDEM algorithm on the artificial dynamic network N_3. With the linear growth of the number of nodes on N_3, the running time basically shows a linear growth trend. The time complexity of the DCDEM algorithm is able to be approximated as linear, which is consistent with the result of the previous complexity analysis in this paper. In fact, DEDCM with more improvements, such as merge strategies, can lead to more runtime.

6 | CONCLUSIONS AND FUTURE WORK

This paper proposes a new dynamic community discovery algorithm based on the evolutionary matrix and link analysis. First of all, the DCDEM algorithm can detect abnormal evolution when the number of edges changing dramatically with time passing. Meanwhile, experimental results on artificial datasets and real datasets show that the DCDEM algorithm can effectively detect overlapping communities in dynamic network. In addition, in order to conquer community debris, the DCDEM algorithm also proposes a new community merging strategy. Visualization of tracing community evolution and the evolutionary link community mining in dynamic heterogeneous networks will be in the future work. Furthermore, graph neural networks are powerful tools for learning the structure of nodes and links, so the community structure could be detected by graph neural networks in the future.

ACKNOWLEDGMENTS

This work is partly supported by the National Natural Science Foundation of China under grants 61300104, 61300103, and 61672158; the Fujian Province High School Science Fund for Distinguished Young Scholars under grant JA12016; the Program for New Century Excellent Talents in Fujian Province University under grant JA13021; the Fujian Natural Science Funds for Distinguished Young Scholar under grants 2014J06017 and 2015J06014; the Major Production and Research Project of Fujian Scientific and Technical Department; the Technology Innovation Platform Project of Fujian Province under grants 2009J1007 and 2014H2005; the Fujian Collaborative Innovation Center for Big Data Applications in Governments; the Natural Science Foundation of Fujian Province under grants 2013J01230 and 2014J01232; Industry-Academy Cooperation Project under grants 2014H6014 and 2017H6008; Haixi Government Big Data Application Cooperative Innovation Center; and the Natural Science Foundation of Fujian Province of China under the project of 2018J01791 and 2017J01752.

ORCID

Ling Wu https://orcid.org/0000-0001-5293-8701

REFERENCES

1. Wang L, Cheng XQ. Dynamic community in online social networks. *Chin J Comput*. 2015;38:219-237.
2. Chen X, Jian C. Gene expression data clustering based on graph regularized subspace segmentation. *Neurocomputing*. 2014;143:44-50.
3. Guo K, Chen E, Guo W. Overlapping community detection based on edge density. *Pattern Recognit Artif Intell*. 2018;31(8):693-703.
4. Guo K, Guo W, Chen Y, et al. Community discovery by propagating local and global information based on the MapReduce model. *Information Sciences*. 2015;323:73-93.

5. Guo K, Zhang Q. Detecting communities in social networks by local affinity propagation with grey relational analysis. *Grey Syst Theory Appl*. 2015;15(1):31-40.
6. Guo K, Zhang Q. Fast clustering-based anonymization approaches with time constraints for data streams. *Knowl-Based Syst*. 2013;46(95C108).
7. Zhang Q, Qiu Q, Guo W, et al. A social community detection algorithm based on parallel grey label propagation. *Computer Networks*. 2016;107:133-143.
8. Rossetti G, Cazabet R. Community discovery in dynamic networks: a survey. *ACM Comput Surv*. 2018;51(2):35.
9. Bóta A, Krész M, Pluhár A. Dynamic communities and their detection. *Acta Cybernetica*. 2011;20(1):35-52.
10. Falkowski T, Spiliopoulou M. Data mining for community dynamics. Paper presented at: Fourteenth European Conference on Information Systems; 2006; Göteborg, Sweden.
11. Hopcroft J, Khan O, Kulis B, Selman B. Tracking evolving communities in large linked networks. *Proc Natl Acad Sci USA*. 2004;101(Suppl 1):5249-5253.
12. Palla G, Derényi I, Farkas I, Vicsek T. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*. 2005;435(7043):814.
13. Matias C, Miele V. Statistical clustering of temporal networks through a dynamic stochastic block model. *J R Stat Soc Ser B Stat Methodol*. 2017;79(4):1119-1141.
14. Sarkar P, Moore AW. Dynamic social network analysis using latent space models. *SIGKDD Explor*. 2005;7(2):31-40.
15. Aynaud T, Guillaume JL. Static community detection algorithms for evolving networks. Paper presented at: 8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks; 2010; Avignon, France.
16. Chakrabarti D, Kumar R, Tomkins A. Evolutionary clustering. Paper presented at: 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2006; Philadelphia, PA.
17. Chi Y, Song X, Zhou D, Hino K, Tseng BL. On evolutionary spectral clustering. *ACM Trans Knowl Discov Data*. 2009;3(4):1-30.
18. Kim MS, Han J. A particle-and-density based evolutionary clustering method for dynamic networks. *Proc VLDB Endow*. 2009>;2(1):622-633.
19. Lin YR, Zhu S, Sundaram H, Tseng BL. Analyzing communities and their evolutions in dynamic social networks. *ACM Trans Knowl Discov Data*. 2009;3(2):1-31.
20. Shang J, Liu L, Xie F, Chen Z, Miao J, Fang X, Wu C. A real-time detecting algorithm for tracking community structure of dynamic networks. Paper presented at: 6th SNA-KDD Workshop on Social Network Mining and Analysis; 2014; Beijing, China.
21. Rossetti G, Pappalardo L, Pedreschi D, Giannotti F. Tiles: an online algorithm for community discovery in dynamic social networks. *Machine Learning*. 2017;106(8):1213-1241.
22. Wang S, Guo W. Robust co-clustering via dual local learning and high-order matrix factorization. *Knowl-Based Syst*. 2017;138:176-187.
23. Wang Y, Wu B, Pei X. CommTracker: a core-based algorithm of tracking community evolution. *J Front Comput Sci Technol*. 2008;5139:229-240.
24. Xiong Z. An Increment-and-Density Based Community Detection Algorithm for Dynamic Networks [master thesis]. Xi'an, China: Xidian University; 2012.
25. Chen Z, Wilson KA, Jin Y, Hendrix W, Samatova NF. Detecting and tracking community dynamics in evolutionary networks. Paper presented at: IEEE International Conference on Data Mining Workshops; 2011; Vancouver, Canada.
26. Greene D, Doyle D, Cunningham P. Tracking the evolution of communities in dynamic social networks. Paper presented at: International Conference on Advances in Social Networks Analysis and Mining; 2010; Odense, Denmark.
27. Ahn YY, Bagrow JP, Lehmann S. Link communities reveal multiscale complexity in networks. *Nature*. 2010;466(7307):761-764.
28. Evans TS, Lambiotte R. Line graphs, link partitions, and overlapping communities. *Phys Rev E*. 2009;80(1).
29. Aynaud T, Fleury E, Guillaume JL, et al. Communities in evolving networks: definitions, detection, and analysis techniques. In: Mukherjee A, Choudhury M, Peruani F, Ganguly N, Mitra B, eds. *Dynamics On and Of Complex Networks*. Vol. 2. New York, NY: Birkhäuser; 2013:159-200.
30. Enron Email Dataset. http://snap.stanford.edu/data/email-Enron.html
31. Leskovec J, Kleinberg J, Faloutsos C. Graphs over time: Densification laws, shrinking diameters and possible explanations. Paper presented at: 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining; 2005; New York, NY.
32. Danon L, Diaz-Guilera A, Duch J, Arenas A. Comparing community structure identification. *J Stat Mech Theory Exp*. 2005;2005(9):P09008-09008.
33. Shen H, Cheng X, Cai K, Huc MB. Detect overlapping and hierarchical community structure in networks. *Phys A Stat Mech Appl*. 2009;388(8):1706-1712.

How to cite this article: Wu L, Zhang Q, Guo K, Chen E, Xu C. Dynamic community detection method based on an improved evolutionary matrix. *Concurrency Computat Pract Exper*. 2021;33:e5314. https://doi.org/10.1002/cpe.5314