Detecting and Evolving Microblog Community Based on Structure and Gravity Cohesion

JINRONG HU1,2, YAJUN DU2, AND JIA LIU2
1Department of Computer Science, Chengdu University of Information Technology, Chengdu 610225, China
2School of Computer and Software Engineering, Xihua University, Chengdu 610039, China
Corresponding author: Yajun Du (duyajun@mail.xhu.edu.cn)

This work was supported in part by the National Nature Science Foundation under Grant 61872298 and Grant 61532009, and in part by the Sichuan Science and Technology Program under Grant 2019YFG0196.

ABSTRACT Microblogs are open and real-time online social network platforms used by people to make posts about their moods, experiences, and interests. It will be very significant to gather microblog users who have similar interests and hobbies into the same community. In this paper, we propose novel approaches for detecting and evolving dynamic microblog communities. First, inspired by the universal gravitation law, we redefinition the gravitation relationships among microblog users. Based on the structure of the microblog social network, we define the basic nodes and their gravity tendency and propose the microblog community detection algorithm. Second, we determine the community changes in the microblog social networks at times $t$ and $t+1$ and propose a microblog community evolution algorithm. Third, we define the mutual transformation probability between communities at times $t$ and $t+1$ and propose the microblog community evolution behavior algorithm. The experiment includes a comparison and evaluation of the microblog community detection, evolution, and behavior extraction algorithms and the optimal ranges of the parameters involved in these algorithms. The experimental results indicate that our proposed algorithms have good performance compared to other benchmarking methods.

INDEX TERMS Microblog, universal gravity, community detection, dynamic community evolution, evolution behavior.

I. INTRODUCTION

Social networks are one of the most popular applications of modern times. Famous social networking websites include Facebook, Twitter, MySpace, Microblog, WeChat, Baidu tieba, Zhihu, etc. A microblog is an open and real-time online social networking platform that is based on the platform of users and their relationships and allows people to post about daily events, their own moods and so on. The most popular microblog service platforms are Twitter and Sina microblogs. Twitter, the world’s most popular online social network platform, allows registered users to post “tweets” of no more than 140 characters. The number of active Twitter users has grown rapidly in the past year, with the number of active users jumping 11% in the first quarter of 2019. The most popular online social networking platform in China is Sina microblog, the daily active users of which reached 165 million, up 25% from a year ago. Thus far, microblogging, in which the kind of online social networking platform has become an indispensible part of people’s lives, has become a daily topic of discussion, and access to such information, such as current political reviews of important social platforms, is also a way for people to make friends.

In the era of big data, microblog users can generate an enormous amount of data every day. These data include microblog posts, thumbs up, retweets, and comments. Microblog features include originality, innovative interaction, immediacy and viral transmission [1].

Microblogs allow people to post about daily events, their moods and more. Among the large number of microblog users, some actually have similar interests and hobbies. Thus, it will be very significant to gather these microblog users who have similar interests and hobbies, such as advertising accurately and recommending friends, into the same community. Community refers to a group of nodes, and community detection is the detection of such a group with a special relationship. The prior approaches for community detection are analyzed from the community structure. However, the community is not static over time. Therefore, the study of community evolution is also indispensable. The existing methods...
of community evolution mainly study the explicit evolution of the community, such as “split”, “merge” and “shrink”. For the microblog social network, we can not only analyze the community from the community structure but also extract the microblog user interest features from the microblog content. For the evolution of microblog communities, some existing community evolution methods only analyze the explicit evolution behavior of the community. However, microblog communities still have implicit evolution behavior. Based on the existing shortcomings of community detection methods and community evolution methods, we will conduct our research, our main contributions of which are as follows:

- We redefine the microblog social network and the gravity relationship in the microblog social network. By using the random walk method combined with game theory, we mine the gravity tendency of microblog users.
- We propose the microblog community detection algorithm to find the community. We determine the community changes in microblog social networks at times \( t \) and \( t + 1 \) and propose a microblog community evolution algorithm.
- We define the interconversion probability between communities at times \( t \) and \( t + 1 \) and propose the community evolution behavior extraction algorithm for microblog social networks.
- The experimental results show that our proposed community detection and evolution algorithms outperform the CNM (Clauset-Newman–Moore) algorithm, COPA (community overlap propagation algorithm), Infomap (information map), NRW (nonhomogeneous random walk) in average gravity and modularity, and F-score.

The outline of the paper is as follows. Section II presents the related works. In Section III, we discuss the detection of the microblog dynamic community based on structure and gravitational cohesion. Section IV discusses evolution behavior extraction. The experiment and result analysis are presented in Section V. We provide conclusions and future works in Section VI.

II. RELATED WORKS

A microblog community is the common structure in a network; it shows that users are similar in one or more aspects. As an important structure in social networks, the community has attracted extensive attention.

A. COMMUNITY DETECTION

Kernighan and Lin [2] first proposed the graph segmentation method in 1970, a period in which the community concept was still unknown. Community detection has developed rapidly since the community was proposed [3]. In the following decades, researchers proposed some classic community detecting (CCD) algorithms [4].

1) MODULARITY-DEGREE-BASED CCD ALGORITHMS

The main idea of these algorithms is to find the community by using the optimized modularity \( Q \) value. We classify the algorithms into three classes: (1) top-down hierarchical clustering approaches: the CNM [5], FN (Fast-Newman) [6], and Msg-mv [7] are classic algorithms; (2) bottom-up hierarchical clustering approaches: these complex approaches split social networks into different communities, and the GN (Girvan-Newman) algorithm [3] is a typical representative; and (3) direct optimization approaches: the EO (Extremal Optimization) algorithm [8] is representative. Cao et al. [9] found vulnerable and fuzzy communities in undirected and unweighted networks by maximizing the weighted modularization. Newman [10] proved the exact equivalence relation between module degree optimization and the maximum likelihood algorithm.

2) SPECTRAL-ANALYSIS-BASED CCD ALGORITHMS

In this type of algorithm, the node characteristic component matrix contains the spatial coordinates. The network nodes are mapped to a multidimensional vector space and then gathered into the community by the traditional clustering algorithm. Chin et al. [11] proposed and analyzed a simple and robust random block model based on a spectral algorithm, which is mainly aimed at sparse graphs. Bordemave et al. [12] confirmed the “spectral redemption conjecture”. Community detection is carried out on the basis of the leading eigenvectors above the feasibility threshold. Feng et al. [13] proposed a local spectral method to find small community structures in large social networks.

3) INFORMATION-THEORY-BASED CCD ALGORITHMS

Rosvall and Bergstrom [14] used the simulated annealing optimization algorithm and random walk to effectively detect communities. Lancichinetti and Fortunato [15] tested to show that this method is the most accurate in the current nonoverlapping community discovery algorithm. Banks et al. [16] presented the upper and lower limits of the information theory threshold for community detection.

4) LABEL-PROPAGATION-BASED CCD ALGORITHMS

Raghavan et al. [17] proposed the label propagation algorithm (LPA). Each node updates its own labels to the highest number of occurrences of its neighbors. Nodes with the same label are grouped into a community. Based on random walks, Chang et al. [18] proposed the RWLPA algorithm to detect communities by using the location probability distribution of random walkers. Hu et al. [19] reported that various community-oriented nodes differently affect the spread of labels and proposed a role-based label propagation algorithm. Fang et al. [20] proposed structure balance to measure the edge of the balance in the local network and a new type of structural balance signature network label propagation algorithm.

From the community detection perspective, the above four methods only consider the community structure of the microblog network to extract user or user-group features. However, our research utilizes a large amount of microblog data to extract the interest characteristics of microblog users,
which enables us to conduct research on microblog content and community structure. We propose the microblog community discovery algorithm combining two aspects of user features of microblog content and community structure into the gravity relationship. Our proposed community detection algorithm is similar to the spectral-analysis-based, information-theory-based, label-propagation-based CCD algorithms in that it uses the bottom-up strategy to find the center point through the gravity relationship. However, we use the random walk method combined with game theory to find the relationships regarding gravity tendency among microblog users. The modularity-degree-based CCD algorithm adopts the top-down strategy.

B. COMMUNITY EVOLUTION
Community detection is used to format the law of the community in a static network. In fact, social networks are evolving continuously over time [21], [22]. Recently, a large number of evolving community approaches have been formed.

1) DYNAMIC CLUSTERING APPROACHES
Evolving clustering (EC) approaches for communities take the time stamp as the cluster sample unit and calculate the node distribution at times \( t \) and \( t + 1 \). Chi et al. [23] proposed the protection clustering quality (PCQ) and protection cluster member (PCM) algorithm based on EC. Lin et al. [24] proposed a FacetNet approach based on EC, which establishes a time stamp cost function based on community distribution at time \( t \). Kim and Han [25] took into account the fact that FacetNet needs to preset the number of communities and does not allow us to change its disadvantages. Hong et al. [26] divided the community at every moment based on the idea of timetables and overlapping communities.

2) OBJECTIVE FUNCTION OPTIMIZATION APPROACHES
These approaches estimate the structural changes in evolving communities by using community density functions or modular optimization. Blondel et al. [27] applied the FN algorithm [4] to detect dynamic communities. They combined regional optimization and hierarchical clustering. To detect dynamic communities, they designed four dynamic event (node\(^+\), node\(^-\), link\(^+\), and link\(^-\)) strategies to improve the community density function and modularization. Zhou et al. [28] proposed multiobjective optimization approaches for dynamic community detection, which modified new or changed nodes in the community during network updating. To overcome the modular optimization, Guo et al. [29] found small communities, increasing the sequence of highly sensitive issues such as networks. They proposed a new dynamic community discovery algorithm based on distance dynamics.

3) REPRESENTATIVE NODE/COMMUNITY DETECTION APPROACHES
Duan et al. [30] used the transfer probability matrix to establish the correlation measurement between nodes and proposed a compact model to assess the closeness of the regional community. Chen et al. [31] proposed the evolution tracking method based on the representative community to evaluate the detected communities and find stable communities. When the network structure changes, it only tracks the stable group. Liu et al. [32] proposed a fast incremental community evolution tracking (FICET) framework to find communities and track evolving communities. The core subgraph can quickly capture community evolution events, including formation, decomposition, and division. Hu et al. [33] found the dynamic community by exploring the local view of changing nodes.

4) DYNAMIC PROBABILISTIC MODEL APPROACHES
These approaches assume that the community distribution in each timestamp is a sample of the potential community distribution. Sarkar and Moore [34] constructed the probability of the relation between nodes and then used each timestamp in nodes and the probability of the relationship between community to build a Bayesian model between nodes and the potential community. Sun et al. [35] constructed a Bayesian model for each node in the timestamp sampling of DMM (Dirichlet mixture model) and in the adjacent timestamp between nodes, which can be found by the LDA (latent Dirichlet allocation) community. Yu et al. [36] proposed the ART model to adjust the community structure adaptively according to the changes in the semantic cohesion of the community.

In community evolution aspects, on the basis of the community found at the \( t \) time point, the change in the social network is found by using the microblog data of the \( t + 1 \) time point. According to the community and the change in microblog users and their relationship, the evolution algorithm of the microblog community is proposed to find the community of \( t + 1 \) time. We define the interconversion probability between the \( t \) time point and the \( t + 1 \) time point community and propose the microblog social evolution behavior extraction algorithm to extract the evolution behavior of the community in the microblog social network. Our proposed algorithm focuses on implicit and explicit evolution behaviors and identifies the implicit and explicit evolution community. However, the above dynamic clustering, objective function optimization, representative node/community detection, and dynamic probabilistic models focus on the explicit evolution community. They focus on designing different clustering algorithms and objective functions to perform community evolution from the change in community structure features. Our proposed algorithm focuses on discussing community evolution caused by changes in community evolution behaviors.

III. DETECTING A MICROBLOG DYNAMIC COMMUNITY BASED ON STRUCTURE AND GRAVITATIONAL COHESION
In microblog social networks, large numbers of microblog users, microblog posts, and the interactions between them are available. Since microblog social networks can change at any time [22], we use multiple static microblog social networks to define dynamic microblog social networks.
**Definition 1:** The static microblog social network $MSN^t$ (Eq. (1)) represents the status of the microblog social network at time $t$.

$$MSN^t = \langle U^t, E^t, C^t \rangle, \quad t = 1, 2, \cdots,$$  \hspace{1cm} (1)

where $U^t$, $E^t$, and $C^t$ are the microblog users, their interaction set between any two microblog users, and microblog communities at time $t$, respectively.

Because microblog social networks have characteristics such as timeliness and interactive innovation, they can occur at any time change, so we use multiple static dynamic Weibo microblog social networks to define the social network. A dynamic microblog social network is the representation of multiple static microblog social networks; namely, the time series is added on the basis of the static microblog social network.

**Definition 2:** The dynamic microblog social network $MSN$ (Eq. (2)) refers to the social network of microblog users, microblog user relationships and communities with time changes.

$$MSN = \langle MSN^0, \cdots, MSN^t, MSN^{t+1}, \cdots \rangle$$

$$\Delta MSN^{t\rightarrow t+1} = \langle \Delta U^{t\rightarrow t+1}, \Delta E^{t\rightarrow t+1}, \Delta C^{t\rightarrow t+1} \rangle,$$  \hspace{1cm} (2)

where $\Delta MSN^{t\rightarrow t+1}$ represents the content of changes in the microblog social network between times $t$ and $t+1$, including changes $\Delta U^{t\rightarrow t+1}, \Delta E^{t\rightarrow t+1}$ and $\Delta C^{t\rightarrow t+1}$ of microblog users and the relationships between microblog users and microblog communities.

Newman [3] pointed out that community refers to dense regions in the network and that the connections among users within each community are relatively close, but those between communities are relatively sparse. We use the gravity relationship between microblog users to measure the degree of connection between them.

**A. GRAVITY RELATIONSHIP OF THE MICROBLOG COMMUNITY**

In the universe, a large number of celestial bodies have formed some stable structures under the action of universal gravity $f$. Given two objects, $m_i$ and $m_j$, their distance is $g = 9.8$ is the gravitational constant, and the universal gravity $f$ of two objects can be computed by Eq. (3).

$$f_{ij} = g \frac{m_i \times m_j}{r^2}.$$  \hspace{1cm} (3)

The dynamic community evolving algorithm (GCEA) [37] reveals gravity relationships by considering only the degrees of each node in a dynamic social network. They only set $g$ to 1, and $m_i$ and $m_j$ is the degree of node $i$ and node $j$, respectively. The node degree is its intrinsic property and can reflect the local influence of the node. The greater the degree of the node is, the more other nodes it connects. This indicates that the node has some influence. This is similar to the gravity theory law of universal gravitation. An edge exists between two nodes, and the distance $r_{ij}$ between them is 0. Yin redefined the gravity between two objects as in Eq. (4).

$$f_{ij} = m_i \times m_j \times a_{ij}.$$  \hspace{1cm} (4)

When an edge exists between nodes $i$ and $j$, $a_{ij} = 1$; otherwise, $a_{ij} = 0$. In microblog social networks, the connection strengths between any two edges are different. Obviously, $a_{ij} = 1$ is not suitable for any two nodes. Inspired by Yin’s ideas, we reconsider gravity cohesion relationships and apply them to microblog social networks.

**B. RECONSTRUCTING THE GRAVITY RELATIONSHIP IN THE MICROBLOG SOCIAL NETWORK**

We build a vector of microblog users and use it to represent microblog users. At time $t$, microblog user $i$ represents $\overrightarrow{u_i}$ = $(u_{i1}, u_{i2}, \cdots)$. To construct the gravity relationship in the microblog social network, we redefined the distance $r_{ij}$ in the gravity law on two theoretical points of the complex network gravity relationship. The greater the degree of similarity between two objects is, the stronger the link between these two objects. The smaller the degree of similarity between two objects is, the weaker the link between these two objects. Considering microblog user interests, we improve the Tanimoto coefficient to compute the similarity between them by Eqs. (5), (6), and (7).

$$w_{ij} = \frac{\overrightarrow{u_i} \cdot \overrightarrow{u_j}}{|\overrightarrow{u_i}|^2 + |\overrightarrow{u_j}|^2 - \overrightarrow{u_i} \cdot \overrightarrow{u_j}},$$  \hspace{1cm} (5)

where $w_{ij}$ denotes the similarity between two microblog users at time $t$. $\overrightarrow{u_i}$ is the representation of microblog users on the Boolean dimension. $\overrightarrow{u_i} \cdot \overrightarrow{u_j}$ is the number of the same elements between $\overrightarrow{u_i}$ and $\overrightarrow{u_j}$.

$$\overrightarrow{u_i} = (u_{i1}, u_{i2}, \cdots, u_{ik}), \quad k = 1 \cdots \text{count}(\overrightarrow{u_i} \cup \overrightarrow{u_j}),$$  \hspace{1cm} (6)

where $\text{count}(\overrightarrow{u_i} \cup \overrightarrow{u_j})$ is the sum of the interest features of microblog user vectors $\overrightarrow{u_i}$ and $\overrightarrow{u_j}$. $u_{ik}$ is the element in the microblog user vector $\overrightarrow{u_i}$ represented by the Boolean dimension.

$$u_{ik} = \begin{cases} 1 & \text{if } r_k \in \overrightarrow{u_i} \\ 0 & \text{otherwise} \end{cases}.$$  \hspace{1cm} (7)

If the element $r_k$ in $\overrightarrow{u_i}$, $u_{ik}$ is 1, then the microblog user $u_i$ has the interest $u_{ik}$. Instead of 0, the microblog user $u_i$ has no interest $u_{ik}$.

**Example 1:** We illustrate how to calculate the similarity between two microblog users $u_i$ and $u_j$. Let $\overrightarrow{u_i} = (\text{Music, Movie, Food, Travel})$ and $\overrightarrow{u_j} = (\text{Movie, Travel, Automobile, Basketball})$. $\overrightarrow{u_i} \cup \overrightarrow{u_j} = \{\text{Music, Movie, Food, Travel, Automobile, Basketball}\}$.

$$\overrightarrow{u_i} = (1, 1, 1, 1, 0, 0), \quad \overrightarrow{u_j} = (1, 1, 0, 0, 1, 1).$$

Finally, we calculate the similarity $w_{ij} = 0.0667$ between the two microblog users by Eq. (5).
The greater the similarity between two microblog users is, the shorter the distance between them. We measure the relationship among similarity $w_{ij}$, $d_{ij}$ and distance $r_{ij}$ by Eq. (8).

$$r_{ij}^t = \begin{cases} \infty & \text{if } d_{ij}^t = 0, \\ \infty & \text{if } d_{ij}^t = 1 \text{ and } w_{ij}^t = 0, \\ \text{and } (w_{ij}^t)^{-1} & \text{otherwise}. \end{cases} \tag{8}$$

After obtaining the similarity $w_{ij}$ between two microblog users, we combine it with gravity theory of two points and then reconstruct the gravity relationship through the combination of community structure and interest characteristics.

$$f_{ij}^t = \begin{cases} 0 & \text{if } d_{ij}^t = 0 \text{ or } w_{ij}^t = 0, \\ \text{and } m_{ij}(w_{ij}^t)^2 & \text{otherwise}, \end{cases} \tag{9}$$

where $f_{ij}^t$ denotes the gravitational relationship between the user and the user at time $t$.

### C. CONSTRUCTING A MICROBLOG SOCIAL NETWORK GRAPH

After establishing the gravity relationship between any two microblog users, we can set up a social network graph of microblogs to accurately express the real social network. We use each microblog user as the node of the microblog social network graph. An edge exists between two microblog user nodes under the three following cases:

- Microblog user $i$ focuses on microblog user $j$;
- Microblog user $j$ focuses on microblog user $i$; and
- Microblog users $i$ and $j$ pay attention to each other.

According to the gravity relationships between any two microblog users, we regard the gravity relationships as gravity weights of edges. Compared with the microblog social network graphs at time $t$, the nodes and edges of the microblog social network graphs at time $t + 1$ are represented by $\Delta U^{t+1}$ and $\Delta E^{t+1}$, respectively.

### D. GRAVITY TENDENCY

We use $N_{ij}^t$ to represent the neighbor collection of microblog user $u_i^t$ at time $t$. Fig. 1 shows that the social network graph consists of two communities $C_1^t$ and $C_2^t$, and microblog user nodes $u_i^t$, $u_j^t$ and $u_k^t$ are the boundaries of the community. We assume that user node $u_i^t$ is a random walk of the initial node, and the random walk user $u_i^t$ can choose three sides $e_{i\rightarrow 4}$, $e_{i\rightarrow 8}$ and $e_{i\rightarrow 9}$ to travel. Because nodes $u_i^t$ and $u_j^t$ belong to the same community $C_2^t$, $u_i^t$ will have a greater probability of choosing edge $e_{i\rightarrow 9}$. Because microblog users in the same community have more similarities (namely, the gravity relationship between them), the probability of $e_{i\rightarrow 9}$ being wandered is greater than $e_{i\rightarrow 4}$ and $e_{i\rightarrow 8}$. When the walking user wanders in the community $C_1^t$, it will have a very small probability of getting out of the community $C_2^t$ into the community $C_1^t$. In contrast, when walking users travel to community $C_2^t$ by edge $e_{i\rightarrow 4}$ or edge $e_{i\rightarrow 8}$, it will have a very small probability of moving to community $C_2^t$. Moving from node $u_i^t$ to node $u_j^t$ by edges $e_{i\rightarrow 4}$, $e_{i\rightarrow 5}$, $e_{i\rightarrow 8}$ and $e_{i\rightarrow 1}$ does not represent a strong gravity relationship between microblog users $u_i^t$ and $u_j^t$.

**FIGURE 1. Microblog network structure based on gravity.**

**Definition 3:** Assume that $u_i^t$ is the initial node of the walking user. In the $N$ random walk, the walking user has $n$ times return to node $u_i^t$ and $n'$ times return to node $u_j^t$ by side $e_{i\rightarrow 4}$. The returning probability of node $u_i^t$ is $n/N$, and the returning probability of edge $e_{i\rightarrow 4}$ is $n'/N$.

To determine the random walk number, Yin et al. [37] used a probability function to calculate the random walk step number. They set up the following random walk: $a \in (0, 1)$, $b = f(x)$, $x$ is the current number of the walking user, and $f(x)$ is a probability function used to determine whether to move on. At the current node, if $a < b$, then the walking user continues; otherwise, it will terminate. The random walk probability function $f(x)$ is shown below:

$$f(x) = \frac{1}{\exp((x - \frac{1}{\eta d^{m+1}})^{\beta} - \beta) + 1}, \tag{10}$$

where parameters $\delta$ and $\beta$ control the migration distribution and the farthest node distribution of the walking user, respectively. According to the principle of “six degrees of separation”, the diameter of a community is not greater than 6. Thus, $\beta$ is set to 6. $x - \frac{1}{\eta d^{m+1}}$ is a structural gain function based on the logic function, which takes the degree of node as the inputting parameter of the probability function $f(x)$. $d$ is the degree of the node. $\omega$ and $\eta$ are adjustment parameters of logical functions. The physical meaning of Eq. (10) is that when the walking user moves to a node with a high degree, the topological gain function assigns the greater probability of the walking user to travel.

In the random walk process, we propose the weighted random walk benefit maximization algorithm to measure the importance of other nodes to the initial node and extract the gravity relationship tendency of microblog user nodes. The algorithm can grasp the nodes that are more useful for the initial nodes. Even though the initial nodes and their neighboring nodes do not have a strong gravity relationship, the initial nodes may obtain more useful nodes through their neighboring nodes. As shown in Fig. 1, compared with $u_i^t$ and $u_j^t$, the earnings of $u_i^t$ choosing node $u_k^t$ are lower than those of $u_i^t$ choosing node $u_j^t$. Thus, node $u_k^t$ may choose $u_j^t$ because $u_k^t$ can find nodes $u_i^t$ and $u_j^t$ with a larger probability.
in the same community. In our proposed algorithm, when the walking user walks a step, we use the gravity value between two nodes as the profit for this travel and all the nodes’ average normalized negative gravity as the loss of value. The weighted random walk benefit maximization is presented in Eq. (11).

\[
S_{\nabla k}^f = \varphi(\Sigma f_{ij} - \beta' \times f_{ij}^{|loss|}) + (1 - \varphi)(1 - \frac{\text{Path}_\min(u_0', u_{ter}')}{\beta}),
\]

(11)

where \(S_{\nabla k}^f\) represents the actual maximum benefit of a neighbor node \(u_k'\) of \(u_0\). \(\beta'\) represents the actual number of walking steps, and \(\beta' \leq \beta, f_{ij}' = 1 - \frac{\sum f_{ij}}{2n}\) is the loss of each random walk, which means the average normalized negative gravity of all nodes. Nodes \(u_0'\) and \(u_{ter}'\) represent the initial node and terminal node of the walking user, respectively. Due to the exclusion of remote nodes in the microblog social network, we use \(1 - \frac{\text{Path}_\min(u_0', u_{ter}')}{\beta}\) to express this feature. This means that the shortest distance between \(u_0'\) and \(u_{ter}'\) is very large, and the probability that walking users return to \(u_0'\) is very small. Parameter \(\varphi\) is a weight value of the characteristics of the microblog user’s gravity and social network structure. Because they are equally important in our proposed approach, the optimal value of \(\varphi\) is 0.5, which will be proven in the experiment section.

We will find the gravity tendency of \(u_0'\) after the walking user has \(N\) random walk steps passing by \(u_0'\). First, we find \(u_0'\) with the maximum average benefit as the first gravity tendency of the initial node \(u_0'\) by Eq. (12).

\[
R(u_0') = \{u_0', u_k' | u_k' = \text{argmax}(\Sigma u_k') \land d_{0k} = 1\},
\]

(12)

where \(R(u_0')\) denotes the set of gravity tendency nodes of the initial node \(u_0'\). \(\Sigma u_k'\) indicates that the walking user has \(n(n \leq N)\) random walk times to pass through node \(u_k'\) and gain the average maximum benefit. Thus, we use the node with the maximum average benefit as the first gravity tendency of \(u_0'\). Of course, if the number of nodes with the maximum average benefit is greater than 1, then these nodes are the gravity tendency of the initial node \(u_0'\). If the average benefit of a node is closer to the maximum average benefit, then this node is the gravity tendency of the initial node \(u_0'\). The gravity tendency node set of \(u_0'\) can be updated by Eq. (13).

\[
R(u_0') = \{u_0', u_k' | u_k' = \text{argmax}(\Sigma u_k'), \ (\Sigma u_k' - \Sigma u_k' < \varepsilon, \cdots)\},
\]

(13)

where \(\varepsilon\) is a threshold. This indicates that the nodes with similar average benefits with node \(u_k'\) have a gravity tendency of \(u_0'\). Algorithm 1 shows the whole process of the weighted random walk benefit maximization algorithm. Algorithm 2 shows the process of the random walk algorithm. \(--\) and \(\rightarrow\) in the algorithms represent the assignment and the value added, respectively.

In algorithm 1, steps 10-12 perform \(n\) random walks on the initial node \(u_0'\). Step 13 finds all neighbor nodes of node \(u_0'\).

| Algorithm 1: Weighted Random Walk Benefit Maximization |
|--------------------------------------------------------|
| 01 Input                                              |
| 02 MSN\({}^t\); // Social networks at \(t\) time       |
| 03 \(\eta, \omega\);                                   |
| 04 \(\delta\); // The distribution of random walk steps |
| 05 \(\varphi\);                                       |
| 06 Output                                             |
| 07 FR_{\delta}'; // Gravitational tendencies for each node in MSN\({}^t\) |
| 08 For each \(u_{0}'\) in MSN\({}^t\) do               |
| 09 FR_{\delta}'; \rightarrow \varphi;                 |
| 10 while \(n > 0\) do                                  |
| 11 \(\text{walkRes} \leftarrow \text{RandomWalk}(u_t', \varphi, \eta, \omega, \delta)\) |
| 12 End While;                                         |
| 13 Neiborhoods \leftarrow \text{findNeiborhoods}(u_t'); |
| 14 NeiAveEar \leftarrow \text{Average} (\Sigma u_{0}') of the neighbor nodes of \(u_t'\); |
| 15 FR_{\delta}'; \rightarrow u_t' / \| maximum gravitational tendency by Eq. (12) |
| 16 For each \(u_{0}'\) in Neiborhoods and \(u_t' \neq u_{0}'\) do; |
| 17 If \(\Sigma u_{0}' - \Sigma u_{0}' < \varepsilon\) by Eq. (13) then; |
| 18 FR_{\delta}'; \rightarrow u_{0}' ;                 |
| 19 End If                                             |
| 20 End For                                            |
| 21 FR_{\delta}'; \rightarrow FR_{\delta}';             |
| 22 End For                                            |
| 23 Return FR_{\delta}';                               |
| 24 End                                                |

Step 14 calculates the average score \(\Sigma u_{0}'\) of the neighboring nodes of \(u_t'\). Step 15 finds the node \(u_{0}'\) with the highest gravity tendency by Eq. (12). Steps 16-20 determine the more gravity tendency node \(u_{0}'\) by Eq. (13). If there is a node, \(u_{0}'\), of which the average benefit and maximum average benefit are very close, then we put the node \(u_{0}'\) into the gravity tendency set of \(u_0'\). Step 21 stores the gravity tendency nodes of \(u_0'\) to FR_{\delta}'.

In algorithm 2, steps 9-13 are random walks. Step 9 uses Eq. (10) to decide whether to continue the next step. If it needs to continue to walk, then step 10 makes a step walk. Step 11 obtains the neighbor nodes of the first walk. Step 12 calculates the current walk score by Eq. (11). Step 14 calculates the shortest distance between the initial node \(u_0'\) and the destination node \(u_{ter}'\). Step 15 calculates the final score of the walk. Step 14 saves gravity tendency.

E. COMMUNITY DETECTION

In physics, a gravity attraction exists between two planets, and a planet with a massive mass is likely to attract a smaller planet. In a social network, a node has a large degree, which proves that it has an edge connection with many other nodes. Thus, these nodes are very important for the social network. Finding these important nodes is a key part of our proposed microblog community detection algorithm, so we identify these nodes as the base nodes.
Algorithm 2 Random Walk

01 Input
02 $u'_i$; //Microblog user nodes at $t$ time;
03 $\eta, \omega$;
04 $\delta$/$\text{The distribution of random walk steps}$
05 $\varphi$;
06 Output
07 walkRes; //Corresponding neighbor nodes and their scores
08 $x \leftarrow 0$; $d(x)$ is the current walking number
09 while $f(x) > a$ and $x \leq 6$ do//Eq. 10
10 user walk a step;
11 Get the neighbor nodes for the first walk $u'_j$;
12 $x \leftarrow x + 1$;
13 End While;
14 Calculate $Path_{min}(\overrightarrow{u'_i}, \overrightarrow{u'_{f_{er}}})$;
15 calculate $ear_{final}$ by Eq. (11);
16 walkRes $\leftarrow (u'_j, ear_{final})$;
17 Return walkRes;
18 End.

Definition 4: Let $d(u'_i)$ represent the degree of node $u'_i$; if $d(u'_i) > d(u'_j) \land u'_i \in N(u'_j)$, $u'_i$ becomes the base node.

After obtaining the basic nodes in the microblog social network, we detect the initial community by using the gravity tendency. We start with a basic node as an initial community. If more basic nodes connect with each other, then we consider them as a basic node chain of an initial community. Then, we incorporate these nodes into the corresponding initial community according to the gravity tendency of the nodes. The gravity tendency directions of two nodes have three situations:

- Nonbasic node $u'_i$ and basic node $u'_j$ have gravity tendencies. We add node $u'_j$ to the community in which the base node $u'_i$ is located.
- Nonbasic node $u'_j$ does not tend to the basic node $u'_i$, but basic node $u'_j$ tends to nonbasic node $u'_i$. We add $u'_j$ to the community in which $u'_i$ is located. Because $u'_i$ tends to $u'_j$, it means that the nonbasic node $u'_j$ has great significance for the basic node $u'_i$.
- Nonbasic node $u'_j$ tends to basic node $u'_i$, but the basic node $u'_j$ does not tend to nonbasic node $u'_i$. We discuss in detail how it is handled as follows.

For case (3) above, we use the gravity relationship between the community and the user node to allocate the remaining nodes into communities. If node $u'_i$ is only attracted by community $C'_k$, then $u'_i$ is put into community $C'_k$. If node $u'_i$ is attracted to multiple communities, then $u'_i$ is placed in the community with the largest attraction to $u'_i$. We measure the gravity tendency relationship between node $u'_i$ and community $C'_k$ by calculating the average gravity of each node in community $C'_k$ by Eq. (14).

$$f(u'_i, C'_k) = \frac{\sum M'_i M'_j (w'_i) \delta}{\sum \delta i (j)}, \quad u'_i \in C'_k \land u'_j \in C'_k,$$  \hspace{1cm} (14)

where $f(u'_i, C'_k)$ represents the gravity relationship between node $u'_i$ and community $C'_k$. As walking through the remaining nodes, we obtain their communities. Microblog community detection is shown in algorithm 3.

Algorithm 3 Microblog Community Detection

01 Input
02 $MSN'^t$; //Social networks at time $t$
03 Output
04 $C'_t = \{C'_1, \ldots, \}$ //Community collection at time $t$
05 Find the basic node set $Bnodes$;
06 For each $u'_i \in bnodes$ do
07 If $u'_i, u'_j \in bnodes$ and $Tendency(u'_i, u'_j)$ or $u'_k, u'_j$ then
08 Build community $C'_k$, and put $u'_i, u'_j$ into $C'_k$;
09 Else;
10 Build community $C'_k$, and put $u'_j$ into $C'_k$;
11 End If;
12 End For;
13 For each $u'_i$ not in $bnodes$ do
14 If $u'_i$ not in $bnodes$ of $u'_j$, then
15 put $u'_j$ into $C'_k$ which $u'_j$ belong to;
16 Else;
17 Put $u'_j$ into $C'_k$ which $f(u'_i, C'_k)$ is largest;
18 End If;
19 End For;
20 End.

In algorithm 3, step 5 finds the base node set $Bnodes$. Steps 6-12 find communities for basic nodes in the $Bnodes$. If more than two basic nodes connect each other, then steps 7-8 build a new community $C'_k$ for them. A basic node $u'_i$ does not connect to other basic nodes; steps 7-8 create a community $C'_k$ for $u'_i$. Steps 13-15 identify nodes that are not included in $Bnodes$. Step 14 uses Eq. (14) to calculate $f(u'_i, C'_k)$ for these nodes $u'_i$'s and to determine their communities with their largest gravity.

F. COMMUNITY EVOLUTION

On the basis of detecting the community, we develop the community $\Delta C^t \rightarrow^t C^{t+1}$ evolution algorithm to determine the structural changes $\Delta U^t \rightarrow^t U^{t+1}$ and $\Delta E^t \rightarrow^t E^{t+1}$ of $MSN^t \rightarrow MSN^{t+1}$.

Two microblog social networks at adjacent times $t$ and $t + 1$ undergo some changes, such as user nodes increasing (nodes$^+$), user node reducing (nodes$^-$), edge increasing (edges$^+$) and edge reducing (edges$^-$). We analyze the community evolution by using these changes.

To analyze the social network community at time $t + 1$, we find $\Delta MSN^t \rightarrow^t MSN^{t+1} = < \Delta U^t \rightarrow^t U^{t+1}, \Delta E^t \rightarrow^t E^{t+1} >$ by combining $MSN^t$ with $MSN^{t+1}$. Then, we update the gravity relationship of the changed nodes $\Delta U^t \rightarrow^t U^{t+1}$. Finally, we analyze these four cases nodes$^+$, nodes$^-$, edges$^+$, and edges$^-$ of community evolution.

- **Microblog user nodes increasing (nodes$^+$).** We first judge whether a new microblog user node $u^{t+1}$ affects the base nodes of $MSN^t$. If the base nodes do not change,
then we place node $u_{i}'^{t+1}$ into community $C_{j}'^{t+1}$, where the gravity relationship $f(u_{i}'^{t+1}, C_{j}'^{t+1})$ is the largest. If the base node changes, then we can find the community of the new microblog user node by algorithm 3.

- **Microblog user node reducing**($nodes^-$). If the deleted node $u_{i}'^{t+1}$ is a base node, then we reassign the affected nodes into the community $C_{j}'^{t+1}$, which has the largest average gravity of these affected nodes. If the deleted node does not belong to the base node, then the affected node is redistributed directly to the community in which the average gravity of these affected nodes is the largest. Otherwise, we use algorithm 3 to find communities of the affected nodes.

- **Microblog user edge increasing**($edges^+$). We judge whether the new edge causes changes in the base nodes. If the base nodes do not change, then the nodes affected by the increasing edge are put into the community in which the average gravity of these affected nodes is the largest. Otherwise, we use algorithm 3 to find communities of the affected nodes.

- **Microblog user edge reducing**($edges^-$). We judge whether the reduced edge causes changes in the base nodes. If the base nodes do not change, then the nodes affected by the reducing edges are placed in the community in which the average gravity of these affected nodes is the largest. Otherwise, we use algorithm 3 to find communities of the affected nodes.

In algorithm 4, $\Delta U^{t\rightarrow t+1}$ represents the changed nodes in $MSN^t$ and $MSN^{t+1}$. Step 7 updates the gravity relationship of the nodes in $\Delta U^{t\rightarrow t+1}$. Steps 8-14 analyze community evolution in the case of node reduction. If the base nodes are reduced, then steps 9-10 update $Bnodes$, and these nodes in $\Delta U^{t\rightarrow t+1}$ are assigned to the communities with the largest gravity. If the base nodes do not change, steps 11-12 assign these nodes in $\Delta U^{t\rightarrow t+1}$ to the community with the largest gravity. Steps 16-24 process community evolution in the case of $nodes^+$ or $edges^+$ or $edges^-$. Step 17 updates the base node set $Bnodes$. If the basic node set $Bnodes$ changes, then steps 18-19 find the community for these nodes in $\Delta U^{t\rightarrow t+1}$ by calling algorithm 3. If the base node set does not change, then steps 20-21 assign these nodes in $\Delta U^{t\rightarrow t+1}$ to the community with the largest gravity.

### Algorithm 4 Microblog Community Evolution

01 Input
02 $MSN^t$ //Social networks at time $t$
03 $MSN^{t+1}$ //Social networks at time $t+1$
04 $C^t = \{C^t_1, \ldots, C^t_i, \ldots\}$ //Community collection at time $t$
05 Output
06 $C^{t+1} = \{C^{t+1}_1, \ldots, C^{t+1}_j, \ldots\}$
07 Update gravity of nodes in $\Delta U^{t\rightarrow t+1}$;
08 If $nodes^-$
09 If $nodes^- \in Bnodes$ then
10 Update $Bnodes$
11 Assign nodes to most gravity community;
12 Else;
13 Assign nodes in $\Delta U^{t\rightarrow t+1}$ to most gravity community;
14 End If;
15 End If;
16 If $nodes^+ or edges^+ or edges^-$ then;
17 Update the base node set $Bnodes$
18 If $Bnodes$ change then
19 Call Algorithm 3;
20 Else
21 Assign nodes in $\Delta U^{t\rightarrow t+1}$ to most gravity community;
22 End If.
23 End If.
24 End.

- **Splitting** A community $C^t_i$ at time $t$ is split into two or more communities at time $t+1$. The nodes of community $C^{t+1}_i$ at time $t+1$ come from community $C^t_i$ at time $t$.

- **Merging** By combining two or more communities at time $t$ to form a new community $C^{t+1}_i$ at time $t+1$.

- **Dissolving** A community $C^t_i$ at time $t$ vanishes and does not appear at the $t+1$ time point.

- **Forming** At time $t+1$, $C^{t+1}_i$ is a novel community that does not exist at $t$ time.

Based on these evolutionary behaviors, we propose the implicit evolutionary behaviors of the social network communities. In explicit evolution behaviors, such as “continuing”, although nodes and sides of the communities at times $t$ and $t+1$ do not change, the gravity relationship between the nodes in the different communities may change. We mainly analyze implicit evolution behaviors of “continuing”, “narrowing”, and “increasing”. We do not discuss the other four kinds of implicit evolution behaviors because the gravity relationship of its corresponding community also changes greatly. Therefore, we present Eq. (15) to express gravity changes between $C^t_i$ and $C^{t+1}_i$.

$$f(C^t_i) = \Sigma M^t_i M^t_i (w_{jk})^2, u_j \in C^t_i \land u_j \in C^t_i \land j \neq k. \quad (15)$$
where \( f(C_i^t) \) represents gravity in community \( C_i^t \).

\[
f(C_i^t, C_j^{t+1}) = \frac{|f(C_i^t) - f(C_j^{t+1})|}{\Sigma d_{mn}^t - \Sigma d_{xy}^{t+1}},
\]

(16)

where \( f(C_i^t, C_j^{t+1}) \) represents the average gravity change from \( C_i^t \) to \( C_j^{t+1} \). \( u_m^t \) and \( u_n^t \) are nodes in \( C_i^t \), and \( u_j^{t+1} \) and \( u_n^{t+1} \) are nodes in \( C_j^{t+1} \). \( \Sigma d_{mn}^t \) and \( \Sigma d_{xy}^{t+1} \) express the total number of edges of community \( C_i^t \) and \( C_j^{t+1} \) at times \( t \) and \( t + 1 \), respectively.

For convenience of discussion, we denote \( \Delta(x^t, x^{t+1}) \) to change the function of \( x \) from \( t \) to \( t + 1 \).

**Definition 5**: Social network community evolution can be described as community transformation from time \( t \) to \( t + 1 \). That is, if \( \Delta(U^t, U^{t+1}) = 0 \lor \Delta(E^t, E^{t+1}) > 0 \lor \Delta(f(C^t, f(C^{t+1}))) > 0 \), then we consider the community to be evolving.

### A. COMMUNITY RELATIONSHIP GRAPH

To extract the evolutionary behavior of social network communities, we defined a community relationship graph at two adjacent times \( t \) and \( t + 1 \). Social network changes are represented through the relationship between community and users.

Fig. 2 clearly indicates the changes from time \( t \) to \( t + 1 \) and the relationship between the communities of the adjacent times \( t \) and \( t + 1 \), respectively. The left part and right part shows the social network communities at time \( t \) and \( t + 1 \), respectively. The middle part represents nodes changed from communities at time \( t \) to communities at time \( t + 1 \). If two communities share the same microblog user node, then an edge exists between the two communities.

![FIGURE 2. Relationship between C_i^t to C_j^{t+1}](image)

**FIGURE 2. Relationship between C_i^t to C_j^{t+1}**

In Fig. 3, an edge connects community \( C_i^t \) and community \( C_j^{t+1} \). In the same way, two edges connect communities \( C_i^2, C_j^2 \) and \( C_i^3, C_j^3 \). The weights on these edges represent the transformation probability between two communities at two times \( t \) and \( t + 1 \). We define \( P(C_i^t \rightarrow C_j^{t+1}), P(C_i^{t+1} \rightarrow C_j^t) \) as the transformation probability of communities from \( C_i^t \) to \( C_j^{t+1} \) and from communities from \( C_i^{t+1} \) to \( C_j^t \).

At time \( t \), microblog user nodes \( u_1^t \), \( u_2^t \), and \( u_3^t \) belong to community \( C_i^t \). Whereas at time \( t + 1 \), the microblog user nodes \( u_2^{t+1} \) and \( u_3^{t+1} \) belong to \( C_j^{t+1} \). The mutual transformation probability \( P(C_i^t \rightarrow C_j^{t+1}) \) is 2/3. The microblog user nodes \( u_2^t \) and \( u_3^t \) in \( C_j^{t+1} \) are both in \( C_i^t \). The mutual transformation probability \( P(C_i^{t+1} \rightarrow C_j^t) \) is 1.

![FIGURE 3. Mutual transformation probability of C_i^t and C_j^{t+1}](image)

**FIGURE 3. Mutual transformation probability of C_i^t and C_j^{t+1}**

### B. COMMUNITY EVOLUTION BEHAVIOR EXTRACTION

Table 1 shows the types, conditions, reasons, and possible factors for different community evolution behaviors. If communities \( C_i^t \) and \( C_j^{t+1} \) satisfy the corresponding conditions, then the corresponding evolution behavior occurs. We present the reason numbers and concrete reasons in Table 2. \( \kappa \) and \( \lambda \) are the thresholds for mutual transforming probability and gravity changes. We redefine the community evolution behaviors in social networks as follows:

- **1) Explicit continuing**. This evolution behavior satisfies the mutual transforming probability \( \min(P(C_i^t \rightarrow C_j^{t+1}), P(C_j^{t+1} \rightarrow C_i^t)) \leq \kappa \). The possible reasons for the “explicit continuing” are that the microblog user nodes and their concerning edges in the community \( C_i^t \) remain consistent with community \( C_j^{t+1} \).

- **2) Implicit continuing**. Based on explicit continuing evolution behavior, the gravity changes of the microblog user nodes have changed considerably. The implicit continuing evolution behaviors satisfy \( \min(P(c_i^t \rightarrow c_j^{t+1}), P(c_j^{t+1} \rightarrow c_i^t)) \leq \kappa \) and \( \Delta(f(C^t, f(C^{t+1}))) > \lambda \). That is, the mutual transforming probability between the two adjacent communities \( C_i^t \) and \( C_j^{t+1} \) is greater than \( \kappa \). At the same time, their gravity change is greater than \( \lambda \).

- **3) Explicit narrowing**. In this case, a certain community has lost some microblog user nodes. The evolution behaviors satisfy \( 1 \geq P(C_i^{t+1} \rightarrow C_i^t) > P(C_j^{t+1} \rightarrow C_j^t) \) \( \leq \kappa \). The mutual transforming probability that \( C_i^{t+1} \) transforms into \( C_i^t \) is greater than the mutual transforming probability that \( C_j^{t+1} \) transforms into \( C_j^t \). The reasons for the evolution behavior are “deleting microblog user nodes” or their concerning edges “edges”.

- **4) Implicit narrowing**. Based on the explicit narrowing evolution behavior, the gravity changes in microblog user nodes in the community are more than \( \lambda \).
TABLE 1. Community evolution behaviors.

| Evolution behavior | Type of evolution | Condition of evolution | Reason numbers | Possible factors of evolution |
|-------------------|-------------------|------------------------|---------------|-----------------------------|
| Continuing        | Explicit          | \(\min\{P(c_i^t \rightarrow c_{i+1}^t), P(c_{i+1}^t \rightarrow c_i^t)\} \geq \kappa\) | R1            | nodes \(^+\); edges \(^+\) |
|                   | Implicit          | \(\min\{P(c_i^t \rightarrow c_{i+1}^t), P(c_{i+1}^t \rightarrow c_i^t)\} \geq \kappa\) | R2            | nodes \(^-\); edges \(^-\) |
| Shrinking         | Explicit          | \(1 \geq P(c_i^t \rightarrow c_{i+1}^t) > P(c_i^t \rightarrow c_{i+1}^t) \geq \kappa\) | R3            | nodes \(^-\); edges \(^-\) |
|                   | Implicit          | \(1 \geq P(c_i^t \rightarrow c_{i+1}^t) > P(c_i^t \rightarrow c_{i+1}^t) \geq \kappa\) | R4            | nodes \(^+\); edges \(^+\) |
| Growing           | Explicit          | \(1 \geq P(c_i^t \rightarrow c_{i+1}^t) > P(c_i^t \rightarrow c_{i+1}^t) \geq \kappa\) | R5            | nodes \(^+\); edges \(^-\) |
|                   | Implicit          | \(1 \geq P(c_i^t \rightarrow c_{i+1}^t) > P(c_i^t \rightarrow c_{i+1}^t) \geq \kappa\) | R6            | nodes \(^+\); edges \(^-\) |
| Splitting         | Explicit/implicit | \(\exists c_i^t, c_{i+1}^t, c_{i+2}^t, \ldots \text{satisfy}\) | R7            | nodes \(^-\); nodes \(^+\); edges \(^-\); edges \(^+\) |
|                   |                   | \(P(c_i^t \rightarrow c_{i+1}^t) \geq \kappa \wedge P(c_i^t \rightarrow c_{i+1}^t) \geq \kappa \wedge \cdots\) | R8            | nodes \(^-\); nodes \(^-\); edges \(^+\); edges \(^+\) |
| Merging           | Explicit/implicit | \(\exists c_i^t, c_{i+1}^t, c_{i+2}^t, \ldots \text{satisfy}\) | R9            | nodes \(^-\); edges \(^-\) |
|                   |                   | \(P(c_i^t \rightarrow c_{i+1}^t) \geq \kappa \wedge P(c_i^t \rightarrow c_{i+1}^t) \geq \kappa \wedge \cdots\) | R10           | nodes \(^-\); edges \(^-\) |

Evolution behaviors satisfy \(1 \geq P(C_{i+1}^t \rightarrow C_i^t) > P(C_i^t \rightarrow C_{i+1}^t) \leq \kappa\), and \(\Delta f(C_i^t, f(C_{i+1}^t)) > \lambda\). The mutual transforming probability that \(C_{i+1}^t\) transforms into \(C_i^t\) is greater than the mutual transforming probability that \(C_i^t\) transforms into \(C_{i+1}^t\). The reasons for the explicit growing evolutionary behavior are “increasing microblog user nodes (nodes \(^+\)) or their concerning edges (edges \(^+\))”.

(5) Explicit growing. There are some new microblog user nodes to add a community. The evolution behaviors satisfy \(1 \geq P(C_i^t \rightarrow C_{i+1}^t) > P(C_i^t \rightarrow C_{i+1}^t) \leq \kappa\). The mutual transforming probability that \(C_i^t\) transforms into \(C_{i+1}^t\) is greater than the mutual transforming probability that \(C_{i+1}^t\) transforms into \(C_i^t\). The reasons for explicitly growing evolutionary behavior are “increasing microblog user nodes (nodes \(^+\)) or their concerning edges (edges \(^+\))”.

(6) Implicit growing. Based on the explicit growing evolution behavior, the gravity changes of microblog user nodes in the community are more than \(\lambda\). The evolution behaviors satisfy \(1 \geq P(C_i^t \rightarrow C_{i+1}^t) > P(C_i^t \rightarrow C_{i+1}^t) \leq \kappa\), and \(\Delta f(C_i^t, f(C_{i+1}^t)) > \lambda\). The mutual transforming probability that \(C_i^t\) transforms into \(C_{i+1}^t\) is greater than the mutual transforming probability that \(C_{i+1}^t\) transforms into \(C_i^t\). The reasons for the implicit growing evolution behavior are “increasing microblog user nodes (nodes \(^+\)) or their concerning edges (edges \(^+\))”.

(7) Explicit/implicit splitting. In this case, a community at time \(t\) is split into many communities at time \(t+1\), and the microblog user nodes in these communities at time \(t+1\) come from the previous community at time \(t\). The evolution behaviors satisfy \(P(C_{i+1}^t \rightarrow C_i^t) \geq \kappa \wedge P(C_{j+1}^t \rightarrow C_j^t) \geq \kappa \wedge \cdots\) and \(P(C_j^t) \rightarrow C_{i+1}^t \cup C_{j+1}^t \cup \cdots \geq \kappa\). The transforming probability that \(C_{i+1}^t\) transforms into \(C_i^t\) is greater than \(\kappa\). The transforming probability that \(C_i^t\) transforms into many communities \(C_{j+1}^t, C_{j+2}^t, \ldots\) is more than \(\kappa\). The reasons for this evolution behavior are “increasing, reducing the microblog user nodes (nodes \(^+\)/nodes \(^-\)), or their concerning edges (edges \(^+\)/edges \(^-\))”.

(8) Explicit/implicit merging. Many communities \(C_{i_1}^t, C_{i_2}^t, \ldots\) merge into a community \(C_{i+1}^t\), and the microblog user nodes of \(C_{i+1}^t\) come from these communities \(C_{i_1}^t, C_{i_2}^t, \ldots\). The evolution behaviors satisfy \(P(C_{i_1}^t \rightarrow C_{i+1}^t) \geq \kappa \wedge P(C_{i_2}^t \rightarrow C_{i+1}^t) \geq \kappa \wedge \cdots\) and \(P(C_{i_1}^t \rightarrow C_{i+1}^t) \geq \kappa \wedge \cdots\). The transforming probability that \(C_{i_1}^t, C_{i_2}^t, \ldots\) transform into \(C_{i+1}^t\) is greater than \(\kappa\). The transforming probability that \(C_{i+1}^t\) transforms into \(C_{i_1}^t, C_{i_2}^t, \ldots\) is more than \(\kappa\). The reasons for this evolution behavior are “increasing or reducing microblog user nodes (nodes \(^+\)/nodes \(^-\)), or their concerning edges (edges \(^+\)/edges \(^-\))”.

VOLUME 8, 2020 176653
• (9) Explicit/implicit dissolving. In this case, a community $C_i^t$ may disappear at time $t + 1$. The evolution behaviors satisfy $\max\{P(C_i^t \rightarrow C_j^{t+1}), P(C_j^{t+1} \rightarrow C_i^t)\} \leq \kappa$ at time $t$. The mutual transforming probabilities between $C_i^t$ and $C_j^{t+1}$ are less than $\kappa$. The reasons for this evolution behavior are ‘‘reducing microblog user nodes (nodes~)’’ and their concerning edges (edges~). Considering the above 10 cases of community evolution behaviors, we propose community evolution behavior extraction algorithm 5. $\Delta C_{i,t+1}$ indicates communities belonging to $\Delta U_{t-t+1}$. Step 7 establishes social network community graphs for the communities. Steps 8-11 judge whether communities belonging to $\Delta C_{i,t+1}$ satisfy the evolution condition in Table 1, and the corresponding evolution behaviors are outputted. If the gravity changes in the community are more than $\lambda$, then Steps 12-15 proceed and output the explicit community evolution behaviors.

**TABLE 2. Reasons for community evolution behavior conditions.**

| Reason | Description |
|--------|-------------|
| R1     | Based on R1, average gravity changes from community $C_i^t$ to $C_j^{t+1}$ are more than $\lambda$. |
| R2     | Based on R1, mutual transforming probability $P(C_i^t \rightarrow C_j^{t+1})$ is more than $P(C_i^t \rightarrow C_j^{t+1})$. |
| R3     | Based on R3, average gravity change from community $C_i^t$ to $C_j^{t+1}$ is more than $\lambda$. |
| R4     | Based on R5, average gravity change from community $C_i^t$ to $C_j^{t+1}$ is more than $\lambda$. |
| R5     | Based on R5, average gravity change from community $C_i^t$ to $C_j^{t+1}$ is more than $\lambda$. |
| R6     | Some communities $C_{i1}^{t+1}, C_{i2}^{t+1}, \ldots$ exist. Mutual transforming probabilities from each of them to community $C_i^t$ are more than $\kappa$, and mutual transforming probabilities from community $C_i^t$ to each of them are more than $\kappa$. |
| R7     | Some communities $C_{i1}^{t+1}, C_{i2}^{t+1}, \ldots$ exist. Mutual transforming probabilities from each of them to community $C_j^{t+1}$ are more than $\kappa$, and mutual transforming probabilities from community $C_j^{t+1}$ to each of them are more than $\kappa$. |
| R8     | Mutual transforming probabilities of two any communities $C_i^t$ and $C_j^{t+1}$ are less than $\kappa$. |
| R9     | Mutual transforming probabilities of two any communities $C_i^t$ and $C_j^{t+1}$ are less than $\kappa$. |
| R10    | Mutual transforming probabilities of two any communities $C_i^t$ and $C_j^{t+1}$ are less than $\kappa$. |

**Algorithm 5 Extracting Community Evolution Behavior**

```
01 Input
02 $SN^t = \{c_1^t, c_2^t, \ldots, c_n^t\}$,
03 $SN^{t+1} = \{c_1^{t+1}, c_2^{t+1}, \ldots, c_n^{t+1}\}$
04 Output
05 Community evolution behaviors
06 Begin
07 Construct community relationship Graph $\Delta C_{i,t+1}$ at $t$ and $t + 1$ time.
08 For $c_i^t, c_j^{t+1} \in \Delta C_{i,t+1}$
09 Judge $c_i^t, c_j^{t+1}$ satisfy the conditions of evolution behavior in Table 1.
10 Output corresponding evolution behavior.
11 EndFor
12 For $c_i^t, c_j^{t+1} \notin \Delta C_{i,t+1}$
13 Judge $\Delta f(c_i^t, c_j^{t+1}) > \lambda$.
14 Output corresponding evolution behavior.
15 EndFor.
16 End.
```

V. EXPERIMENT AND RESULT ANALYSIS

In this section, our main tasks are to carry out experiments and to analyze the performance of our proposed algorithm. The specific experimental environment indicates a Windows 64 bit operation system, and the processor is an Intel Core i7-5600U (3.40 GHz/L3 4M) and 16 G running memory. We use the Python program language and anaconda compiler. We save experimental data (TXT or CSV format) on a local PC machine. We extract the features of microblog users from microblog posts by using the language cloud technology platform (LTP) system developed by the social computing and information retrieval research center of Kazakhstan.

A. MICROBLOG DATASET

Two microblog datasets are used to carry out our experiments. We use the Sina API open platform to extract microblog user data (the relationship between microblog users and their published microblog texts) and finally arrange them into dataset 1. Dataset 2 comes from the Fifth National Social Media Conference 2016 (SMP2016). It is the microblog dataset of user portrait competition. In Table 3, we show the two datasets and the number of microblog users (UN) at 12 time points (Months, Mon), the number of posts (PN) posted by microblog users and the number of relationships (RN) between them.

B. EVALUATION INDEX

We evaluate the quality of community detection through the relationship between community structures and their node gravities. In terms of community structures, we use the modularity proposed by [3]. In terms of the gravity relationship, we define the average community gravity. Modularity [3] is a commonly used method to measure the strength of network community structure. The nodes with high modularity tightly connect in the intracommunity and sparsely connect with nodes in other communities. Therefore, we use modularity...
TABLE 3. Two experimental datasets.

| Dataset | Mon | 1   | 2   | 3   | 4   | 5   | 6   |
|---------|-----|-----|-----|-----|-----|-----|-----|
| UN      | 780 | 780 | 768 | 764 | 762 | 789 |
| PN      | 704 | 664 | 555 | 507 | 730 | 811 |
| RN      | 42369 | 42864 | 37415 | 39165 | 52900 | 58571 |

**TABLE 4. Comparison of average gravity.**

| Dataset | Alg. | CNM | COPRA | Infomap | NRW | Our method |
|---------|-----|-----|-------|---------|-----|------------|
|         |     |     |       |         |     |            |
|         | Num(C'_t) | 37  | 89    | 51     | 58  | 64         |
|         | Num(U'_t) | 213 | 80    | 97     | 105 | 81         |
|         | f^t_{SN}  | 0.5557 | 0.1217 | 0.6863 | 0.5388 | 0.7227 |
|         | √s(f^t_{SN}) | 1.1024 | 0.0351 | 0.5545 | 0.5387 | 0.0295 |

**TABLE 5. Comparison of average modularity.**

| Dataset | Alg. | CNM | COPRA | Infomap | NRW | Our method |
|---------|-----|-----|-------|---------|-----|------------|
|         |     |     |       |         |     |            |
|         | Max(Q'_t) | 0.0872 | 0.0110 | 0.0323 | 0.0294 | 0.0345 |
|         | Q'_t | 0.3680 | 0.1667 | 0.1912 | 0.2459 | 0.3572 |
|         | √s(Q'_t) | 0.0172 | 0.0467 | 0.0582 | 0.0671 | 0.0625 |
|         | Q'_t | 0.5124 | 0.3023 | 0.3741 | 0.3908 | 0.4099 |

(C. COMPARISONS OF COMMUNITY DETECTION)

To evaluate the performance of our proposed community detection algorithm, we conduct experiments on two real datasets and compare them with other classic community detection methods, such as CNM [38], COPRA [39], Infomap [14] and NRW [40].

Table 4 shows the number of communities Num(C'_t) in the social network at time t. Num(U'_m) represents the number of microblog users in C'_t with the largest number of microblog users. f^t_{SN} represents the average gravity at time t in SN. √s(f^t_{SN}) is a statistical deviation of f^t_{SN}. Our proposed method, NRW, and Infomap algorithms are similar to the number of communities. The reasons are that these three methods are based on the random walk method. CNM has the largest number of microblog users. √s(Q'_t) indicates that the number of microblog users in the communities found by the CNM algorithm has an imbalanced distribution. The number of microblog users in one community is far greater than the number of microblog users in the other community. In contrast, the √s(Q'_t) of our proposed method is relatively low, which indicates that the community detection by our proposed method has a balanced distribution. In addition, f^t_{SN} of our proposed method is higher than other algorithms because our proposed method is a combination of community structure and gravity.

(D. COMPARISONS OF COMMUNITY EVOLUTION)

We compare our method with other classic community evolution methods, such as the LDM-CET (Local Dynamic Method for Community Evolution Track) [33], GCEA [37] and iLCD (improvement Local Community Detection) [41]. Fig. 4 shows the number of communities at each time on the two datasets. The iLDC algorithm discovers a large part of the duplicated communities. Comparing our proposed algorithm with the LDM-CET algorithm, the number of communities detected by GCEA is relatively stable. Our proposed algorithm discovers duplicated and stable communities.
Figs. 5 and 6 represent the average gravities $f_{SN}$ and modularities $Q_{SN}$ for the four algorithms, respectively. The community qualities of these four algorithms are maintained at a high level. As shown in Fig. 5, compared with the other three algorithms, our proposed algorithm has high average community gravities. The GCEA algorithm has low average community gravities, as clearly shown on dataset 2. The higher the average community gravities are, the higher the similarities between microblog users. The average community gravities in dataset 1 are higher than those in dataset 2. Fig. 6 indicates that the communities discovered by these four algorithms are similar. Obviously, the average community modularities in dataset 1 are higher than the average community modularities in dataset 2. Compared with the other three algorithms, the average community modularities of the iLCD algorithm are relatively low.

In Table 6, “O” represents our proposed algorithm, “G” represents GCEA, “L” represents LDM-CET, and “I” represents iLCD. It shows the recall of different algorithms for extracting community evolution behaviors. We conclude that our proposed algorithm outperforms the other three algorithms in the most adjacent times. The GCEA follows closely. The iLCD performs the worst. The number of communities that the iLCD algorithm finds at each time is too large, and microblog users in many communities are very small.

**Table 6.** Recall of community evolution for different methods.

| Alg. | Dataset1 | Dataset2 |
|------|----------|----------|
|      | M1 | M2 | M3 | M4 | M1 | M2 | M3 | M4 |
| 1-2  | 0.83 | 0.97 | 0.50 | 0.00 | 0.80 | 0.70 | 0.67 | 0.00 |
| 2-3  | 1.00 | 0.88 | 0.50 | 0.83 | 1.00 | 1.00 | 0.83 | 0.00 |
| 3-4  | 0.75 | 0.88 | 0.80 | 1.00 | 0.90 | 0.83 | 0.67 | 0.00 |
| 4-5  | 0.83 | 0.33 | 0.25 | 0.00 | 0.80 | 0.33 | 0.67 | 0.25 |
| 5-6  | 0.80 | 0.17 | 0.80 | 0.00 | 0.83 | 0.17 | 0.83 | 0.50 |
| 6-7  | 0.83 | 1.00 | 0.70 | 1.00 | 0.90 | 1.00 | 0.67 | 0.50 |
| 7-8  | 1.00 | 1.00 | 1.00 | 0.17 | 1.00 | 1.00 | 1.00 | 0.00 |
| 8-9  | 1.00 | 1.00 | 0.75 | 0.00 | 0.80 | 1.00 | 1.00 | 1.00 |
| 9-10 | 1.00 | 0.67 | 0.50 | 0.00 | 1.00 | 0.88 | 0.75 | 0.00 |
| 10-11| 0.80 | 1.00 | 0.33 | 0.00 | 0.90 | 0.88 | 0.67 | 0.00 |
| 11-12| m1:Our method, m2:GCEA, m3:LDM-CET, m4:iLCD |

Figs. 7 and 8 demonstrate the F value of four algorithms for extracting community evolution behavior on dataset 1 and dataset 2, respectively. Our proposed algorithm and GCEA algorithm perform best on dataset 1. The recall difference between our proposed algorithm and the two other algorithms is not obvious on dataset 2. The F values of the iLCD algorithm on dataset 2 are slightly higher than those on dataset 1. The F values of our proposed algorithm on dataset 2 are slightly lower than those on dataset 2. However, the average F values of our proposed algorithm are higher than those of the other three algorithms.

**Figure 7.** F-score of evolution behavior on dataset 1.

To evaluate the extracting evolution behavior algorithm, we only focused on the explicit and implicit “continuing”,
“shrinking” and “growing” evolution behaviors. As shown in Fig. 9, we can conclude that implicit evolution behaviors occur every two adjacent times. In adjacent times 7-8, many communities show implicit evolution behaviors, indicating that the interests of many microblog users have changed during this time period. However, in two adjacent times 9-10, only a small portion of the communities display implicit evolution behaviors, which indicates that only a few microblog users have changed their interest in this time period.

E. PARAMETER ANALYSIS IN THE ALGORITHM

Our proposed algorithms 1 and 2 involve several parameters that need to be identified. In this section, our primary task is to discuss important parameters $\varphi$ and $\varepsilon$. The ranges of these two parameters $\varphi$ and $\varepsilon$ are determined by experiments.

In Eq. (11), $\varphi \in [0, 1]$ is a weight to measure the gravities of microblog users and the characteristics of microblog network structure. Fig. 10 shows that the average gravity and modularity change with $\varphi$. When $\varphi$ increases from 0 to 3.5, the average gravity of the social network increases. When $\varphi > 3.5$, the average gravity of the social network remains almost stable. For the parameter $\varphi > 4.5$, the average modularities in the social network begin to decrease. The changes in average gravities and modularities with $\varphi$ are similar on datasets 1 and 2. To make our proposed algorithm good, $\varphi$ should be approximately 3.5-4.5.

In Eq. (13), $\varepsilon$ is a relatively small positive number. When the parameter $\varepsilon$ is equal to 0.0, each microblog user has only one gravity orientation. When $\varepsilon$ is equal to 1.0, each microblog user’s neighbor node is the gravity orientation of this microblog user. Fig. 11 shows the number of communities in the social network with different values of $\varepsilon$. When the parameter $\varepsilon$ is close to 0.0, the microblog users in the same community have high similarity, but the number of microblog users is very small. Conversely, if $\varepsilon$ is close to 1.0, a part of the microblog users in the same community have a lower similarity, which leads to the poor results of the algorithm. The optimal interval of $\varepsilon$ needs to be obtained according to the average gravity of the social network.

VI. CONCLUSION AND FUTURE WORKS

Community discovery refers to finding a group with a special relationship. The microblog community means finding out a group of microblog users with the same interests. Existing community detection methods only adopt the community
structure features and ignore the users’ microblog content features. Our proposed community detection algorithms not only analyze the community structure but also extract the microblog users’ interest features (community gravity) from users’ microblog content. The existing community evolution methods mainly study the explicit evolution behavior of the community. Our proposed microblog community evolution algorithm together finds explicit and implicit evolution and behaviors.

We define static and dynamic microblog social networks based on microblog data and reconstruct gravity relationships in complex networks by combining microblog user interest features. We use random walk and game theories to find the gravity tendencies of microblog users. We propose a microblog community discovery algorithm to detect basic features. We use random walk and game theories to find the explicit and implicit evolution and community. Our proposed microblog community evolution methods mainly study the explicit evolution behavior of the community structure features and ignore the users’ microblog content features. However, our proposed algorithms are verified by experimental results may not be sufficient and needs improvement for future works. Then, in the community evolution and detection evaluation methods, the time intervals are one month as a granularity. However, granularities not based on monthly but on daily, seasonal, or annual granularity may affect the experimental results. Thus, our future works will focus on this topic.

REFERENCES

[1] Y. Du, Y. Yi, X. Li, X. Chen, Y. Fan, and F. Su, “Extracting and tracking hot topics of micro-blogs based on improved latent Dirichlet allocation,” Eng. Appl. Artif. Intell., vol. 87, Jan. 2020, Art. no. 103279.

[2] B. W. Kernighan and S. Lin, “An efficient heuristic procedure for partitioning graphs,” Bell Syst. Tech. J., vol. 49, no. 2, pp. 291–307, Feb. 1970.

[3] M. E. J. Newman, “Detecting community structure in networks,” Eur. Phys. J. B, vol. 38, no. 2, pp. 321–338, 2004.

[4] J. Liu, Y.-J. Du, and J.-Z. Ren, “Micro-blog user community detection by focusing on micro-blog content and community structure,” in Proc. China Conf. Knowl. Graph Semantic Comput., Chengdu, China, vol. 784, 2017, pp. 92–103.

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

[6] M. E. J. Newman, “Fast algorithm for detecting community structure in networks,” Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top., vol. 69, no. 6, Jun. 2004, Art. no. 066133.

[7] P. Schuetz and A. Caffi, “Multistep greedy algorithm identifies community structure in real-world and computer-generated networks,” Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top., vol. 78, no. 2, Aug. 2008.

[8] J. Duch and A. Arenas, “Community detection in complex networks using extremal optimization,” Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top., vol. 72, no. 2, Aug. 2005, Art. no. 027104.

[9] J. Cao, Z. Bu, G. Gao, and H. Tao, “Weighted modularity optimization for crisp and fuzzy community detection in large-scale networks,” Phys. A, Stat. Mech. Appl., vol. 462, pp. 386–395, Nov. 2016.

[10] M. E. J. Newman, “Equivalence between modularity optimization and maximum likelihood methods for community detection,” Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top., vol. 94, no. 5, pp. 05231501–05231508, Nov. 2016.

[11] P. Chin, “Stochastic block model and community detection in sparse graphs: A spectral algorithm with optimal rate of recovery,” in Proc. Conf. Learn. Theory, Paris, France, vol. 40, 2015, pp. 391–423.

[12] C. Bordenave, M. Lelarge, and L. Massouille, “Non-backtracking spectrum of random graphs: Community detection and non-regular Ramanujan graphs,” in Proc. IEEE 56th Annu. Symp. Found. Comput. Sci., Oct. 2015, pp. 1347–1357.

[13] J. Feng, N.-N. Jia, Z.-H. Qi, and X.-F. Zhang, “Identification of function modules in protein-protein interaction networks by modularity optimization method,” in Proc. Int. Conf. Mach. Learn. Cybern. (ICMLC), Guangzhou, China, Jul. 2015, pp. 47–51.

[14] M. Rosvall and C. T. Bergstrom, “An information-theoretic framework for resolving community structure in complex networks,” Proc. Nat. Acad. Sci. USA, vol. 104, no. 18, pp. 7327–7331, 2007.

[15] A. Lancichinetti and S. Fortunato, “Community detection algorithms: A comparative analysis,” Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top., vol. 80, no. 5, Nov. 2009, Art. no. 056117.

[16] J. Banks, C. Moore, J. Neeman, and P. Netrapalli, “Information-theoretic propagation community detection algorithm,” in Proc. 10th Int. Conf. Learn. Theory, New York, NY, USA, vol. 49, Jun. 2016, pp. 383–416.

[17] U. N. Raghavan, R. Albert, and S. Kumara, “Near linear time algorithm to detect community structures in large-scale networks,” Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top., vol. 76, no. 3, Sep. 2007, Art. no. 036106.

[18] C. Su, X. Jia, X. Xie, and Y. Yu, “A new random-walk based label propagation community detection algorithm,” in Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol. (WI-IAT), vol. 1, Singapore, Dec. 2015, pp. 137–140.

[19] X. Hu, W. He, L. Li, Y. Lin, H. Li, and J. Pan, “An efficient and fast algorithm for community detection based on node role analysis,” Int. J. Mach. Learn. Cybern., vol. 10, no. 4, pp. 641–654, Apr. 2019.

[20] L. Fang, Q. Yang, J. Wang, and W. Lei, “Signed network label propagation algorithm with structural balance degree for community detection,” in Proc. Int. Conf. Smart Homes Health Telematics, Wuhan, China, vol. 9677, 2016, pp. 427–435.

[21] D. Eroglu, M. Tanzi, S. van Strien, and T. Pereira, “Revealing dynamics, communities, and criticality from data,” Phys. Rev. X, vol. 10, no. 2, Jun. 2020, Art. no. 021047.

[22] Y. Du, F. Su, A. Yang, X. Li, and Y. Fan, “Extracting deep personas social relations in microblog posts,” IEEE Access, vol. 8, pp. 5488–5501, 2020.

[23] Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng, “Evolutionary spectral clustering by incorporating temporal smoothness,” in Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, San Jose, CA, USA, vol. 10, 2007, pp. 153–162.

[24] 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 Proc. 17th Int. Conf. World Wide Web, Beijing, China, 2008, pp. 685–694.

[25] M.-S. Kim and J. Han, “A particle-and-density based evolutionary clustering method for dynamic networks,” J. Proc. VLDB Endowment, vol. 2, no. 1, pp. 622–633, 2009.

[26] Y. Hong, C. Fu, Q. Huang, Z. Fang, J. Zeng, and L. Han, “Communities evolution analysis based on events in dynamic complex network,” in Proc. 14th Int. Conf. Dependable, Autonomic Secure Comput., Auckland, New Zealand, 2016, pp. 497–503.
[27] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” J. Stat. Mech., Theory Exp., vol. 2008, no. 10, 2008, Art. no. P10008.

[28] X. Zhou, Y. Liu, B. Li, and G. Sun, “Multiobjective bioeconomy based optimization algorithm with decomposition for community detection in dynamic networks,” Phys. A, Stat. Mech. Appl., vol. 436, pp. 430–442, Oct. 2015.

[29] Q. Guo, L. Zhang, B. Wu, and X. Zeng, “Dynamic community detection based on distance dynamics,” in Proc. IEEE/ACM Int. Conf. Adv. Social Natw. Anal. Mining, San Francisco, CA, USA, Aug. 2016, pp. 329–336.

[30] D. Duan, Y. Li, Y. Jin, and Z. Lu, “Community mining on dynamic weighted directed graphs,” in Proc. 1st ACM Int. Workshop Complex Netw. Meet Inf. Knowl. Manage., Hong Kong, 2009, pp. 11–18.

[31] Z. Chen, K. A. Wilson, Y. Jin, W. Hendrix, and N. F. Samatova, “Detecting and tracking community dynamics in evolutionary networks,” in Proc. IEEE Int. Conf. Data Mining Workshops, Dec. 2010, pp. 318–327.

[32] 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. Int. Conf. Data Mining Workshop, Atlantic City, NJ, USA, Nov. 2015, pp. 13–20.

[33] Y. Hu, B. Yang, and C. Lv, “A local dynamic method for tracking communities and their evolution in dynamic networks,” Knowl.-Based Syst., vol. 110, pp. 176–190, Oct. 2016.

[34] P. Sarkar and A. W. Moore, “Dynamic social network analysis using latent space models,” in Proc. Adv. Neural Inf. Process. Syst., Vancouver, BC, Canada, 2005, pp. 1145–1152.

[35] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu, “Graphscope: Parameter-free mining of large time-evolving graphs,” in Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, San Jose, CA, USA, 2009, pp. 687–696.

[36] X. Yu, J. Yang, and Z.-Q. Xie, “A semantic overlapping community detection algorithm based on field sampling,” Expert Syst. Appl., vol. 42, no. 1, pp. 366–375, Jan. 2015.

[37] G. Yin, K. Chi, Y. Dong, and H. Dong, “An approach of community evolution based on gravitational relationship refactoring in dynamic networks,” Phys. Lett. A, vol. 381, no. 16, pp. 1349–1355, Apr. 2017.

[38] 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. 6, Dec. 2004, Art. no. 066111.

[39] S. Gregory, “Finding overlapping communities in networks by label propagation,” New J. Phys., vol. 12, no. 10, 2010, Art. no. 103018.

[40] Y. Xin, Z.-Q. Xie, and J. Yang, “The adaptive dynamic community detection algorithm based on the non-homogeneous random walking,” Phys. A, Stat. Mech. Appl., vol. 450, pp. 241–252, May 2016.

[41] R. Cazabet and F. Amblard, “Simulate to detect: A multi-agent system for community detection,” in Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol., Lyon, France, vol. 2, Aug. 2011, pp. 402–408.

JINrong Hu received the B.Sc. and M.Sc. degrees in computer science from Sichuan Normal University, Chengdu, China, in 2005 and 2008, respectively, and the Ph.D. degree in computer science from Sichuan University, Chengdu, in 2012. She is an Associate Professor with the Chengdu University of Information Technology. Her main research interests include information retrieval, image processing, and artificial intelligence.

YAJUN DU received the Ph.D. degree from the School of Computer and Communication Engineering, Southwest Jiaotong University, in 2005. He is currently a Professor in computer science with Xihua University. He has published several articles in the fields of information retrieval and search engines. His experience and research work focus on information retrieval, search engines, Web mining, and computer networks. He is serving on the committees of Chinese information and a PC Member of some international conferences such as WISE, ICIC, CCIR, and WMSE.

Jia Liu received the M.S. degree in computer science and technology from the School of Computer and Software Engineering, Xihua University, in 2018. He has published two articles in the field of information retrieval such as search engines, focused crawlers, and knowledge. His experience and research work focus on social networks and software engineering.

VOLUME 8, 2020