

Research Article

Community Vitality in Dynamic Temporal Networks

Fu Cai,¹ Li Min,¹ Zou Deqing,¹ Qu Shuyan,¹ Han Lansheng,¹ and James J. Park²

¹ School of Computer Sci & Tech, Huazhong University of Science and Technology, Wuhan 430074, China

² Department of Computer Science and Engineering, Seoul National University of Science and Technology, Seoul 139-742, Republic of Korea

Correspondence should be addressed to Zou Deqing; deqingzou@hust.edu.cn

Received 3 October 2013; Accepted 26 November 2013

Academic Editor: Neil Y. Yen

Copyright © 2013 Fu Cai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Current researches on temporal networks mainly tend to detect community structure. A number of community detection algorithms can obtain community structure on each time slice or each period of time but rarely present the evolution of community structure. Some papers discussed the process of community structure evolution but lacked quantifying the evolution. In this paper, we put forward the concept of *Community Vitality* (CV), which shows a community's life intensity on a time slice. In the process of computing CV, the "dead communities" can also be distinguished. Moreover, CV cannot only be used to quantify the life intensity of a community but also be used to describe the process of community evolution over time. More specifically, the change of community's structure will be found if CVs for different time slices of a community were compared, while the community with big value of CV can be selected if CVs for different communities were compared. Furthermore, *community vitality* change rate (CVCR) is proposed for revealing communities' structure change. The results of our experiments show that community vitality is a novel and effective way to understand or model the community evolution.

1. Introduction

As we all know, many systems, such as traffic system or information system, can be modeled as networks.

By studying structure and features of networks we can realise and predict networks' behavior and then discover the laws of corresponding real system. For instance, with the help of evolution research, *ad hoc* networks' structure can be optimized in order to improve their transmit efficiency. Virus immunization strategies aimed to avoid virus outbreak in a large scale can also be optimized through the study on virus propagation networks. In addition, by researching production sales networks we can discover the consumers who prefer specific productions and then achieve the goal of maximizing the benefits of goods promotion.

The above-mentioned networks can be considered as temporal networks, in which active time of network connections is limited and sequenced, and events are propagated along temporal paths [1]. Up to now, the research on temporal networks are mainly focused on three aspects [2]: detecting temporal community structure; analyzing characteristics of

community structure, such as temporal node centrality, temporal distances, and temporal clustering coefficient; and modelling temporal networks, such as TVG framework proposed by Arnaud and Paola and TRG framework proposed by John Whitbeck.

Detecting community structure in temporal networks is the foundation of studying networks' structure, which includes static community detection [3] and dynamic community detection [4]. As networks' structure changes over time, it is hard for static community detection algorithm to deal with noises in networks and insignificant communities might be detected. Therefore, the dynamic methods have been put forward currently. However, they cannot find "dead communities" and present the process of evolution. From Figure 1 we can find that the four communities have almost the same complexity and activeness (Figure 1(a)) when the final accumulating data is considered. But when the data is shown on time slices, we can find that the four communities are completely different: some are in the state of "growing" and others in the state of "dying" (Figure 1(b)). Although there are a few researches on the evolution of temporal

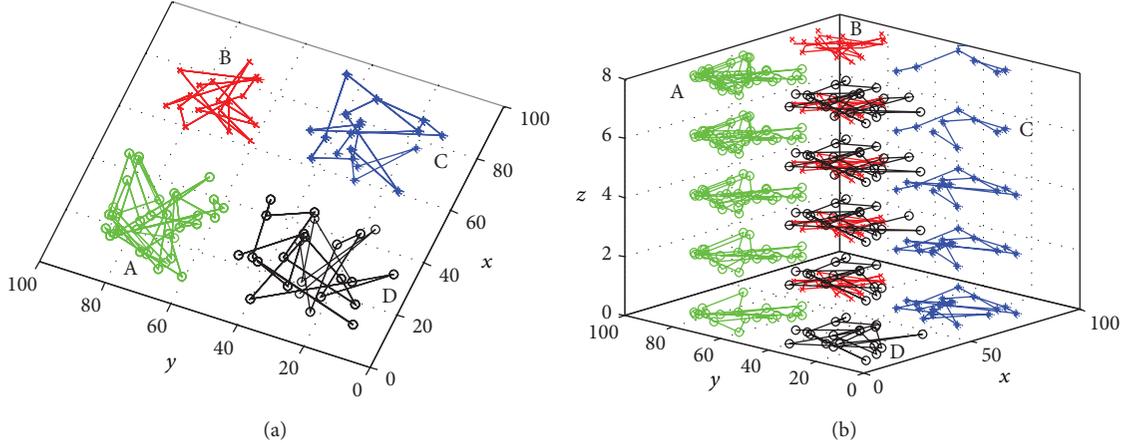


FIGURE 1: Temporal communities.

networks [5, 6], they did not quantify the process of evolution. These researches cannot quantify the process of community structure changed over time and also cannot recognize the process of a community's structure changing from "birth" to "death."

In this paper, we put forward the conception of *Community Vitality* (CV) designed for describing the life intensity of communities, on the basis of static community detection. What's more, CV can reflect communities' structure change quantitatively. We can analyze a community's status, structure changes, and life intensity between different communities also. And in the process of computing CV, we cannot only detect community structure changes, but also distinguish between the shrinking and increasing communities. By analyzing a time slice, we can get a community's life intensity, the complexity between different communities and then find out the "dead" communities.

The rest of the paper is organized as follows. In Section 2, different community detection methods are compared, and the definition of *Community Inheritability* (CI), the decision rules, and algorithm of *Community Inheritability* are given. In Section 3, the core conception of CV and CVCR which describes the change process of a community's CV are defined. In Section 4, experiments on two real datasets are given. In the experiments the values of CV are calculated, and inner structure and some laws of the real system which represented by the datasets are discovered. Applications in *ad hoc* networks were also analyzed in this section. A conclusion and some future works are given in Section 5.

2. Community Detection and Inheritability

Before computing CI and CV, the initial communities of temporal networks must be detected first. Recently, lots of static community detection algorithms have been put forward. The algorithms are mainly divided into two categories: one is graphs segmentation algorithm, such as the most representative algorithms Kernighan-Lin algorithm [7] and spectral bisection method [8] and the other one is hierarchical clustering algorithm, such as GN algorithm [9],

Fast algorithm [10], and CNM algorithm [11]. Considering CNM algorithm has low time complexity which almost equal to $O(n)$ and high algorithm efficiency, we choose CNM algorithm as temporal networks' static community detection algorithm.

After detecting communities on the given time slice, *Community Similarity* (CSS) is defined to judge the state of a community on the next time slice. CI can show us when a community is "born" or "die", that is, the start or end of a community's lifecycle, and serve for computing community vitality. Below we analyze three types of community changes (Keep, Separation, and Mergence) to judge inheritability between communities on consecutive time slices, by computing CSS between different communities.

Definition 1 (critical events). Critical events are those events who can cause community structures changes. A community's nodes or edges must have been changed if its structure changed. Thus, the four critical events N_Join , N_Leave , E_Appear , and $E_Disappear$ are defined as follows.

(N_Join). N_Join is an event that a node joins into a community, which includes two situations: a new created node come into the community and a node from another community that moves to the community. Node v joins into community C on the time slice t represented as follows:

$$N_{Join(v,t)} = 1 \quad \text{iff } v \notin V_{t-1}, v \in V_t. \quad (1)$$

(N_Leave). N_Leave is an event that a node leaves from a community, which includes two situations: a node disappears and a node from one community that moves to another community. Node v leaves from community C on the time slice t represented as follows:

$$N_{Leave(v,t)} = 1 \quad \text{iff } v \in V_{t-1}, v \notin V_t. \quad (2)$$

(E_Appear). E_Appear is an event that an edge appears in a community, which includes two situations: a new edge that appears in the community and an edge that comes from

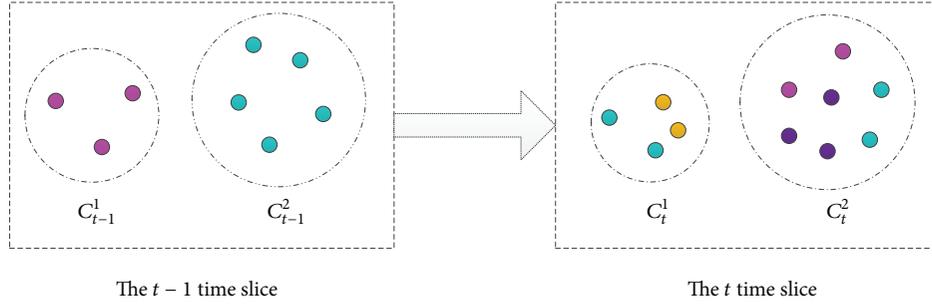


FIGURE 2: Community structure on adjoining time slices.

another community. Edge e appears in community C on the time slice t is represented as follows:

$$E_{\text{Appear}(e,t)} = 1 \quad \text{iff } e \notin E_{t-1}, e \in E_t. \quad (3)$$

(E_Disappear). $E_Disappear$ is an event that an edge disappears from a community, which includes two situations: an edge that disappears from the community and an edge that moves to another community. Edge e disappears from community C on the time slice t represented as follows:

$$E_{\text{Disappear}(e,t)} = 1 \quad \text{iff } e \in E_{t-1}, e \notin E_t. \quad (4)$$

In the definitions above, V_{t-1} and V_t are the set of C 's nodes on time slice $t-1$ and t ; E_{t-1} and E_t are the set of C 's edges on time slice $t-1$ and t .

Definition 2 (community structure change events). *Community structure change events* (CSCE) are used to reflect communities' relationship between adjacent time slices. Three types of CSCEs, which are Keep, Separation, and Mergence are defined as follows.

(Keep). A community C_{t-1}^p on time slice $t-1$ and another community C_t^q on time slice t has a relationship called Keep, when and only when C_{t-1}^p only has the same nodes with C_t^q on time slice t and C_t^q only has the same nodes with C_{t-1}^p on time slice $t-1$. In Figure 2, C_{t-1}^1 and C_t^2 have the relationship called Keep. The event can be formalized as follows:

$$\begin{aligned} \text{Keep}(C_{t-1}^p, C_t^q) &= 1, \quad \text{iff } V_{t-1}^p \cap V_t^q \neq \Phi, \\ V_{t-1}^p \cap V_t^i &= \Phi, \quad V_{t-1}^j \cap V_t^q = \Phi, \\ \text{here } i &= 1, \dots, q-1, q+1, \dots, n, \\ j &= 1, \dots, p-1, p+1, \dots, m; \end{aligned} \quad (5)$$

(Separation). A community C_{t-1}^p on time slice $t-1$ and k ($k \geq 2$) communities $C_t^{k_1}, C_t^{k_2}, \dots, C_t^{k_k}$ on time slice t has a relationship called Separation when C_{t-1}^p have the same nodes with all of the k communities on time slice t . In Figure 2, C_{t-1}^2

and C_t^1, C_t^2 has the relationship called Separation. The event can be formalized as follows:

$$\begin{aligned} \text{Separation}(C_{t-1}^p, C_t^{k_1}, C_t^{k_2}, \dots, C_t^{k_k}) &= 1, \\ \text{iff } V_{t-1}^p \cap V_t^i &\neq \Phi, \quad V_{t-1}^p \cap V_t^j = \Phi, \end{aligned} \quad (6)$$

here $i = k_1, k_2, \dots, k_k, j \in [1, n], j \neq k_1, k_2, \dots, k_k$;

(Mergence). k ($k \geq 2$) communities $C_{t-1}^{k_1}, C_{t-1}^{k_2}, \dots, C_{t-1}^{k_k}$ on time slice $t-1$ and a community C_t^q on time slice t has a relationship called Mergence when all of the k communities on time slice $t-1$ have the same nodes with C_t^q . In Figure 2, C_t^2 and C_{t-1}^1, C_{t-1}^2 have the relationship called Mergence. The event can be formalized as follows:

$$\begin{aligned} \text{Mergence}(C_{t-1}^{k_1}, C_{t-1}^{k_2}, \dots, C_{t-1}^{k_k}, C_t^q) &= 1, \\ \text{iff } V_{t-1}^i \cap V_t^q &\neq \Phi, \quad V_{t-1}^j \cap V_t^q = \Phi, \end{aligned} \quad (7)$$

here $i = k_1, k_2, \dots, k_k, j \in [1, m], j \neq k_1, k_2, \dots, k_k$.

In the definitions above, m and n , respectively, represent the number of community on time slice $t-1$ and t .

Definition 3 (influence of a node). Influence of a node indicates the importance of the node in its community. A node with big influence can attract a new connection with larger probability.

Let a community on time slice t be C_t^p , whose node ID is denoted by $1, 2, \dots, n$ and degree of nodes is denoted by d_1, d_2, \dots, d_n ; then the influence of node i can be calculated as

$$\text{Inf}(i, C_t^p) = \frac{d_i}{\sum_{j=1}^n d_j}. \quad (8)$$

Definition 4 (community structure similarity). CSS can quantify the similarity of two communities on consecutive time slices. And it is the criterion of CI: bigger similarity have higher probability inheritability.

Let a community on time slice $t-1$ be C_{t-1}^p , whose number of node is n_1 and number of edge is e_1 . Another community

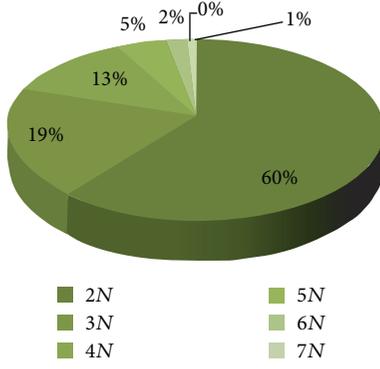


FIGURE 3: Community lifecycle distribution in CDBLP.

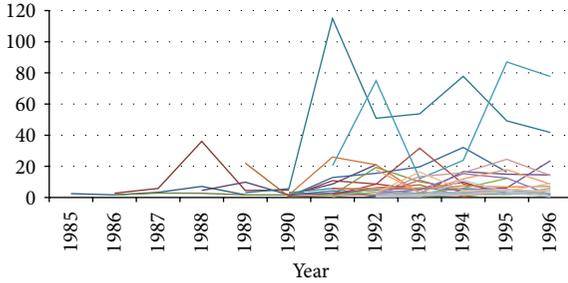


FIGURE 4: CV change in CDBLP.

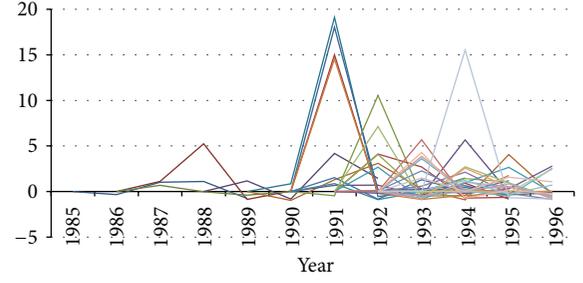


FIGURE 5: CVCR change over time in CDBLP.

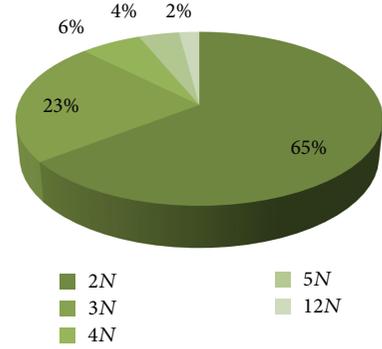


FIGURE 6: Community lifecycle distribution in EED.

C_t^q on time slice t whose number of node is n_2 and number of edge is e_2 . The number of the same nodes between C_{t-1}^p and C_t^q is denoted by n , and the n nodes' ID is denoted by $1, 2, \dots, n$. Also, the number of the same edges between C_{t-1}^p and C_t^q is denoted by e . The weight of node and edge of networks' community structure are denoted by w_1 and w_2 , respectively. Then we have the following conclusions.

- (a) If two communities have the relationship Keep, CSS can be calculated as follows:

$$\begin{aligned} \text{Sim}(C_{t-1}^p, C_t^q) &= w_1 \times \frac{n}{n_1} \times \frac{\sum_{i=1}^n \text{Inf}(i, C_{t-1}^p)}{n} \\ &+ w_2 \times \frac{e}{e_1} \\ &= w_1 \times \frac{\sum_{i=1}^n \text{Inf}(i, C_{t-1}^p)}{n_1} + w_2 \times \frac{e}{e_1}. \end{aligned} \quad (9)$$

- (b) If two communities have the relationship Separation, CSS can be calculated as follows:

$$\begin{aligned} \text{Sim}(C_{t-1}^p, C_t^q) &= w_1 \times \frac{n}{n_1} \times \frac{\sum_{i=1}^n \text{Inf}(i, C_{t-1}^p)}{n} \\ &+ w_2 \times \frac{e}{e_1} \\ &= w_1 \times \frac{\sum_{i=1}^n \text{Inf}(i, C_{t-1}^p)}{n_1} + w_2 \times \frac{e}{e_1}. \end{aligned} \quad (10)$$

- (c) If two communities have the relationship Mergence, CSS can be calculated as follows:

$$\begin{aligned} \text{Sim}(C_{t-1}^p, C_t^q) &= w_1 \times \frac{n}{n_2} \times \frac{\sum_{i=1}^n \text{Inf}(i, C_t^q)}{n} \\ &+ w_2 \times \frac{e}{e_2} \\ &= w_1 \times \frac{\sum_{i=1}^n \text{Inf}(i, C_t^q)}{n_2} + w_2 \times \frac{e}{e_2}. \end{aligned} \quad (11)$$

- (d) Two communities has any of these relationships above, CSS is equal to 0; that is, $\text{Sim} = 0$.

All parameters appeared in the paper (including w_1, w_2 and w_3, w_4, w_5 appeared in chapter III) are decided by entropy method, which can be used to determine evaluation indicators of arbitrary evaluation problem and remove the indicators having slight effect on evaluation result, then reflect the importance of every indicator objectively.

Definition 5 (community inheritability). CI represents that a community on last time slice whose state and position on current time slice. Next we will analyze all cases of communities' CI between consecutive time slices.

- (a) If a community C_{t-1}^p on time slice $t-1$ and another community C_t^q on time slice t has a relationship called Keep; then C_t^q inherits from C_{t-1}^p .
- (b) If a community C_{t-1}^p on time slice $t-1$ and another k ($k \geq 2$) communities $C_t^{k_1}, C_t^{k_2}, \dots, C_t^{k_k}$ on time

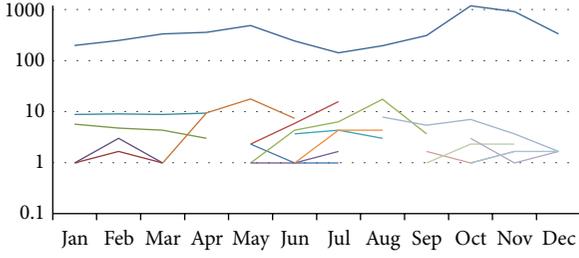


FIGURE 7: CV change in EED.

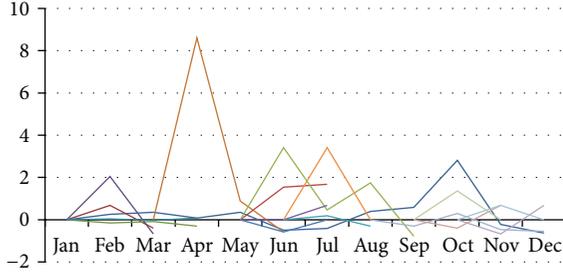


FIGURE 8: CVCR change over time in EED.

slice t have a relationship called Separation; then community structure similarity $\text{Sim}_1, \text{Sim}_2, \dots, \text{Sim}_k$ between community C_{t-1}^p and k communities on time slice t should be computed. After that, the maximum and the second largest value Sim_i and Sim_j ($i, j \in [k_1, k_k]$) should be selected from these values. If $\text{Sim}_i \neq \text{Sim}_j$; then C_t^i inherits from C_{t-1}^p and other communities C_t^x ($x = k_1, k_2, \dots, i-1, i+1, \dots, k_k$) are all treated as new communities or else, C_{t-1}^p is “dead” and communities $C_t^{k_1}, C_t^{k_2}, \dots, C_t^{k_k}$ are all treated as new community.

(c) If k ($k \geq 2$) communities $C_{t-1}^{k_1}, C_{t-1}^{k_2}, \dots, C_{t-1}^{k_k}$ on time slice $t-1$ and a community C_t^q on time slice t has a relationship called Mergence; then community structure similarity $\text{Sim}_1, \text{Sim}_2, \dots, \text{Sim}_k$ between k communities and community C_t^q should be computed. And next, the maximum and the second largest value Sim_i and Sim_j ($i, j \in [k_1, k_k]$) should be selected from these values. If $\text{Sim}_i \neq \text{Sim}_j$, then C_t^q inherits from C_{t-1}^i and other communities C_{t-1}^x ($x = k_1, k_2, \dots, i-1, i+1, \dots, k_k$) are all “dead” or else, communities $C_{t-1}^{k_1}, C_{t-1}^{k_2}, \dots, C_{t-1}^{k_k}$ are all “dead” and C_t^q are treated as a new community.

(d) If a community C_{t-1}^p on time slice $t-1$ and another community C_t^q on time slice t has none of these relationships above; then C_t^q does not inherit from C_{t-1}^p .

After defining the method that CSS under three cases of CSCE, judgement method of CI in each case are defined.

Furthermore, we design Algorithm 1 called CI decision algorithm to decide communities’ CI between consecutive time slices. Time complexity of the algorithm is $O(n * m)$. Here n is the number of communities on time slice $t-1$ and m is the number of communities on time slice t .

3. Community Vitality

We define the concept of CI in order to determine when a community is “born or dies”, so we can obtain a community’s lifecycle. Also the concept is the precondition of computing CV. Furthermore, the concept of CV is used to quantify the life intensity of communities on every time slice, and if a community’s different CVs in its lifecycle are compared, the evolution of community structure can be described quantificationally. CV can be used to forecast community structure changes to optimize *ad hoc* networks and to analyze virus propagation and so on.

Definition 6 (community vitality). CV reflects the life intensity of a community on a time slice. Considering a community’s CV change over time, the evolution process of community structure can be obtained quantificationally. Besides, a community with bigger CV than others has more complex structure. As we all know, a community’s structure change is equal to some of critical events that happened, which leads to the community’s CV change. Thus, it can be concluded that a community’s CV change is decided by the change of critical events and finally puts down to the change of number of nodes, edges, and compactness of structure in the community. In fact, clustering coefficient [3] describes the aggregation of nodes in the network, namely, compactness of the network, so it’s a influence factor of CV.

Let a community on time slice t be C_t^q , whose number of nodes is n , number of edges is e , and clustering coefficient is C . Then C_t^q ’s CV is defined as

$$\text{VI}(t, q) = w_3 \times n + w_4 \times e + w_5 \times C. \quad (12)$$

Here parameters w_3 , w_4 , and w_5 respectively, present the degree of importance that nodes, edges, and community structure to CV.

Definition 7 (community vitality change rate). In order to see a community’s state and change degree directly, CVCR is put forward. CVCR shows a community’s community structure change degree between consecutive time slices.

If the community C_t^q on time slice t inherits from the community C_{t-1}^p on time slice $t-1$, then CVCR can be calculated as follows:

$$\Delta \text{VI}(C_{t-1}^p, C_t^q) = \frac{\text{VI}(t, q) - \text{VI}(t-1, p)}{\text{VI}(t-1, p)}. \quad (13)$$

If the community C_t^q on time slice t is a newly created community, then CVCR can be calculated as

$$\Delta \text{VI}(\Phi, C_t^q) = 0. \quad (14)$$

```

Input:
    Communities on time slice  $t - 1$ :  $C_{t-1}^1, C_{t-1}^2, \dots, C_{t-1}^m$ 
    Communities on time slice  $t$ :  $C_t^1, C_t^2, \dots, C_t^n$ 
Output: continuity_matrix  $[n] [m]$ 
%  $n$  is the number of communities on time slice  $t$ 
%  $m$  is the number of communities on time slice  $t - 1$ 
% node_num_matrix  $[n] [m]$  saves the number of same
% nodes between communities on adjacent time slices
% continuity_matrix  $[n] [m]$  saves the result of CI
% between communities on adjacent time slices
% merger_matrix  $[n] [m]$  saves the Mergence
% relationships and results of CI in the case
(1)   for  $i \in [0, n)$  do //keep
(2)       if (only one element in the  $i$ th row of
(3)           node_num_matrix is not equal to 0) then
(4)               continuity_matrix  $[i] [j] = 1$ ;
(5)       end if
(6)   end for
(7)   for  $i \in [0, n)$  do //mergence
(8)       if (more than one element in the  $i$ th row of
(9)           node_num_matrix are not equal to 0) then
(10)           Modifying continuity_matrix as
(11)           Definition 3(d) and relevant elements of
(12)           merger_matrix;
(13)       end if
(14)   end for
(15)   for  $i \in [0, m)$  do //separation
(16)       if (more than one element in the  $i$ th column
(17)           of node_num_matrix are not equal to 0) then
(18)           Modifying continuity_matrix based on
(19)           merger_matrix and results of CI computed as
(20)           Definition 3(c);
(21)       end if
(22)   end for

```

ALGORITHM 1: CI decision algorithm.

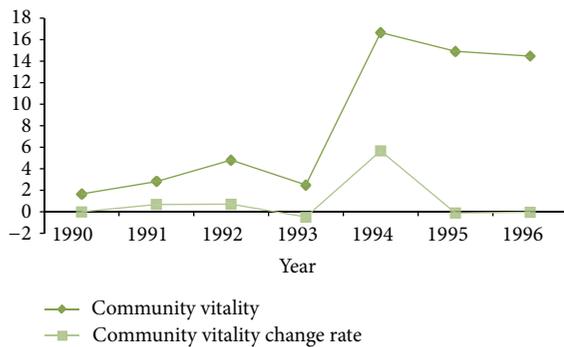


FIGURE 9: The 10th series of CDBLP CV and CVCR.

The value of CVCR $\Delta VI(C_{t-1}^p, C_t^q)$ has the following meanings.

If $\Delta VI(C_{t-1}^p, C_t^q) > 0$, then C_t^q is in the state of growing and the multiple of growth is $\Delta VI(C_{t-1}^p, C_t^q)$;

If $\Delta VI(C_{t-1}^p, C_t^q) = 0$, then C_t^q has steady community structure;

If $\Delta VI(C_{t-1}^p, C_t^q) < 0$, then C_t^q is in the state of shrink and the shrink multiple is $|\Delta VI(C_{t-1}^p, C_t^q)|$.

4. Experiments and Analysis

In this section, experiments are carried out on two real datasets, Chinese DBLP Dataset (CDBLP) and Enron Email Dataset (EED). And then our method is analyzed and compared with some existing algorithms and methods.

4.1. Experiments. CDBLP was published by “Automation Discipline Innovation Method” research group of Chinese Academy of Sciences Institute of automation and it derived from the network of Computer Chinese Journal. The part of the data from 1985 to 1996 is used in our experiment. We set a year as the time interval and integrate annual data as a time slice data.

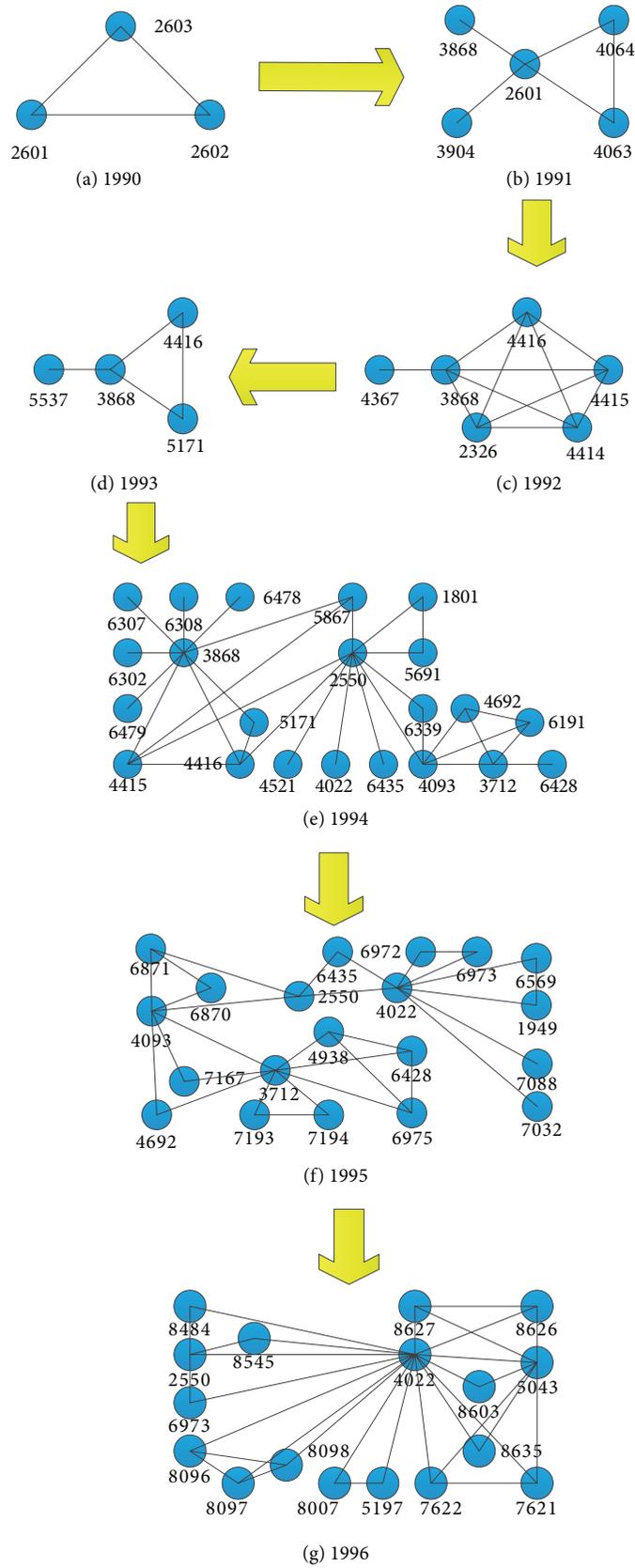


FIGURE 10: Change of 10th series of community structure.

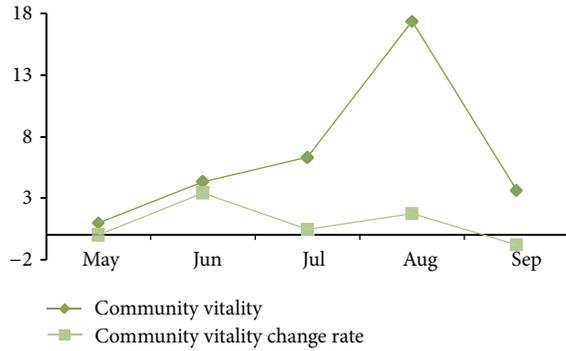


FIGURE 11: The 9th series of EED CV and CVCR.

EED was collected by CALO Project (a cognitive assistant that learns and organizes). The dataset represents the connections between employees. The part of the data in 2001 is used in our experiment. We set a month as the time interval and integrate each month data as a time slice data.

In Figures 3 and 6, N is the number of time slices in a community's lifecycle.

From the results (Figures 3, 4, 5, 6, 7, and 8), we can find the distribution of community's lifecycle. In fact, from the lifecycle distribution and CV change process of all communities in the network, structure features or rules of real system corresponded by the dataset are mined.

Figures 4 and 5 shows the change process of all communities' CV and CVCR during their lifecycle. Those communities in Figures 4 and 5 are formed from 12 years data of CDBLP and the communities whose lifecycle less than five time slices are ignored. Figures 7 and 8 show the change process of all communities' CV and CVCR during their lifecycle. Those communities in Figures 7 and 8 are formed from one year data of EED and the communities whose lifecycle are less than three time slices are ignored. By analyzing Figure 3 to Figure 8, we can get the following conclusions.

- (1) The communities' lifecycles are generally short, which show that the relationships between users of the real systems are sustained for a short period of time.
- (2) It can show the structure features or rules of the real system. Analyzing CDBLP, the number of papers published in the Computer Chinese Journal network is increasing over time. In other words, theoretical research is more and more popular, it actually has achieved outstanding progress. Analyzing EED, the communication between employees of Enron Corporation is in a stable state without obvious change over time in general.
- (3) A community with big CV tends to have long lifecycle, which means that core leaders or core structure may be included in the community. And more attention need to be paid on this kind of community.
- (4) In Figures 4–8, each line presents the evolution process of a community structure, in which the line's start point and terminal point represent the beginning and end of community lifecycle, respectively. In Figures

4 and 7, a point means a community's CV on a time slice. And in Figures 5 and 8, a point beside the first point represents the extent of communities' structure enlargement or decrease comparing with the community on the last time slice. Furthermore, we can find "dead" communities if a community's lifecycle ends; for instance, dead communities are the communities represented by the 3rd series and the 5th series on the time slice of April in Figure 7. Therefore, compared with other community detection algorithms, our algorithm cannot only find communities' structure and analyze its evolution process, but also quantify communities' life intensity on each time slice by analyzing CV.

In order to verify whether CV can correctly describe the change of growing or shrinking community structure, we randomly select two communities in the two real datasets (communities represent by the 10th series of Chinese DBLP Dataset and the 9th series of Enron Email Dataset), and then we compare the change of the community structure with the change process of CV and CVCR. Finally, we find the change process of community structure is consistent with the change process of CV. Now we use a series of the following diagrams (Figures 9, 10, 11, and 12) to give a specific verification.

Figure 10 shows structures of the community represented by the 10th series of Chinese DBLP Dataset on each time slice. By comparing arbitrary adjacent two pictures, we can obviously find that changes in the community's community structure can be described as increase, increase, decrease, increase sharply, decrease, decrease a little, and almost constant. On the last three time slices, the community's structure shows a steady state. On the whole, the change process of the community's structure is completely consistent with the change rules described in Figure 9 which shows the change process of CV.

Figure 11 shows structures of the community represented by the 9th series of EED on each time slice. By comparing arbitrary adjacent two pictures, we can obviously find that changes in the community's community structure can be described as increase, increase slightly, increase sharply, and decrease. Because a community in EED whose community structure always centers on a node, the community's structure relies on the central node to survive; the aggregation of the community is poor. In fact, the change process of the community's structure is completely consistent with the change rules described in Figure 12 which shows the change process of CV.

4.2. Analysis. Our method is devote to study community structure's evolution based on CV quantificationally. Then a comparison of some proposed algorithms and methods is given as Table 1.

Furthermore, CV can quantify the change process of dynamic networks. On the one hand, it can be applied to mining change pattern of dynamic networks. On the other hand, we can use these laws to optimize applications, such as

TABLE 1: A comparison of some proposed algorithms and methods.

| Algorithm | Static community detection | Dynamic community detection | | Community evolution | |
|--|----------------------------|-----------------------------|----------------------------|------------------------------|------------|
| | | CQA [4] | Incremental algorithm [12] | An event-based framework [5] | Our method |
| Obtaining community structure | ✓ | ✓ | ✓ | ✓ | ✓ |
| Obtaining insignificance community | ✓ | ✗ | ✗ | ✗ | ✗ |
| Detecting death community | ✗ | ✗ | ✗ | ✓ | ✓ |
| A community structure's change | ✗ | ✗ | ✗ | ✗ | ✓ |
| Change process of communities on consecutive time slices | ✗ | ✗ | ✗ | ✓ | ✓ |
| Quantifying community state | ✗ | ✗ | ✗ | ✗ | ✓ |
| Analysis community evolution quantitatively | ✗ | ✗ | ✗ | ✗ | ✓ |

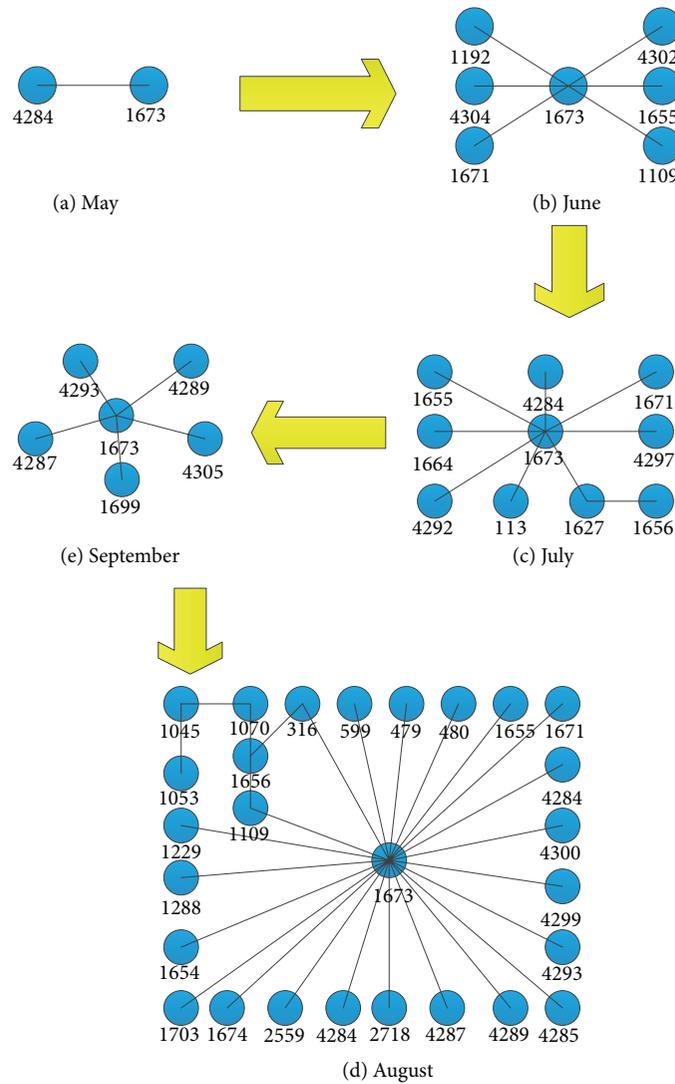


FIGURE 12: Change of 9th series of community structure.

optimizing communication of *ad hoc* networks. Most practicable *ad hoc* networks adopt the cluster-based routing protocols and a cluster can be treated as a relatively concentrated communication community. As the sample data showed in Figure 1, we assume that community B communicates with community D through community A or community C, and we acquire the change in communication condition of each cluster, shown as Figure 1(b) by setting appropriate monitoring points and collecting the communication between nodes at regular time interval. Via CV, community B can perceive that community A is a communication cluster in the state of steady and active gradually and community C is a communication cluster in the state of “dying”. Thus, community B can adjust the communication routing from community B to community D all through community A and communication efficiency of the communication routing will be better than that through community C.

5. Conclusion

In this paper, we put forward the concept of CV and design CI decision algorithm based on existing static community detection algorithm CNM, to calculate CVs. CV is defined for studying communities’ life intensity and describing the process of community evolution quantitatively. Furthermore, the concept CV has many practical applications. The main contributions are as follows.

- (1) Four critical events are proposed to describe community structure changes. And they are the basic of the research on the evolution of community structure.
- (2) Three types of CSCEs (Keep, Separation, and Merger) are defined, and then the computing method of CSS is designed under the three circumstances. By computing the value of CSS, inheritability between communities on consecutive time slices can be found out.
- (3) Decision rules of CI are put forward in each type of CSCE and CI decision algorithm is designed. CI is designed to determine when a community is “born” or “die”; namely, a community’s lifecycle can be presented. Also “dead” community can be mined.
- (4) The core concept of CV is defined. CV can quantify a community’s life intensity and describe the evolution process of community structure dynamically. Then CVCR is defined to find a community’s state and change degree directly.
- (5) We utilize two real datasets which are Chinese DBLP Dataset and Enron Email Dataset to perform experiments. And we use the figures to express the evolution process of community structure and randomly select two communities to verify the correctness of CV. In the end, simulated data is used to show that CV can be applied to communication of *ad hoc* networks.

In future researches, we will focus on studying the applications of CV.

Acknowledgments

The paper is supported by China 973 Program (2014CB340600), NSF (60903175, 61272405, 61272033, and 61272451), and University Innovation Foundation (2013TS102 and 2013TS106).

References

- [1] P. Holme and J. Saramäki, “Temporal networks,” *Physics Reports*, vol. 519, no. 3, pp. 97–125, 2012.
- [2] H. Kim and R. Anderson, “Temporal node centrality in complex networks,” *Physical Review E*, vol. 85, no. 2, Article ID 026107, 8 pages, 2012.
- [3] A. Clauset, M. E. J. Newman, and C. Moore, “Finding community structure in very large networks,” *Physical Review E*, vol. 70, no. 6, Article ID 066111, 6 pages, 2004.
- [4] N. P. Nguyen, T. N. Dinh, Y. Xuan, and M. T. Thai, “Adaptive algorithms for detecting community structure in dynamic social networks,” in *Proceedings of the IEEE International Conference on Computer Communication (INFOCOM ’11)*, pp. 2282–2290, Shanghai, China, April 2011.
- [5] S. Asur, S. Parthasarathy, and D. Ucar, “An event-based framework for characterizing the evolutionary behavior of interaction graphs,” *ACM Transactions on Knowledge Discovery from Data*, vol. 3, no. 4, article 16, 2009.
- [6] Y.-R. Lin, H. Sundaram, Y. Chi, S. Zhu, and B. L. Tseng, “Facetnet: a framework for analyzing communities and their evolutions in dynamic networks,” in *Proceedings of the 17th International Conference on World Wide Web (WWW ’08)*, pp. 685–694, Beijing, China, April 2008.
- [7] B. W. Kernighan and S. Lin, “An efficient heuristic procedure for partitioning graph,” *Bell System Technical Journal*, vol. 49, no. 2, pp. 291–307, 1970.
- [8] A. Pothen, H. D. Simon, and K. Liou, “Partitioning sparse matrices with eigenvectors of graphs,” *SIAM Journal on Matrix Analysis and Applications*, vol. 11, no. 3, pp. 430–452, 1990.
- [9] M. Girvan and N. E. J. Newman, “Community structure in social and biological networks,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, 2001.
- [10] M. E. J. Newman, “Fast algorithm for detecting community structure in networks,” *Physical Review E*, vol. 69, no. 6, Article ID 066133, 5 pages, 2004.
- [11] U. N. Raghavan, R. Albert, and S. Kumara, “Near linear time algorithm to detect community structures in large-scale networks,” *Physical Review E*, vol. 76, no. 3, Article ID 036106, 11 pages, 2007.
- [12] H.-F. Wang, L.-P. Huang, and S. Yu, “An incremental community discovering approach,” *Computer Simulation*, vol. 25, no. 1, 2008.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

