LETTER

An Evolutionary Approach Based on Symmetric Nonnegative Matrix Factorization for Community Detection in Dynamic Networks

Yu PAN[†], ***Student Member***, **Guyu HU**[†a)], **Zhisong PAN**[†], **Shuaihui WANG**[†], ***and*** **Dongsheng SHAO**[†], ***Nonmembers***

SUMMARY Detecting community structures and analyzing temporal evolution in dynamic networks are challenging tasks to explore the inherent characteristics of the complex networks. In this paper, we propose a semi-supervised evolutionary clustering model based on symmetric nonnegative matrix factorization to detect communities in dynamic networks, named sEC-SNMF. We use the results of community partition at the previous time step as the priori information to modify the current network topology, then smooth-out the evolution of the communities and reduce the impact of noise. Furthermore, we introduce a community transition probability matrix to track and analyze the temporal evolutions. Different from previous algorithms, our approach does not need to know the number of communities in advance and can deal with the situation in which the number of communities and nodes varies over time. Extensive experiments on synthetic datasets demonstrate that the proposed method is competitive and has a superior performance.

key words: *dynamic networks, community detection, symmetric nonnegative matrix factorization*

1. Introduction

Community structure is considered as a significant property of complex networks, which can provide insight into the functionality of networks and uncover the underlying correlations. Recently, there has been an increasing number of studies in developing methods to detect community. However, these studies mainly focused on static networks and cannot detect communities in dynamic networks. Temporal networks are pervasive in the real-world, including social networks, person-to-person communication networks, and protein-protein relation networks. The analysis of temporal networks can reveal the important characteristics and potential rules within dynamic networks. Therefore, it has become increasingly important to develop methods to detect community structures in dynamic networks.

One of the main problems in detecting temporal communities is the instability of the solutions. Therefore, we cannot determine whether the change in community detection is caused by the evolution of the community or the instability of the algorithm. To address this problem, a large number of solutions have been proposed and their ultimate goal is to smooth the evolution of communities. Chakrabarti [1] proposed the evolutionary clustering framework, which assumes that sudden changes of clustering in a short time are probably caused by noise and abrupt changes of clustering are not expected. Based on the framework, several algorithms for dynamic community detection have been developed. A classic method, the FacetNet [2] algorithm, adopted a stochastic block model for detecting communities. The DYNMOGA [3] algorithm presented a multiobjective method that maximized the modular structures at the current time step, and minimized the differences between the community structures at a successive time step. Ma et al. [4] proposed two evolutionary nonnegative matrix factorization frameworks and then incorporated a priori information into the frameworks. Furthermore, Jiao et al. [5] proposed a method based on nonnegative matrix factorization (NMF) from a fully probabilistic perspective. Yu et al. [6] proposed an evolutionary clustering framework based NMF by combining the first-order varying information of the microstructure. Liu et al. [7] used symmetric nonnegative matrix factorization (SNMF) and introduced node weight matrices to improve the algorithmic performance. However, most of the proposed methods did not utilize the important prior information to improve the accuracy of community detection. Furthermore, they cannot detect the temporal communities and explore the evolutionary pattern of communities synchronously. And some of these methods cannot handle the situation in which the number of communities and nodes varies over time.

To address the above issues, in this paper, we propose a novel temporal community detection framework based on SNMF incorporating a priori information, named sEC-SNMF. Our main contributions in this work can be summarized as follows: Firstly, we use the results of the community partition at the previous time step as the priori information to modify the current network topology. In this way, we can reduce the influence of noise and improve the accuracy of community detection. Secondly, we introduce a community transition probability matrix to track and analyze the communities and temporal evolutions over time. Furthermore, owing to the property of our framework, which quantifies the degree of each individual's participation in the community, our framework can be easily extended to detect the overlapping community structure. More importantly, our approach can deal with the situation in which the number of nodes and communities changing over time. In addition,

Manuscript received March 7, 2019.
Manuscript revised July 10, 2019.
Manuscript publicized September 2, 2019.
[†]The authors are with Institute of Command and Control Engineering, Army Engineering University, Nanjing, 210007, China.
a) E-mail: guyu_hu36@163.com (Corresponding author)
DOI: 10.1587/transinf.2019EDL8046

Copyright © 2019 The Institute of Electronics, Information and Communication Engineers

our method does not need to know the number of communities in advance.

2. Evolutionary Approach Based on Symmetric Nonnegative Matrix Factorization

2.1 Notion

In this paper, we use $Tr(A)$ to denote the trace of a matrix, $\|A\|_F$ to denote the Frobenius norm of matrix A. In addition, the operator $(A)^T$ stands for matrix transposition.

Formally, the network at time t can be modelled as a graph $G_t = \langle V_t, E_t \rangle$, where V_t is the set of nodes and E_t is the set of edges. The dynamic network G is captured as a sequence of networks $G = \{G_1, G_2, \ldots, G_T\}$. In this paper, we assume that the network is unweighted and undirected. We use the adjacency matrix A to denote the dynamic network G, where $A_{i,j,t} = 1$ denotes there exists a link between v_i and v_j at time t, and $A_{i,j,t} = 0$ otherwise. Here, k represents the number of communities in the dynamic networks.

The evolutionary clustering framework assumes that most changes of links in short time periods could be caused by noise and abrupt changes of clustering are not expected [1]. The framework consists of two sub-costs: snapshot cost (CS) and temporal cost (CT). The CS represents how well the community structure fits the network at the current time step and the CT represents the degree of similarity between the community structures at the successive time step. The cost function is defined as follows:

$$Cost = \alpha \cdot CS + (1 - \alpha) \cdot CT \tag{1}$$

where $\alpha \in (0, 1)$ is a parameter used to control the contributions of the two objectives.

2.2 The Dynamic sEC-SNMF Model Formulation

Because of the excellent scalability and high computational efficiency, NMF algorithm is widely used for data clustering in machine learning. The NMF algorithm aims to find a pair of nonnegative low-dimensional matrices W and H to approximate the factorization of A. Given a matrix $A \in R^{n \times n}$, we use HH^T to approximate the factorization. The cost function is constructed by minimizing the Frobenius norm between A and HH^T.

$$\min_{H \in R^{n \times k}} \left\| A - HH^T \right\|_F^2 \quad s.t.\ H \geq 0 \tag{2}$$

The matrix $H \in R^{n \times n}$ is the community indicator matrix, which h_{ij} represents the membership strength of the i^{th} node belonging to the j^{th} community. The form HH^T represents the expected number of edges between the i^{th} node and the j^{th} node. For hard partition, nodes will be assigned to only one community with the highest probability. In this paper, our framework can detect overlapping communities by assigning nodes to more than one community whose community indicator is higher than a threshold.

In the evolutionary clustering framework, we first expect to maximize the clustering accuracy at the current time step. In this work, we assume that if two nodes belong to the same community, there is a high probability that they will be an edge between them. Therefore, we want the expected number of edges at time t to be as close as possible to the adjacent matrix, and the matrix A^t could be denoted by $H_t H_t^T$. The snapshot cost is defined as follows:

$$CS = \left\| A_t - H_t H_t^T \right\|_F^2 \tag{3}$$

To detect the temporal communities and explore the evolutionary pattern of communities synchronously, we introduce a community transition probability matrix $G \in R^{k \times k}$ to explore and trace the evolution of communities. The element g_{ij} represents the probability of a node transferring from the i^{th} community to the j^{th} community.

In the evolutionary clustering framework, we also expect to minimize the temporal cost to smooth the evolution between the community structures at a successive time step. Therefore, we assume that if a node at time $t-1$ belongs to the i^{th} community, it has a lower probability of changing its membership at time t. Therefore, $H_{t-1}G_t$ should be as close as possible to H_t. The temporal cost is defined as follows:

$$CT = \|H_{t-1}G_t - H_t\|_F^2 \tag{4}$$

Furthermore, to reduce the impact of noise, we use the result of the community partition at the previous time step as a priori information to modify the current network topology, which is defined as:

$$A_t^* = A_t - \gamma\left(A_{t-1} - H_{t-1}{H_{t-1}}^T\right) \tag{5}$$

where γ is the parameter to control the degree of priori information. Through formula (5), the smoothness between the current and historical network topology can be adjusted to reduce the impact of noise. The overall cost function is as follows:

$$\begin{cases} \min\limits_{H_t \geq 0, G_t \geq 0} \left\| A_t^* - H_t H_t^T \right\|_F^2 + \alpha \left\| H_{t-1}G_t - H_t \right\|_F^2 & if\ t \geq 2 \\ \min\limits_{H_t \geq 0} \left\| A_t - H_t H_t^T \right\|_F^2 & if\ t = 1 \end{cases} \\ s.t.\ (H_t)_{ij} \geq 0,\ (G_t)_{ij} \geq 0,\ \forall i,\ j \tag{6}$$

The final objective function (6) is not a convex function, so we optimize the objectives with respect to one variable while fixing the other variables. We use an iterative update algorithm and introduce the Lagrange multiplier ϕ and ε, respectively. The Lagrange function is written as follows:

For $t = 1$, we have

$$\begin{aligned} L_1 &= \left\| A_t - H_t H_t^T \right\|_F^2 + Tr(\phi H_t) \\ &= Tr\left(A_t A_t^T\right) - 2\,Tr\left(A_t H_t H_t^T\right) \\ &\quad + Tr\left(H_t H_t^T H_t H_t^T\right) + Tr(\phi H_t) \end{aligned} \tag{7}$$

and for $t = 2$

$$L_2 = \left\| A_t^* - H_t H_t^T \right\|_F^2 + \alpha \left\| H_{t-1}G_t - H_t \right\|_F^2$$

$$
\begin{aligned}
&+ Tr(\phi H_t) + Tr(\varepsilon G_t) \\
&= Tr\left(A_t^* \left(A_t^*\right)^T - 2A_t^* H_t H_t^T + H_t H_t^T H_t H_t^T\right) \\
&+ \alpha\, Tr\left(H_{t-1} G_t G_t^T H_{t-1}^T - 2H_{t-1} G_t H_t^T + H_t H_t^T\right) \\
&+ Tr(\phi H_t) + Tr(\varepsilon G_t) \qquad (8)
\end{aligned}
$$

The partial derivative of L_2 with respect to G_t and H_t are:

$$\frac{\partial L_2}{\partial G_t} = 2H_{t-1}^T H_{t-1} G_t - 2H_{t-1}^T H_t + \varepsilon \qquad (9)$$

$$\frac{\partial L_1}{\partial H_1} = -4A_1 H_1 + 4H_1 H_1^T H_1 + \phi \qquad (10)$$

$$\frac{\partial L_2}{\partial H_t} = -4A_t^* H_t + 4H_t H_t^T H_t - 2\alpha H_{t-1} G_t + 2\alpha H_t + \phi \qquad (11)$$

Using the Karush-Kuhn-Tucker (KKT) condition $\varepsilon_{ij} G_{ij} = 0$ and $\phi_{ij} H_{ij} = 0$, we obtain the following equations:

$$\left[2H_{t-1}^T H_{t-1} G_t - 2H_{t-1}^T H_t + \varepsilon\right]_{ij} (G_t)_{ij} = 0 \qquad (12)$$

$$\left[-4A_1 H_1 + 4H_1 H_1^T H_1 + \phi\right]_{ij} (H_1)_{ij} = 0 \qquad (13)$$

$$\left[-4A_t^* H_t + 4H_t H_t^T H_t - 2\alpha H_{t-1} G_t + 2\alpha H_t + \phi\right]_{ij} (H_t)_{ij} = 0 \qquad (14)$$

Finally, we obtain the updating formula of G_t and H_t as follows:

$$(G_t)_{ij} \leftarrow (G_t)_{ij} \left[\frac{\left(H_{t-1} H_t\right)_{ij}}{\left(H_{t-1}^T H_{t-1} G_t\right)_{ij}}\right] \qquad (15)$$

$$(H_1)_{ij} \leftarrow (H_1)_{ij} \sqrt{\frac{\left(A_1 H_1\right)_{ij}}{\left(H_1 H_1^T H_1\right)_{ij}}} \qquad (16)$$

$$(H_t)_{ij} \leftarrow (H_t)_{ij} \left[\frac{\left(2A_t^* H_t + \alpha H_{t-1} G_t\right)_{ij}}{\left(2H_t H_t^T H_t + \alpha H_t\right)_{ij}}\right]^{\frac{1}{4}} \qquad (17)$$

In this paper, we select k at each time step as follows:

$$k = \arg\min_{r^*} \sqrt{\left\| \sum_{i=1}^{r} \lambda_{iT} x_{iT} x'_{iT} \right\| \Big/ \|A_T\|} > \delta \qquad (18)$$

where λ_{iT} is the eigenvalue of matrix A, and x_{iT} is the corresponding eigenvector, δ is a parameter controlling the approximation. According to [4], we set $\delta = 0.55$ in this work. When nodes appear and vanish in networks, we fill 0 and delete the corresponding row to make the matrix become same size. When the number of communities in two successive time steps changes, we randomly initialize H_t instead of initializing H_t with H_{t-1}.

3. Experiment

In this section, we perform extensive experiments on three representative artificial datasets and compare the results with two well-known algorithms to test the validity of the sEC-SNMF algorithm. Firstly, we adopt two representative dynamic Griven Newman synthetic benchmark datasets proposed by Lin et al. [2] and Kim et al. [8]. The datasets have various evolution events that can verify whether our algorithm can accurately detect different evolutionary communities. Furthermore, we also adopt Power-Law networks generated by the LFR benchmark. The LFR benchmark extends the Girvan and Newman benchmark by introducing power-law degree distributions which can generate more realistic and large-scale benchmark data.

3.1 Evaluation Measures and Baseline Algorithms

To test the validity of the sEC-SNMF algorithm, two dynamic community detection algorithms are used as comparison algorithms: the DYNMOGA and FacetNet. In the following experiments, we set the parameters in our algorithm as $\alpha = 0.2$ and $\gamma = 0.1$. For the baseline algorithms, we use the parameters recommended by the authors.

In our experiments, we use the widely-used evaluation metric normalized mutual information (NMI) [9] to evaluate the performance, which is formally defined as:

$$
\begin{aligned}
&NMI(T, C) \\
&= \frac{-2 \sum_{i=1}^{F_T} \sum_{j=1}^{F_C} F_{ij} \log(F_{ij} \cdot n / F_{i\cdot} F_{\cdot j})}{\sum_{i=1}^{F_T} F_{i\cdot} \log(F_{i\cdot}/n) + \sum_{i=1}^{F_C} F_{\cdot j} \log(F_{\cdot j}/n)} \qquad (19)
\end{aligned}
$$

where T and C is the real community detection and the community partition of our algorithm, respectively. The matrix F is the confusion matrix. A higher value indicates a better performance.

3.2 The Evaluation Performance on Synthetic Dataset 1

The first synthetic dataset is introduced by Lin et al. [2]. The network consists of 128 vertices divided into 4 communities and each community contains 32 vertices. Every node has a fixed average degree, and contact z links with the nodes in other communities. In this work, we set the average degree as 20, and set $z = 5$, $z = 6$. The higher the value of z is, the fuzzier the community structure is. Moreover, $C\%$ of the vertices are moved among communities. For each fixed z, we randomly select 10% and 30% of the vertices to change their community at each time step. We consider 50 time steps in synthetic dataset 1.

To illustrate the performance of the sEC-SNMF algorithm, fifty independent runs are conducted on four networks and the averaged NMI values of three algorithms are as shown in Fig. 1. When the noise level increases, the performance of the two baselines decreases dramatically, and our algorithm consistently shows better performance at all noise levels.

3.3 The Evaluation Performance on Synthetic Dataset 2

The second synthetic dataset is introduced in [8]. It

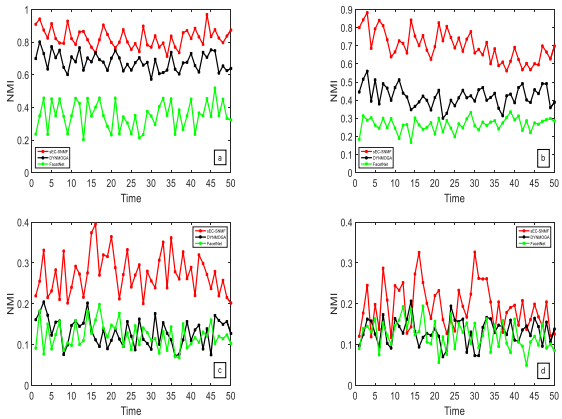

Fig. 1 Compare with DYNMOGA and FacetNet in terms of the NMI value of synthetic dataset 1 (a) z = 5, C = 10% (b) z = 5, C = 30% (c) z = 6, C = 10% (d) z = 6, C = 30%

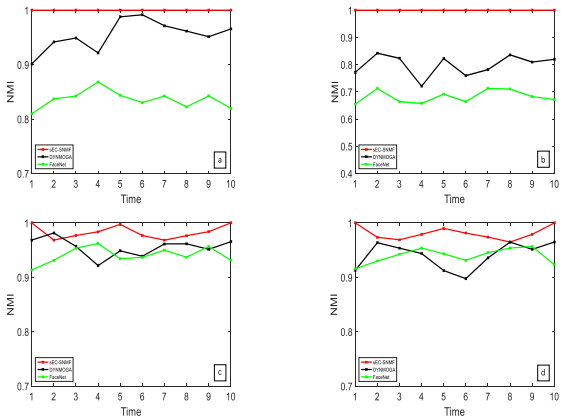

Fig. 2 Compare with DYNMOGA and FacetNet in terms of the NMI value of synthetic dataset 2 (a) SYN-FIX, z = 3 (b) SYN-FIX, z = 5 (c) SYN-VAR, z = 3 (d) SYN-VAR, z = 5

consists of two kinds of datasets. The first network is the SYN-FIX network which consists of 128 vertices divided into 4 communities with 32 vertices each equally. The number of communities is fixed. Similar to synthetic dataset 1, 3 vertices are randomly selected to change their community. The second network is the SYN-VAR network which consists of 256 vertices divided into 4 communities with 64 vertices and the number of communities is variable. The number of communities for the 10 timestamps is 4, 5, 6, 7, 8, 8, 7, 6, 5, and 4.

As can be seen from Fig. 2, it is obvious that the sEC-SNMF algorithm performs the best among all the methods on all datasets. It is worth noting that the NMI values obtained by our algorithm are always 1 for the two SYN-FIX datasets for all timestamps.

3.4 The Evaluation Performance on Synthetic Dataset 3

The third synthetic dataset is Power-Law networks which is generated by the LFR benchmark [10]. In Network 1, the numbers of nodes, links, and communities are 1858, 10635 and 26, respectively. The number of communities is fixed. To introduce dynamics, 6 vertices are randomly selected to join other communities at each time step. In Network 2, the number of nodes and links is 1000 and 7692, respectively. Analogously, we select two communities randomly to merge them into one community, and select one community to divide it into two communities. Overall, the number of communities for the 5 timestamps is 28, 27, 26, 27, and 28. The majority of the existing algorithms performed well on the Girvan and Newman benchmark, but poorly on the LFR benchmark. As can be seen from Fig. 3, all the results demonstrate that the sEC-SNMF algorithm is superior to all compared methods on the two Power-Law networks. The values of NMI obtained by the sEC-SNMF algorithm on the two datasets are all higher than 0.8 for 5 time steps.

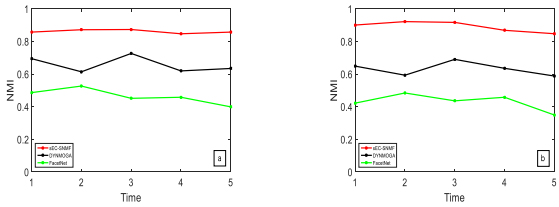

Fig. 3 Normalized mutual information values of synthetic dataset 3 (a) Network 1 (b) Network 2.

4. Conclusions

This paper presents a semi-supervised temporal community detection model based on SNMF. We use the results of the community partition at the previous time step as the priori information to modify the current network topology. Furthermore, a community transition probability matrix is introduced to track and analyze the temporal evolutions. Extensive experiments on synthetic datasets demonstrate that the sEC-SNMF algorithm has excellent performance in mining accurate community structures in dynamic networks.

References

[1] D. Chakrabarti, R. Kumar, and A. Tomkins, "Evolutionary clustering," ACM Sigkdd International Conference on Knowledge Discovery & Data Mining, 2006.

[2] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B.L. Tseng, "Analyzing communities and their evolutions in dynamic social networks," ACM Transactions on Knowledge Discovery from Data, vol.3, no.2, pp.1–31, 2009.

[3] F. Folino and C. Pizzuti, "An Evolutionary Multiobjective Approach for Community Discovery in Dynamic Networks," IEEE Trans. Knowl. Data Eng., vol.26, no.8, pp.1838–1852, 2014.

[4] X. Ma and D. Di, "Evolutionary nonnegative matrix factorization algorithms for community detection in dynamic networks," IEEE Trans. Knowl. Data Eng., vol.29, no.5, pp.1045–1058, 2017.

[5] P. Jiao, et al., "Exploring temporal community structure and constant evolutionary pattern hiding in dynamic networks," Neurocomputing, vol.3, no.14, pp.224–233, 2018.

[6] W. Yu, P. Jiao, W. Wang, Y. Yu, X. Chen, and L. Pan, "A novel evolutionary clustering via the first-order varying information for dynamic networks," Physica A: Statistical Mechanics and its Applications, vol.520, no.15, pp.507–520, 2019.

[7] H. Liu and L.M.Z. Yuan, "Community detection in temporal networks using triple nonnegative matrix factorization," DEStech Transactions on Computer Science and Engineering, vol.19, no.2, pp.68–76, 2017.
[8] M.S. Kim and J. Han, "A particle-and-density based evolutionary clustering method for dynamic networks," Proc. Vldb Endowment, vol.1, no.2, pp.622–633, 2009.
[9] A. Strehl and J. Ghosh, "Cluster ensembles — a knowledge reuse framework for combining multiple partitions," The Journal of Machine Learning Research, vol.3, no.3, pp.583–617, 2003.
[10] A. Lancichinetti and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities," Physical Review E., vol.80, no.1, 016118, 2009.