Hindawi
Complexity
Volume 2021, Article ID 7215888, 13 pages
https://doi.org/10.1155/2021/7215888

Research Article

Modeling Community Evolution Characteristics of Dynamic Networks with Evolutionary Bayesian Nonnegative Matrix Factorization

Wei Yu, [1,2] Xiaoming Li, [1,2] Huaming Wu, [3] Xue Chen, [4] Minghu Tang, [5] Yang Yu, [5] and Wenjun Wang [5]

[1] *School of International Business, Zhejiang Yuexiu University, Shaoxing 312069, China*
[2] *Shaoxing Key Laboratory of Intelligent Monitoring and Prevention of Smart City, Shaoxing 312069, China*
[3] *Center for Applied Mathematics, Tianjin University, Tianjin 300072, China*
[4] *Law School, Tianjin University, Tianjin 300072, China*
[5] *College of Intelligence and Computing, Tianjin University, Tianjin 300350, China*

Correspondence should be addressed to Xiaoming Li; lxm696@tju.edu.cn and Huaming Wu; whming@tju.edu.cn

Received 16 September 2021; Accepted 15 October 2021; Published 5 November 2021

Academic Editor: M. De Aguiar

In most cases, the block structures and evolution characteristics always coexist in dynamic networks. This leads to inaccurate results of temporal community structure analysis with a two-step strategy. Fortunately, a few approaches take the evolution characteristics into account for modeling temporal community structures. But the number of communities cannot be determined automatically. Therefore, a model, Evolutionary Bayesian Nonnegative Matrix Factorization (EvoBNMF), is proposed in this paper. It focuses on modeling the temporal community structures with evolution characteristics. More specifically, the evolution behavior, which is introduced into EvoBNMF, can quantify the transfer intensity of communities between adjacent snapshots for modeling the evolution characteristics. Innovatively, the most appropriate number of communities can be determined autonomously by shrinking the corresponding evolution behaviors. Experimental results show that our approach has superior performance on temporal community detection with the virtue of autonomous determination of the number of communities.

1. Introduction

Dynamic network analysis, as an important branch of complex network science, has attracted wide attention in recent years [1]. Temporal community structure analysis is one of the important research problems, which includes two subproblems: temporal community detection [1] and community evolution analysis [2]. However, community structure analysis of dynamic networks first detects the community structures and then analyzes the corresponding evolution pattern with a heuristic strategy in most cases. These works ignore the evolution characteristics of community structures when doing the temporal community detection. In fact, the block structures and evolution characteristics always coexist in dynamic networks. This leads to inaccurate results of temporal community detection. Therefore, it is very necessary to propose a model which describes the community structures with evolution characteristics for improving the accuracy of temporal community detection.

At present, temporal community detection has been widely concerned, which focuses on mining the meaningful block structures or functional modules hiding in the network snapshots of dynamic networks. In the beginning, two-stage approaches are introduced into temporal community detection, which first detect communities on each snapshot with a static method and then match them across different snapshots [3]. These types of approaches detect the communities of current snapshots ignoring the historical community structures from last snapshots, which take away the evolution characteristics of temporal community structures and are usually sensitive to noise. Later, evolutionary clustering- [4] based approaches are proposed, which effectively make up for this shortcoming by detecting the communities at the current snapshot with not only the current topology structure but also the previous community structures [5]. However, most of these works ignore determining the number of communities at each snapshot automatically and need to be specified in advance. In fact, the determination of the number of communities is a model selection problem, which is a common challenge for community detection. In addition, most of these works just focus on identifying temporal communities accurately but ignore analyzing the corresponding community evolution.

In addition, community evolution analysis exposes the evolution behaviors, which quantify the transition relationships of communities between adjacent snapshots, and helps to trace the change trends of dynamic networks. At first, the heuristic-based approaches [3] are proposed, which usually summarize the changing laws over time for analyzing the evolution pattern after detecting communities. Then, the feature engineering-based approaches [6] are proposed, which extract the evolution feature based on the detected temporal communities with a machine learning algorithm. The first two types of methods tend to analyze the evolution laws after detecting communities so that the results of the evolution analysis rely too much on the results of community detection. They ignore that community structures and evolution characteristics coexist in dynamic networks. Fortunately later, generative model-based approaches [7], which model the generative mechanism of community structure and community evolution synchronously, are able to make up for that shortcoming. However, most of these existing approaches just describe the evolution behaviors qualitatively but not quantitatively. And few of these approaches can deal with model selection problems automatically.

For addressing the above issues, we pay attention to model the community structures with evolution characteristics for boosting temporal community detection and determining autonomously the number of communities at each snapshot of dynamic networks synchronously. In this paper, a model called Evolutionary Nonnegative Matrix Factorization (EvoBNMF) is proposed based on a Bayesian probabilistic model. In detail, we introduce the evolution behaviors to model the evolution characteristics of community structures with Bayesian Nonnegative Matrix Factorization (BNMF) [8] in an evolutionary clustering framework [4]. Then, we develop a gradient descent algorithm to optimize the parameters of EvoBNMF by maximizing the posterior estimate. Innovatively, the most appropriate number of communities can be determined autonomously by shrinking the corresponding evolution behaviors. Last but not least, experimental results from synthetic networks and real-world networks over several state-of-the-art methods show that our proposed EvoBNMF has superior performance on temporal community detection with the virtue of autonomous determination of the number of communities. It is worthwhile to highlight several contributions of this work here:

(i) A model called Evolutionary Nonnegative Matrix Factorization (EvoBNMF) is proposed by modeling community detection with evolution characteristics for improving the performance of temporal community detection

(ii) The proposed EvoBNMF catches the most appropriate number of communities autonomously by shrinking the corresponding evolution behaviors of each snapshot network

(iii) An effective algorithm is developed to optimize the objective function of EvoBNMF, of which the time complexity can be degraded to be linear

(iv) Extensive experiments on synthetic and real-world dynamic networks demonstrate that our proposed EvoBNMF has superior performance on temporal community detection in comparison with state-of-the-art methods

2. Related Works

According to the core idea of the related works, the methods for temporal community detection can be divided into three categories: the snapshot matching-based methods [9, 10], the historical structural dependency-based methods [11, 12], and the community evolution model-based methods [13, 14].

The basic idea of the snapshot matching-based method is first to discover each network snapshot independently based on the static community detection algorithm and then to match the communities between snapshots based on some similarity strategy. For example, Seifikar et al. [9] proposed a new Louvain-based dynamic community detection algorithm that relied on the derived knowledge of the previous steps of the network evolution. Mishra et al. [10] proposed a tree-based community detection algorithm exploiting two important properties, connectedness and influence, for finding communities in the network. However, this kind of method pays more attention to discover the community structures on each network snapshot but ignores the smoothness of the evolution characteristics of dynamic network structure, splitting the evolution of community structure into multiple snapshots.

The basic idea of the historical structural dependency-based method is derived from the assumption of smoothness of the evolution of dynamic complex network structure. It is believed that the community structure of the current snapshot is evolved slowly from the previous network snapshot so that the result of the current community structure depends on the community structure of one or more historical network snapshots. For example, Yin et al. [12] proposed an efficient and effective multiobjective method via modifying the traditional evolutionary clustering framework and the particle swarm algorithm. Rossetti et al. [5] proposed an online incremental clustering dynamic community detection algorithm (Tiles) based on modularity incremental optimization. The calculation of the network substructure is local and the number of nodes and communities involved is limited, thus speeding up the updating efficiency. In addition, Wang et al. [15] constructed a novel similarity combining structural perturbation theory and network topology characteristics and proposed a dynamic community mining algorithm based on evolutionary clustering. All of these methods can avoid the matching of community structure between snapshots and incorporate the smoothness of dynamic community evolution.

The community evolution model-based method simulates the generation rules and the attribute characteristics hiding in the networks from the perspective of a dynamic complex network generation mechanism. At the same time, a reasonable network evolution mechanism is embedded to construct a parameterized dynamic community evolution generation model. Finally, the model parameters are solved to obtain the optimal community structure and the evolution pattern. For example, Ting et al. [13] proposed a novel framework for fitting the multilayer stochastic block model (SBM) that builds on multislice modularity maximization. It can discover a common community partition of all snapshots simultaneously. In addition, Yu et al. [16] constructed a matrix decomposition model containing an edge evolution time function for mining the evolution patterns in the microstructure (edge) of dynamic networks, which can be applied to structural trend prediction, link prediction, and anomaly detection of dynamic networks. Li et al. [14] proposed a method to learn the graph embedding and dynamic community detection via joint learning graph representation and NMF. In general, this kind of method abandons the two-step strategy in the snapshot matching-based method and continues the evolution smoothness property in the historical structural dependency-based method. It makes up for the lack of evolution mechanism in the historical structural dependency-based methods and gradually becomes the most popular method in the field of dynamic community detection.

3. Methodology

3.1. Notations. A dynamic network is usually cut as a series of network snapshots according to a fixed time window. It can be expressed as $\mathcal{G} = (V^{(t)}, E^{(t)}), t \in [1, 2, \ldots, T]$, where $V^{(t)}) = \{1, \ldots, N^{(t)}\}$ is the entity or node sets, $E^{(t)}$ is edge sets at snapshot t, and T is the number of snapshots. Network snapshot t is represented with an $N^{(t)} \times N^{(t)}$ adjacency matrix $A^{(t)}$, where the element at snapshot t is

$$A_{ij}^{(t)} = \begin{cases} 1, & (i,j) \in E^{(t)}, \\ 0, & (i,j) \notin E^{(t)}. \end{cases} \tag{1}$$

In addition, we summarize the main notations in Table 1.

As community number is unknown in most of the real cases, from the perspective of the Bayesian model, we assume that the community membership degree and the evolution tendency follow some probability distribution, respectively. At the same time, we assume that the parameters of the probability distribution of the evolution tendency follow a given prior distribution. Then, we construct the dynamic community evolution model EvoBNMF under the framework of the NMF and transform the optimization problem from maximizing the a posteriori probability of minimizing the negative logarithm of the posterior probability. Therefore, the tasks of the dynamic community evolution model EvoBNMF are summarized as follows:

(i) Input: the adjacency matrix sequence of the dynamic complex network $A(t)(t \in [1, T])$, and the hyperparameters a and b

(ii) Output: the dynamic community structures $C(t \in [1, T])$, the community evolution matrix sequence $Z(t)(t \in [2, T])$, and the number of communities $K(t)(t \in [1, T])$

3.2. EvoBNMF Model. Here, we design the generative graphical model of EvoBNMF (see Figure 1) with a Bayesian probabilistic model using the core idea of evolutionary clustering. For snapshot 1, as there is no historical structure information, EvoBNMF can be constructed similarly according to the static BNMF. Accordingly, the observed adjacency matrix $A^{(1)}$ is influenced by an unobserved expectation snapshot network $\widehat{A}^{(1)}$, of which element $\widehat{A}_{ij}^{(1)}$ denotes the expected link weight that takes place between i and j at snapshot 1. The expectation snapshot network can be composed of a basis matrix $W^{(1)} \in R_{+}^{N^{(1)} \times K^{(1)}}$ and a community membership matrix $H^{(1)} \in R_{+}^{K^{(1)} \times N^{(1)}}$ so that $A_{ij}^{(1)} \approx \widehat{A}^{(1)} = W^{(1)} H^{(1)}$, where $H_{kj}^{(1)}$ captures the propensity of node j belonging to community k and $K^{(1)}$ is the unknown number of communities. Similar to [8], we assume that $\widehat{A}_{ij}^{(1)}$ is drawn from a Poisson distribution with rate $\widehat{A}_{ij}^{(1)} = \sum_{k=1}^{K^{(1)}} W_{ik}^{(1)} H_{kj}^{(1)}$. And $W_{ik}^{(1)}$ and $H_{kj}^{(1)}$ are both drawn from a half-normal distribution with scale parameters $\beta^{(1)} = \{\beta_k^{(1)}\}$. In addition, we consider that the conjugate prior [17] of half-normal distribution is Gamma distribution. Therefore, $\beta^{(1)}$ is drawn from Gamma distribution with two hyperparameters $a^{(1)}$ and $b^{(1)}$. According to the graphical model in Figure 1, the model of snapshot 1 is the same as that of [8], and the corresponding posterior of the model at snapshot 1 is

$$P\left(W^{(1)}, H^{(1)}, \beta^{(1)} | A^{(1)}\right) = \frac{P\left(A^{(1)} | W^{(1)}, H^{(1)}\right) \cdot P\left(W^{(1)} | \beta^{(1)}\right) \cdot P\left(H^{(1)} | \beta^{(1)}\right) \cdot P\left(\beta^{(1)}\right)}{P\left(A^{(1)}\right)}. \tag{2}$$

The task of minimizing the negative log posterior, which is equivalent to the task of maximizing the posterior, can be regarded as the objective function of snapshot 1, and its specific form is as follows [8]:

Table 1: Notations.

Symbol	Definition
t	The snapshot label, and $t \in\in [1, T]$
$V^{(t)}$	The node sets of snapshot t, and $V^{(t)} = \{1, 2, \ldots, N^{(t)}\}$
$E^{(t)}$	The edge sets of snapshot t
$A^{(t)}$	The adjacent matrix at snapshot t, and $A^{(t)} \in R_{+}^{N^{(t)} \times N^{(t)}}$
$W^{(t)}$	The basis matrix at t, and $W^{(t)} \in R_{+}^{N^{(t)} \times K^{(t)}}$
$H^{(t)}$	The community membership matrix at t, and $H^{(t)} \in R_{+}^{K^{(t)} \times N^{(t)}}$
$Z^{(t)}$	The evolution behavior matrix at snapshot t, and $Z^{(t)} \in R^{K^{(t)} \times K^{(t-1)}}$
$C^{(t)}$	The community label matrix of snapshot t
$K^{(t)}$	The number of communities of snapshot t

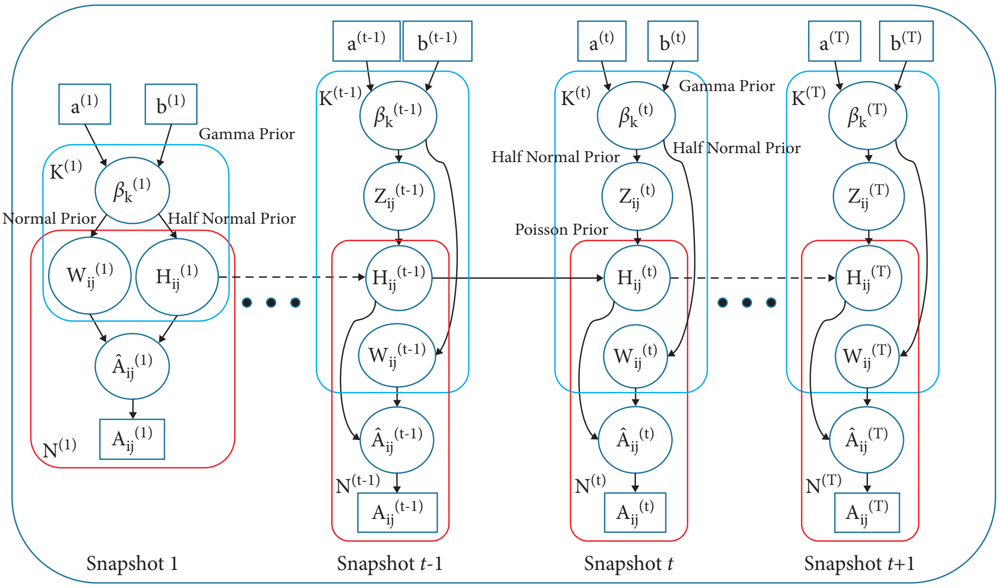


Figure 1: Graphical model of the proposed EvoBNMF.

$$\begin{aligned} L^{(1)} &= -\log P\left(A^{(1)} \mid \widehat{A}^{(1)}\right) - \log P\left(W^{(1)} \mid \beta^{(1)}\right) \\ &\quad - \log P\left(H^{(1)} \mid \beta^{(1)}\right) - \log P\left(\beta^{(1)}\right) \\ &= -\sum_{i=1}^{N^{(1)}} \sum_{j=1}^{N^{(1)}} \left(A_{ij}^{(1)} \log \frac{A_{ij}^{(1)}}{\widehat{A}_{ij}^{(1)}} + \widehat{A}_{ij}^{(1)} - A_{ij}^{(1)} \right) \\ &\quad + \frac{1}{2} \sum_{k=1}^{K} \left(\sum_{i=1}^{N^{(1)}} \beta_k^{(1)} W_{ik}^{(1)2} + \sum_{j=1}^{N^{(1)}} \beta_k^{(1)} H_{kj}^{(1)2} \right) \\ &\quad + \sum_{k=1}^{K} \left(\beta_k^{(1)} b^{(1)} - \left(a^{(1)} - 1\right) \log \beta_k^{(1)} \right) \\ &\quad - 2N \log \beta_k^{(1)} \Big) + \mathscr{C}, \end{aligned} \tag{3}$$

where $\mathscr{C}$ denotes a constant.

For the case of snapshot $t\,(2 \le t \le T)$, similarly, the observed adjacency matrix $A^{(t)}$ is influenced by an unobserved expectation snapshot network $\widehat{A}^{(t)}$, of which element $\widehat{A}_{ij}^{(t)}$ denotes the expected link weight that takes place between i and j at snapshot t. The expectation snapshot network can be composed of a basis matrix $W^{(t)} \in R_{+}^{N^{(t)} \times K^{(t)}}$ and a community membership matrix $H^{(t)} \in R_{+}^{K^{(t)} \times N^{(t)}}$ so that $A_{ij}^{(t)} \approx \widehat{A}^{(t)} = W^{(t)} H^{(t)}$, where $H_{kj}^{(t)}$ captures the propensity that node j belongs to community k and $K^{(t)}$ is the unknown number of communities. Differently, we consider the historical structure information into the model of the current snapshot according to the core idea of evolutionary clustering. In addition, we introduce an evolution matrix $Z^{(t)}$ to model the evolution behaviors of communities synchronously. The element $Z_{lk}^{(t)}$ denotes the propensity that nodes of community l of snapshot $t-1$ transfer into community k of snapshot t. Here, we think that the current community membership $H^{(t)}$ is evolved from $H^{(t-1)}$ by modeling some evolution behaviors $Z^{(t)}$ and introduce a penalty term to force that $H^{(t)} \approx \widehat{H}^{(t)} = Z^{(t)T} H^{(t-1)}$. In detail, we assume that $H_{jk}^{(t)}$ is drawn from a Poisson distribution with rate $\widehat{H}_{kj}^{(t)} = \sum_{l=1}^{K^{(t-1)}} Z_{lk}^{(t)} H_{lj}^{(t-1)}$, $W_{ik}^{(t)}$ and $Z_{lk}^{(t)}$ are both drawn from a half-normal distribution with scale parameters $\beta^{(t)} = \{\beta_k^{(t)}\}$, and $\beta^{(t)}$ is drawn from Gamma distribution with two hyperparameters $a^{(t)}$ and $b^{(t)}$ [17]. According to the graphical model in Figure 1, the joint distribution at snapshot t can be represented as follows:

$$\begin{aligned} &P\left(A^{(t)}, H^{(t-1)}, W^{(t)}, H^{(t)}, Z^{(t)}, \beta^{(t)}, \alpha\right) \\ &= P\left(A^{(t)} | W^{(t)}, H^{(t)}\right) \cdot P\left(H^{(t)} | H^{(t-1)}, Z^{(t)}, \alpha\right) \\ &\cdot P\left(Z^{(t)} | \beta^{(t)}\right) \cdot P\left(W^{(t)} | \beta^{(t)}\right) \cdot P\left(\beta^{(t)}\right), \end{aligned} \tag{4}$$

where α is a balance parameter. Therefore, the corresponding posterior is

$$\begin{aligned} &P\left(W^{(t)}, H^{(t)}, Z^{(t)}, \beta^{(t)} | A^{(t)}, H^{(t-1)}, \alpha\right) \\ &= \frac{P\left(A^{(t)}, H^{(t-1)}, W^{(t)}, H^{(t)}, Z^{(t)}, \beta^{(t)}, \alpha\right)}{P\left(A^{(t)}, H^{(t-1)}, \alpha\right)}. \end{aligned} \tag{5}$$

The task of minimizing the negative log posterior, which is equivalent to the task of maximizing the posterior, can be regarded as the objective function of snapshot t, and its specific form is as follows:

$$\begin{aligned} L^{(t)} = &-\log P\left(A^{(t)} | \widehat{A}^{(t)}\right) - \log P\left(H^{(t)} | H^{(t-1)}, Z^{(t)}, \alpha\right) \\ &- \log P\left(Z^{(t)} | \beta^{t)}\right) - \log P\left(\boldsymbol{\beta}^{(t)}\right). \end{aligned} \tag{6}$$

We assume that $A_{ij}^{(t)}$ is drawn from a Poisson distribution with rate $\widehat{A}^{(t)} = \sum_{k=1}^{K^{(t)}} W_{ik}^{(t)} H_{kj}^{(t)}$, $H_{ik}^{(t)}$ is drawn from a Poisson distribution with rate $\widehat{H}_{kj}^{(t)} = \sum_{l=1}^{K^{(t-1)}} Z_{lk}^{(t)} H_{lj}^{(t-1)}$, $W_{ik}^{(t)}$ and $Z_{lk}^{(t)}$ are both drawn from a half-normal distribution with scale parameters $\beta^{(t)} = \left\{\beta_k^{(t)}\right\}$, and $\beta^{(t)}$ is drawn from Gamma distribution with $a^{(t)}$, and $b^{(t)}$. Then, $L^{(t)}$ can be rewritten as

$$\begin{aligned} L^{(t)} = &\sum_{i=1}^{N^{(t)}} \sum_{j=1}^{N^{(t)}} \left(A_{ij}^{(t)} \log \frac{A_{ij}^{(t)}}{\sum_{k=1}^{K_{\text{initial}}} W_{ik}^{(t)} H_{kj}^{(t)}} \right. \\ &\left. + \sum_{k=1}^{K_{\text{initial}}} W_{ik}^{(t)} H_{kj}^{(t)} - A_{ij}^{(t)} \right) \\ &+ \alpha \sum_{i=1}^{N^{(t)}} \sum_{k=1}^{K_{\text{initial}}} \left(H_{ki}^{(t)} \log \frac{H_{ki}^{(t)}}{\sum_{l=1}^{K^{(t-1)}} H_{li}^{(t-1)} Z_{lk}^{(t)}} \right. \\ &\left. + \sum_{l=1}^{K^{(t-1)}} H_{li}^{(t-1)} Z_{lk}^{(t)} - H_{ki}^{(t)} \right) \\ &+ \sum_{l=1}^{K^{(t-1)}} \sum_{k=1}^{K_{\text{initial}}} \left(\frac{1}{2} \beta_k^{(t)} Z_{lk}^{(t)2} \right) - \frac{K^{((t-1))}}{2} \log \beta_k^{(t)} \\ &+ \sum_{i=1}^{N^{(t)}} \sum_{k=1}^{K_{\text{initial}}} \left(\frac{1}{2} \beta_k^{(t)} W_{ik}^{(t)2} \right) - \frac{N^{(t)}}{2} \log \beta_k^{(t)} \\ &+ \sum_{k=1}^{K_{\text{initial}}} \left(\beta_k^{(t)} b^{(t)} - \left(a^{(t)} - 1\right) \log \beta_k^{(t)} \right) + \mathscr{C}, \end{aligned} \tag{7}$$

where $\mathscr{C}$ is a constant.

3.3. Updating Rules. For snapshot 1, EvoBNMF is reduced to BNMF. So, the updating rules of the objective function $L^{(1)}$ are the same as those of [8]; in detail,

$$\begin{aligned} H^{(1)} \longleftarrow &\left(\frac{H^{(1)}}{W^{(1)T} 1 + \operatorname{diag}\left(\beta^{(1)}\right) H^{(1)}} \right) \\ &\odot \left(W^{(1)T} \frac{A^{(1)}}{W^{(1)} H^{(1)}} \right), \end{aligned} \tag{8}$$

$$\begin{aligned} W^{(1)} \longleftarrow &\left(\frac{W^{(1)}}{1 H^{(1)T} + W^{(1)} \operatorname{diag}\left(\beta^{(1)}\right)} \right) \\ &\odot \left(\frac{A^{(1)}}{W^{(1)} H^{(1)}} H^{(1)T} \right), \end{aligned} \tag{9}$$

$$\beta_k^{(1)} \longleftarrow \frac{N^{(1)} + a^{(1)} - 1}{1/2\left(\sum_i W_{ik}^{(1)2} + \sum_j H_{kj}^{(1)2}\right) + b^{(1)}}. \tag{10}$$

Similarly, for snapshot $t\,(t \in [2, T])$, we optimize equation (7) for $W^{(t)}$, $H^{(t)}$, $Z^{(t)}$, and $\beta^{(t)}$ with a gradient descent algorithm, and the updated rules are as follows:

$$\begin{aligned} H^{(t)} \longleftarrow &\left(\frac{H^{(t)}}{W^{(t)T} 1 + \alpha} \right) \odot \left(W^{(t)T} \frac{A^{(t)}}{W^{(t)} H^{(t)T}} \right. \\ &\left. + \alpha \frac{Z^{(t)T} H^{(t-1)}}{H^{(t)}} \right), \end{aligned} \tag{11}$$

$$\begin{aligned} W^{(t)} \longleftarrow &\left(\frac{W^{(t)}}{1 H^{(t)T} + W^{(t)} \operatorname{diag}\left(\beta^{(t)}\right)} \right) \\ &\odot \left(\frac{A^{(t)}}{W^{(t)} H^{(t)}} H^{(t)T} \right), \end{aligned} \tag{12}$$

$$\begin{aligned} Z_{lk}^{(t)} \longleftarrow &\left(\frac{Z^{(t)}}{Z^{(t)} \operatorname{diag}\left(\boldsymbol{\beta}^{(t)} + \alpha H^{(t-1)} 1\right.} \right) \\ &\odot \left(\alpha H^{(t-1)} \left(\frac{H^{(t)}}{Z^{(t)T} H^{(t-1)}} \right)^T \right), \end{aligned} \tag{13}$$

$$\beta_k^{(t)} \longleftarrow \frac{K^{(t-1)} + N^{(t)} + 2a^{(t)} - 2}{\sum_l^{K^{(t-1)}} Z_{lk}^{(t)2} + \sum_{l=1}^{N^{(t)}} W_{ik}^{(t)2} + 2b^{(t)}}. \tag{14}$$

We update iteratively $W^{(t)}$, $H^{(t)}$, $Z^{(t)}$, and $\beta^{(t)}$ according to the above rules until converges. We determine automatically the most appropriate number of communities of each snapshot with a statistical model selection method. In detail, we set a large value (e.g., $K_{\text{initial}}^{(t)} = N^{(t)}/4$) as the initial number of communities. After parameter optimization, we shrink $W^{(t)}$, $H^{(t)}$, $Z^{(t)}$ to $W^{(t)*}$, $H^{(t)*}$, and $Z^{(t)*}$ by removing the irrelevant rows or columns of which sum is zero or very close to zero. The pseudocode of the solving algorithm of EvoBNMF is presented in Algorithm 1. The returned community label vectors $C^{(t)}\,(t \in [1, T])$ are the results of temporal community detection. And the returned evolution matrices $Z^{(t)*}$ are the results of quantifying evolution behaviors.

According to Algorithm 1, iteratively updating of $H^{(t)}$ is most time-consuming. The time complexity of each iterative is $O(N^2 K_{\text{initial}} + N K_{\text{initial}}^2)$. Here, we set the average number of iterations as ρ, and the whole time complexity is about $O(\rho T (N^2 K_{\text{initial}} + N K_{\text{initial}}^2))$. As we know, dynamic networks are usually very sparse in real cases. Then, N^2 can be replaced with the average number of edges $\widehat{E}$ approximatively at each snapshot. In addition, K_{initial} can be ignored as it is usually much less than N. Naturally, the time complexity of the optimization algorithm of EvoBNMF can be degraded to $O(\rho T (\widehat{E} + N))$.

4. Experiments

In order to verify the principle and the effectiveness of EvoBNMF, we design the comparison experiments on synthetic networks and real-world networks. In this section, we mainly introduce the experimental settings, discuss the experimental results, and analyze the parameter sensitivity and algorithm convergence.

4.1. Settings

4.1.1. Datasets. We test the performance of our EvoBNMF on the eight dynamic networks. Four networks are generated according to SYN-FIX [18], and the other four networks are from real-world KIT-mail (https://i11www.iti.uni-karlsruhe.de/en/projects/spp1307/emaildata). We show their statistical information including the number of snapshots T, the average number of nodes $|\overline{V}|$, the average number of edges $|\overline{E}|$, and the average number of $\overline{K}$ in Table 2. The details are described as follows:

(1) SYN-FIX [19]: this type of dataset is generated one snapshot by one snapshot with Girvan Newman benchmark. During the specific generation process, the parameters of network properties are set as follows: the number of snapshots T is 10, the number of nodes N is 128, the number of communities K is 4, the mixed parameter Z is 3 which is used to control the degree of the noise, the average degree of the nodes d is 16 and 20, and the community transfer parameter NC is used to control the dynamic level of nodes moving from the current community to other communities, which is set at 10% and 30%.

(2) LFR [20]: the classical synthetic network LFR is mainly to describe the dynamics of networks by considering some community evolution events, which include Birth, Death, Growth, Contraction, Merging, and Splitting. Here, we select the Mergesplit event to generate datasets. We set different probabilities P_1 and P_2 during network generation to control the probability of community merging and splitting.

(3) KIT-Email: it is a mail communication network from the Information Department of Karlsruhe Institute of Technology (KIT) in Germany. The members are nodes, and the number of mail communication times is the weight of the edge. Different research groups are the corresponding communities. Here, the network data of 48 from September 2006 to August 2010 are divided into different dynamic complex networks. Specifically, we construct each snapshot by shading consecutive 2, 3, 4, and 6 months, respectively, and get four different dynamic networks.

4.1.2. Evaluation Metrics. The performance of community detection is evaluated with two widely used indexes: the Normalized Mutual Information (NMI) and Error Rate (ER) [7]. In detail,

$$\text{NMI} = \frac{2I(\widehat{G}, G)}{\mathscr{H}(\widehat{G}) + \mathscr{H}(G)}, \qquad \text{ER} = \left\| \widehat{G}\widehat{G}^T - GG^T \right\|_F^2, \tag{15}$$

where $\widehat{G}$ denotes the community structures detected from the algorithm and G denotes the ground truth. $\mathscr{H}(\widehat{G})$ and $\mathscr{H}(G)$ denotes the entropy of $\widehat{G}$ and G, and $I(\widehat{G}, G)$ denotes the mutual information between $\widehat{G}$ and G, respectively [18]. Here, the entropy and the mutual information are computed with the equation $\mathscr{H}(X) = \sum_{k=1}^{K} N_k \log N_k / N$, and $I(X;Y) = \sum_{k=1}^{K^X} \sum_{l=1}^{K^Y} N_{kl} \log N \cdot N_{kl} / N_k^X \cdot N_l^Y$, respectively, where N is the number of nodes, and K is the number of communities, respectively. NMI as an entropy measure restrained in $[0, 1]$ is usually used to measure the consistency between two partitions. ER is usually used to measure the difference between two different partitions, and the smaller it is, the better its performance is. Universally, ER tends to increase with the scale of networks.

And the accuracy of the autonomous determination of the number of communities is evaluated with KA [21]; in detail,

$$\text{KA} = 1 - \frac{\left\| C^* | - |\widehat{C} \right\|}{2\left| C^* \right|}, \tag{16}$$

where $|C^*|$ denotes the grand truth of the number of communities and $|\widehat{C}|$ denotes the number of communities detected by the methods.

4.1.3. Comparison Approaches. In this work, five state-of-the-art approaches are chosen for detecting communities as compared groups as follows:

(i) BNMF [8]: it is a two-steps strategy, which segments the snapshots into discrete time steps and community detection with the static Bayesian NMF on each snapshot, respectively. When the snapshot label $t = 1$, EvoBNMF is reduced to BNMF.

(ii) Dyluvain [3]: it optimizes the temporal modularity with a greedy heuristic method. The resolution

```
Initialize $W^{(t)}$, $H^{(t)}$, $Z^{(t)}$, $\beta^{(t)}$, where $t \in [1,T]$;
while not converge do
    Update $W^{(1)}$, $H^{(1)}$, $\beta^{(1)}$ according to equations (8)–(10);
for $t \in [2,T]$ do
    while not converge do
        $W^{(t)}$, $H^{(t)}$, $Z^{(t)}$, $\beta^{(t)}$ according to equations (11)–(14);
for $t \in [1,T]$ do
    shrink $W^{(t)}$, $H^{(t)}$, and $Z^{(t)}$ to $W^{(t)*}$, $H^{(t)*}$, and $Z^{(t)*}$;
    $C_i^{(t)} = \arg\max(H_{ik}^{(t)*})$;
return $C^{(t)}$, $Z^{(t)*}$.
```

Algorithm 1: EvoBNMF.

Table 2: Statistical information of dynamic networks.

Datasets	Networks	T	$\lvert\overline{V}\rvert$	$\lvert\overline{E}\rvert$	$\overline{K}$
SYN-FIX [18]	Net. 1	10	128	2048	4
	Net. 2	10	128	2374	4
	Net. 3	10	128	1977	4
	Net. 4	10	128	2419	4
LFR [20]	Net. 5	10	128	2048	4
	Net. 6	10	128	2374	4
	Net. 7	10	128	1977	4
	Net. 8	10	128	2419	4
KIT-mail	Net. 9	24	138	29481	23
	Net. 10	16	170	29963	25
	Net. 11	12	195	29788	25
	Net. 12	8	231	27883	27

parameter γ and couple parameter omiga are set to 1 and 0.5, which are the commonly used parameter settings in the related works.

(iii) PisCES [22]: it is a temporal community detection model based on global spectral clustering, which is derived from the idea of evolutionary clustering and degree modification. The parameter α is set to 0.1, and the maximum community number defaults to 1/10 of the number of nodes.

(iv) DYNMO [18]: it is a multiobjective approach based on evolutionary clustering, which is formalized as a multiobjective optimization problem to be optimized by a genetic algorithm.

(v) ESPRA [15]: it is an evolutionary clustering algorithm based on the fusion of structural perturbation and network topological features, which can automatically determine the number of communities. In this model, the smoothness balance parameter alpha is set to 0.8, and the perturbation and similarity information balance parameter β is set to 0.5.

4.2. Experimental Results

4.2.1. Illustrative Example. To clarify the working principle of EvoBNMF, we take an illustrative example on the results of Net. 1 in Figure 2. Due to space constraints, we just show the results of snapshots $1-3$. At snapshot 1, the learned matrices $W^{(1)}$ and $H^{(1)}$ are decomposed from the observed $A^{(1)}$. Obviously, there are just four columns of $W^{(1)}$ and four rows of $H^{(1)}$ have a high value. And the number of rows of $H^{(1)*}$ is the targeted number of communities after the adaptive compression of the rows with a low value. Then, the compressed $H^{(1)*}$ and the observed $A^{(2)}$ are both the input of the model at snapshot 2. For snapshot $t\,(t \geq 2)$, the matrices $W^{(t)}$ and $H^{(t)}$ are decomposed from the observed $A^{(t)}$ and the matrices $H^{(t-1)*}$ and $Z^{(t)}$ are decomposed from the $H^{(t)}$ synchronously in a unified model. It is worth noting that we can obtain the evolution matrices $Z^{(t)*}$ after the adaptive compression of the rows of $Z^{(t)}$. And the evolution matrices correspond to quantitative results of the evolution behavior of communities.

The community evolution matrix $Z^{(t)}$ mined from the dynamic network through EvoBNMF can represent its evolution pattern and reflect the community evolution relationship between adjacent snapshots. Because the scale of the evolution matrix of different snapshots is different, the row normalization of $Z^{(t)}$is carried out, so that $\forall l, \sum_k Z_{lk}^{(t)} = 1$. Then, $Z_{lk}^{(t)}$ can be regarded as the propensity that the node transfer from community C_l to community C_k between snapshot $t-1$ and snapshot t. In Figure 3, the community evolution relationship between adjacent snapshots of Net. 1 is given, and the shade of color in the figure represents the node transition probability between communities. In each subgraph, the vertical coordinate represents the community label C_l of

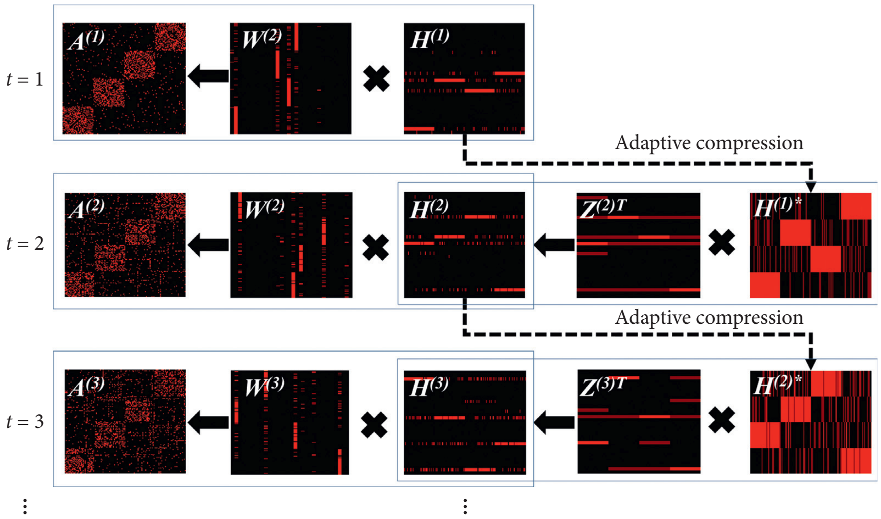


Figure 2: An illustration of EvoBNMF on network 1.

the current network snapshot, the horizontal coordinate represents the community label C_k of the next snapshot, and the shade of the color represents the transition probability, which corresponds to the value of $Z_{lk}^{(t)}$ of the community label in the current snapshot t. On the whole, the diagonals of these subgraphs always show a yellow (light) color, which means that most of the internal nodes of the communities are likely to remain in the current communities. This phenomenon reflects that the evolution of the dynamic network is slow.

It is worth noting that the subgraphs of $Z^{(6)}$, $Z^{(7)}$, $Z^{(8)}$, and $Z^{(9)}$ have taken place an anomaly about the evolution patterns. For example, in the subgraph of $Z^{(6)}$, the new community C_5 is derived when the network evolves from snapshot 6 to 7. And the probability that nodes transfer from community C_2 to community C_5 is relatively high, and the probability that nodes keep in community C_2 is obviously relatively low. Therefore, it can be speculated that community C_5 is split from community C_2. In addition, in the subgraph of $Z^{(7)}$, it can be found that community C_5 disappears again when the network snapshot evolves from 7 to 8. And its nodes have a high probability to move to community C_1. Similarly, it can be found community C_6 and C_5 appear in the subgraphs of $Z^{(8)}$ and $Z^{(9)}$ and then disappear. This phenomenon indicates that the evolution of community structure is unstable after snapshot 6. The Birth/Death of community structure is actually corresponding to the dynamic community evolution events, which is first defined by Palla et al. [23]. In real-world social networks, if there are a large number of similar community evolution events, it can indicate that there are real events behind them, which proves that EvoBNMF can be applied to event detection with real-world social networks.

In order to more vividly represent the evolution process of a dynamic community, corresponding to the community evolution of dynamic network Net. 1 in Figure 3, a visual schematic diagram of evolution and its transition over time is given in Figure 4. In Figure 4, the horizontal axis represents the network snapshot label t, the vertical axis represents the dynamic community label k, the colored circle represents different communities, the circle size represents the relative size of communities, and the dotted arrow represents the node transfer relationship between communities. The community evolution intensity in Figure 3 corresponds to the node transfer relationship between communities and the changing situation of communities in Figure 4. For example, in Figure 3, there is a high diagonal strength in many subgraphs, corresponding to Figure 4, and the community changes little over time in most cases. In addition, new communities appear in snapshots 7 and 9 and disappear again in snapshots 8 and 10, respectively. This phenomenon echoes strongly with Figure 4. In general, Figures 3 and 4 show the evolution pattern of a dynamic community and give a visual diagram of evolution, which can effectively deepen people's understanding of the evolution of temporal communities in the networks.

4.2.2. Temporal Community Detection. In order to investigate the effectiveness, we compare the accuracy of our proposed EvoBNMF with five state-of-the-art methods on temporal community detection, including BNMF [8], Dyluvain [3], PisCES [22], DYNMO [18], and ESPRA [15]. The hyperparameters are set as $\alpha = 0.2$, $a = 8$, and $b = 5$ in experiments.

At first, the results are shown in Table 3 over NMI, ER, and KA of the five methods on Net. $1-4$ of SYN-FIX. The best results, which are bolded out, demonstrate that Dyluvain and our proposed EvoBNMF are comparable in performance and both are better than others. The reason is that Dyluvain optimizes the temporal modularity with a greedy heuristic method and is suitable to the synthetic data SYN-FIX. In addition, the results of BNMF, which is the static version of EvoBNMF, are obviously lower than EvoBNMF. The phenomenon verifies the validity of the proposed EvoBNMF.

Furthermore, we show the results over NMI, ER, and KA of the five methods on network $5-8$ from top to bottom in Figure 5, respectively. These results are the average results of

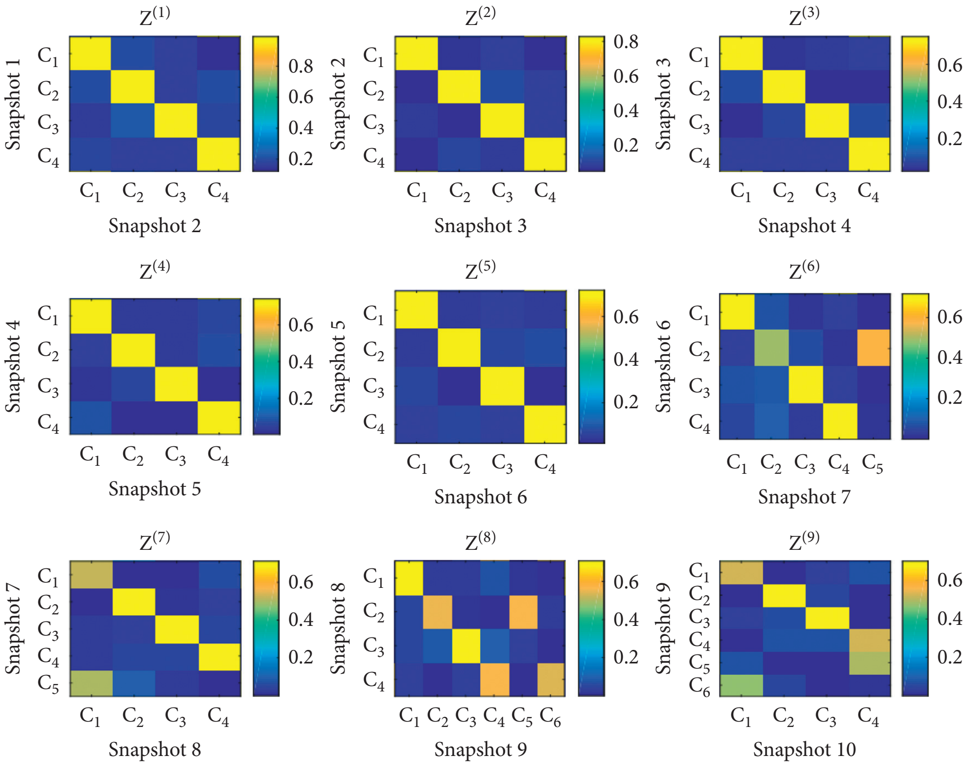


FIGURE 3: The propensity of the evolution $(Z^{(t)})$ between adjacent snapshots on Net. 1.

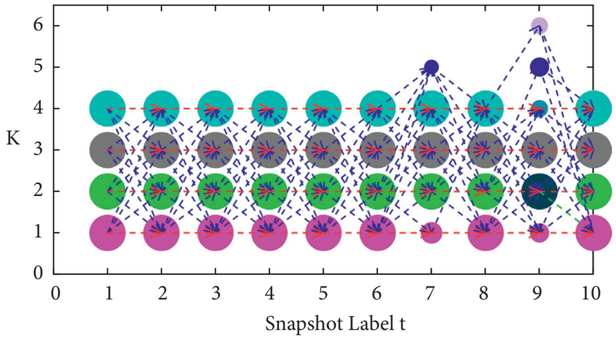
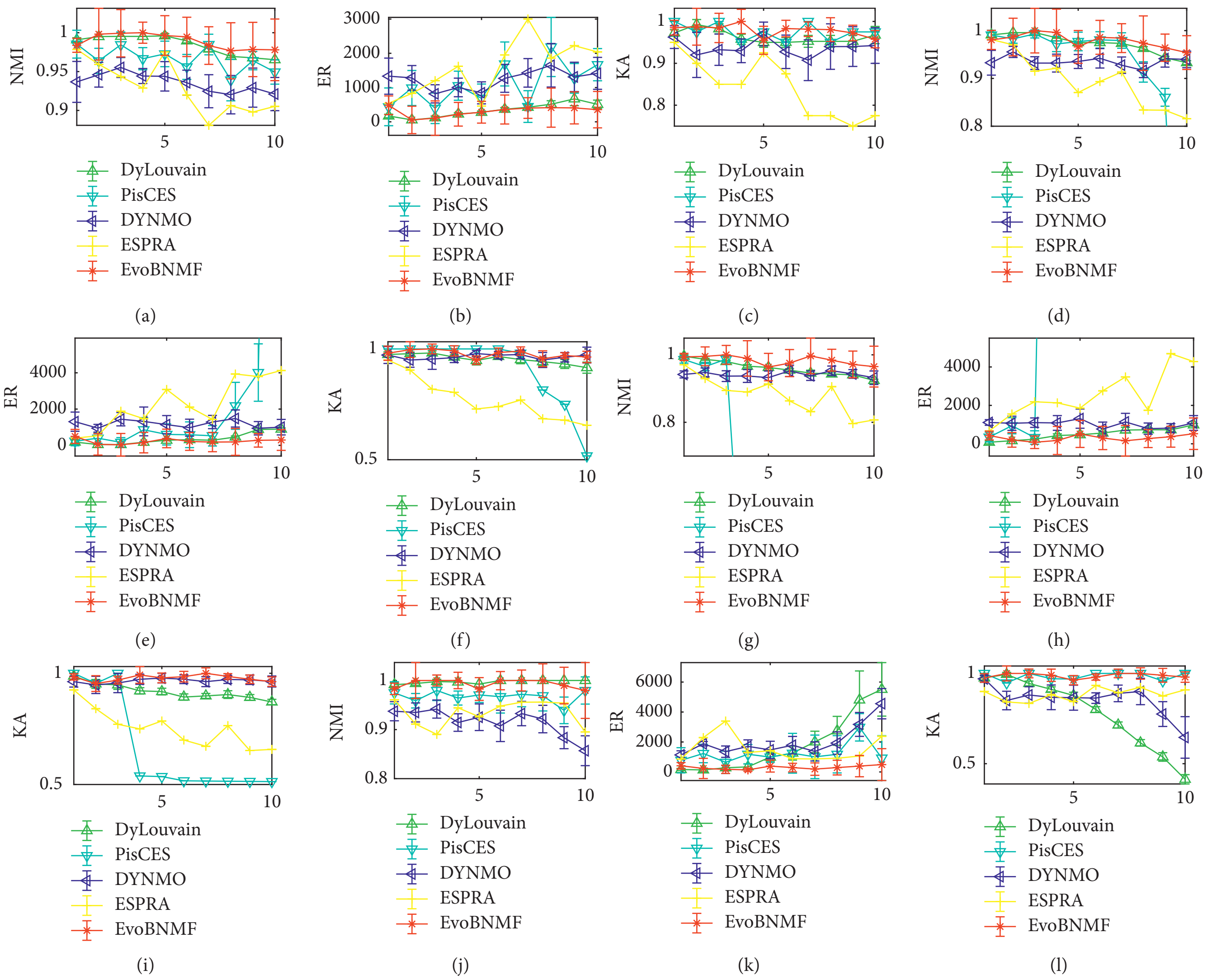


FIGURE 4: The visualization of the nodes transition among communities at snapshots of Net. 1.

TABLE 3: The comparison results of five methods on Net. 1 – 4.

Net.	Index	BNMF	Dyluvain	PisCES	DYNMO	ESPRA	EvoBNMF
	NMI	0.65	1	0.98	0.59	0.97	0.99
1	ER	2956.22	0	229.94	3256.91	138	61.56
	KA	0.44	1	1	0.38	1	1
	NMI	0.61	1	0.97	0.56	0.39	1
2	ER	3015.26	0	244.34	3340.43	4125	0
	KA	0.41	1	1	0.38	0.56	1
	NMI	0.26	0.44	0.30	0.23	0.28	0.45
3	ER	4189.56	3630.99	7615.98	4436.61	4467.80	3576.38
	KA	0.078	0.54	0.75	0.069	0.71	0.69
	NMI	0.39	0.97	0.92	0.41	0.29	0.92
4	ER	4285.79	128.96	559.14	4024.73	4942.20	500.30
	KA	0.29	1	1	0.31	0.61	0.99



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 5: The comparison result of five methods on Net. 5 – 8.

ten repetitions including the corresponding variance bar. In addition, the x-axis is the snapshot label t, and the y-axis is NMI or ER or KA values. From all the subfigures, DyLouvain and EvoBNMF have higher NMI and ER values, and DYNMO and EvoBNMF have higher KA values. This is a strong indication that the proposed EvoBNMF has superior performance not only on temporal community detection but also on autonomous determination of the number of communities.

Similarly, the subfigures in Figure 6 show the results over NMI, ER, and KA of the five methods on network 9 – 12 from top to bottom, respectively. From all the subfigures, we found that the results over NMI, ER, and KA of EvoBNMF have the highest accuracy in most cases, but not at the first snapshot. The main reason is that there is no historical structure information for the first snapshot, and EvoBNMF degenerates to BNMF. In addition, there is significantly improved accuracy from snapshot 1 to snapshot 2, which fully demonstrates the effectiveness of EvoBNMF.

4.2.3. Parameter Sensitivity and Algorithm Convergence. We test the sensitivity of the balance parameter α of EvoBNMF on Net. 9 – 12 over NMI by ranging $\alpha \in [0, 0.5]$ with a step length of 0.02. As shown in Figure 7(a), the performance of EvoBNMF is not sensitive when parameter $\alpha \geq 0.1$, which is the best on about 0.2.

In addition, we verify the convergence of EvoBNMF on Net. 12 with $\alpha = 0.2$. Figure 7(b) shows the convergence of $L^{(t)}$ at snapshots 2 – 5 of Net. 12. We find that the value of $L^{(t)}$ always tends to converge when the times of iterations n_{iter} are no more than 50, which demonstrates that the convergence rate is relatively fast.

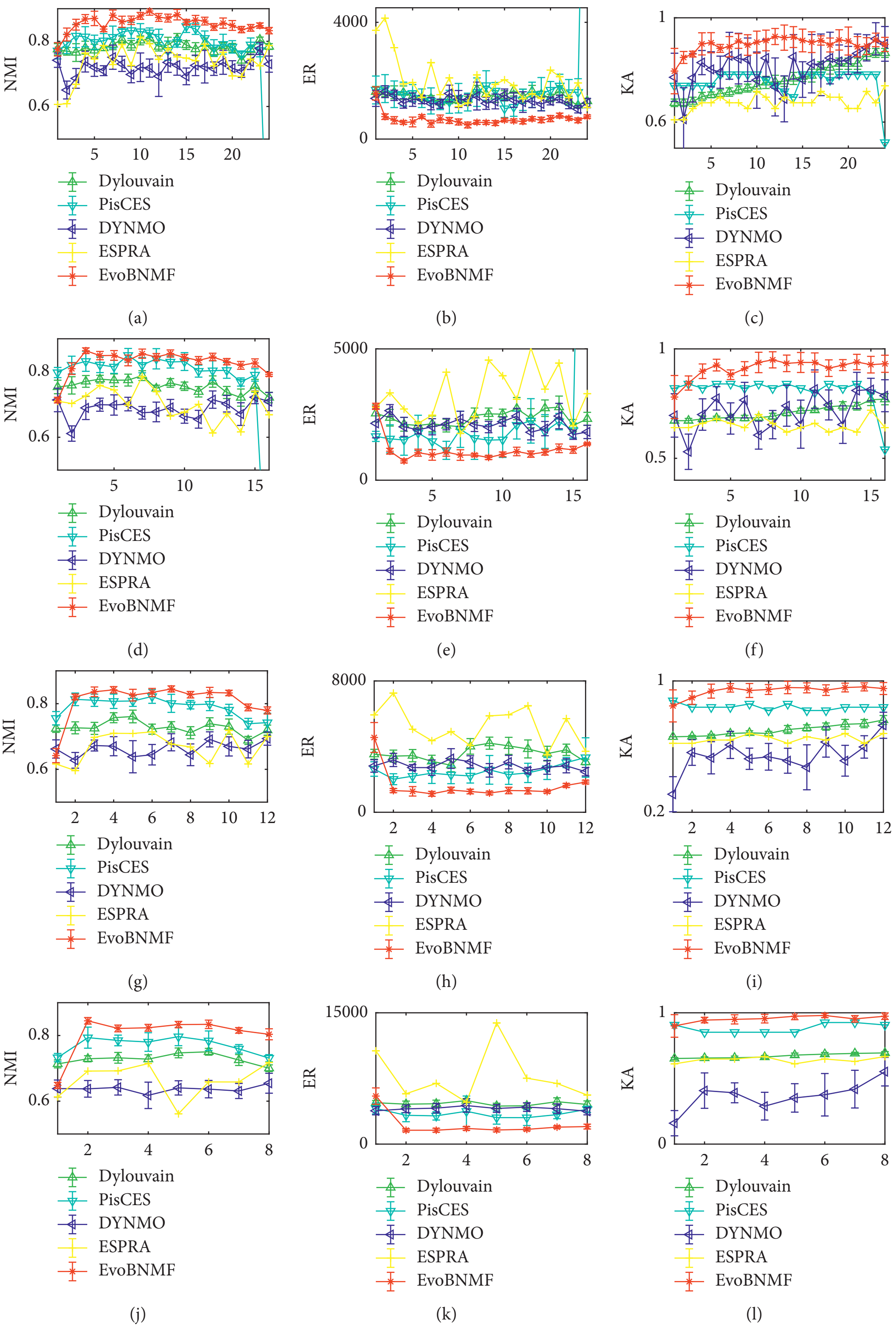
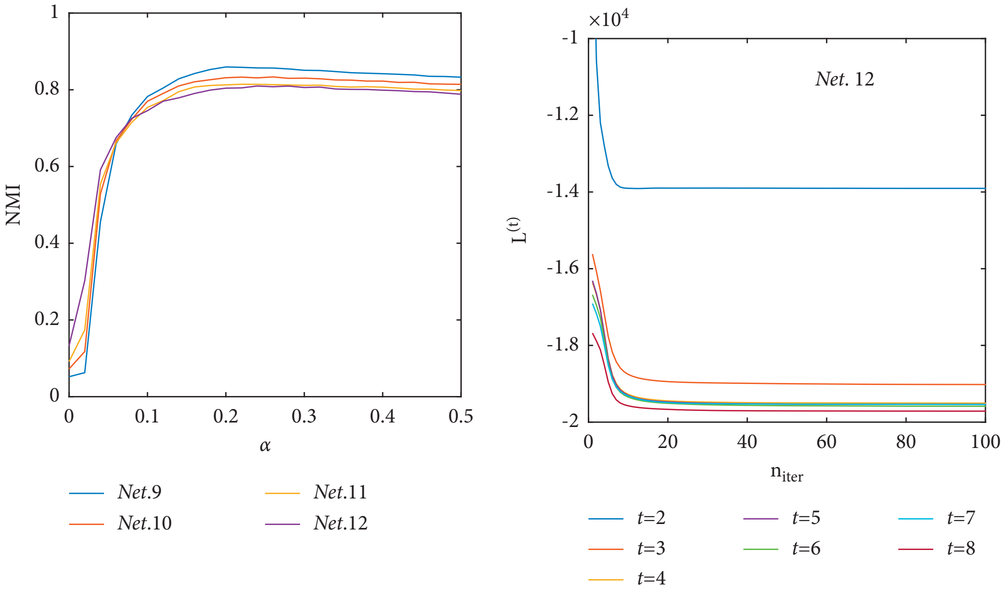


FIGURE 6: The comparison result of five methods on Net. 9 – 12.



(a) (b)

Figure 7: (a) Parameter analysis of the balance parameter α on Net. 9 – 12; (b) the convergence analysis of $L^{(t)}$ on Net. 12.

5. Conclusion

In this paper, we focus on modeling temporal community structure with evolution characteristics for boosting community detection and propose the EvoBNMF model which can trace the corresponding evolution behaviors synchronously in dynamic networks. In addition, a gradient descent algorithm is developed to optimize our model. Importantly, the number of communities of each snapshot can be determined automatically by shrinking the evolution behavior in EvoBNMF. Finally, experimental results on synthetic and real-world networks demonstrate the effectiveness of EvoBNMF. In the future, we will do some predictive tasks of dynamic networks (e.g., links or community structures prediction), which are of great practical significance and application value.

Data Availability

The data sets used to support the results of this study are available from the corresponding author upon request.

Disclosure

This manuscript is an extension of a conference paper [24].

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Key Research Project of Zhejiang Yuexiu University (D2020003) and the Finance Science and Technology Project of the 12th Division of Xinjiang Construction Corps (SR202103).

References

[1] G. Rossetti and R. Cazabet, “Community discovery in dynamic networks: a survey,” *ACM Computing Surveys*, vol. 51, no. 2, p. 35, 2018.

[2] N. Dakiche, F. Benbouzid-Si Tayeb, Y. Slimani, and K. Benatchba, “Tracking community evolution in social networks: a survey,” *Information Processing & Management*, vol. 56, no. 3, pp. 1084–1102, 2019.

[3] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela, “Community structure in time-dependent, multiscale, and multiplex networks,” *Science*, vol. 328, no. 5980, pp. 876–878, 2010.

[4] D. Chakrabarti, R. Kumar, and A. Tomkins, “Evolutionary clustering,” in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 554–560, ACM, Philadelphia PA USA, August 2006.

[5] G. Rossetti, L. Pappalardo, D. Pedreschi, and F. Giannotti, “Tiles: an online algorithm for community discovery in dynamic social networks,” *Machine Learning*, vol. 106, no. 8, pp. 1213–1241, 2017.

[6] N. İlhan and Ş. G. Öğüdücü, “Feature identification for predicting community evolution in dynamic social networks,” *Engineering Applications of Artificial Intelligence*, vol. 55, pp. 202–218, 2016.

[7] W. Yu, W. Wang, P. Jiao, and X. Li, “Evolutionary clustering via graph regularized nonnegative matrix factorization for exploring temporal networks,” *Knowledge-Based Systems*, vol. 167, pp. 1–10, 2019.

[8] I. Psorakis, S. Roberts, M. Ebden, and B. Sheldon, “Overlapping community detection using bayesian non-negative matrix factorization,” *Physical review. E, Statistical,*

Nonlinear, and Soft Matter Physics, vol. 83, no. 6, Article ID 066114, 2011.

[9] M. Seifikar, S. Farzi, and M. Barati, “C-blondel: an efficient louvain-based dynamic community detection algorithm,” *IEEE Transactions on Computational Social Systems*, vol. 7, no. 2, pp. 308–318, 2020.

[10] S. Mishra, S. S. Singh, S. Mishra, and B. Biswas, “Tcd2: tree-based community detection in dynamic social networks,” *Expert Systems with Applications*, vol. 169, Article ID 114493, 2021.

[11] Z. Zhao, C. Li, X. Zhang, F. Chiclana, and E. H. Viedma, “An incremental method to detect communities in dynamic evolving social networks,” *Knowledge-Based Systems*, vol. 163, pp. 404–415, 2019.

[12] Y. Yin, Y. Zhao, H. Li, and X. Dong, “Multi-objective evolutionary clustering for large-scale dynamic community detection,” *Information Sciences*, vol. 549, pp. 269–287, 2021.

[13] C.-M. Ting, S. B. Samdin, M. Tang, and H. Ombao, “Detecting dynamic community structure in functional brain networks across individuals: a multilayer approach,” *IEEE Transactions on Medical Imaging*, vol. 40, no. 2, pp. 468–480, 2020.

[14] D. Li, Q. Lin, and X. Ma, “Identification of dynamic community in temporal network via joint learning graph representation and nonnegative matrix factorization,” *Neurocomputing*, vol. 435, pp. 77–90, 2021.

[15] P. Wang, L. Gao, and X. Ma, “Dynamic community detection based on network structural perturbation and topological similarity,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2017, no. 1, Article ID 013401, 2017.

[16] W. Yu, C. C. Aggarwal, and W. Wang, “Temporally factorized network modeling for evolutionary network analysis,” in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pp. 455–464, Cambridge, UK, February 2017.

[17] M. N. Murty and V. S. Devi, “Introduction to pattern recognition and machine learning,” *World Scientific*, vol. 5, 2015.

[18] F. Folino and C. Pizzuti, “An evolutionary multiobjective approach for community discovery in dynamic networks,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1838–1852, 2014.

[19] 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.

[20] D. Greene, D. Doyle, and P. Cunningham, “Tracking the evolution of communities in dynamic social networks,” in *Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining*, pp. 176–183, IEEE, Odense, Denmark, August 2010.

[21] J. Yang and J. Leskovec, “Overlapping community detection at scale: a nonnegative matrix factorization approach,” in *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, pp. 587–596, ACM, Rome Italy, February 2013.

[22] F. Liu, D. Choi, L. Xie, and K. Roeder, “Global spectral clustering in dynamic networks,” *Proceedings of the National Academy of Sciences*, vol. 115, no. 5, pp. 927–932, 2018.

[23] G. Palla, A.-L. Barabási, and T. Vicsek, “Quantifying social group evolution,” *Nature*, vol. 446, no. 7136, pp. 664–667, 2007.

[24] W. Yu, W. Wang, X. Chen, H. Wu, M. Tang, and Y. Yu, “Boosting temporal community detection via modeling community evolution characteristics,” in *Proceedings of the 2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pp. 1291–1296, IEEE, Xiamen, China, December 2019.