Dynamic Community Evolution Analysis Framework for Large-Scale Complex Networks Based on Strong and Weak Events

Shaojie Qiao, Nan Han, Yunjun Gao, *Member, IEEE*, Rong-Hua Li, Jianbin Huang, Heli Sun, and Xindong Wu, *Fellow, IEEE*

***Abstract*—Community evolution remains a heavily researched and challenging area in the analysis of dynamic complex network structures. Currently, the primary limitation of traditional event-based approaches for community evolution analysis is the lack of strict constraint conditions for distinguishing evolutionary events, which entails that as the cardinality of discovered events increases, so does the number of redundant events. Another limitation of existing approaches is the lack of consideration for weak events. Weak events can be generated by small changes in communities, which are empirically prevalent, and are typically not captured by traditional events. To manage these two aforementioned limitations, this research aims to formalize a weak and strong events-based framework, which includes the following newly discovered events: "weak shrink," "weak expand," "weak merge," and "weak splity" predicated on the community overlapping degree and community degree membership, this article refines these traditional strong events, as well as new constraints for weak events. In addition, a community evolution mining framework, which is based on both strong and weak events, is proposed and denoted by a weak-event-based community evolution method (WECEM). The framework can be summarized by the following: 1) communities in complex networks with adjacent time-stamps are compared to determine the community overlapping degree and community membership degree; 2) the values of the community overlapping degree and membership degree meet the definition of events; and 3) weak events are effectively identified. Extensive experimental results, on real and synthetic data sets consisting of dynamic complex networks and online social networks, demonstrate that WECEM is able to identify weak events more effectively than traditional frameworks. Specifically, WECEM outperforms traditional frameworks by 22.9% in the number of discovered strong events. The detection accuracy of evolutionary events is approximately 12.2% higher than that of traditional event-based frameworks. It is also worth noting that, as the cardinality of the data grows, the proposed framework, when compared with traditional frameworks, can more effectively, and efficiently, mine large-scale complex networks.**

***Index Terms*—Community detection, community evolution analysis, complex networks, event-based framework, weak events.**

Manuscript received March 28, 2019; revised September 23, 2019; accepted December 12, 2019. Date of publication January 7, 2020; date of current version September 16, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61772091, Grant 61802035, Grant 61962006, and Grant 71701026, in part by the Sichuan Science and Technology Program under Grant 2018JY0448, Grant 2019YFG0106, and Grant 2019YFS0067, in part by the Natural Science Foundation of Guangxi under Grant 2018GXNSFDA138005, and in part by the Guangdong Province Key Laboratory of Popular High Performance Computers under Grant 2017B030314073. This article was recommended by Associate Editor J. Lu. *(Corresponding author: Nan Han.)*

Shaojie Qiao is with the School of Software Engineering, Chengdu University of Information Technology, Chengdu 610225, China.

Nan Han is with the School of Management, Chengdu University of Information Technology, Chengdu 610103, China (e-mail: hannan@cuit.edu.cn).

Yunjun Gao is with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China.

Rong-Hua Li is with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China, also with the Guangdong Province Key Laboratory of Popular High Performance Computers, Shenzhen University, Shenzhen 518060, China, and also with the Guangdong Province Engineering Center of China-made High Performance Data Computing System, Shenzhen University, Shenzhen 518060, China.

Jianbin Huang is with the School of Computer Science and Technology, Xidian University, Xi'an 710071, China.

Heli Sun is with the School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China.

Xindong Wu is with the Mininglamp Academy of Sciences, Mininglamp Technology, Beijing 100084, China, and also with the Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology, Hefei 230009, China.

Digital Object Identifier 10.1109/TSMC.2019.2960085

I. Introduction

COMPLEX networks are distinct from simple networks, such as lattices or random graphs, such that they have nontrivial topological features. These complex networks are ubiquitous in our everyday lives, and examples include online social networks, such as Facebook and Twitter. Community structures are generally inherent in complex networks, and can be best exemplified by groups of nodes in which the network connections are dense, but between which connections are sparser [1]. Community structures, representing a mesoscale structure of networks, accordingly, are viewed as one of the most important characteristics of complex networks. Such structures provide immense social and economic value in understanding the processes of network formation, growth and shrinkage, information dissemination and public opinion analysis. Dynamic analysis of complex networks, especially assessing the evolution of communities, can provide insights into: 1) detecting a drastic change in the interaction patterns; 2) understanding the latent structures of complex networks; and 3) forecasting the future trends of networks [2], [3].

Motivation: In real-world applications, community structures represent a dynamically changing phenomenon. Accordingly, communities are in constant flux: growing, shrinking, emerging, and disappearing all together. Examples include human migration in social networks, seasonal animal migrations, and topic transfers in blogs. Due to the cardinality of nodes in dynamic complex networks, static community discovery approaches cannot be effectively applied to analyze this evolution of communities. Thus, it follows that research in this area remains fundamentally important in precision marketing, crime prevention, traffic flow forecasting, and network congestion prediction. The following example illustrates the importance of research in community evolution.

Example 1: Sina microblog is the largest blogging system in China. The distribution of the userbase is in constant flux, with users joining and leaving on a regular basis. This dynamic is indicative of a constantly changing network structure. In addition, the userbase represents a multitude of interests on a vast array of topics, which in itself contributes to the ever-changing structure. Mining communities with such a userbase can provide insight and understanding into how the networks change, and identify users' points of interests. This knowledge can then be applied to social networking platforms to recommend services across various communities.

Challenges: Current research relevant to mining complex and dynamic networks has focused on event-based frameworks, and variants, as proposed by Asur *et al.* [4]. However, the existing methods have the following disadvantages.

1) Defining an evolutionary event is not straight forward. Given the scenario in which a constraint condition is loosely defined, the potential for events to grow quickly increases, leading to a large number of redundant events.
2) Traditional event-based frameworks work ideally given a single type of event in one community over a given period of time. However, in practice, multiple events often occur within a community simultaneously. In addition, traditional methods are poorly equipped to deal with weak events, which can be defined as events triggered by small changes in the community. These events are not considered evolutionary due to the strict constraints of strong events. Finally, traditional methods generally return low accuracy in evolutionary event discovery.
3) The high time complexity of traditional event-based frameworks makes it difficult to efficiently implement discovery in large-scale dynamic complex networks.

Contributions: In an effort to improve the efficiency and accuracy of traditional event-based frameworks for discovering community evolution, this article focuses on the detection of various types of events occurring in the same communities, as well as the discovery of comparatively higher quality strong events. In this article, a new method for community evolution analysis in mining dynamic networks is proposed, which is based on newly discovered weak events. The contributions of this article are given as follows.

New Concept: We introduce a new concept of *weak event* based on the community overlapping degree and membership degree. According to this concept, we refine the events of community evolution, and then improve the traditional event-based framework for community evolution.

New Framework: Evolutionary events are categorized into strong and weak events, and the application scenarios of weak events are introduced. Weak events are formally defined, as well as a method for determining such events. In addition, a weak event-based mining framework is proposed, referred as WECEM.

Extensive Experimental Results: Extensive experiments are conducted in real and synthetic large-scale dynamic networks. We compare the proposed WECEM framework with state-of-the-art community evolution discovery methods to verify the quality, mining accuracy, and runtime performance.

II. Related Work

In recent years, due to the popularity and ubiquity of online social networks and large-scale complex networks, community discovery, and evolutionary event mining have attracted a lot of attention [5]–[8]. Most specifically, dynamic networks operate as a powerful signal for forecasting the behavior of individuals, route planning, personalized recommendations, and so on. Within this field, community detection remains fundamental in community evolution analysis. Existing research has done much to progress the field of study, as noted in the following examples. The modularity-based approach [1], [9] widely used for discovering communities in complex networks, and several improved algorithms [10] have been proposed successively, e.g., Fast-Newman [9], and the Clauset, Newman, and Moore's algorithm (CNM) [11]. Recently, Hao *et al.* [12] proposed a technique which integrates formal concept analysis with the clique percolation method, and works to improve the accuracy of community discovery. Mahmood *et al.* [13] combined complex networks and spatial data mining techniques by mapping network nodes into a geometric space and encoding the position of each node with its geodesic distances from all the other nodes. Palla *et al.* [14] proposed a clique percolation method that effectively identifies overlapping communities in complex networks. Parsa *et al.* [15] presented a new method for detecting communities based on an estimation of distribution algorithm. Bouguessa *et al.* [16] aggregated similar nodes to form small communities, then iteratively combined these small communities until a maximum modularity is reached. Lyzinski *et al.* [17] obtained a low-dimensional representation by mapping networks to an Euclidean space.

Recent research has been focused on predicting the trend of network development. The most common approach is the event-based framework proposed by Ausr *et al.* [4]. It first detects communities over time, and then mines evolutionary events by comparing overlapping and membership degrees of communities relative to time. Takaffoli *et al.* [2], [18] formally defined a series of events for community evolution, and proposed a community matching algorithm to identify similar communities. In addition, the concept of "meta-community" was proposed, which entails a series of similar communities with different timestamps. İlhan and Öguducu [19] used the autoregressive integrated moving average model to predict how particular community features change on the next time horizon. Zhu *et al.* [20] proposed a multi-mode co-clustering approach to detect the hierarchical and overlapping communities in location-based social networks. Tajeuna *et al.* [21] formulated the number of nodes shared between two communities as a matrix. This approach could efficiently track the changes of communities during evolution. Falkowski *et al.* [22] proposed an incremental graph mining algorithm based on the idea of static density clustering, which partitions evolutionary events into positive and negative changes. The method can discover detailed information about evolutionary events. Wang *et al.* [23] proposed a new method to calculate the importance of core nodes based on the degrees of nodes. The changes are then compared to core nodes over adjacent timestamps to determine the evolution of the given networks. This method uncovered two important phenomenon: growing and metabolic processes in networks. Zhang *et al.* [24] studied the evolutionary game dynamics of multiple community networks. Liu *et al.* [25] proposed a fast community evolution tracking model, which uses an improved PageRank algorithm to find the core nodes in a network. As a result, the evolutionary events can be detected by adding nodes and edges into core communities over time.

The disadvantages of the aforementioned community evolution analysis methods over event-based frameworks can be summarized as follows.

1) These methods maintain a strict definition for strong events, which greatly limits the scope. This leads to events, which may be equally important, not qualifying under the given criteria. This is the motivation behind the proposed weak events in this article. Existing work, based on the event-based framework, does not take into account the effects of overlapping events, and that of weak events.
2) The computational complexity of the event-based framework is extremely costly, most notably being the Takaffoli framework, which requires the calculation of the difference of each community at all timestamps.

In order to overcome the challenges associated with the traditional event-based frameworks, this article proposes a new weak event-based community evolution analysis algorithm based on strong and weak events. In this article, the algorithm is referred to as WECEM. In discovery, WECEM takes into consideration the overlapping and membership degrees of communities, which allows events between overlapping communities to be distinguished more effectively. Most notably, this enables WECEM to perform without influencing the identification accuracy.

The proposed dynamic community evolution analysis framework can be applied in several applications, for example, the identification of crucial genes in biological network [26], [27], the design, synthesis, and re-engineering of biological networks for biomedical purpose [28], and networked medicine and biological network control [29].

III. Community Evolution Analysis Framework Based on Strong and Weak Events

In large-scale complex dynamic networks, community structures evolve slowly once the relationships between nodes in networks are formed. Consequently, it becomes difficult

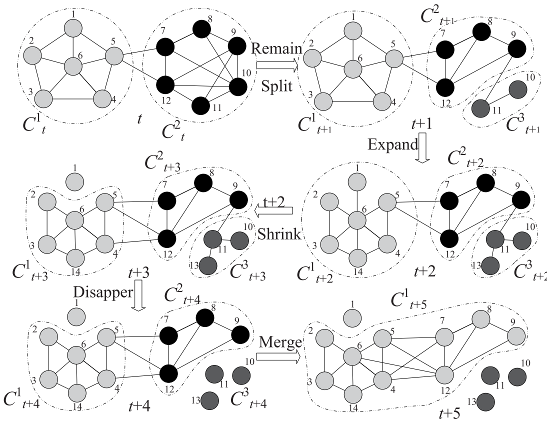


Fig. 1. Example of strong events in community evolution.

to change community structures over a comparatively short period of time. Thus, large overarching changes are infrequent, however, this does not imply that small changes are not frequently occurring during the periods between larger changes. It follows then that if traditional methods are designed only to detect broad, sweeping changes in complex networks, those smaller, weak events are not being detected. This is the motivation behind our proposed weak-event-based approach.

A. Preliminaries

Definition 1 (Dynamic Community): Let $C_t = (v_1^t, v_2^t, \ldots, v_n^t)$ be a community at time t, and v represents the node in C_t. From t to $t+m$, several events may occur in C_t. $\{C_t, \Delta A_1, \Delta A_2, \ldots, \Delta A_m\}$ represents a dynamic community, denoted by $C_{t:t+m}$. The community $C_{t+m} = C_t + \Delta A_1 + \Delta A_2 + \cdots + \Delta A_m$, in which $\Delta A_1 \sim \Delta A_m$ are evolutionary events happening in C_t from time t to $t+m$.

With time passing by, the relationships between nodes in the network have gradually changed, forming some new communities with different evolutionary events. Here, we will first give the definition of traditional evolutionary events, which is called "strong event."

Definition 2 (Strong Event): The nodes in the network interact with each other, which causes the network structure to change at the next timestamp, and this process is called a strong event. Strong events include: *Remain*, *Shrink*, *Expand*, *Split*, *Merge*, *Form*, and *Disappear*. The structures and features of the network will change as strong events occur. An illustrative example of strong event is shown in Fig. 1.

As shown in Fig. 1, this network has two communities, C_t^1 and C_t^2, at time t. At time $t+1$, C_{t+1}^2 *splits* into two smaller communities as a result of the relationships change between nodes 10, 11 and other nodes. Simultaneously, the structure of C_{t+1}^1 remains unchanged, so C_{t+1}^1 is viewed to *remain* as an event. At time $t+2$, nodes 13 and 14 join this network, and node 14 establishes a relationship with nodes 3, 4, and 6. At the same time node 13 establishes a relationship with node 11. The cardinalities of community C_{t+1}^1 and C_{t+1}^3 *expand* due to these changes. At time $t+3$, community C_{t+3}^1 *shrinks* because node 1 exits from C_{t+3}^1. At time t+4, community C_{t+4}^3 *disappears* completely since all of its nodes

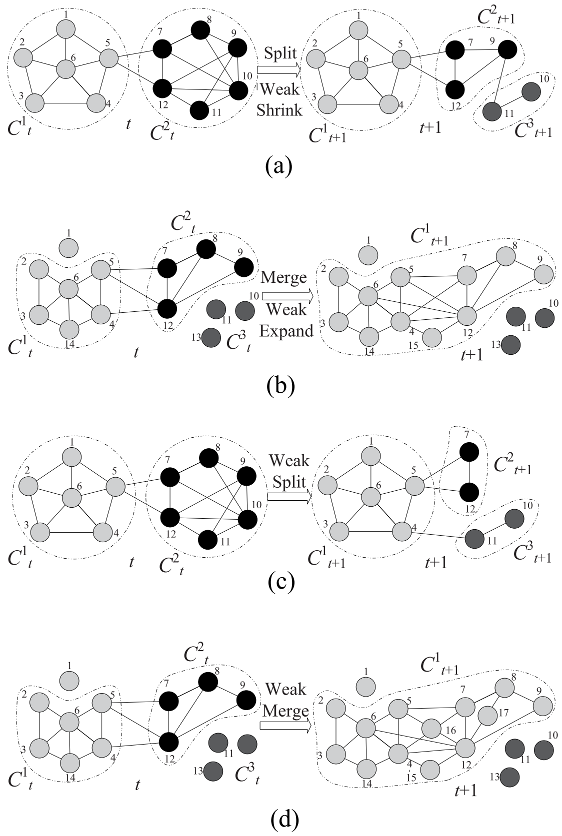


Fig. 2. Examples of weak events in community evolution. (a) Weak shrink. (b) Weak expand. (c) Weak split. (d) Weak merge.

maintain no relationship with external communities. At time t+5, the relationships between nodes 4, 5, 7, and 12 become so strong that it causes C^1_{t+4} to merge with C^2_{t+4}.

Remark 1: Due to the characteristics of nonrealtime evolving communities in complex networks, measurable change can be a slow occurring process. There may also be multiple evolutionary events occurring at the same time in similar communities.

Definition 3 (Weak Event): A weak event is triggered by small changes in the community. It is not detected by strong events, yet occur together with strong events, including: *Weak Shrink*, *Weak Expand*, *Weak Split*, and *Weak Merge* events.

As shown in Fig. 2(a), community C^2_t at time t splits and forms communities C^2_{t+1} and C^3_{t+1}, as seen at time $t+1$. Simultaneously, node 8 exists in C^2_t, and disappears from the network at time t+1. It can be observed then that a weak shrink event occurs from t to $t+1$ in C^2_t. According to Fig. 2(b), communities C_1 and C_2 aggregate at time $t+1$, and node 15 joins as well. This leads to the occurrence of a weak expanding event. According to Fig. 2(c), communities C^2_t and C^3_t, at time t, belong to C^2_t, and nodes 8 and 9 disappear at time t+1. Thus, it can be deduced that a weak splitting event occurs at time $t+1$ in C^2_t at time t. By Fig. 2(d), though communities C^1_t and C^2_t at time t belong to C_1, at time $t+1$, nodes 15–17 join in this network, and belong to C^1_{t+1} at time $t+1$. Thus, a weak merging event occurs at $t+1$ for communities C^1_t and C^2_t at t. The formal definition of these four weak events is described in Section III-B2.

TABLE I
DESCRIPTION OF IMPORTANT SYMBOLS

Symbol	Description
R, F, D E, SH SP, M	Community set having Remain, Form, Disappear, Expand, Shrink, Split, Merge events, respectively
WSH,*WE* *WSP*,*WM*	Community set having Weak Shrink, Weak Expand, Weak Split, Weak Merge events, respectively
O	Community overlapping degree
S	Community membership degree
C^p_t	Community at time t with label p
C^q_{t+1}	Community at time t+1 with label q
θ	Parameter for determining community existence
γ	Parameter for judging community changing size
ξ	Parameter for judging multi-community changes

Definition 4 (Community Overlapping Degree): Given community C^p_t at time t and community C^q_{t+1} at time $t+1$, the community overlapping degree of these two communities is defined as the proportion of the number of nodes in the intersection of communities to the number of nodes in the union of communities, as follows [30]:

$$O\left(C^p_t, C^q_{t+1}\right) = \frac{\left|C^p_t \bigcap C^q_{t+1}\right|}{\left|C^p_t \bigcup C^q_{t+1}\right|}. \tag{1}$$

Equation (1) is used to determine the persistence of relationships of nodes between communities at different time.

Definition 5 (Community Membership Degree): The community membership degree of community C^q_{t+1} at time $t+1$ and C^p_t at time t is equal to the proportion of the number of nodes in the interaction of these two communities to the number of nodes in C^p_t, which is defined as follows [30]:

$$S\left(C^p_t, C^q_{t+1}\right) = \frac{\left|C^p_t \bigcap C^q_{t+1}\right|}{\left|C^p_t\right|}. \tag{2}$$

Equation (2) implies that the degree of the community C^q_{t+1} belongs to community C^p_t. The community membership degree is used to determine whether a community belongs to another one. Specifically, it can be used to discover evolutionary events, such as splitting and merging events.

An event plays a fundamental role in community evolution analysis. The following section provides a detailed description of evolutionary events.

B. Definitions of Evolutionary Events

The nomenclature is provided in Table I.

1) Strong Events: According to the concepts given by Asur *et al.* [4] and Takaffoli *et al.* [18], strong events are defined as follows.

Definition 6 (Remain): Suppose there is a community C^q_{t+1} at time $t+1$ and another community C^p_t at time t, and they are highly overlapped and share many similar nodes. This phenomenon is called "Remain," and can be formalized by the following formula:

$$R\left(C_t^p, C_{t+1}^q\right) = 1$$
$$\text{iff.}\exists C_{t+1}^q \in C_{t+1}, O\left(C_t^p, C_{t+1}^q\right) \geq \theta \tag{3}$$

where C_{t+1} represents a dynamic network at time $t+1$.

Definition 7 (Form): If the community overlapping degree between community C_t^p and community C_{t-1}^q at time $t-1$ is very low, that is, the community C_t^p at time t has no relationship with other communities at time $t-1$, then a "Form" event occurs. This event can be formalized as

$$F\left(C_{t-1}^q, C_t^p\right) = 1$$
$$\text{iff.}\forall C_{t-1}^q \in C_{t-1}, O\left(C_t^p, C_{t-1}^q\right) < \theta. \tag{4}$$

Definition 8 (Disappear): If the community overlapping degree between community C_t^p and community C_{t+1}^q at $t+1$ is very low, C_t^p at time t has no relationship with other communities at time $t+1$, then a "Disappear" event occurs, which can be modeled as

$$D\left(C_t^p, C_{t+1}^q\right) = 1$$
$$\text{iff.}\forall C_{t+1}^q \in C_{t+1}, O\left(C_t^p, C_{t+1}^q\right) < \theta. \tag{5}$$

Definition 9 (Expand): If community C_t^p at time t belongs to another community at $t+1$, denoted by $C_t^p \subset C_{t+1}^q$, and the number of nodes in C_t^p is less than that of C_{t+1}^q, we call C_t^p "Expands" at time $t+1$ by the following formula:

$$E\left(C_t^p, C_{t+1}^q\right) = 1$$
$$\text{iff.}\exists C_{t+1}^q \in C_{t+1}, 1-\gamma \leq S\left(C_{t+1}^q, C_t^p\right) < 1. \tag{6}$$

Definition 10 (Shrink): If community C_{t+1}^q at time $t+1$ belongs to community C_t^p at time t, and the number of nodes in C_{t+1}^q is less than that of C_t^p, then a "Shrink" event occurs at time $t+1$, which is described as follows:

$$SH\left(C_t^p, C_{t+1}^q\right) = 1$$
$$\text{iff.}\exists C_t^p \in C_t, 1-\gamma \leq S\left(C_t^p, C_{t+1}^q\right) < 1. \tag{7}$$

Because $\gamma \in (0, 1]$, in order to detect Expand and Shrink events effectively, the left-hand constraints of Expand and Shrink events with respect to S are specified to 1-γ, rather than γ. While the value of S only changes within a limited range.

Definition 11 (Split): If there are $k(>1)$ communities $X=\{C_{t+1}^q, \ldots, C_{t+1}^{q+k}\}$ at time $t+1$, and each community in X almost belongs to community C_t^p, and the community overlapping degree between the union of communities in X and C_t^p is very high, C_t^p is viewed to "split" into different communities as follows:

$$SP\left(C_t^p, X\right) = 1$$
$$\text{iff}\begin{cases} O\left(C_t^p, X\right) \geq \xi \\ S\left(C_{t+1}^i, C_t^p\right) \geq \xi \quad \forall C_{t+1}^i \in X. \end{cases} \tag{8}$$

Definition 12 (Merge): If there are many communities $Y=\{C_t^p, \ldots, C_t^{p+k}\}$ at time t, each community in Y almost belongs to community C_{t+1}^q, and the community overlapping degree between the union set of communities in Y and C_{t+1}^q is very high, then a "merge" event occurs, which is represented by the following equation:

$$SP\left(C_{t+1}^q, Y\right) = 1$$
$$\text{iff}\begin{cases} O\left(Y, C_{t+1}^q\right) \geq \xi \\ S\left(C_t^i, C_{t+1}^q\right) \geq \xi \quad \forall C_t^i \in Y. \end{cases} \tag{9}$$

In Definitions 6–12, the parameters of θ, γ, and ξ are tuned by experiments in order to discover as many events as possible. Different from traditional event-based frameworks, the WECEM framework uses these three parameters to control the occurring conditions of each event.

Observation 1: Strong events can be classified into the following types.

1) *Form*, *Disappear*, and *Remain* involve the existence events of communities.
2) *Shrink* and *Expand* events are relevant to the change of community sizes.
3) *Split* and *Merge* involve the change of multiple communities.

For the aforementioned three kinds of evolutionary events, if only one parameter as shown in the definition of each event to control the evolution of communities, it is difficult to accurately detect evolutionary events. Therefore, we use three parameters θ, γ, and ξ as constraints for events.

Theorem 1: The size of the union of two communities is larger than that of each community, that is

$$\left|C_t^p \cup C_{t+1}^q\right| \geq \left|C_t^p\right| \tag{10}$$
$$\left|C_t^p \cup C_{t+1}^q\right| \geq \left|C_{t+1}^q\right|. \tag{11}$$

It is worth noting that (1) is used to determine the overlapping degree of two communities between adjacent timestamps and the persistent relationship between two communities. By Theorem 1, it is unlikely, and most probably a byproduct of randomness, when in real-world situations the union set of two communities can be used to determine the overlapping degree. Equation (1) takes into account the growth and shrink of different communities over time.

2) Weak Events: It can be assumed that strong events may be accompanied by weak events, which can lead to small changes in communities. However, it can not be viewed to constitute the change necessary to trigger traditional events as observed by event-based frameworks. At the same time, there exist some changes that do not satisfy the requirements of strong events, despite there being measurable changes in the network. This phenomenon is referred to as a "weak event," which can serve as a complement for strong events. A formal definition is as follows.

Definition 13 (Weak Shrink): The phenomenon of a slight or measurably small shrink of nodes in communities is called a weak shrink event. This event occurs at the same time as a strong event. Weak shrink events appear in the following three scenarios.

1) When a Remain event occurs, community C_t^p at time t belongs to community C_{t+1}^q at time $t+1$, and the size of the intersection of the two communities, at various time intervals, is less than that of the communities at time t.

2) When a Form event occurs, resulting from a community at time t shrinking but not splitting, and event is not observed even though a new community has formed at time $t+1$.
3) When a Split event occurs, the size of the intersection of X, which represents the union of communities at time $t+1$ and community C_t^p at time t, is less than the size of C_t^p at time t.

The above three cases can be formulated as follows:

$$WSH = \begin{cases} S\left(C_t^p, C_{t+1}^q\right) < 1, & \left(C_t^p, C_{t+1}^q\right) \in R \\ S\left(C_{t+1}^q, C_t^p\right) \geq \theta, & \begin{cases} \left(C_t^p, C_{t+1}^q\right) \in F \\ \left(C_t^p, C_{t+1}^q\right) \notin SP \end{cases} \\ S\left(C_t^p, X\right) < 1, & \left(C_t^p, X\right) \in SP \end{cases} \tag{12}$$

Definition 14 (Weak Expand): A weak expanding event occurs along side strong events, which indicates a slow growth in communities. Weak expanding events appear in the following three scenarios.

1) When a Remain event occurs, the size of the intersection of two communities at time t is less than that of communities at the next timestamp.
2) When a Form event occurs, since the community at time t expands without the occurrence of a Merge event, a new community nevertheless forms at time $t+1$.
3) When a Merge event occurs, the size of the interaction set of Y (which represents the union of communities at time t) and community C_{t+1}^q at time $t+1$, is less than the size of C_{t+1}^q at time $t+1$.

The above three cases can be modeled as follows:

$$WE = \begin{cases} S\left(C_{t+1}^q, C_t^p\right) < 1, & \left(C_t^p, C_{t+1}^q\right) \in R \\ S\left(C_t^p, C_{t+1}^q\right) \geq \theta, & \begin{cases} \left(C_t^p, C_{t+1}^q\right) \in B \\ \left(C_t^p, C_{t+1}^q\right) \notin M \end{cases} \\ S\left(Y, C_{t+1}^q\right) < 1, & \left(Y, C_{t+1}^q\right) \in M. \end{cases} \tag{13}$$

Definition 15 (Weak Split): If a community at time $t+1$ belongs to another community at time t, but the union of these communities cannot represent the one at time t, then this phenomenon is called a "weak split," which can be described further with the following formula:

$$WSP = \begin{cases} \forall C_{t+1}^i \in X, S\left(C_{t+1}^i, C_t^p\right) \geq \xi \\ O\left(X, C_t^p\right) < \xi. \end{cases} \tag{14}$$

Definition 16 (Weak Merge): If some communities at time t belong to one community at time $t+1$, but the union of these communities cannot constitute the community at time $t+1$, this phenomenon is called "weak merge," which is defined as follows:

$$WM = \begin{cases} \forall C_t^i \in Y, S\left(C_t^i, C_{t+1}^q\right) \geq \xi \\ O\left(C_t^p, Y\right) < \xi. \end{cases} \tag{15}$$

In Definitions 6–16, the parameter θ is specified to the same value, and γ and ξ as well. These three parameters are determined empirically by experimentation.

Detailed descriptions of various events are given in Table II. The complexity represents the cardinality of changing nodes in two communities, when events are detected by the framework. The absolute value in the last column represents the size of the corresponding communities.

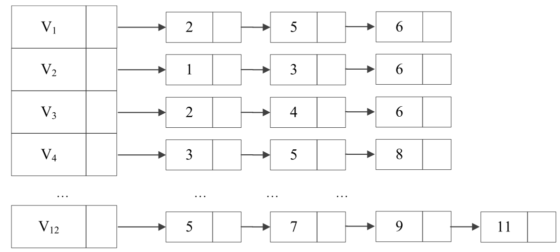


Fig. 3. Linked-list storage structure.

Remark 2: There is a difference between splitting and merging events in terms of time sequences. For a splitting event, we compare communities over different time sequentially. Contrarily, for a merging event, we have to compare communities in a reverse time sequence.

Weak events are the manifestations of small changes in communities. Since these changes are not apparent, we cannot detect them via traditional event-based frameworks. More generally, vast networks evolve slowly with myriad small changes, which are difficult to detect, but nonetheless serve as the catalyst for strong events. Therefore, a case can be made that detecting weak events can be of equal, if not of greater importance, for successfully detecting changing trends in dynamic networks. This can be paramount in helping service providers predict future developments of communities.

Remark 3: When compared with strong events, weak events occur more frequently, and the occurrences of a large number of weak events are an indicator for an eventual strong event.

IV. Community Evolution Detection Algorithm Based on Weak Events

WECEM includes the following steps: 1) detecting Remain, Disappear, and their accompanying events, including Weak Expand and Weak Shrink; 2) detecting Expand and Shrink events; 3) detecting Split, Weak Split and Weak Shrink events; 4) detecting Form, Weak Shrink, and Weak Expand events; and 5) detecting Merge, Weak Merge and Weak Expand events. Before discovering evolutionary events, duplicated edges are eliminated and indices of nodes are reordered. In particular, we use the linked-list storage structure as shown in Fig. 3.

We apply the above data structure because there are a huge volume of network data generated at different time, it is difficult to use a very large matrix to store the big network structure. Contrarily, linked lists with head nodes can help greatly compress the storage space in order to reduce the cost of determining whether an edge does exist.

A. Remain, Disappear, and Accompanying Event Detection

Algorithm 1 can be summarized as follows.

1) For each community at time t, if there is a community at time $t+1$ in which the community overlapping degree with it is bigger than θ, a Remain event occurs (lines 1–4).
2) If these two communities do not have complete membership relationship, Weak Shrink, and Weak Expand events have occurred (lines 5–8).

TABLE II
DESCRIPTION OF EVENTS

Event	Formal definition	Computational complexity
Remain	$\exists C_{t+1}^q \in C_{t+1}, O(C_t^p, C_{t+1}^q) \ge \theta$	$\lvert C_t^p\rvert * \lvert C_{t+1}^q\rvert$
Form	$\forall C_{t-1}^q \in C_{t-1}, O(C_t^p, C_{t-1}^q) < \theta$	$\lvert C_t^p\rvert * \lvert C_{t+1}^q\rvert$
Disappear	$\forall C_{t+1}^q \in C_{t+1}, O(C_t^p, C_{t+1}^q) < \theta$	$\lvert C_t^p\rvert * \lvert C_{t+1}^q\rvert$
Expand	$\exists C_{t+1}^q \in C_{t+1}, 1-\gamma \le S(C_{t+1}^q, C_t^p) < 1$	$\lvert C_t^p\rvert * \lvert C_{t+1}^q\rvert$
Shrink	$\exists C_t^p \in C_t, 1-\gamma \le S(C_t^p, C_{t+1}^q) < 1$	$\lvert C_t^p\rvert * \lvert C_{t+1}^q\rvert$
Split	$\begin{cases} O(C_t^p, X) \ge \xi \\ S(C_{t+1}^i, C_t^p) \ge \xi, \forall C_{t+1}^i \in X \end{cases}, C_t^p \in C_t$	$\lvert X\rvert * \lvert C_t^p\rvert * \lvert C_{t+1}^q\rvert$
Merge	$\begin{cases} O(Y, C_{t+1}^q) \ge \xi \\ S(C_t^i, C_{t+1}^q) \ge \xi, \forall C_t^i \in Y \end{cases}, C_{t+1}^p \in C_{t+1}$	$\lvert Y\rvert * \lvert C_t^p\rvert * \lvert C_{t+1}^q\rvert$
Weak Shrink	$\begin{cases} S(C_t^p, C_{t+1}^q) < 1, (C_t^p, C_{t+1}^q) \in R \\ S(C_{t+1}^q, C_t^p) \ge \theta, \begin{cases} (C_t^p, C_{t+1}^q) \in B \\ (C_t^p, C_{t+1}^q) \notin SP \end{cases} \\ S(C_t^p, X) < 1, (C_t^p, X) \in SP \end{cases}$	$(\lvert R\rvert + \lvert F\rvert + \lvert SP\rvert) * \lvert C_t^p\rvert * \lvert C_{t+1}^q\rvert$
Weak Expand	$\begin{cases} S(C_{t+1}^q, C_t^p) < 1, (C_t^p, C_{t+1}^q) \in R \\ S(C_t^p, C_{t+1}^q) \ge \theta, \begin{cases} (C_t^p, C_{t+1}^q) \in B \\ (C_t^p, C_{t+1}^q) \notin M \end{cases} \\ S(Y, C_{t+1}^q) < 1, (Y, C_{t+1}^q) \in M \end{cases}$	$(\lvert R\rvert + \lvert F\rvert + \lvert M\rvert) * \lvert C_t^p\rvert * \lvert C_{t+1}^q\rvert$
Weak Split	$\begin{cases} \forall C_{t+1}^i \in X, S(C_{t+1}^i, C_t^p) \ge \xi \\ O(X, C_t^p) < \xi \end{cases}$	$\lvert X\rvert * \lvert C_t^p\rvert * \lvert C_{t+1}^q\rvert$
Weak Merge	$\begin{cases} \forall C_t^i \in Y, S(C_t^i, C_{t+1}^q) \ge \xi \\ O(C_t^p, Y) < \xi \end{cases}$	$\lvert Y\rvert * \lvert C_t^p\rvert * \lvert C_{t+1}^q\rvert$

Algorithm 1 Remain, Disappear, Weak Expand, and Weak Shrink Event Detection

Input: The community set C_t at t, and C_{t+1} at t+1.
Output: *R*, *WE*, *WSH*, *D*.
1. **for** each $c_t \in C_t$ **do**
2. c_{t+1}=*find*(C_{t+1}, $O(c_t, c_{t+1})$);
3. **if** $c_{t+1} \neq \emptyset$ **then**
4. *R*=*insert*(c_t);
5. **if** $S(c_t, c_{t+1})<1$ **then**
6. *WSH*=*insert*(c_t);
7. **if** $S(c_{t+1}, c_t)<1$ **then**
8. *WE*=*insert*(c_t);
9. **else**
10. *D*=*insert*(c_t);
11. **output** *R*, *WE*, *WSH*, *D*.

3) If there is no community at time $t+1$ similar to a community at t, a Disappear event has occurred (lines 9 and 10).
4) Finally, it outputs identified events (line 11).

B. Expand and Shrink Event Detection

The main steps of Algorithm 2 include as follows.

1) It compares communities at time t with communities at time $t+1$, if the community membership degree of communities at time $t+1$ meets with the communities at time t in (6), an Expand event occurs (lines 1–4).
2) If the community membership degree meets with Equation (7), a Shrink event occurs (lines 5 and 6).

Algorithm 2 Expand and Shrink Event Detection

Input: The community set C_t at t, and C_{t+1} at t+1.
Output: *E*, *SH*.
1. **for** each $c_t \in C_t$ **do**
2. **for** each $c_{t+1} \in C_{t+1}$ **do**
3. **if** 1-$\gamma \le S(c_{t+1},c_t)<1$ **then**
4. *E*=*insert*(c_t);
5. **else if** 1-$\gamma \le S(c_t,c_{t+1})<1$ **then**
6. *SH*=*insert*(c_{t+1});
7. **output** *E*, *SH*.

Algorithm 3 Split and Its Accompanied Event Detection

Input: The community set C_t at t, and C_{t+1} at t+1.
Output: *SP*, *WSP*, *WSH*.
1. **for** each $c_t \in C_t$ **do**
2. **for** each $c_{t+1} \in C_{t+1}$ **do**
3. **if** $S(c_{t+1}, c_t) \ge \xi$ **then**
4. $C'=C' \cup c_{t+1}$;
5. **if** $O(c_t, C') \ge \xi$ **then**
6. *SP*=*insert*(c_t);
7. **if** $S(c_t, C')<1$ **then**
8. *WSH*=*insert*(c_t);
9. **else**
10. *WSP*=*insert*(c_t);
11. **output** *SP*, *WSP*, *WSH*.

3) Finally, it outputs identified events (line 7).

C. Split and Accompanying Event Detection

Algorithm 3 includes the following important steps.

Algorithm 4 Form and Its Accompanied Event Detection

Input: The community set C_t at t, and C_{t+1} at t+1.
Output: *F*, *WSH*, *WE*.
1. **for** each $c_{t+1} \in C_{t+1}$ **do**
2. c_t=*find*(c_t, $O(c_{t+1}, c_t) \geq \theta$)
3. **if** $c_t = \emptyset$ **then**
4. *F*=*insert*(c_{t+1});
5. **if** $S(c_{t+1}, c_t) \geq \theta$ **then**
6. *WSH*=*insert*(c_t);
7. **if** $S(c_t, c_{t+1}) \geq \theta$ **then**
8. *WE*=*insert*(c_t);
9. **output** *F*, *WSH*, *WE*.

Algorithm 5 Merge and Its Accompanying Events Detection

Input: The community set C_t at t, and C_{t+1} at t+1.
Output: *M*, *WM*, *WE*.
1. **for** each $c_{t+1} \in C_{t+1}$ **do**
2. **for** each $c_t \in C_t$ **do**
3. **if** $S(c_t, c_{t+1}) \geq \xi$ **then**
4. $C' = C' \cup c_t$;
5. **if** $O(c_{t+1}, C') \geq \xi$ **then**
6. *M*=*insert*(c_{t+1});
7. **if** $S(C', c_{t+1}) < 1$ **then**
8. *WE*=*insert*(c_{t+1});
9. **else**
10. *WM*=*insert*(c_{t+1});
11. **output** *M*, *WM*, *WE*.

1) It compare communities at time t with all communities at time $t+1$, if a community at time $t+1$ belongs to the community at time t, stores this community (lines 1–4).
2) If the community overlapping degree between the union set of these communities and the community at time t satisfies the condition of Split, a Split event occurs (lines 5 and 6). If the community membership degree between the union of these communities and the community at time t is very small, a Weak Shrink event occurs (lines 7 and 8).
3) Otherwise, Weak Split occurs (lines 9 and 10).
4) Finally, it outputs identified events (line 11).

D. Form and Accompanying Event Detection

The main steps of Algorithm 4 include as follows.

1) For each community at time $t+1$, if there is no community at time t satisfying the forming condition, a Form event occurs (lines 1–4).
2) When a Form event occurs, if the community membership degree of c_{t+1} to c_t is bigger than θ, a Weak Shrink event occurs (lines 5 and 6). If the community membership degree of c_t to c_{t+1} is bigger than θ, a Weak Expand event occurs (lines 7 and 8).
3) Finally, it outputs identified events (line 9).

E. Merge and Accompanying Events Detection

The main steps of Algorithm 5 are as follows.

1) It compares a community at time $t+1$ with all communities at t. If multiple communities at time t belong to the community at $t+1$, store these communities (lines 1–4). If the community overlapping degree between the union of these communities at time t and the community at t+1 satisfies the merging condition, a Merge event occurs (lines 5 and 6); otherwise, a Weak Merge event does occur (lines 9 and 10). For communities with Merge events, if the union set of these communities does not belong to the community at time t, a Weak Expand event occurs (lines 7 and 8).
2) Finally, it outputs identified events (line 11).

F. Algorithm Complexity Analysis

For a network $G(V, E)$ with n nodes and m edges, in Algorithms 1–5, each algorithm visits all nodes in the network at adjacent timestamps, after which they determine whether the number of nodes changes, in order to determine which event happens, similar to visiting the Cartesian product of nodes at adjacent timestamps. Therefore, the time complexity of Algorithms 1–5 is $O(n^2)$.

V. Experiments

A. Experimental Setup

In order to verify the accuracy and efficiency of the proposed community evolution analysis framework, we conduct experiments using real data as well as large-scale synthetic network data sets, including: 1) two types of synthetic dynamic networks generated by the data generator [31] and 2) real dynamic networks, including DBLP data set [32] and Facebook data set from New Orleans in 2008 [33]. The details of these data are shown in Table III.

The first type of synthetic data is generated by the dynamic network D3, with parameters listed in Table III(a), without specifying the number of evolutionary events. The second type of synthetic data is generated by the dynamic networks D1 and D2 in Table III(a), in order to estimate the correctness of the WECEM framework, where the D1 dataset generates 50 Form and Disappear events, 10 Merge and Split events, 50 Shrink and Expand events, while the D2 dataset is specified to have 200 Form and Disappear events, 50 Split and Merge events, and 200 Shrink and Expand events.

As we can see from Table III(a), for the synthetic dynamic network datasets D1, D2, and D3, the dynamic community evolution events were analyzed across 5 time steps, and the time steps is determined based on the following rules: The networks began at $t = 1$ with around 400 embedded communities, which were constrained to have sizes between [20, 60]. In these three synthetic datasets, twenty percent of node memberships were randomly permuted at each step to simulate users' movement across communities over time. Then, events were added by the generator. As for the real DBLP dataset in Table III(b), the number of time steps for community evolution analysis is 5 (years). Similarly, for the real Facebook dataset in Table III(c), the number of time steps for community evolution analysis is 12 (months).

TABLE III
DESCRIPTION OF EXPERIMENTAL DATA SETS. (a) SYNTHETIC DYNAMIC NETWORK DATA. (b) DBLP NETWORK DATA. (c) FACEBOOK NETWORK DATA

(a)

Parameter	Description	D1	D2	D3
s	No. of timestamps	5	5	5
N	No. of nodes	5000	10000	15000
k	Average degree	10	5	10
max_k	Maximum degree	20	20	30
max_c	Max. community size	30	30	60
min_c	Min. community size	10	5	10
μ	Mixed parameter	0.2	0.2	0.2

(b)

Year	No. of nodes(V)	No. of edges(E)	Average degree
2005	1061	1254	2.363
2006	1325	1572	2.372
2007	1379	1574	2.283
2008	1411	1629	2.309
2009	1243	1343	2.161

(c)

Month	No. of nodes	No. of edges	Average degree
1	10990	12462	2.268
2	11193	12371	2.21
3	18360	30700	3.344
4	17360	32270	3.718
5	17382	30987	3.565
6	18606	31607	3.398
7	21465	38540	3.591
8	23908	45409	3.799
9	24237	42959	3.545
10	27462	53014	3.861
11	29904	56342	3.768
12	34624	70307	4.061

The proposed WECEM framework is implemented by Java programming language, and we compare it with classic Asur framework [4] and Takaffoli framework [2], where the parameter k of the Asur and the Takaffoli framework is set to 0.5 based on experimental studies. The hardware environment includes the Intel Corei7-4710HQ processor, and 8G memory. Each framework executes 3 times on each data set, and we take the average value to show their performance.

Definition 17 [Event Mining Accuracy (EMA)]: EMA represents the accuracy of event detection, which equals the proportion of the number of correctly identified communities to the actual number of communities with events happening

$$\mathrm{EMA}_P = \frac{\sum_{t\in T}\left\{\left|C_t^P \cap C_t^{P'}\right|\right\}}{\sum_{t\in T}\max\left\{\left|C_t^P\right|, \left|C_t^{P'}\right|\right\}} \tag{16}$$

where EMA_P represents EMA of a particular event P, T represents the set of timestamps, C_t^P represents the community where P occurs at time t detected by algorithms, $C_t^{P'}$ represents the true community where P happens at time t, and EMA is used to evaluate the accuracy of each framework for detecting evolutionary events.

B. Parameter Specification

In order to take into account the quantity and quality of discovered events, it is necessary to determine the value of the community existence parameter θ, the community scale changing parameter γ and the multiple community changing parameter ξ. The accuracy of WECEM is relevant to these three parameters, thus choosing appropriate parameter values can help discover more reliable events. θ and γ are specified by experiments on the D1 data set. Because the D1 data set contains only 10 Split and Merge events and the scale of data is small, the accuracy of event mining cannot be accurately displayed on D1, ξ is determined by experiments on the D2 data set. The results are shown in Fig. 4, while Fig. 4(a)–(d) demonstrate the experimental results by changing θ and γ parameters on the D1 data set, and Fig. 4(e) and (f) show the results by specifying different ξ values on the D2 data set.

In Fig. 4(a), as θ grows, except for Form and Disappear events, the number of other events decreases, because Form and Disappear events require that the community overlapping degree between communities at adjacent timestamps is less than θ. When the value of θ is very small, it is difficult to find Form and Disappear events. However, as the value of θ grows, the constraint for Form and Disappear events becomes increasingly loose, and it can discover more such events. On the contrary, with the value of θ growing, the constraint for other events becomes strict, so the number of discovered events decreases.

As shown in Fig. 4(b), when $\theta < 0.4$, as the value of θ grows, EMA of Form and Disappear events grows gradually. Because as θ grows, the number of Form and Disappear events changes to the actual situation of these two events. But when $\theta > 0.4$, as θ increases, redundant events increase gradually, which leads to the decrease of EMA. According to the experimental results of discovered events on the D1 data set, θ is specified to 0.4 in the following experiments so as to avoid the overlapping of strong events and enable WECEM to find as many events as possible.

As shown in Fig. 4(c), as γ increases, the number of Expand and the Shrink events grows gradually, because the constraint range of Expand and Shrink events becomes large with γ increasing. However, the capability of distinguishing these events becomes weak, and the discovered events will be overlapped by other events, especially for Shrink events. When $\gamma = 1$, the bound constraint is between [0, 1], and the Shrink event happens in almost all communities.

In Fig. 4(d), as γ grows, EMA of Expand and Shrink events increases at first and then drops, because when γ is small, the number of events is small, but, when γ is very large, there exists several redundant events. In order to identify more events in an effective fashion, the numbers of Shrink and Expand events on the D1 data set are increased to 50 and γ is specified to 0.3.

By Fig. 4(e), the number of Split and Merge events decreases with ξ, because when ξ is set to be large, the

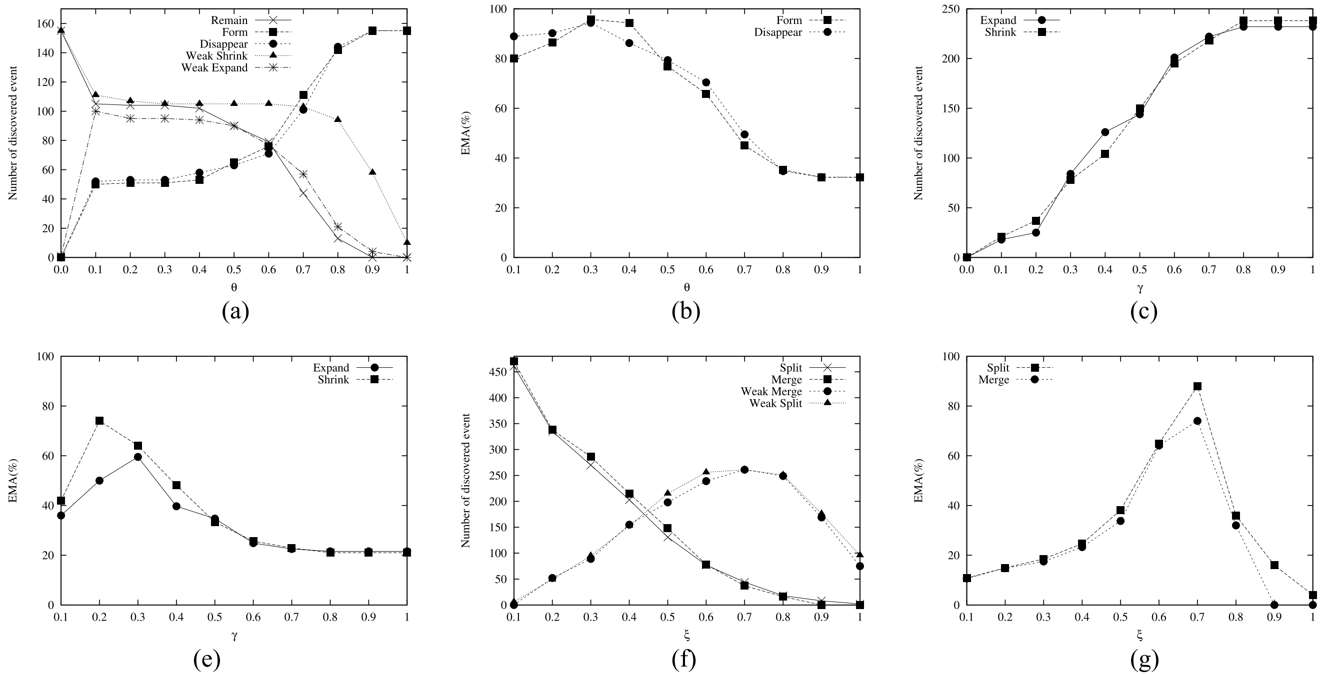


Fig. 4. Number of discovered events and EMA by the WECEM framework with different θ, γ and ξ values. (a) Number of discovered events as θ changes. (b) EMA as θ changes. (c) Number of discovered events as γ changes. (d) EMA as γ changes. (e) Number of discovered events as ξ changes. (f) EMA as ξ changes.

number of communities meeting the constraints of overlapping degree and membership degree decreases, and the corresponding events will reduce gradually. When $0 < \xi \leq 0.6$, as ξ grows, the number of communities grows which satisfies the requirement of membership degree without satisfying the requirement of community overlapping degree, thus the numbers of Weak Split and Weak Merge events grow gradually. When $0.6 < \xi \leq 1$, these two conditions cannot be met, so the changing trends of Weak Split and Weak Merge events (Split and Merge events) are nearly the same.

According to Fig. 4(f), when 0< ξ <0.7, EMA increases with ξ. When 0.7< ξ <1, EMA degrades with ξ. When ξ is small, with ξ growing, the number of discovered events is approximate to the number of actually occurred events. When ξ is specified to a large value, the number of redundant events grows, which leads to the decrease of EMA. Based on the above discussion, in order to accurately identify evolutionary events, ξ is set to 0.6 in experiments.

C. Quantity Analysis of Detected Events

In this section, we compare the number of correctly detected events among different community detection frameworks.

1) Quantity Comparison of Detected Events on DBLP: Table IV shows the number of detected events on the DBLP data set, and the following observations can be drawn.

1) For each framework, the numbers of Form and Disappear events are the largest compared with other events. This is because DBLP is a coauthor network, with some scholars publishing papers and other scholars may not publishing papers each year. Consequently, several small communities are formed, thus the numbers of Form and Disappear events are larger than that of other events.
2) As shown in Table IV(a), although there are many weak events in the DBLP network, the change of nodes and edges relevant to weak events is so small, which does not cause a qualitative change in the community. Additionally, because the weak events are accompanied with strong events, even weak events overlap with strong events, discovering weak events can help accurately predict the variation tendency of network structures. However, Asur, Takaffoli, and other event-based frameworks does not work for identifying weak events, while they only focus on strong events which are easy to be found. Actually, the phenomenon of overlapping events rarely appears, and it is difficult for traditional event-based frameworks to detect events with a slowly changing tendency in dynamic networks.
3) By comparing Table IV(a) with Table IV(b), the numbers of events detected by WECEM are larger than that mined by the Asur framework and WECEM outperforms traditional frameworks by 22.9% in the number of discovered strong events. Because the definition of events by Asur is strict, which makes the Asur framework difficult to detect events. Taking the Remain event as an example, it requires the number of nodes in a community should be exactly the same at adjacent timestamps. Moreover, as for Form events, there should be no similar nodes in a community at time t and $t+1$.
4) By comparing Table IV(a) with Table IV(c), we can find that the number of events identified by WECEM is almost the same as Takaffoli, since WECEM uses multiple parameters to deal with different events. In

TABLE IV
QUANTITY COMPARISON ON DBLP DATASETS. (a) NUMBER OF EVENTS DETECTED BY WECEM. (b) NUMBER OF EVENTS DETECTED BY ASUR. (c) NUMBER OF EVENTS DETECTED BY TAKAFFOLI

(a)

Year	Remain	Form	Disappear	Shrink	Expand	Split	Merge	Weak Merge	Weak Shrink	Weak Split	Weak Expand
05-06	31	251	196	0	0	1	0	17	31	14	42
06-07	22	286	260	0	0	0	0	10	36	13	31
07-08	21	283	287	0	0	1	1	9	21	9	29
08-09	13	264	291	0	0	0	0	4	25	9	20

(b)

Year	Remain	Form	Disappear	Split	Merge
05-06	0	229	177	0	0
06-07	0	263	238	0	0
07-08	0	263	267	0	0
08-09	0	241	274	0	0

(c)

Year	Remain	Form	Disappear	Split	Merge
05-06	30	262	197	3	0
06-07	19	288	263	0	0
07-08	21	285	287	0	0
08-09	7	261	297	0	2

addition, community overlapping degree and membership degree works to accurately detect evolutionary events. For a small scale of events, Takaffoli can discover more events than WECEM, because Takaffoli focuses on the detection of multiple networks before and after the time slice, the event that meets the condition is identified, and the corresponding event is considered to have occurred.

2) Quantity Comparison of Detected Events on D3 Data: Table V shows the number of events detected by each framework. The most important difference from the D3 data set to the D1 and D2 data sets lies in we do not need to manually specify the number of events, which is approximate to the real-world dynamic network structures.

According to [31], at every moment, nodes are selected from the D3 synthetic data set to simulate the changes of the network. This kind of network evolves in a continuous and stable fashion, which is similar to the real dynamic network. Consequently, the numbers of events discovered by these three frameworks over time are relatively stable.

As shown in Table V, the Asur framework cannot handle large-scale synthetic network data because the node changes in each time slice of the simulated network are extracted proportionally from the last moment, and many communities have small changes, which is not sufficient to satisfy the definition of events in the Asur framework, resulting in zero for each event in the Asur framework. On the contrary, Tables V(a) and (c) show that the WECEM framework discovers the similar numbers of Remain, Form and Disappear events to the Takaffoli framework, which verifies the effectiveness of the WECEM framework. For the Split and Merge events, Takaffoli discovers more events than WECEM. Because the values of parameters in WECEM is set to be small, which results in many redundant events. On the other hand, the Takaffoli framework needs to compare the number of nodes at each time slice in event detection.

In Tables IV(a) and V(a), there are a large number of weak events on each data set. We can conclude that weak events are very common in complex networks. For two networks evolving during a short period of time, there are a large number of weak events because of the slow changes in the number of nodes and edges. Finally, the frequent occurrences of weak events result in strong events.

D. Event Detection Quality Analysis

For the D1 and D2 datasets, we generate a fixed number of events. The EMA measurement is used to verify the accuracy of event detection by comparing the number of discovered events with the actual number of events on these three frameworks. It is noteworthy that we cannot manually specify the number of events in community evolution on DBLP and Facebook datasets that are from real networks. It is difficult to accurately estimate the accuracy of event mining, so we conduct experiments on the D1 and D2 datasets and the experimental results are shown in Fig. 5.

Fig. 5 shows the accuracy of event detection on the D1 and D2 datasets by each framework. Given that Asur and Takaffoli do not define Shrink and Expand events, we only show the results about Shrink and Expand events from WECEM. The following conclusions are made by Fig. 5.

1) WECEM is more accurate than Asur and Takaffoli in discovering Form and Disappear events that occur in various communities. The reason behind is that although Asur can guarantee the accuracy of event detection, it is too strict to define different events and the number of events discovered by the Asur framework is very small, so its EMA is very low. On the other hand, the Takaffoli framework considers the change of the number of nodes at all timestamps when mining events and the change of communities in synthetic networks are regular, the redundant nodes mined by the Takaffoli framework are higher than that of the WECEM framework.
2) As for the Split and Merger events, the WECEM event discovery accuracy is lower than that of the Asur framework. This can be explained by the fact that we use the data generator designed by Greene *et al.* [31] to generate dynamic synthetic network datasets in order to mine Split and Merge events, and this data generator

TABLE V
QUANTITY COMPARISON OF EVENTS ON D3 DATASETS. (a) NUMBER OF EVENTS DETECTED BY WECEM. (b) NUMBER OF EVENTS DETECTED BY ASUR. (c) NUMBER OF EVENTS DETECTED BY TAKAFFOLI

(a)

Time	Remain	Form	Disappear	Shrink	Expand	Split	Merge	Weak Merge	Weak Shrink	Weak Split	Weak Expand
T1-T2	534	3	3	244	255	50	49	213	500	217	511
T2-T3	528	9	9	238	253	41	45	21	496	220	511
T3-T4	528	9	9	231	236	42	46	213	501	220	506
T4-T5	526	11	11	227	224	43	38	221	496	218	498

(b)

Time	Remain	Form	Disappear	Split	Merge
T1-T2	0	0	0	0	0
T2-T3	0	0	0	0	0
T3-T4	0	0	0	0	0
T4-T5	0	0	0	0	0

(c)

Time	Remain	Form	Disappear	Split	Merge
T1-T2	534	3	3	0	0
T2-T3	529	8	8	0	0
T3-T4	527	11	10	0	0
T4-T5	528	9	9	0	0

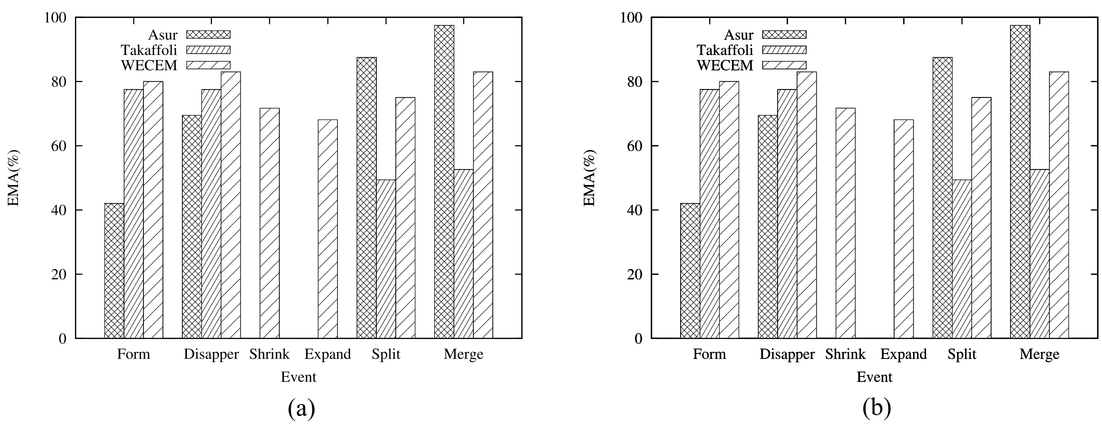


Fig. 5. Event detection accuracy comparison on different datasets. (a) EMA on D1 dataset. (b) EMA on D2 dataset.

is developed based on Asur framework. On one hand, WECEM can identify all of Split and Merge events due to the relaxed definition of evolutionary events when compared to the Asur framework. On the other hand, more redundant events will be found by the WECEM framework, which leads to the lower accuracy of discovering Merger and Split events compared with the Asur framework. Although some of the events found by WECEM are redundant, these communities with these events actually have changed in the network, which mainly constitutes weak events. Similarly, the Takaffoli framework has a lower accuracy and higher redundancy rate than the WECEM framework, although the Takaffoli framework can identify all Split and Merge events. The reasons of its higher redundancy are twofold: a) the synthetic network data agrees with some regularity without considering the distribution of events at each timestamp and b) the Takaffoli framework uses the uniform parameter k to control community similarity and discover events in the network, whereas WECEM takes into account three parameters, which plays important roles in accurately mining evolutionary events.

3) EMA of WECEM is averagely 2.13% higher than that of the Asur framework and 12.2% higher than that of the Takaffoli framework. The accuracy of WECEM is higher than that of the Asur framework. This is because WECEM uses multiple parameters to distribute different kinds of events in order to avoid the overlap of events as well as reduce the redundancy rate of mining events.

E. *Efficiency Analysis of Detecting Events*

In this section, we compare the execution time of each framework on these five datasets, including DBLP, Facebook, D1, D2, and D3. The experimental results are shown in Fig. 6, where the x-axis represents the time interval of two adjacent timestamps. As shown in Fig. 6(e), the y-axis represents the execution time of each framework.

As demonstrated in Fig. 6, with the number of nodes growing, the runtime of these three frameworks increases gradually, and the following conclusions can be made.

1) According to Fig. 6(a), (c), (d), and (e), the Asur framework runs first on the DBLP, D1, D2, and D3 datasets, followed by the Takaffoli framework, with the WECEM framework being the lowest one. This can be explained

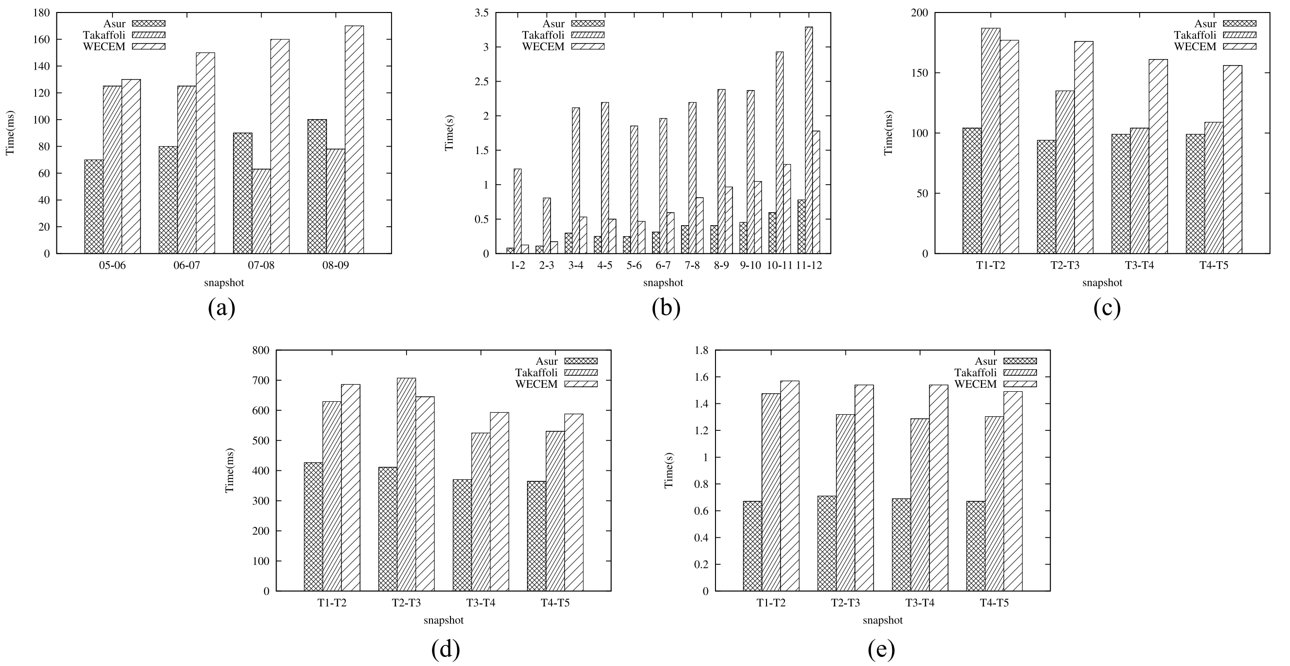


Fig. 6. Execution time comparison of different frameworks on different datasets. (a) DBLP. (b) Facebook. (c) D1. (d) D2. (e) D3.

by the reason that WECEM needs to simultaneously identify 11 kinds of events composed of strong events and weak events, while the Asur framework and the Takaffoli framework only need to mine five kinds of elementary events.

2) From Fig. 6(b), the most efficient framework is Asur, followed by WECEM, and the slowest one is Takaffoli on the Facebook dataset. Because the Facebook datasets have a large number of nodes involved over several timestamps, the Takaffoli framework detects every event by comparing the number of nodes over all timestamps. However, WECEM and Asur only need to compare the number of nodes at adjacent timestamps. So the time complexity of Takaffoli is higher than WECEM and Asur. The efficiency of Asur is higher than that of WECEM, as Asur framework does not need to detect Shrink and Expand events and it cannot detect weak events as well. For Facebook datasets, the efficiency of WECEM is 48.83% less than Asur and is 67.73% higher than Takaffoli. In summary, the proposed WECEM framework can deal with large-scale dynamic networks over several timestamps, which is more flexible and generic than the Takaffoli framework.

VI. Conclusion

In this article, we have explored the fundamental principle and working mechanism of the WECEM framework for weak event mining in the community evolution of dynamic complex networks. WECEM classifies events into strong events and weak events. Two measurements, community overlapping degree, and community membership degree, are used to determine the continuity of dynamic communities in complex networks. To calculate the community overlapping degree and community membership degree, the WECEM framework first compares each community at consecutive timestamps, respectively, and then discovers different events based on these two measurements. The experimental results have indicated that the WECEM framework is effective at mining events. Particularly, WECEM can discover weak events which cannot be handled by other frameworks. In addition, the experimental results have also shown that the WECEM framework is effective at detecting strong as well as weak events. As for mining large-scale dynamic networks, the advantage of WECEM is apparent, since it can detect small changes in the network. Given that WECEM needs much time to mine several kinds of events, its efficiency is less than the traditional frameworks in some cases. In our future work, we will continue to improve the accuracy of event mining by reducing redundant events. Because traditional serial community evolution analysis methods cannot handle a big network data, and we will parallel the WECEM framework to mine larger complex networks with a huge number of nodes and complex relationships.

References

[1] M. E. J. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 2, 2004, Art. no. 026113.

[2] M. Takaffoli, F. Sangi, J. Fagnan, and O. R. Zaiane, “Community evolution mining in dynamic social networks,” *Soc. Behav. Sci.*, vol. 22, no. 22, pp. 49–58, 2011.

[3] S. Qiao, N. Han, J. Wang, R.-H. Li, L. A. Gutierrez, and X. Wu, “Predicting long-term trajectories of connected vehicles via the prefix-projection technique,” *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 7, pp. 2305–2315, Jul. 2018.

[4] S. Asur, S. Parthasarathy, and D. Ucar, “An event-based framework for characterizing the evolutionary behavior of interaction graphs,” *ACM Trans. Knowl. Disc. Data*, vol. 3, no. 4, pp. 1–36, 2009.

[5] S. Qiao *et al.*, “A fast parallel community discovery model on complex networks through approximate optimization,” *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1638–1651, Sep. 2018.

[6] S. Y. Bhat and M. Abulaish, "HOCTracker: Tracking the evolution of hierarchical and overlapping communities in dynamic social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 4, pp. 1019–1031, Apr. 2015.

[7] S. Qiao, D. Shen, X. Wang, N. Han, and W. Zhu, "A self-adaptive parameter selection trajectory prediction approach via hidden Markov models," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 284–296, Feb. 2015.

[8] S. Qiao, N. Han, W. Zhu, and L. A. Gutierrez, "TraPlan: An effective three-in-one trajectory-prediction model in transportation networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1188–1198, Jun. 2015.

[9] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 70, no. 2, 2004, Art. no. 066111.

[10] M. E. J. Newman, *Networks: An Introduction*. Oxford, U.K.: Oxford Univ. Press, 2010.

[11] A. Clauset, "Finding local community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 72, no. 2, 2005, Art. no. 026132.

[12] F. Hao, G. Min, Z. Pei, D.-S. Park, and L. T. Yang, "K-clique community detection in social networks based on formal concept analysis," *IEEE Syst. J.*, vol. 11, no. 1, pp. 250–259, Mar. 2017.

[13] A. Mahmood, M. Small, S. A. Al-Maadeed, and N. Rajpoot, "Using geodesic space density gradients for network community detection," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 4, pp. 921–935, Apr. 2017.

[14] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, "Network community detection based on spectral clustering," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[15] M. G. Parsa, N. Mozayani, and A. Esmaeili, "An EDA-based community detection in complex networks," in *Proc. IEEE 7th Int. Symp. Telecommun.*, 2014, pp. 476–480.

[16] M. Bouguessa, R. Missaoui, and M. Talbi, "A novel approach for detecting community structure in networks," in *Proc. IEEE 26th Int. Conf. Tools Artif. Intell.*, 2014, pp. 469–477.

[17] V. Lyzinski, M. Tang, A. Athreya, Y. Park, and C. E. Priebe, "Community detection and classification in hierarchical stochastic blockmodels," *IEEE Trans. Netw. Sci. Eng.*, vol. 4, no. 1, pp. 13–26, Jan.–Mar. 2017.

[18] M. Takaffoli, R. Rabbany, and O. Zaiane, "Community evolution prediction in dynamic social networks," in *Proc. IEEE/ACM Int. Conf. Adv. Soc. Netw. Anal. Min.*, 2014, pp. 9–16.

[19] N. İlhan and Ş. G. Öguducu, "Predicting community evolution based on time series modeling," in *Proc. IEEE/ACM Int. Conf. Adv. Soc. Netw. Anal. Min.*, 2015, pp. 1509–1516.

[20] W. Zhu, D. Zhang, X. Zhou, D. Yang, and Z. Yu, "Discovering and profiling overlapping communities in location-based social networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 4, pp. 499–509, Apr. 2017.

[21] E. G. Tajeuna, M. Bouguessa, and S. Wang, "Tracking the evolution of community structures in time-evolving social networks," in *Proc. IEEE Int. Conf. Data Sci. Adv. Anal.*, 2015, pp. 1–10.

[22] T. Falkowski, A. Barth, and M. Spiliopoulou, "Studying community dynamics with an incremental graph mining algorithm," in *Proc. 14th Americas Conf. Inf. Syst.*, 2008, pp. 1–11.

[23] Y. Wang, B. Wu, and N. Du, "Community evolution of social network: Feature algorithm and model," *Phys. Soc.*, vol. 804, p. 4356, Apr. 2008.

[24] J. Zhang, Y. Zhu, and Z. Chen, "Evolutionary game dynamics of multiagent systems on multiple community networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.

[25] Y. Liu, H. Gao, X. Kang, Q. Liu, R. Wang, and Z. Qin, "Fast community discovery and its evolution tracking in time-evolving social networks," in *Proc. IEEE Int. Conf. Data Min. Workshop (ICDMW)*, 2015, pp. 13–22.

[26] P. Wang, J. Lü, and X. Yu, "Identification of important nodes in directed biological networks: A network motif approach," *PLoS ONE*, vol. 9, no. 8, 2014, Art. no. e106132.

[27] P. Wang, C. Yang, H. Chen, Q. Leng, S. Li, and D. Wang, "Exploring transcription factors reveals crucial members and regulatory networks involved in different abiotic stresses in Brassica Napus L," *BMC Plant Biol.*, vol. 18, p. 202, Sep. 2018.

[28] P. Wang, J. Lü, X. Yu, and Z. Liu, "Duplication and divergence effect on network motifs in undirected bio-molecular networks," *IEEE Trans. Biomed. Circuits Syst.*, vol. 9, no. 3, pp. 312–320, Jun. 2015.

[29] P. Wang, Y. Chen, J. Lu, Q. Wang, and X. Yu, "Graphical features of functional genes in human protein interaction network," *IEEE Trans. Biomed. Circuits Syst.*, vol. 10, no. 3, pp. 707–720, Jun. 2016.

[30] J. Rao, H. Du, X. Yan, and C. Liu, "Detecting overlapping community in social networks based on fuzzy membership degree," in *Proc. 5th Int. Conf. Comput. Soc. Netw.*, 2016, pp. 99–110.

[31] D. Greene, D. Doyle, and P. Cunningham, "Tracking the evolution of communities in dynamic social networks," in *Proc. IEEE Int. Conf. Adv. Soc. Netw. Anal. Min.*, 2011, pp. 176–183.

[32] Datatang. (2017). *DBLP Co-Author Data Sets*. Accessed: Apr. 27, 2017. [Online]. Available: http://www.datatang.com

[33] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, "On the evolution of user interactions in Facebook," in *Proc. ACM Workshop Online Soc. Netw.*, 2009, pp. 37–42.

Shaojie Qiao received the B.S. and Ph.D. degrees in computer science from Sichuan University, Chengdu, China, in 2004 and 2009, respectively.

From 2007 to 2008, he worked as a Visiting Scholar with the School of Computing, National University of Singapore, Singapore. He is currently a Professor with the School of Software Engineering, Chengdu University of Information Technology, Chengdu. He has authored more than 40 high quality papers, and coauthored more than 90 papers. His research interests include community discovery and artificial intelligence.

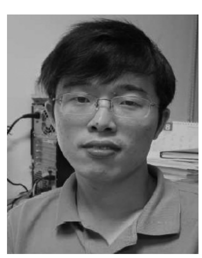
Nan Han received the M.S. and Ph.D. degrees in science of prescriptions from the Chengdu University of Traditional Chinese Medicine, Chengdu, China, in 2009 and 2012, respectively.

She is an Associate Professor with the School of Management, Chengdu University of Information Technology, Chengdu. She has authored of more than 20 papers and she has participated in several projects supported by the National Natural Science Foundation of China. Her research interests include complex networks and data mining.

Yunjun Gao (Member, IEEE) received the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2008.

He is currently a Professor with the College of Computer Science and Technology, Zhejiang University. His research interests include spatial and spatio-temporal databases and spatio-textual data processing.

Prof. Gao is a member of ACM and a Senior Member of CCF.



Rong-Hua Li received the Ph.D. degree in computer science from the Chinese University of Hong Kong, Hong Kong, in 2013.

He is currently an Associate Professor with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China, also with the Guangdong Province Key Laboratory of Popular High Performance Computers, Shenzhen University, Shenzhen, China, and also with the Guangdong Province Engineering Center of China-made High Performance Data Computing System, Shenzhen University. His research interests include social network analysis and graph data management.

Jianbin Huang received the Ph.D. degree in pattern recognition and intelligent systems from Xidian University, Xi'an, China, in 2007.

He is currently a Professor with the School of Computer Science and Technology, Xidian University. His research interests are in data mining and knowledge discovery.

Heli Sun received the Ph.D. degree in computer science from Xi'an Jiaotong University, Xi'an, China, in 2011.

She is currently an Associate Professor with the School of Computer Science and Technology, Xi'an Jiaotong University. She is a Visiting Scholar with the University of California at Los Angeles, Los Angeles, CA, USA, from 2017 to 2018. Her research interests are in graph mining, social network analysis, and smart city.

Xindong Wu (Fellow, IEEE) received the Ph.D. degree in artificial intelligence from the University of Edinburgh, Edinburgh, U.K., in 1993.

He is the President of Mininglamp Academy of Sciences, Mininglamp Technology, Beijing, China, and the Professor with the Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology, Ministry of Education, Hefei, China. His research interests include data mining and knowledge-based systems.

Prof. Wu is a fellow of the AAAS.