Weighted-Group-Density Based Community Discovery Algorithm for Dynamic Weighted Networks

Dongming Chen, Xinyu Huang, Yunkai Wang, Dongqi Wang

Software College, Northeastern University, China
chendm@mail.neu.edu.cn, neuhxy@163.com, rickwyk@qq.com, wangdq@swc.neu.edu.cn

Abstract

Aiming at solving the problem of community detection in weighted dynamic networks, this paper defines a Weighted-Group-Density metric to evaluate the community closeness. By analyzing the dynamic changes of weighted networks, we propose a novel community detection algorithm based on weighted group density for dynamic weighted networks. To validate the performance of the proposed algorithm, several experiments are conducted, where the datasets are extracted from the novel 'A Song of Ice and Fire'. Experimental results show the proposed algorithm outperforms the competitive algorithms, which is of great significance to the dynamic research of complex networks.

Keywords: Complex network, Community discovery, Modularity, Group density, Dynamic network

1 Introduction

Network science is interdisciplinary research, which is widely used to represent the connections of complex systems such as social networks, transportation networks, protein molecular networks, etc. A network is composed of nodes and edges, and most networks are holding the community structures. The so-called community structure is known as a group of nodes that are closely connected inside and sparsely linked outside. The research of community detection has become a popular topic, since it is a powerful tool to divide the network into different groups (also known as clusters or modules), according to the topology of the network. However, community detection is an NP problem and we can hardly get a precise definition of community structure yet. Hence, there are no clear criteria for evaluating the performance of different algorithms. The study of community structures in dynamic networks is helpful to understand the network structure and conduct further research. Take social networks for example, community detection benefits revealing the distributions and behaviors of individuals.

Numerous attempts have been made to uncover the community structures in static networks, such as Girvan-Newman (GN) and Fast-Newman algorithm [1]. With the development of research on complex networks, many scholars have devoted themselves to studying dynamic networks based on time snapshots. By conducting the classic static algorithms on every snapshot, we can obtain the dynamic community structures. Representative algorithms such as Hopcroft's method for tracking and processing paper link networks [2], and dynamic community partitioning algorithm based on clique penetration [3] has exhibited remarkable performance. However, the real-world networks are dynamic and differ from one to another, which has brought enormous challenging to the analyzing task. For instance, a large community is divided into several small communities, or two communities are directly merged, which may lead to inconsistency in partition process. Besides, the processing of dynamic changes is also time-consuming. Thus, it's really difficult to balance a reasonable partition result and acceptable efficiency.

In 2006, Chakrabarti et al. [4] proposed an evolutionary clustering framework, with its purpose at partitioning network based on both current and previous snapshots in dynamic networks. The cost function introduced in their framework aims to optimize the compromise processing and simultaneously obtain higher flexibility. Most of the community detection algorithms for static networks can also be employed in this framework, such as the similarity matrix clustering system based on historical information with current information by Chi [5] et al., and the FacetNet dynamic community partition framework by Lin et al. [6], and so on. However, it is criticized by the incapability of dealing with large-scale networks. Another representative method is proposed by Lancichinetti and Fortunato [7] in 2012, i.e., Consensus Clustering, which integrates different snapshots by setting dynamic sliding windows to find stable dynamic community structure. The method is effective and stable but suffering from the selection of parameters. Other solutions such as probability model [8], soft overlapping [9], multi-objective optimization method [10] are proposed in these years to reveal underlying community structures in dynamic networks

*Corresponding Author: Xinyu Huang; E-mail: neuhxy@163.com
DOI: 10.3966/160792642020092105026

or even multilayer networks. Besides, to study the community structures in decentralized online social networks, Guidi et al. [11] utilized a real Facebook dataset to uncover the *birth*, *death*, *growth*, *contraction*, *merge*, *split*, *resurgence* and *continue* processes, which proves the social network has high instability and distributed solutions to manage the dynamism are necessary.

The study of dynamic community detection is not only a powerful tool to analyze online social networks, but also benefits some other fields, such as biological systems and bibliographic networks. Jin et al. [12] proposed a multi-scale community detection method for brain functional networks using the Dynamic Time Wrapping, which reveals the motor state of the human brain involved in functional activities and provides an effective method for the study of human brain pathologies. Zhang et al. [13] proposed a dynamic topical community detection (DTCD) method to detect communities and their topical meanings in dynamic networks. They considered a community as a mixture of topics and documents with time stamps and utilized the proposed DTCD algorithm to find topics, community and temporal variations. The proposed method was verified on two real-world network datasets: online forum Reddit and academic cooperation network DBLP, which shown superiority on competitors. In brief, the study on detecting communities in dynamic networks is significant and valuable in a variety of subjects.

The remaining part of this paper is organized as follows. Section 2 introduces the improved group density method for community detection in weighted networks. Section 3 extends the weighted group density algorithm to proceeding dynamic changes. The fourth part analyzes the datasets experimentally, and the last section summarizes the whole paper.

2 Modelling and Definitions

Group Density (GD) [14] is a non-global single community structure evaluation index and defined as

$$\boldsymbol{GD} = \sqrt{\left(\frac{k_{p-in}}{k_{p-in}+k_{p-out}}\right)^{\alpha} \times \frac{k_{p-in}}{k_{p-in}+k_{p-out}} \times \left(\frac{k_{p-in}}{p}\right)^{\beta}}, \quad \textbf{(1)}$$

where $k_{l\text{-}in}$ represents the number of edges within community k, $k_{l\text{-}out}$ represents the number of external edges of community k, $k_{p\text{-}in}$ represents the number of nodes within community k, $k_{p\text{-}out}$ represents the number of nodes outside community k, $\alpha$is the influence of positive power coefficient to weigh the proportion of community edges in the equation, p represents the total number of nodes in the network, and β is the fractional exponential power coefficient, which is to prevent the function value from being too small due to excessive small size of the community in a large-scale network, so as to balance the ratio of the number of community nodes to the total number of nodes in the network. The range of group density GD is [0, 1].

In real-world networks, there is no simple 0 or 1 relationship between the nodes. In order to express this relationship, the concept of weighted networks is employed. When the network changes constantly, we call this varied process dynamic network. According to the characteristics of group density, this paper replaces the number of edges in the original definition of group density function with the sum of the weights of the edges, within and outside a community respectively, which can solve the problem of applying the weights of group density. The weighted group density is defined as

$$\boldsymbol{GD^w} = \sqrt{\frac{k_{p-in}}{k_{p-in}+k_{p-out}} \left(\frac{\Sigma w_{l-in}}{\Sigma w_{l-in}+\Sigma w_{l-out}}\right)^{\alpha} \times \left(\frac{k_{p-in}}{p}\right)^{\beta}}, \quad \textbf{(2)}$$

where $w_{l\text{-}in}$ is the internal edge weight, $w_{l\text{-}out}$ is the external edge weight, $k_{p\text{-}in}$ represents the number of nodes inside community k, $k_{p\text{-}out}$ represents the number of nodes connected to community k which is outside the community, p denotes the total number of nodes in the network, and α and β are balance parameters.

At the same time, the range of weighted group density is also fixed. When there is only one node in the community, w_{l-in} equals 0, and the calculated GD value is 0. When the community contains all the edges and nodes of the network, that is, the whole network is regarded as a community, where w_{l-out} equals 0, k_{p-out} equals 0, k_{p-in} equals p, and the calculated GD value is 1. In the process of community enlargement, w_{l-in} is less than or equal to the sum of w_{l-in} and w_{l-out}, and k_{p-in} is less than the sum of k_{p-in} and $k_{p\text{-}out}$, so the value range of equation (2) is [0, 1].

3 Methods

By combining weighted group density metric and GD algorithm in our previous work [14], we propose a weighted-group-density algorithm for community detection in weighted networks. The process is described as follows:

Step1: Initialize the network, each node is regarded as a community.

Step2: Go to Step 5 if each sub-network is a community; go to Step3, otherwise.

Step3: Select one community with the smallest size and the minimum weighted group density among the current communities as the initial community.

Step4: Select all the neighboring communities of the initial community, then calculate the weighted group density of each community merged with the initial community, further merging the community with the maximum value-added into the initial community as the target community, calculating the modularity, and saving the round result. After this, go back to Step 2.

Step5: The final community partitioning result is obtained when the modularity reaches its maximum.

Initially, the algorithm marks n communities with the n nodes, which needs complexity of $O(n)$. The traversing process requires $O(n+n(n+n))$, thus, the total complexity of Weighted-Group-Density algorithm is $O(2n^2+n)$, which can be simplified as $O(n^2)$. The weighted-group-density algorithm is capable of detecting communities in weighted networks, while it achieves a complexity close to Fast-Newman algorithm $(O(N^2))$ and superior to the GN algorithm $(O(NM^2))$. Inspired by the classic Fast-Newman algorithm, we have improved the judgment metrics of the traversing process. The calculation of the modularity is based on the global structure, which is very time-consuming, while the proposed weighted density is a local metric and saves running time.

Suppose to be given a toy network as shown in Figure 1, there are mainly two groups, i.e., (1, 2, 3, 4) and (6, 7, 8, 9, 10), respectively. However, how to divide node 5 between the two groups is a difficult problem. We are considering whether it is simpler when the edges are weighted? What's the partitioning strategy for node 5?

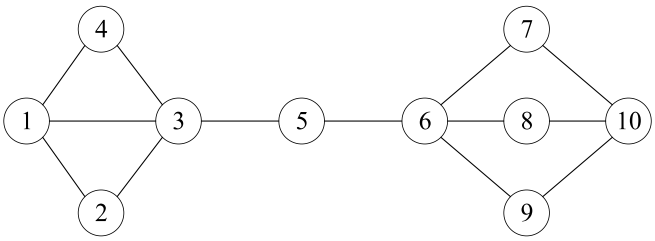

Figure 1. Topology of a toy network

There are two assumptions in the weight's assignment of the toy network, as described in the following aspects:

Case 1: The weight of edge (3, 5) is 2, while the weight of the other edges is 1;

Case 2: Edge (5, 6) has a weight of 2, whereas the other edges have a weight of 1. The weighted-group-density algorithm is applied in the above cases, and the following results are obtained.

In case 1, the partitioning process can be plotted as a tree diagram as shown in Figure 2. The algorithm divides node 5 into group (1, 2, 3, 4), and obtains the maximum modularity in the eighth round, which is 0.4231. Thus, the weighted group density of group (1, 2, 3, 4, 5) is 0.7069, and the weighted group density of community (6, 7, 8, 9, 10) is 0.6853.

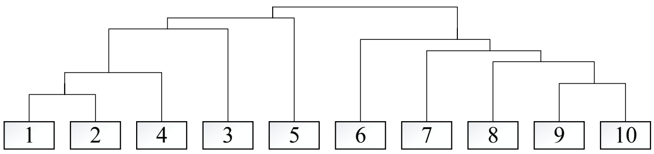

Figure 2. Topology of a toy network in Case 1

In case 2, edge (5, 6) has a weight of 2, and the partitioning process can be plotted as a tree as shown in Figure 3. The algorithm divides node 5 into group (6, 7, 8, 9, 10), and obtains the maximum modularity 0.4112. Now the weighted group density of community (1, 2, 3, 4) is 0.6323, and the weighted group density of community (5, 6, 7, 8, 9, 10) is 0.7448.

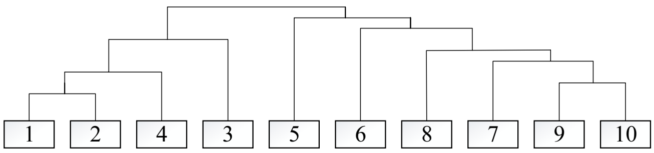

Figure 3. Topology of a toy network in Case 2

It can be concluded that node 5 is more inclined to select the community with higher weights, which is basically consistent with the experimental hypothesis. By adding edge weights and using community detection algorithm based on weighted group density, the problem of marginal node partition in simple networks can be well solved. When the network changes, the static community detection algorithms cannot be directly conducted on the new time slice but needed to consider the network structure at the previous time and the current network situation simultaneously. Moreover, in order to achieve a smooth transition effect, network partition cannot be directly converted from one snapshot to the next. By employing a threshold to determine the minimum network changes and combining with the community structure of the previous time, the weighted-group-density algorithm for local partition is proposed. The pseudo-code of the proposed algorithm is described in Algorithm 1. Figure 4 shows an example of dynamic changes from time (a) to time (b).

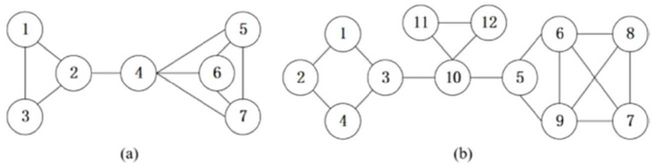

Figure 4. Sample networks

Algorithm 1. Weighted-Group-Density Based Community Discovery Algorithm

Input: original network G= {*n* nodes, *m* edges}, original community *Cr*, newly added *x* pieces of change data e_x, threshold *th*

Output: Results of network partitioning in each phase and the results

for $i = 1, 2, \ldots, x$ **do**
if $type\text{-}e_i$ = reduce **do**
$C_i = e_i$
if e_i in one community **do**
calculate *averD, delGD*
if *delGD*>*averD**(1+threshold) **do**
remove e_i **or** reduce weight
C_r remove C_i
C_r append n_i in C_i
$C_r = GroupDensity(C_r)$
else do
remove e_i **or** reduce weight
end if
else do
remove e_i **or** reduce weight
end if
if create island **do**
delete n_i
end if
else if *type-ei* = add **do**
$C_1 = e_{i1}$, $C_2 = e_{i2}$
if e_i not in one community **do**
calculate *average GD*
calculate GD_1, GD_2
if max$(GD_1,GD_2) > averaged \,||\, |GD_1 – GD_2| > (1\text{-}th) * averD$ **do**
add e_i **or** add weight
else if $|GD_1 – GD_2| > (1+th)*averD$
merge C_1, C_2
add e_i **or** add weight
else do
C_r remove C_1, C_2
C_r append n_i in C_1, C_2
add e_i **or** add weight
$C_r = GroupDensity(C_r)$
end if
else do
add e_i **or** add weight
end if
end if
end for

Conducting the proposed algorithm on the sample network, the dynamic changes and data processing type during each round of community detection are recorded in detail. After the algorithm finished, the modularity of this round process and the average weighted group density are shown in Table 1.

Table 1. Processing flow of the proposed algorithm on a toy network

Node	Type	Process Type	Modularity	<*WGD*>
2, 4	del	unchanged	0.4444	0.9453
1, 4	add	Local repartition	0.355	0.6022
7, 5	del	Local repartition	0.3642	0.5881
12, 10	add	Local repartition	0.4650	0.6798
1, 3	del	Local repartition	0.4259	0.6520
9, 7	add	Community Merge	0.4050	0.6642
7, 4	del	Local repartition	0.4630	0.4720
10, 5	add	Community Merge	0.3550	0.3882
4, 5	del	Local repartition	0.4383	0.4056
9, 5	add	Local repartition	0.4050	0.4022
6, 4	del	unchanged	0.5000	0.5926
8, 6	add	Community Merge	0.4850	0.6087
7, 8	add	unchanged	0.4670	0.6181
9, 8	add	unchanged	0.4479	0.6252
11, 10	add	Local repartition	0.5000	0.6781
10, 3	add	Local repartition	0.4617	0.4913
9, 6	add	unchanged	0.4511	0.4968
3, 4	add	unchanged	0.4785	0.5149
11, 12	add	unchanged	0.5069	0.5405

Note: <*WGD*> represents the average weighted-group-density.

Table 1 shows that out of 19 partitions there are 9 local repartitions, 3 community mergers and 7 network structure unchanged, which have less complexity than the static algorithm using snapshots directly after each round of change. At the same time, the network modularity after each round of change is recorded, and the modularity during the whole process keeps above 0.35, which indicates that the partition result of the algorithm also maintains a preferable level. After recording the final partition result of the proposed algorithm, its outcome is consistent with the community structure using weighted-group-density algorithm directly on the changed network of Figure 4(b), that is, three communities are obtained, which verifies the accuracy of the proposed algorithm. Meanwhile, the proposed algorithm saves the change trajectory of the whole network and provides fast and reliable data for subsequent extensibility analysis.

When the network is changing, the proposed algorithm proceeds according to the changing type and then checks whether the changes are in the current community among total *C* communities, which needs the complexity of $O(C)$. If the re-dividing process occurs, the algorithm needs complexity of $O(C^2 n)$. The total processing *x* changes require complexity of $O(xC^2 n)$, which is acceptable.

4 Experimental Analysis

'*A Song of Ice and Fire*', a fantasy novel written by American writer George Martin, focuses on a series of court struggles and magical adventures in the fictional magical land of Westeros. It is planned to consist of seven volumes in total and has been published its five volumes so far. In 2011, HBO adapted the book into the TV series ‘Game of Thrones’, which received extensive attention.

Andrew Beveridg [15], a professor at McAllister University, collates the data of this book by treating each character as a node and adding an edge between two nodes if two characters appear between 15 words and adding weight to the edge if one already exists. With a total of seven weighted network datasets including volume 1 to volume 5 respectively, collection of volume 4 and volume 5, the whole book is constructed. The statistics of the five volumes are shown in Table 2.

Table 2. The statistics of five real-world datasets

Volume	Nodes	Edges	Average degree
V_1	187	684	7.3155
V_2	259	775	5.9846
V_3	303	1008	6.6535
V_4	274	682	4.9781
V_5	317	760	4.7950

To compare the performance of the proposed algorithm, here we employ the Label Propagation Algorithm (LPA) [16] as a benchmark. LPA algorithm is widely used to detect communities in large-scale networks. The main idea is to initialize each node and give it a unique label, and each node updates its own label based on the label that the maximum number of their neighbors have in one iteration. The algorithm stops when the label is no longer updated, and the nodes with the same last label belong to the same community. The comparison of the proposed algorithm with LPA algorithm on the 5 datasets is shown in Table 3.

Table 3. Comparison with LPA algorithm on the “*A Song of Ice and Fire*” datasets

	V_1	V_2	V_3	V_4	V_5
$\boldsymbol{Q_{\mathrm{WGD}}}$	0.4617	0.5618	0.5830	0.6439	0.6719
$\boldsymbol{Q_{\mathrm{LPA}}}$	0.4667	0.5713	0.5870	0.6142	0.6525

Note: Q_{WGD} represents the modularity obtained by the proposed WGD algorithm, Q_{LPA} is the modularity by the LPA algorithm. V_1 to V_5 represents the network at Volume 1 to Volume 5, respectively.

Table 3 shows that the proposed Weighted-Group-Density algorithm outperforms the classic LPA algorithm, especially. Although the performance is close to LPA algorithm at the previous volumes, when conducting on volume 4 and volume 5, the modularity obtained by WGD algorithm is much higher than that of LPA algorithm. There are several reasons for this comparison result. Firstly, the initial relationships of characters are not very clear, especially some characters are playing bridge role in the seven kingdoms. Secondly, the connections among the characters are collected by the co-occurrence within 15 words, which may lead to minor unreasonable edges. Thus, the proposed algorithm is not outstanding in the previous volumes, and show superiority in the latter volumes. In addition, to eliminate this bias, we have also conducted a comparison experiment on another five real-world datasets, the statistics are shown in Table 4.

Table 4. The statistics of five real-world datasets

Dataset name	Nodes	Edges	Average degree
Southern women’s activity network [17]	32	89	5.5625
Dolphin social network [18]	62	159	5.1290
Scottish enterprise network [19]	217	348	3.2074
European airport network [20]	232	701	6.0431
Scientist cooperation network [21]	379	914	4.8232

Likewise, we compare the modularities obtained by the two algorithms, and the result is shown in Table 5.

Table 5. Modularity Comparison with LPA algorithm on five real-world datasets

Dataset name	Q_{WGD}	Q_{LPA}
Southern women’s activity	0.3110	0.3249
Dolphin social network	0.5153	0.5253
Scottish enterprise network	**0.8210**	0.7676
European airport network	**0.3752**	0.2114
Scientist cooperation network	**0.6641**	0.5679

As shown in Table 5, with the size of the dataset increasing, the modularity obtained by the proposed algorithm is significantly greater than that of the LPA algorithm, as marked in boldface. In above, the proposed algorithm is capable of dealing with detecting communities in social networks, especially when the network is with more nodes and edges.

4.1 Experiment on Networks from V_1 to V_2

In this paper, the dynamic network datasets are constructed by randomly sorting the difference between volumes, which are Volume 1 to Volume 2 network, volume 2 to volume 3 network and volume 3 to Collection of volume 4 and volume 5 network.

Initialize the two networks, and compares the differences. As shown in Table 2, volume 1 has 187 nodes and 684 weighted edges, while volume 2 has 259 nodes and 775 weighted edges. The comparison shows that the nodes’ increment rate in the network is 65.25%, the nodes deletion rate is 51.87%, the edges increment rate is 81.29%, and the edges deletion rate is 85.53%. In order to simulate the dynamic changes of the network, all the added and deleted edges are marked, and the order is disrupted. One edge is randomly selected as input at a time. A total of 459 weighted edges were deleted and 126 edges were reduced in weight. Simultaneously a total of 550 weighted edges were added, and the weights of 80 existing edges were increased, amounting to a totally of 1215 dynamic changes. Records regarding the results of each partition and calculations of the modularity of each round to obtain the overall change trend are shown in Figure 5.

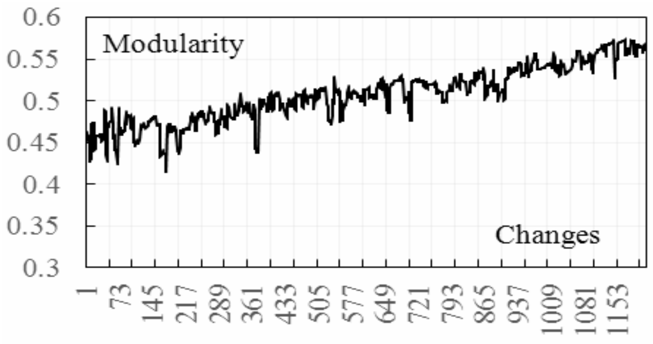

Figure 5. Experimental results of Volume 1 to Volume 2 obtained by the proposed algorithm

According to the changing trend of the whole network, with the input data of each round, the network structure changes correspondingly. In the dynamic process, the modularity of each partition result of the proposed algorithm keeps above 0.4, which is comparatively good and stable. At the same time, the changing trend of the modularity of the proposed algorithm shows that the network structure tends to become more stable and obvious. We can conclude that during 1215 dynamic changes, there are 464 times local repartition, 751 times direct community merge or no modification of the network structure, and so obtain the local repartition rate 38%. Compared with repartitioning the whole network in each round, the algorithm divides the whole dynamic process more accurately at less cost and obtains the modularity of 0.5608 in the last round of partition. The modularity is only 0.1% less than the modularity 0.5618 obtained by employing static weighted-group-density algorithm directly, which proves the accuracy of the dynamic algorithm.

4.2 Experiment on Networks from V_2 to V_3

Form network Volume 2 to network Volume 3, the nodes' increment rate is 55.12%, the nodes deletion rate is 47.49%, the edges increment rate (weight increment) is 85.52%, and the edges deletion rate (weight decrement) is 75.35%. Conducting the proposed algorithm on this dynamic process, the changing trend is shown in Figure 6.

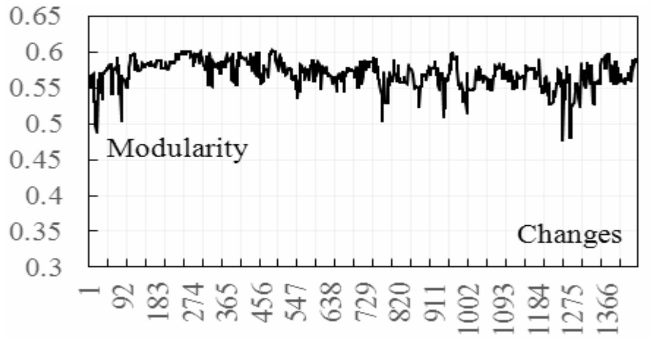

Figure 6. Experimental results of Volume 2 to Volume 3 based on the proposed algorithm

Analyzing the operating records of Volume 2 network to Volume 3 network, 567 times local repartition and 669 times directly community merge or no modification are obtained during 1446 dynamic changes, and the local repartition rate is 39%. The processing cost is still small in the whole operation process, the modularity of the algorithm is kept above 0.45, and the last round modularity is 0.5941. If the weighted-group-density algorithm is directly used for static partition, the modularity is 0.5744. Compared with the static algorithm, the result of the proposed algorithm is 1.9% higher than the Weighted-Group-Density algorithm in modularity, obtaining a better partition result. At the same time, analyzing the overall trend of the network changes, we can see that the overall chance process from volume 2 network to volume 3 network is more volatile, and the stability of the network structure is only slightly improved.

4.3 Experiment on Networks from V_3 to $V_{4\text{-}5}$

The nodes increment rate, nodes deletion rate, edges increment rate (weight increment) and edges deletion rate (weight decrement) in this network are 66.01%, 43.23%, 86.30% and 87.00%, respectively. Conducting the proposed algorithm on this dynamic network, the changing trend is shown in Figure 7.

As Volume 4-5 network is relatively large, there are more changes from Volume 3 to Volume 4-5, including a total of 823 times local repartitions and 1201 times directly community merge or no modification of the network structure during 2024 dynamic changes. The local repartition rate is 40%, and the overall modularity keeps above 0.5. The modularity of the proposed algorithm is 0.6744 in the result, which is 1.4% lower than that of the Weighted-Group-Density algorithm (0.6837), while it still maintains a pretty good result.

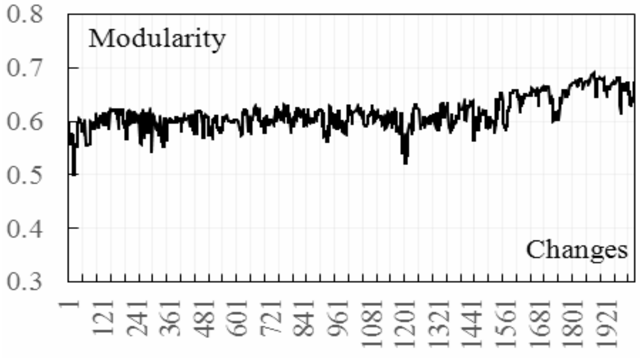

Figure 7. Experimental results of Volume 3 to Collection4/5 with the proposed algorithm

Finally, the processing results of the proposed algorithm on the dynamic network – *A Song of Ice and Fire* are shown in Table 6.

Table 6. Experimental results of the proposed algorithm

Result Parameters	V_1-V_2	V_2-V_3	V_3-$V_{4\text{-}5}$
Nodes increment	65.25%	55.12%	66.01%
nodes deletion	51.87%	47.49%	43.23%
edges increment	81.29%	85.52%	86.30%
edges deletion	85.53%	75.35%	87.00%
Average treatment	1.33 s	1.33 s	2.81 s
local repartition	464/1009	567/1447	823/2024
Final Q of the proposed Q	0.5608	0.5914	0.6744
Q_{WGD}	0.5618	0.583	0.6837

Table 6 demonstrates that the proposed algorithm can obtain comparatively better partition results with lower cost in each process for the network with large changes in the case of a single machine. Even in some cases, the community structure of the final network is superior to the direct partition results of the static algorithm, such as Volume 2 network to Volume 3 network. At the same time, the proposed algorithm processing is relatively smooth to record the dynamic change of the network via taking the dynamic change snapshot with minimum granularity and processing the algorithm on the minimum granularity, and is further capable of obtaining the structure and partition of the network under dynamic changes more intuitively and in more detail simultaneously.

5 Conclusion

To address the limitation of modularity, this paper analyzes the usability of group density in weighted and dynamic networks. Analogically, a dynamic community detection algorithm based on the proposed Weighted-Group-Density is proposed, and the accuracy and usability of this algorithm are theoretically verified. We collected the dynamic network datasets within five volumes of the fantasy novel '*A Song of Ice and Fire*', and the datasets are utilized to verify the performance of the proposed algorithm. Experimental results demonstrate that the proposed algorithm is capable and efficient.

The study of dynamic community detection is of remarkable significance in many other applications [22-24]. It is helpful to study the mechanism behind the community evolving in social networks and get more reasonable results from the various dynamic changes, which is helpful to optimize transfer routes in urban transportation systems, provide accurate candidates for recommending friends in online social networks or co-authorship networks and so on.

Acknowledgments

This work is partially supported by Liaoning Natural Science Foundation under Grant No.20170540320, the Doctoral Scientific Research Foundation of Liaoning Province under Grant No. 20170520358, the Fundamental Research Funds for the Central Universities under Grant No. N161702001, No. N1724 10005-2.

References

[1] M. E. J. Newman, Fast Algorithm for Detecting Community Structure in Networks, *Physical Review E*, Vol. 69, No. 6, Article number 066133, June, 2004.

[2] J. Hopcroft, O. Khan, B. Kulis, B. Selman, Tracking Evolving Communities in Large Linked Networks, *Proceedings of the National Academy of Sciences*, Vol. 101, No. Supplement 1, pp. 5249-5253, April, 2004

[3] G. Palla, A.-L. Barabasi, T. Vicsek, Quantifying Social Group Evolution, *Nature*, Vol. 446, No. 7136, pp. 664-667, April, 2007.

[4] D. Chakrabarti, R. Kumar, A. Tomkins, Evolutionary Clustering, *12th ACM SIGKDD International Conference on Knowledge Discovery and DATA MININg*, Philadelphia, PA, USA, 2006, pp. 554-560.

[5] Y. Chi, X. Song, D. Zhou, K. Hino, B. L. Tseng, Evolutionary Spectral Clustering by Incorporating Temporal Smoothness, *13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Jose, California, USA, 2007, pp. 153-162.

[6] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, B. L. Tseng, FacetNet: A Framework for Analyzing Communities and Their Evolutions in Dynamic Networks, *the 17th international conference on World Wide Web*, ACM, Beijing, China, 2008, pp. 685-694.

[7] A. Lancichinetti, S. Fortunato, Consensus Clustering in Complex Networks, *Scientific Reports*, Vol. 2, No. 13, Article number 336, March, 2012.

[8] T. P. Peixoto, Inferring the Mesoscale Structure of Layered, Edge-valued and Time-Varying Networks, *Physical Review E*, Vol. 92, No. 4, Article number 042807, October, 2015.

[9] R. Márquez, R. Weber, Overlapping Community Detection in Static and Dynamic Social Networks, *Proceedings of the Twelfth ACM International Conference on Web Search and*

Data Mining, Melbourne, Australia, 2019, pp.822-823
[10] X. Niu, W. Si, K. She, Evolutionary Community Detection in Dynamic Networks, *Journal of Software*, Vol. 28, No. 07, pp. 1773-1789, July, 2017.
[11] B. Guidi, A. Michienzi, G. Rossetti, Towards the Dynamic Community Discovery in Decentralized Online Social Networks, *Journal of Grid Computing*, Vol. 17, No. 1, pp. 23-44, March, 2019
[12] D. Jin, R. Li, J. Xu, Multiscale Community Detection in Functional Brain Networks Constructed using Dynamic Time Warping, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, Vol. 28, No. 1, pp. 52-61, January, 2020.
[13] Y. Zhang, B. Wu, N. Ning, C. Song, J. Lv, Dynamic Topical Community Detection in Social Network: A Generative Model Approach, *IEEE Access*, Vol. 7, pp. 74528-74541, June, 2019.
[14] D. Chen, Y. Wang, X. Huang, D. Wang, Community Detection Algorithm for Complex Networks Based on Group Density, *Journal of Northeastern University Natural Science*, Vol. 40, No. 2, pp. 186-191, February, 2019.
[15] A. Beveridge, J. Shan, Network of Thrones, *Math Horizons*, Vol. 23, No. 4, pp. 18-22, April, 2016.
[16] U. N. Raghavan, R. Albert, S. Kumara, Near Linear Time Algorithm to Detect Community Structures in Large-scale Networks, *Physical Review E*, Vol. 76, No. 3, Article number 036106, September, 2007.
[17] S. Brin, L. Page, The Anatomy of a Large-scale Hypertextual Web Search Engine, *Computer Networks and ISDN Systems*, Vol. 30. No. 1-7, pp. 107-117, April, 1998.
[18] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, S. M. Dawson, The Bottlenose Dolphin Community of Doubtful Sound Features a Large Proportion of Long-lasting Associations, *Behavioral Ecology and Sociobiology*, Vol. 54, No. 4, pp. 396-405, September, 2003.
[19] Y. Xu, L. Chen, Community Detection on Bipartite Networks Based on Ant Colony Optimization, *Journal of Frontiers of Computer Science and Technology*, Vol. 8, No. 03, pp. 296-304, March, 2014.
[20] A. Cardillo, J. Gomez-gardenes, M. Zanin, M. Romance, D. Papo, F. Pozo, S. Boccaletti, Emergence of Network Features from Multiplexity, *Scientific Reports*, Vol. 3, No. 1, Article number 1344, February, 2013.
[21] M. E. J. Newman, Finding Community Structure in Networks Using the Eigenvectors of Matrices, *Physical Review E*, Vol. 74, No. 3, Article number 036104, September, 2006.
[22] H. Yang, H. Huang, Sentiment Classification for Web Search Results, *Journal of Internet Technology*, Vol. 20, No. 7, pp. 2043-2053, December, 2019.
[23] S. Roy, R. Chopra, K. Lee, C. Spampinato, B. Mohammadi-ivatlood, Random Forest, Gradient Boosted Machines and Deep Neural Network for Stock Price Forecasting: A Comparative Analysis on South Korean Companies, *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 33, No.1, pp. 62-71, January, 2020.
[24] Y. Fan, X. Qi, B. Li, L. Liu, Fast Clustering-based Multidimensional Scaling for Mobile Networks Localization, *IET Communications*, Vol. 14, No. 1, pp. 135-143, January, 2020.

Biographies

Dongming Chen is a professor of software college at Northeastern University, China. He is a member of ACM, IEEE Computer Society, senior member of China Computer Federation (CCF), Senior member of China Institute of Communications (CIC). His research areas include complex networks, social network analysis and information security.

Xinyu Huang is a Ph.D. candidate in software college of Northeastern University, China. His research interests include complex network, multilayer network, community detection and social network analysis.

Yunkai Wang graduated in 2016 from Northeastern University at Qinhuangdao, China, and then continued with postgraduate education from 2016 to 2019 in software engineering. His research interests include complex network, community detection and social network analysis.

Dongqi Wang is a lecturer in Software College, Northeastern University, China. He was awarded Ph.D. degree in 2011. His research areas include network security and social network analysis. He has published more than 10 papers indexed by SCI, EI and ISTP. He is also a member of CCF and ACM.