Research on Evolutionary Community Detection Algorithms in Dynamic Weighted Networks (Postprint)

Authors: Zhang Gaozhen, Zhang Xiankun, Su Jing, Liu Yuanbo

Date: 2018-05-02T00:00:00+00:00

Abstract

Discovering community structures in dynamic networks is a highly complex yet meaningful process that enables enhanced observation and analysis of network evolution. To address the community detection problem in dynamic weighted networks, we propose an algorithm that integrates historical network community structures, termed the Evolutionary Community Detection Algorithm for dynamic weighted networks (ECDA). The algorithm consists of two steps: first, it computes the input matrix for the current time step by incorporating information from historical communities and network structures; then, it derives community partition results that integrate historical time-step information through this input matrix. The algorithm offers the following advantages: it can automatically discover community structures at each time step in dynamic weighted networks; and it exhibits high sensitivity to changes in both network structure and community structure. Experiments were conducted on both synthetic and real-world datasets, and the results demonstrate that the algorithm can effectively discover community structures in dynamic weighted networks, showing strong competitiveness compared to other algorithms.

Full Text

Preamble

Title: Research on Evolutionary Community Discovery Algorithm in Dynamic Weighted Networks

Authors: Zhang Gaozhen, Zhang Xiankun, Su Jing, Liu Yuanbo
Affiliation: School of Computer Science & Information Engineering, Tianjin University of Science & Technology, Tianjin 300457, China

Abstract: Detecting community structure in dynamic networks is a complex yet meaningful process that enables better observation and analysis of network evolution. To address the community detection problem in dynamic weighted networks, this paper proposes an algorithm that incorporates historical network community structures, called the Evolutionary Community Discovery Algorithm (ECDA) in dynamic weighted networks. The algorithm consists of two steps: first, it calculates the input matrix for the current timestep by combining information from historical communities and network structure; then, it obtains community partition results that incorporate historical timestep information through this input matrix. The algorithm offers several advantages: it can automatically discover community structures at each timestep in dynamic weighted networks, and it exhibits high sensitivity to changes in both network structure and community structure. Experiments on synthetic and real-world datasets demonstrate that the proposed algorithm can effectively detect community structures in dynamic weighted networks and shows strong competitiveness compared to other algorithms.

Keywords: dynamic networks; weighted networks; community detection; modularity
Classification Number: TP393

---

0 Introduction

Complex systems can be represented as complex networks, such as social networks, email networks, biological networks, and technological networks, which have been extensively studied and explored by researchers. In recent years, with the deepening of complex network research, it has been discovered that many real-world networks are composed of communities. A community in a complex network is a set of nodes with dense internal connections, where nodes within the same community are closely connected, while connections between communities are relatively sparse. Traditional social network analysis methods focus on static networks, primarily investigating how to identify meaningful community structures within a network. However, in many complex networks, interactions between entities change dynamically over time, and these network changes inevitably lead to corresponding changes in community structures. The dynamic nature of networks increases the complexity of community detection research. Static network analysis, which does not consider the temporal dimension, misses opportunities to capture community evolution patterns and, to some extent, fails to reflect the dynamic characteristics of community changes in dynamic networks, thereby affecting the accuracy of network analysis results.

Consequently, research on dynamic network community detection has attracted widespread attention from scholars and has become a hot topic in community detection research. The primary goal of dynamic network evolutionary community discovery is to identify community structures at different timesteps, focusing on methods for discovering dynamic community structures to reveal the continuously changing community structures implicit in networks. Network dynamics refer to the fact that relationships between network members change over time.

Algorithms for obtaining community structures in dynamic networks can be broadly divided into two categories. The first category consists of community structure evolution methods based on a two-step analysis strategy. Initially, community network data is divided into timesteps and integrated according to a certain time window. Then, the community structure at each timestep is calculated, the evolution characteristics of the same community across consecutive timesteps are tracked, and differences between these community structures are explained over time. In recent years, many two-step strategy algorithms for detecting evolutionary community structures have been proposed. Gergely et al. developed a new algorithm based on clique percolation that, for the first time, allowed large-scale investigation of the temporal dependencies of overlapping communities and described fundamental relationships of community evolution. Hopcroft et al. proposed a tracking clustering method based on node cosine similarity across different timestep snapshots to track evolving communities in large networks. Duan et al. proposed a community mining algorithm that includes community detection and change point detection on dynamic weighted directed graphs. Takaffoli et al. proposed a community matching algorithm to effectively identify and track similar communities over time. In these studies, community detection and community evolution are analyzed in two separate stages, without considering historical community structures that contain valuable information related to the current community structure.

The second category of methods considers the influence of historical network community structure information. Based on the observation that dynamic networks evolve slowly, Chakrabarti et al. first proposed the evolutionary clustering framework. Under this framework, evolutionary clustering should simultaneously optimize two potentially conflicting conditions: first, the clustering at any timestep should remain as faithful (or consistent) as possible to the current data; second, the clustering results should not change significantly from one timestep to the next. Thus, the dynamic community detection problem can be transformed into an optimization problem of a cost function. Chi proposed an evolutionary spectral clustering algorithm that mines dynamic community structures by optimizing a cost function composed of snapshot cost and temporal cost, where snapshot cost measures clustering quality at the current moment, and temporal cost measures the similarity between clustering results at the current and previous moments. Lin proposed the classic FacetNet framework model for dynamic community discovery, which employs non-negative matrix factorization to simultaneously analyze communities and their evolution. The iterative solution approach used in FacetNet ensures that the algorithm converges to a local optimum. However, a limitation of FacetNet is that the quality of its partitioning results depends on the number of communities in the network. Elham Havvaei et al. proposed a game-theoretic approach for discovering overlapping communities in dynamic complex networks. Guo et al. proposed an evolutionary community structure discovery algorithm for dynamic weighted networks, but this algorithm requires manual threshold setting at each evolutionary step of initial community, community expansion, and community merging, with different threshold settings significantly impacting the algorithm' s results.

This paper proposes an Evolutionary Community Discovery Algorithm (ECDA) for dynamic weighted networks. Drawing inspiration from fast community evolution algorithms in large networks, ECDA effectively addresses the problem of manual intervention in threshold setting for initial communities, community expansion, and community merging steps in evolutionary community structure discovery algorithms. The proposed ECDA algorithm can automatically discover community structures at each timestep in dynamic weighted networks and exhibits high sensitivity to changes in network structure and community structure. The algorithm has been tested on synthetic and real-world datasets, demonstrating its effectiveness and competitiveness compared to other algorithms.

---

1 Related Concepts

1.1 Dynamic Weighted Networks

A dynamic network with T timesteps can be represented as a network sequence $\mathcal{G} = \{G_1, G_2, ..., G_T\}$. Let $G_t = (V_t, E_t)$ denote the network structure at timestep t ($1 \le t \le T$), where $V_t = \{v_1^t, v_2^t, ..., v_{N_t}^t\}$ represents the set of nodes in the network, and N_t denotes the number of nodes; $E_t = \{e_1^t, e_2^t, ..., e_{M_t}^t\}$ represents the set of edges in the network, and M_t denotes the number of edges. If the edge weights in set E_t are not all 1, then G_t is a weighted network. If there exists a timestep t in the dynamic network $\mathcal{G}$ where the network G_t is weighted, then $\mathcal{G}$ is a dynamic weighted network.

Generally, at each timestep t, the network G_t can be represented by an adjacency matrix $A^{(t)} = (w_{ij}^{(t)})_{N_t \times N_t}$, where $w_{ij}^{(t)}$ represents the weight of the edge connecting node i and node j; if there is no edge between node i and node j, then $w_{ij}^{(t)} = 0$. For a network G_t with N_t nodes at timestep t, the degree of the i-th node is defined as the sum of weights of all edges connected to node i, formulated as $d_i^{(t)} = \sum_{j \in N_t} w_{ij}^{(t)}$.

1.2 Input Adjacency Matrix

To discover evolutionary communities in dynamic weighted networks, two potential conditions must be considered: the clustering result at any timestep should remain as faithful (or consistent) as possible to the current data; and the clustering results should not change significantly from one timestep to the next. For a weighted network G_t at timestep t (represented by adjacency matrix $A^{(t)}$), this paper defines an input matrix $U^{(t)} = (u_{ij}^{(t)})_{N_t \times N_t}$, where each element is defined as follows:

$$u_{ij}^{(t)} = \begin{cases} \alpha w_{ij}^{(t)} + (1-\alpha) u_{ij}^{(t-1)} & \text{if } t \geq 2 \\ w_{ij}^{(t)} & \text{if } t = 1 \end{cases}$$

where α is a user-defined parameter in the range $[0, 1]$, and $\delta(C_i^{(t-1)}, C_j^{(t-1)})$ is defined as:

$$\delta(C_i^{(t-1)}, C_j^{(t-1)}) = \begin{cases} 1 & \text{if } C_i^{(t-1)} = C_j^{(t-1)} \\ 0 & \text{otherwise} \end{cases}$$

indicating whether nodes i and j belong to the same community at timestep $t-1$.

This paper provides the following interpretation of $U^{(t)}$. For a dynamic weighted network $\mathcal{G}$, the network at timestep $t = 1$ is the initial weighted network, where no historical network exists; therefore, the input matrix $U^{(1)}$ equals the adjacency matrix $A^{(1)}$ of the weighted network. When $t \geq 2$, the input matrix $U^{(t)}$ must comprehensively consider information from the adjacency matrix $A^{(t)}$ of the weighted network at the current timestep t, as well as information from historical weighted networks and historical community structures. The parameter α is a user-defined parameter that balances the two conditions mentioned above: a larger α places more emphasis on historical network and community structure information. As α increases, the influence of historical network and community structure information becomes smaller.

1.3 Modularity

Modularity, also known as the modularization metric, was proposed by Newman and Girvan as a method to measure the strength of community structure in networks. It is defined as the proportion of the total number of edges within communities to the total number of edges in the network, minus an expected value that represents the same proportion in a randomly generated network with the same community assignment. For convenience of calculation, Newman rewrote the modularity function based on the adjacency matrix. Below, this paper presents the modularity for dynamic weighted networks at timestep t:

$$Q_t = \frac{1}{2m_t} \sum_{i,j \in I_t} \left(u_{ij}^{(t)} - \frac{d_i^{(t)} d_j^{(t)}}{2m_t} \right) \delta(C_i^{(t)}, C_j^{(t)})$$

where $d_i^{(t)}$ represents the degree of node i in the input matrix; $d_j^{(t)}$ represents the degree of node j in the input matrix; and m_t represents the sum of edge weights in the input matrix $U^{(t)}$.

The modularity value is less than 1 and may also be negative. A larger modularity value indicates a more stable community partition. When the modularity value reaches its maximum and approaches 1, it indicates that the community structure is at its strongest. When the entire network is considered as a single community, the modularity value is 0. For a network partition, when the number of edges within communities is less than the expected number of edges in a random network, the modularity value may be 0 or negative; in such cases, according to Newman and Girvan' s definition, the modularity value is 0. For most networks, the modularity value falls within the interval $[0.3, 0.7]$.

The modularity gain formula ΔQ_t was proposed by Vincent et al. in their fast community evolution algorithm for large networks. Below, this paper presents the modularity gain $\Delta Q_{t,i\to C}$ in dynamic weighted networks when assigning node i to community C at timestep t:

$$\Delta Q_{t,i\to C} = \frac{d_{i,in}^{(t)}}{2m_t} - \frac{d_i^{(t)} \cdot d_{C,tot}^{(t)}}{2(2m_t)^2}$$

where $d_{i,in}^{(t)}$ represents the sum of weights of edges between node i and nodes within community C at timestep t; $d_{C,tot}^{(t)}$ represents the sum of weights of edges between nodes in community C and all other nodes at timestep t; and $d_i^{(t)}$ represents the sum of edge weights between node i and nodes in community C at timestep t. This paper identifies the neighbor node that maximizes modularity gain before and after node assignment. If $\Delta Q_{t,i\to C} > 0$, the node i is assigned to the community of that neighbor node; otherwise, it remains unchanged.

In Step 5, this paper only aims to use modularity gain to determine whether a node can join community C; therefore, in practice, it is not necessary to calculate the exact modularity value, only to know whether the current operation' s modularity gain is greater than 0. Consequently, this paper derives the relative modularity gain formula:

$$\Delta Q'_{t,i\to C} = \frac{d_{i,in}^{(t)}}{d_i^{(t)}} - \frac{d_{C,tot}^{(t)}}{2m_t}$$

Here, the relative modularity gain value may be greater than 1, but it is not the true modularity gain value; however, its sign indicates whether the current operation increases modularity. Therefore, using this relative modularity gain formula can significantly reduce the algorithm' s time complexity.

Algorithm Complexity Analysis. The computational complexity for calculating the input matrix is $O(e)$, where e represents the number of edges in the network. The computational complexity for evolutionary community discovery based on input matrix $U^{(t)}$ is also $O(e)$. Therefore, the time complexity per timestep is $O(e)$, and the computational complexity for the entire dynamic network is $O(T \cdot e)$, where T represents the number of timesteps.

---

2 Algorithm Description

Based on the input matrix defined in Chapter 1, this paper proposes an Evolutionary Community Discovery Algorithm (ECDA) for dynamic weighted networks, which consists of two main components: computing the input matrix and performing evolutionary community discovery based on the input matrix. The detailed description of the ECDA algorithm is presented in Algorithm 1.

Algorithm 1: Evolutionary Community Discovery Algorithm (ECDA) in Dynamic Weighted Networks

Input: Dynamic network $\mathcal{G} = \{G_1, G_2, ..., G_T\}$, corresponding adjacency matrices $\{A^{(1)}, A^{(2)}, ..., A^{(T)}\}$

Output: Community sequence $\{C^{(1)}, C^{(2)}, ..., C^{(T)}\}$, where $C^{(t)}$ represents the community partition at timestep t

1. for $t = 1$ to T do
2. Compute input matrix $U^{(t)}$;
3. Treat each node in input matrix $U^{(t)}$ as an independent community, where the number of communities equals the number of nodes;
4. For each node i, sequentially attempt to assign node i to the community of its neighbor node j. Calculate the modularity gain $\Delta Q_{t,i\to C_j}$ for each neighbor node j. If $\max \Delta Q_{t,i\to C_j} > 0$, assign node i to the community of the neighbor node with maximum ΔQ; otherwise, keep it unchanged;
5. Repeat Step 4 until the community membership of all nodes no longer changes;
6. Compress communities: compress all nodes within the same community into a new node. Convert edge weights between nodes within a community into self-weights of the new node. Convert edge weights between communities into weights between new nodes;
7. Repeat Steps 3-6 until the modularity of the entire network no longer changes;
8. $t \leftarrow t + 1$;
9. end

---

3 Experimental Results and Analysis

This paper employs two synthetic networks and one real-world network (a scientist research collaboration network dataset) to evaluate the performance of the proposed Evolutionary Community Discovery Algorithm (ECDA) in dynamic weighted networks. Three partition quality evaluation functions are used to measure the obtained community structures. The first function is the Snapshot Quality function, which evaluates the consistency between the current community partition result and the current network data. Higher snapshot quality indicates that the community partition result is more faithful to the network data at that timestep. The Snapshot Quality function is defined as:

$$SQ_t = \frac{1}{2m_t} \sum_{i,j \in I_t, i \neq j} \left(w_{ij}^{(t)} - \frac{d_i^{(t)} d_j^{(t)}}{2m_t} \right) \delta(C_i^{(t)}, C_j^{(t)})$$

where $d_i^{(t)}$ represents the degree of node i in network G_t at timestep t; $d_j^{(t)}$ represents the degree of node j in network G_t at timestep t; and m_t represents the sum of all edge weights in network G_t at timestep t. The meanings of other parameters are the same as above.

The second function is the Historical Quality function, which measures the instantaneous smoothness of the current community partition structure' s fit to historical network information and community structure information. A higher historical quality evaluation result indicates that the community partition structure has not changed significantly between the previous timestep and the current timestep. The Historical Quality evaluation function is defined as:

$$HQ_t = \frac{1}{2m_{t-1}} \sum_{i,j \in I_{t-1}, i \neq j} \left(w_{ij}^{(t-1)} - \frac{d_i^{(t-1)} d_j^{(t-1)}}{2m_{t-1}} \right) \delta(C_i^{(t)}, C_j^{(t)})$$

where $d_i^{(t-1)}$ represents the degree of node i in network G_{t-1} at timestep $t-1$; $d_j^{(t-1)}$ represents the degree of node j in network G_{t-1} at timestep $t-1$; and m_{t-1} represents the sum of all edge weights in network G_{t-1} at timestep $t-1$. The meanings of other parameters are the same as above.

The third function is the Total Quality function, which evaluates the overall partition quality of the network after balancing the two potentially conflicting conditions. A higher overall quality evaluation result indicates higher overall quality of the network community partition. The Total Quality function is defined as:

$$TQ_t = \frac{1}{2m_t^{(U)}} \sum_{i,j \in I_t, i \neq j} \left(u_{ij}^{(t)} - \frac{d_i^{(t,U)} d_j^{(t,U)}}{2m_t^{(U)}} \right) \delta(C_i^{(t)}, C_j^{(t)})$$

where $d_i^{(t,U)}$ represents the degree of node i in input matrix $U^{(t)}$ at timestep t; $d_j^{(t,U)}$ represents the degree of node j in input matrix $U^{(t)}$ at timestep t; and $m_t^{(U)}$ represents the sum of all edge weights in input matrix $U^{(t)}$ at timestep t. The meanings of other parameters are the same as above.

This paper implements and improves the Evolutionary Community Structure Discovery (ECSD) algorithm proposed by Guo et al. The expansion and merging steps of the ECSD algorithm are optimized. In the community expansion step, ECSD is modified to select the community that maximizes modularity contribution and exceeds a threshold for node addition. In the community merging step, ECSD is modified to select the pair of communities that maximizes modularity and exceeds a set threshold for merging. Comparative experiments are conducted between the improved ECSD algorithm and the proposed ECDA algorithm to demonstrate the effectiveness and competitiveness of ECDA.

The proposed ECDA algorithm is also compared with the ECSD algorithm, PDG algorithm, and InfoMap algorithm in terms of modularity quality at each timestep. Since PDG and InfoMap can only be applied to unweighted networks, the per-timestep modularity comparison experiments are conducted only on unweighted synthetic and real-world networks. By analyzing modularity quality evaluation metrics in other literature, it is found that their modularity quality evaluation metric corresponds to the Snapshot Quality (SQ) function described above. To maintain consistency with other literature, this paper also adopts the term "modularity quality evaluation."

3.1 Artificial Networks

This paper uses two synthetic networks to validate the effectiveness of the proposed ECDA algorithm: a small-scale dynamic weighted network dataset used by Guo et al., and an artificial network dataset generated using the LFR benchmark.

3.1.1 Small-scale Dynamic Weighted Network In this small-scale dynamic weighted network, the nodes and edges in each timestep change continuously, and the resulting community structures also evolve over time. This synthetic network is used to test the effectiveness of the proposed ECDA algorithm. The initial dynamic weighted network is shown in [Figure 1: see original paper].

From [Figure 1: see original paper], it can be observed that at timestep 1, the network contains 7 nodes that can be partitioned into 2 communities. However, at timestep 2, the number of nodes, edges, and edge weights all change. For example, the edge between node 2 and node 3 disappears, a new node 8 is added, etc. At timestep 3, the network' s edges and edge weights change again. At timestep 4, the network can be clearly partitioned into 3 communities. Notably, at timestep 2, node 5 could be assigned to either of two communities (the community consisting of nodes {1,2,3,4} or the community consisting of nodes {6,7,8}); however, referencing the network partition at timestep 1, node 5 should be assigned to the second community (consisting of nodes {6,7,8}). In such cases, historical network and community information become crucial.

The partition results obtained using the ECDA algorithm on this synthetic dynamic weighted network are shown in [Figure 2: see original paper]. Detailed information on Snapshot Quality (SQ), Historical Quality (HQ), and Total Quality (TQ) evaluation results is provided in . At timestep 1, since there is no historical network information or historical community structure information, no Historical Quality result is available. Moreover, as introduced in Section 2, the Snapshot Quality equals the Total Quality at this timestep.

Comparative experimental results between ECDA and ECSD algorithms on this synthetic network are shown in [Figure 3: see original paper], demonstrating the effectiveness and competitiveness of the proposed algorithm.

3.1.2 LFR Artificial Networks This paper uses the LFR benchmark to generate four datasets, which are used as four timesteps for experiments. The parameter settings for network generation are shown in , where N represents the number of nodes, k represents the average node degree, $maxk$ represents the maximum node degree, mu represents the mixing parameter, $minc$ represents the minimum community size, and $maxc$ represents the maximum community size. The parameters for the four datasets generated by LFR are listed from top to bottom.

The proposed ECDA algorithm and the comparison ECSD algorithm are applied to these datasets. Detailed information on Snapshot Quality (SQ), Historical Quality (HQ), and Total Quality (TQ) evaluation results for the LFR artificial network is provided in .

As shown in the table, Historical Quality (HQ) results can be negative. This occurs because the networks generated by LFR have no inherent connections between them, meaning the changes between networks are significant, and the resulting community structure changes are also substantial. Therefore, negative Historical Quality (HQ) values are obtained. The Historical Quality (HQ) evaluation result reflects the instantaneous smoothness of the current community partition structure' s fit to historical network and community structure information. If network structure and community structure fluctuate significantly between different timesteps, the Historical Quality evaluation result will be low. Thus, in the LFR network, the Historical Quality (HQ) results obtained by ECDA are lower than those of ECSD, proving that network and community structures change significantly between different LFR networks, which demonstrates the sensitivity of the ECDA algorithm to changes in network structure and community structure.

Additionally, in the LFR network, the proposed ECDA algorithm is compared with PDG, InfoMap, and ECSD algorithms in terms of modularity quality at each timestep. The experimental results are shown in [Figure 5: see original paper].

From [Figure 5: see original paper], it can be seen that the proposed algorithm achieves good overall partition results in the LFR network. Except at the first timestep where ECDA' s modularity is slightly lower than PDG, the proposed

ECDA algorithm obtains better modularity quality in the other three timesteps, indicating that ECDA has good community partition performance.

3.2 Real Networks

This paper uses a real-world network dataset (a scientist research collaboration network dataset) to demonstrate the effectiveness and stability of the proposed ECDA algorithm. The scientist research collaboration network represents inter-university collaborative paper networks among Chinese "985 Project"universities. The "985 Project" is a higher education construction initiative implemented by the Chinese government to build several world-class universities and internationally renowned high-level research universities. The name originates from May 4, 1998, when Jiang Zemin proposed building world-class universities in a speech at Peking University' s centennial celebration. Initially, nine universities were selected for the 985 Project, forming the C9 League. By the end of 2011, there were 39 universities in the 985 Project. This paper collected inter-university collaborative papers from Chinese "985 Project" universities on Web of Science from 1996 to 2003.

The dynamic network is organized with each year as a timestep. The community partition results obtained using the proposed algorithm are shown in [Figure 6: see original paper].

Detailed information on Snapshot Quality (SQ), Historical Quality (HQ), and Total Quality (TQ) evaluation results for the scientist research collaboration network is provided in . For the year 1996, since there is no historical network information or historical community structure information, no Historical Quality result is available. As introduced in Section 2, the Snapshot Quality equals the Total Quality at this timestep.

Comparative experimental results between ECDA and ECSD algorithms on this scientist research collaboration network are shown in [Figure 7: see original paper], demonstrating the performance of the proposed algorithm.

From the evaluation results shown in [Figure 7: see original paper], the proposed ECDA algorithm achieves good results in both Snapshot Quality (SQ) and Total Quality (TQ). Since Historical Quality (HQ) reflects the instantaneous smoothness of the current community partition structure' s fit to historical network and community structure information, if network structure and community structure fluctuate significantly between timesteps, the Historical Quality evaluation result will be low. Therefore, in the scientist research collaboration network, the Historical Quality (HQ) results obtained by ECDA are lower than those of ECSD, indicating that network and community structures changed significantly between different timesteps, which validates that the ECDA algorithm is more sensitive to changes in network structure and community structure.

Additionally, the proposed ECDA algorithm is compared with PDG, InfoMap, and ECSD algorithms in terms of modularity quality at each timestep on this scientist research collaboration network dataset. The experimental results are shown in [Figure 8: see original paper].

As shown in [Figure 8: see original paper], the proposed ECDA algorithm achieves good community partition results. Except at timestep 2001 where the modularity is slightly lower than PDG, ECDA obtains better partition results at other timesteps, verifying that the proposed ECDA algorithm is competitive compared to other algorithms.

---

4 Conclusion

This paper proposes an Evolutionary Community Discovery Algorithm (ECDA) for dynamic weighted networks. The algorithm can automatically discover community structures at each timestep in dynamic weighted networks. Experiments on synthetic and real-world datasets demonstrate that the algorithm can effectively detect community structures in dynamic weighted networks and shows strong competitiveness compared to other algorithms. Moreover, experimental results prove that the algorithm is more sensitive to changes in dynamic weighted network structures and community structures.

However, the algorithm still has limitations: it cannot be applied to directed networks, and analyzing the specific changes in community structures across different timesteps remains a research hotspot.

References

[1] Mao Lan. Research on network traffic flow based on self-similarity characteristics [D]. Wuxi: Jiangnan University, 2014.

[2] Freeman L C. The sociological concept of 'group' : an empirical test of two models [J]. American Journal of Sociology, 1992, 98(1): 152.

[3] Porter M A, Onnela J P, Mucha P J. Communities in networks [J]. Notices of the American Mathematical Society, 2009, 56(1082): 1.

[4] Girvan M, Newman M E J. Community structure in social and biological networks [J]. Proceedings of the National Academy of Sciences, 2002, 99(12): 7821.

[5] Hopcroft J, Khan O, Kulis B, et al. Natural communities in large linked networks [C]// Proc of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2003: 541.

[6] Strogatz S H. Exploring complex networks [J]. Nature, 2001, 410(6825): 268.

[7] Giatsoglou M, Vakali A. Capturing social data evolution using graph clustering [J]. IEEE Educational Activities Department, 2013, 17(1): 74.

[8] Zhou Xu. Research on community discovery algorithms in complex networks [D]. Jilin: Jilin University, 2016.

[9] Wang Li, Cheng Xueqi. Dynamic community discovery and evolution in online social networks [J]. Chinese Journal of Computers, 2015, 38(2): 219.

[10] Palla G, Barabási A L, Vicsek T. Quantifying social group evolution [J]. Nature, 2007, 446(7136): 664.

[11] Hopcroft J, Khan O, Kulis B, et al. Tracking evolving communities in large linked networks [J]. Proceedings of the National Academy of Sciences of the United States of America, 2004, 101(Suppl 1): 5249.

[12] Duan D, Li Y, Jin Y, et al. Community mining on dynamic weighted directed graphs [C]// Proc of ACM International Workshop on Complex Networks Meet Information & Knowledge Management. 2009: 11.

[13] Takaffoli M, Sangi F, Fagnan J, et al. Community evolution mining in dynamic social networks [J]. Procedia-Social and Behavioral Sciences, 2011, 22(22): 49.

[14] Chakrabarti D, Kumar R, Tomkins A. Evolutionary clustering [C]// Proc of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2006: 554.

[15] Chi Y, Song X D, Zhou D, et al. Evolutionary spectral clustering by incorporating temporal smoothness [C]// Proc of the 13th International Conference on Knowledge Discovery and Data Mining. 2007: 153.

[16] Lin Y R, Zhu S, Sundaram H, et al. Analyzing communities and their evolutions in dynamic social networks [J]. ACM Trans on Knowledge Discovery from Data, 2009, 3(2): 1.

[17] Alvari H, Hajibagheri A, Sukthankar G. Community detection in dynamic social networks: a game-theoretic approach [C]// Proc of IEEE//ACM International Conference on Advances in Social Networks Analysis and Mining. 2014: 101.

[18] Guo C, Wang J, Zhang Z. Evolutionary community structure discovery in dynamic weighted networks [J]. Physica A: Statistical Mechanics & Its Applications, 2014, 413(11): 565.

[19] Blondel V D, Guillaume J L, Lambiotte R, et al. Fast unfolding of communities in large networks [J]. Journal of Statistical Mechanics: Theory and Experiment, 2008, 2008(10): 155.

[20] Newman M E J, Girvan M. Finding and evaluating community structure in networks [J]. Physical Review E, Statistical, Nonlinear, and Soft Matter Physics, 2004, 69(2 Pt 2): 026113.

[21] Xu Yuguang, Jiang Fei, Zhu Enqiang, et al. Research on dynamic community discovery algorithm based on individual stability game [J]. Journal of

Electronics & Information Technology, 2017, 39(4): 763.

[22] Rosvall M, Bergstrom C T. Maps of random walks on complex networks reveal community structure [J]. Proceedings of the National Academy of Sciences of the United States of America, 2007, 105(4): 1118.

Note: Figure translations are in progress. See original paper for figures.