Evolutionary nonnegative matrix factorization algorithms for community detection in dynamic networks

Xiaoke Ma and Di Dong

Abstract—Discovering evolving communities in dynamic networks is essential to important applications such as analysis for dynamic web content and disease progression. Evolutionary clustering uses the temporal smoothness framework that simultaneously maximizes the clustering accuracy at the current time step and minimizes the clustering drift between two successive time steps. In this paper, we propose two evolutionary nonnegative matrix factorization (ENMF) frameworks for detecting dynamic communities. To address the theoretical relationship among evolutionary clustering algorithms, we first prove the equivalence relationship between ENMF and optimization of evolutionary modularity density. Then, we extend the theory by proving the equivalence between evolutionary spectral clustering and ENMF, which serves as the theoretical foundation for hybrid algorithms. Based on the equivalence, we propose a semi-supervised ENMF (sE-NMF) by incorporating a priori information into ENMF. Unlike the traditional semi-supervised algorithms, a priori information is integrated into the objective function of the algorithm. The main advantage of the proposed algorithm is to escape the local optimal solution without increasing time complexity. The experimental results over a number of artificial and real world dynamic networks illustrate that the proposed method is not only more accurate but also more robust than the state-of-the-art approaches.

Index Terms—Dynamic networks, community structure, nonnegative matrix factorization, evolutionary clustering

1 Introduction

THE network is powerful for characterizing and analyzing the complex systems by regarding each entity as a vertex and each edge as an interaction between a pair of vertices. There are various networks such as social networks [1]–[3], web networks [4] and biological networks [6]–[9]. It has been shown that many networks have a structure of community (module) that is defined as a subgraph with more edges connecting vertices within the same group and comparatively fewer edges joining vertices across different groups. For example, communities in protein interaction networks correspond to the protein complexes that are critical for biological processes [10]. In scientific collaboration networks, communities correspond to the groups of scientists with same or similar research directions [3].

Thus, community detection has been extensively studied because it provides an insight into the structure and function of the overall system. However, designing effective and efficient algorithms for community detection is still highly non-trivial for many reasons. The main reason is that there is no consensus criterion for quantifying communities in networks. To overcome this problem, Newman *et al.* [12] proposed the well known modularity Q, which implicitly defines communities based on the observation that the random networks are not expected to exhibit modular structure. Based on the normalized cut, Li *et al.* [13] presented the modularity density function Q_D. Even though they have the resolution limit problem [15], many algorithms have been developed via maximizing Q or Q_D function [11]–[22], [28].

However, the algorithms can only discover the communities in static networks. Actually, many networks derived from real world are dynamic [23]–[26]. For instance, in scientific collaboration networks, the communities evolve dynamically because the scientists shift their research directions frequently [27]. In disease networks, cancer cell migration leads to metastasis, which is critical for cancer progression and therapy [29]. Generally speaking, dynamic networks have been becoming more and more popular in many domains since they provide great opportunities for tracing the dynamics of network structure [30]–[35]. In dynamic networks, communities evolve over time by adding new vertices or deleting old ones, which is called dynamic (evolution) communities.

However, it is difficult to design algorithms for dynamic community detection because of two reasons. First of all, there is no clear definition for dynamic communities, as well as no effective way to characterize evolution events. Secondly, the complexity of dynamic networks poses a great challenge for designing algorithms. Even though it is difficult, many approaches have been developed to attack this issue [36]–[39]. Evolutionary clustering is the most widely used strategy for community detection in dynamic networks [40], which produces local clusters for each time step by introducing the *temporal smoothness* framework with the underlying assumption that the abrupt change of clustering within a short period is not desirable. Based on the temporal smoothness framework, various algorithms have been developed for dynamic community detection

- *X. Ma is with the School of Computer Science and Technology, Xidian University, Xi'an, Shaanxi, 710071. D. Dong is with the Key Laboratory of Molecular Imaging, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, PR China*
E-mail: xkma@xidian.edu.cn (X. Ma)

Manuscript received 2016

[2], [37], [38], [41]. For instance, *FacetNet* [2] employs a stochastic block model to obtain dynamic communities. The *Kim-Han* algorithm [41] makes use of the topological structure to discover dynamic communities. *Evolutionary spectral clustering* [37] discovers dynamic communities by extending spectral clustering, while *DYNMOGA* [38] adapts the multiple objective optimization genetic algorithm.

Although great efforts have been made for discovering dynamic communities, there are still many unsolved problems. For instance, what is the theoretical relation among the evolutionary clustering algorithms and how to further improve the accuracy of algorithms. In this paper, we attack the two problems. We first propose two evolutionary nonnegative matrix factorization frameworks (ENMF) for dynamic community detection. Then, we prove the equivalence among the evolutionary spectral clustering, ENMF as well as optimization of evolutionary modularity density. Based on the equivalence, we develop a novel semi-supervised ENMF (sE-NMF) by incorporating a priori information into ENMF, which improves the accuracy without increasing the time complexity.

The main contributions of the paper can be summarized as follows:

- To derive the theoretical relation among the evolutionary clustering algorithms, we prove the equivalence among the evolutionary modularity density, evolutionary spectral clustering and ENMF, which extends the equivalence between evolutionary spectral clustering and kernel K-means [37].
- We propose the sE-NMF by combing ENMF and spectral clustering, in which semi-supervision is incorporated into the objective of ENMF. To the best of our knowledge, sE-NMF is the first NMF-based algorithm for dynamic community detection. sE-NMF can also be considered as a general framework for the semi-supervised evolutionary clustering.
- We demonstrate that our method outperforms the state-of-the-art methods by using both the artificial and real-world dynamic networks without increasing time complexity.

The rest of the paper is organized as follows: Section 2 gives a short review on the algorithms for dynamic networks. Notations and background are briefly introduced in Section 3. In Section 4, we prove the equivalence among the evolutionary modularity algorithms under two different frameworks. In Section 5, the sE-NMF is proposed. The experimental results are illustrated in Section 6. The extension of sE-NMF and conclusion are handled in Section 7 and 8, respectively.

2 RELATED WORK

Mining dynamic patterns has been widely studied [31], [34], [35] and great efforts have been devoted to the analysis of dynamic data, such as data streams [45]–[48].

Recently, dynamic community detection has been receiving increasing attention. We classify the available algorithms for dynamic community detection into two categories: non-evolutionary based and evolutionary based approaches. Non-evolutionary algorithms separate community detection from the temporal analysis. Specifically, these approaches first discover the local communities in the network at each time step and then analyze the involving relationship of the communities at successive time steps. For instance, Kumar *et al.* [42] classified firstly members of network into groups and then studied the dynamics for social networks. Sun *et al.* [43] proposed the parameter-free GraphScope to discover communities in dynamic networks. Asur *et al.* [39] characterized the evolution events for communities in dynamic networks. Tang *et al.* [44] introduced a spectral clustering framework to discover communities and evolving rules. The non-evolutionary based approaches are easy to implement because the algorithms for detecting static community can be used directly.

However, they ignore the connection between subsequent time steps, resulting in undesirable communities. To overcome the problem, evolutionary clustering [40] uses temporal smoothness for dynamic community detection, which balances the communities obtained in networks at two subsequent time points. It consists of *snapshot cost (CS)* and *temporal cost (CT)*, where *CS* quantifies how well the community structure represents the network at the current time step and *CT* measures how similar the community structure at the current time step is to the community structure at the previous time step. The ultimate goal of an evolutionary clustering is to obtain a trade-off between the two sub-costs. The most widely used strategy is to combine *CS* and *CT* via a weighted linear function as

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

where parameter α controls the relevant importance of CS and CT. When α=1, the algorithm captures the clustering for the current network without temporal smoothness from the previous network, while α=0 returns the clustering of the previous network without the snapshot cost. Usually, $\alpha \in$(0,1) controls the preference degree of each sub-cost. The temporal smoothness function in Eq.(1) has been adapted by various algorithms [2], [37], [41].

The evolutionary clustering based algorithms differ greatly on the strategies for discovering dynamic communities. FacetNet [2] employs a stochastic block model to obtain communities at each time step, then uses a probabilistic model based on the Dirichlet distribution to capture evolution communities. The *particle-and-density* algorithm [41] (Kim-Han) makes use of the network topological structure, where the density-based algorithm is used to obtain local clustering at each time step, and uses the nano-community to obtain dynamic communties. Chi *et al.* [37] extended spectral clustering to trace the dynamics of communities, where two frameworks for evolutionary spectral clustering were proposed. Recently, DYNMOGA [38] addresses dynamic community detection by reformulating the temporal smoothness as a multi-objective optimization problem, where the algorithm maximizes the snapshot cost and minimizes the temporal cost simultaneously. Furthermore, Liu *et al.* [68] developed the evolutionary co-clustering algorithms for dynamic networks. More evolutionary algorithms refer to [49].

3 Preliminaries

Prior to giving a detailed description of the proposed methods, we briefly introduce the terminologies that are widely used in the forthcoming sections.

3.1 Notations

Let $\{1, 2, \ldots, T\}$ be a finite set of time steps. For a given variable, the attached subscript t represents the value of the variable at time t. The *dynamic network* $\mathcal{G}$ is defined as a sequence of networks $\mathcal{G} = \{G_1, G_2, \ldots, G_T\}$, where G_t is the network at time t with a vertex set V_t and an edge set E_t. Without loss of generality, we assume that all of the networks in $\mathcal{G}$ have the same vertex set, i.e. $G_t = (V, E_t)$. The dynamic network $\mathcal{G}$ can be represented by a 3-dimensional adjacency matrix $A = (a_{ijt})_{n \times n \times T}$, where n is the number of vertices (i.e. $n = |V|$) and a_{ijt} is 1 if the i-th vertex is connected to the j-th vertex in G_t, 0 otherwise. Actually, $A = [A_1, A_2, \ldots, A_T]$ where $A_t = (a_{ijt})_{n \times n}$ is the adjacency matrix of network G_t. A_t is symmetrical when G_t is undirected. In this paper, we assume that all the networks are undirected and unweighted. The degree of the i-th vertex in network G_t is defined as the number of edges connected to it, i.e. $d_{it} = \sum_j a_{ijt}$. The degree matrix for dynamic networks $\mathcal{G}$ is defined as $D = [D_1, D_2, \ldots, D_T]$, where D_t is the diagonal matrix with the degree sequence of G_t, i.e. $D_t = diag(d_{1t}, d_{2t}, \ldots, d_{nt})$.

Community detection in a network is to partition the vertices into disjointed groups where the number of edges within the same cluster is more than that across the groups. Specifically, given a network $G_t = (V, E_t)$, the community structure corresponds to a hard partitioning $\{C_{1t}, C_{2t}, \ldots, C_{kt}\}$ (denoted by $\{C_{it}\}_{i=1}^{k}$) such that $C_{it} \cap C_{jt} = \emptyset$ if $i \neq j$ and $V = \sum_i C_{it}$. C_{it} is the i-th local community (cluster) at time t. $\{C_{it}\}_{i=1}^{k}$ is represented by an $n \times k$ indicator matrix $Z_t = (z_{ijt})_{n \times k}$ whose element z_{ijt} = 1 if the i-th vertex belongs to the j-th local cluster at time t, 0 otherwise. For the sake of convenience, we normalize the columns of matrix Z_t by the sizes of the local clusters, i.e. $z_{.jt}/\sqrt{|C_{jt}|}$, where $z_{.jt}$ is the j-th column of Z_t. The normalized indicator matrix is denoted by $\widetilde{Z}_t = [\widetilde{z}_{.1t}, \ldots, \widetilde{z}_{.kt}]$ where $\widetilde{z}_{.jt} = z_{.jt}/\sqrt{|C_{jt}|}$. It is easy to validate that $\widetilde{Z}_t$ is column orthonormal, i.e. $\widetilde{Z}_t^{'}\widetilde{Z}_t = I_k$ where I_k is an identity matrix and $\widetilde{Z}_t^{'}$ is the transpose of matrix $\widetilde{Z}_t$.

3.2 Quantitative functions for community

Given a network $G_t = (V, E_t)$ and a hard partitioning $\{C_{it}\}_{i=1}^{k}$, we define $L(C_{it}, C_{jt}) = \sum_{i \in C_{it}, j \in C_{jt}} a_{ijt}$ and $\overline{C}_{it} = V \setminus C_{it}$. The modularity Q [12] is defined as

$$Q(\{C_{it}\}_{i=1}^{k}) = \sum_{i=1}^{k} \frac{L(C_{it}, C_{it})}{L(V, V)} - \left(\frac{L(C_{it}, V)}{L(V, V)}\right)^2. \quad (2)$$

The modularity density Q_D [13] is formulated as

$$Q_D(\{C_{it}\}_{i=1}^{k}) = \sum_{i=1}^{k} \frac{L(C_{it}, C_{it}) - L(C_{it}, \overline{C}_{it})}{|C_{it}|}. \quad (3)$$

The negated average association [37] is defined as

$$NA = Tr(A_t) - \sum_{i=1}^{k} \frac{L(C_{it}, C_{it})}{|C_{it}|}, \quad (4)$$

where $Tr(A_t)$ is the *trace* of matrix A_t, i.e. $\sum_{i=1}^{n} a_{iit}$.

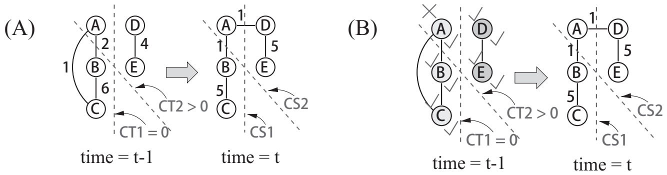

Fig. 1. Two frameworks for evolutionary clustering, where each dashed line corresponds to a partitioning. (A) PCQ framework: The snapshot costs of the two partitionings are the same at time t, but the temporal costs are different at time t-1 because there are more edges crossing partitioning 2 than partitioning 1 at time t-1. (B) PCM framework: at time t, the snapshot costs of the two partitionings are the same; however, when we measure the temporal cost by comparing the partitioning at time t with that at time t-1, the temporal cost of partitioning 2 is greater than partitioning 1. The figures are from Ref. [37].

4 Equivalence of evolutionary clustering algorithms

Based on how the temporal cost CT is defined, evolutionary clusterings are classified into two frameworks: preserving cluster quality (PCQ) as well as preserving cluster membership (PCM). The first one makes use of the consistence between the current clustering result and the network at the previous time step (Fig.1 A), while the second one uses the consistency between the current and historic clustering results (Fig.1 B). The advantage of the PCM framework is that the local communities at the previous time step are used to guide the discovery of communities at the current time step, which is sensitive to community dynamics. The disadvantage of the PCM framework is that the error in the communities at previous time step is passed down to the detected communities at current time step, which affects the accuracy of algorithms. However, the PCQ framework avoids this limitation because the detected communities are determined by the networks at two successive time steps. The dynamic communities discovered by the two frameworks may differ greatly (shown in Section 7).

In this section, we prove the equivalence among the ENMF, evolutionary spectral clustering and modularity density under the two frameworks. At first glance, they appear to be unrelated, while they are equivalent in terms of trace optimization. The key technique is how to transform the objective functions into trace optimization.

4.1 Framework I: Preserving cluster quality (PCQ)

In the PCQ framework, CT is expressed at how well the current partitioning clusters the historic data. As shown in Fig.1 A, partitioning $Z_t^{[1]}$ and $Z_t^{[2]}$ of network G_t are equal well, i.e. $CS1$=$CS2$. However, clustering the historic network G_{t-1}, $Z_t^{[1]}$ is better than $Z_t^{[2]}$ because the temporal cost $CT2$ is larger than $CT1$. In the PCQ framework, $Z_t^{[1]}$ is preferred over $Z_t^{[2]}$.

4.1.1 Evolutionary spectral clustering

The overall cost (Eq.1) for the evolutionary spectral clustering under the PCQ framework using the negated average association is defined as [37]

$$\begin{aligned} Cost_{NA} &= \alpha CS_{NA} + (1-\alpha)CT_{NA} \\ &= \alpha NA_t|_{Z_t} + (1-\alpha)NA_{t-1}|_{Z_t}, \end{aligned} \tag{5}$$

where $|_{Z_t}$ denotes the evaluation by Z_t and the temporal cost $CT_{NA} = NA_{t-1}|_{Z_t}$ penalizes Z_t that does not fit well with network G_{t-1}.

Eq.(5) is transformed into trace optimization as [37]

$$\begin{aligned} Cost_{NA} &= Tr[\alpha A_t + (1-\alpha)A_{t-1}] \\ &\quad -Tr[\widetilde{Z}_t^{'}(\alpha A_t + (1-\alpha)A_{t-1})\widetilde{Z}_t]. \end{aligned} \tag{6}$$

Notice that the first term $Tr[\alpha A_t + (1-\alpha)A_{t-1}]$ is a constant because it is independent of the local partitioning $\widetilde{Z}_t$. Therefore, minimizing $Cost_{NA}$ is equivalent to maximizing the last term, i.e.

$$\min Cost_{NA} \propto \max Tr[\widetilde{Z}_t^{'}(\alpha A_t + (1-\alpha)A_{t-1})\widetilde{Z}_t]. \tag{7}$$

4.1.2 Evolutionary modularity density

We first formulate the overall cost for the evolutionary clustering by optimizing the modularity density Q_D. Following Eq.(5), it is defined as

$$\begin{aligned} Cost_{Q_D} &= \alpha CS_{Q_D} + (1-\alpha)CT_{Q_D} \\ &= \alpha(Q_D)_t|_{Z_t} + (1-\alpha)(Q_D)_{t-1}|_{Z_t}. \end{aligned} \tag{8}$$

To derive the trace optimization of $Cost_{Q_D}$, we first derive the trace optimization of $(Q_D)_t|_{Z_t}$

$$\begin{aligned} (Q_D)_t|_{Z_t} &= \sum_{i=1}^{k} \frac{L(C_{it}, C_{it}) - L(C_{it}, \overline{C}_{it})}{|C_{it}|} \\ &= \sum_{i=1}^{k} \frac{2L(C_{it}, C_{it}) - L(C_{it}, V)}{|C_{it}|} \\ &= \sum_{i=1}^{k} \frac{(z_{.it})^{'}(2A_t - D_t)z_{.it}}{z_{.it}^{'}A_t z_{.it}} \\ &= Tr[\widetilde{Z}_t^{'}(2A_t - D_t)\widetilde{Z}_t]. \end{aligned} \tag{9}$$

Substituting Eq.(9) into Eq.(8), we transform the overall cost $Cost_{Q_D}$ into trace optimization as

$$\begin{aligned} Cost_{QD} &= \alpha Tr[\widetilde{Z}_t^{'}(2A_t - D_t)\widetilde{Z}_t] \\ &\quad +(1-\alpha)Tr[\widetilde{Z}_t^{'}(2A_{t-1} - D_{t-1})\widetilde{Z}_t] \\ &= Tr[\widetilde{Z}_t^{'}(\alpha(2A_t - D_t) \\ &\quad +(1-\alpha)(2A_{t-1} - D_{t-1}))\widetilde{Z}_t] \end{aligned} \tag{10}$$

Comparing Eqs.(7) and (10), we assert that they have the same optimization form except for the matrices used, indicating that the evolutionary modularity density Q_D is equivalent with evolutionary spectral clustering, i.e.

$$\max Cost_{Q_D} \propto \min Cost_{NA}. \tag{11}$$

This result theoretically explains why the topology structure can be used for dynamic community detection, which provides the possible theoretical foundation for Kim-Han [41] and DYNMOGA algorithms [38] because both of them make use of topology to discover the dynamic communities.

4.1.3 Evolutionary nonnegative matrix factorization

Matrix factorization is efficient for mining patterns in graphs [67]. NMF aims at learning the representation parts of the original data [50], [51] by approximating the target matrix into the product of two low-rank matrices. Specifically, given an $n \times m$ matrix F, NMF decomposes F into two non-negative matrices $R_{n\times s}$ and $S_{s\times m}$ such that

$$F \approx RS, s.t. R \geq 0, S \geq 0. \tag{12}$$

Usually, s is much smaller than $\min\{n, m\}$. The Symmetric NMF (SNMF) is an extension of NMF, which approximately factorizes matrix F into an $n \times s$ matrix R such that

$$F \approx RR^{'}, s.t. R \geq 0. \tag{13}$$

Given the adjacency matrix A_t of network G_t, the local clusters $\widetilde{Z}_t$ can be obtained by using SNMF [28], [64], i.e.

$$A_t \approx \widetilde{Z}_t\widetilde{Z}_t^{'}. \tag{14}$$

The above equation is solved via minimizing the squared error, i.e.

$$SNMF_t = \min_{\widetilde{Z}_t \geq 0, \widetilde{Z}_t^{'}\widetilde{Z}_t = I_k} \|A_t - \widetilde{Z}_t\widetilde{Z}_t^{'}\|^2, \tag{15}$$

where $\|B\|$ is the Frobeneus norm of matrix B. We rewrite Eq.(15) as trace optimization, i.e.

$$\begin{aligned} SNMF_t &= \min_{\widetilde{Z}_t \geq 0, \widetilde{Z}_t^{'}\widetilde{Z}_t = I_k} \|A_t - \widetilde{Z}_t\widetilde{Z}_t^{'}\|^2 \\ &= \min_{\widetilde{Z}_t \geq 0, \widetilde{Z}_t^{'}\widetilde{Z}_t = I_k} \|A_t\|^2 - 2trace(\widetilde{Z}_t^{'}A_t\widetilde{Z}_t) + \|\widetilde{Z}_t^{'}\widetilde{Z}_t\|^2 \\ &\propto -\min_{\widetilde{Z}_t \geq 0, \widetilde{Z}_t^{'}\widetilde{Z}_t = I_k} trace(\widetilde{Z}_t^{'}A_t\widetilde{Z}_t) \\ &= \max_{\widetilde{Z}_t \geq 0, \widetilde{Z}_t^{'}\widetilde{Z}_t = I_k} trace(\widetilde{Z}_t^{'}A_t\widetilde{Z}_t). \end{aligned} \tag{16}$$

Relaxing the orthogonality $\widetilde{Z}_t^{'}\widetilde{Z}_t = I_k$, Eq.(16) is rewritten as

$$SNMF_t \propto \max_{\widetilde{Z}_t \geq 0} trace(\widetilde{Z}_t^{'}A_t\widetilde{Z}_t). \tag{17}$$

Following Eq.(5), we define the overall cost for ENMF as

$$\begin{aligned} Cost_{ENMF} &= \alpha CS_{SNMF} + (1-\alpha)CT_{SNMF} \\ &= \alpha SNMF_t|_{Z_t} + (1-\alpha)SNMF_{t-1}|_{Z_t}. \end{aligned}$$

Based on Eq.(17), we reformulate the above equation as

$$\begin{aligned} Cost_{ENMF} &\propto \alpha Tr(\widetilde{Z}_t^{'}A_t\widetilde{Z}_t) + (1-\alpha)Tr(\widetilde{Z}_t^{'}A_{t-1}\widetilde{Z}_t) \\ &= Tr[\widetilde{Z}_t^{'}(\alpha A_t + (1-\alpha)A_{t-1})\widetilde{Z}_t]. \end{aligned} \tag{18}$$

Comparing Eqs.(18), (7) and (10), we conclude that ENMF, evolutionary spectral clustering and evolutionary modularity density are equivalent, i.e.

$$\begin{cases} \min Cost_{ENMF} \propto \min Cost_{NA}, \\ \min Cost_{ENMF} \propto \max Cost_{Q_D}. \end{cases}$$

The proved equivalence between SNMF and evolutionary spectral clustering implies that the NMF algorithm can also be applied for dynamic community detection. Then, we ask whether the equivalence under the PCQ framework also holds under the PCM framework, which is proven in the next subsection.

4.2 Framework II: Preserving clustering membership (PCM)

We illustrate the PCM framework with an example (Fig.1 B). At time t, the partitioning $Z_t^{[1]}$ and $Z_t^{[2]}$ cluster the current network equally well ($CS1 = CS2$). When compared to the historic partition Z_{t-1}, $Z_t^{[1]}$ is more similar to Z_{t-1} than $Z_t^{[2]}$ because $CT1 < CT2$. In such a case, $Z_t^{[1]}$ is preferred over $Z_t^{[2]}$.

4.2.1 Evolutionary spectral clustering

The temporal cost CT under the PCM framework is defined as [37]

$$\begin{aligned} CT &= dist(\widetilde{Z}_t, \widetilde{Z}_{t-1}) \\ &= \frac{1}{2}\|\widetilde{Z}_t\widetilde{Z}_t' - \widetilde{Z}_{t-1}\widetilde{Z}_{t-1}'\|^2. \end{aligned} \tag{19}$$

Note that the evolutionary spectral clustering replaces $\widetilde{Z}_t$ with the top k eigenvectors of A_t. Here, we do not employ the strategy with an immediate purpose to have a stringent mathematical proof. The overall cost of evolutionary spectral clustering under PCM can be reformulated as

$$\begin{aligned} Cost_{NA} &= \alpha Tr(A_t) + (1-\alpha)k \\ &\quad - Tr[\widetilde{Z}_t'(\alpha A_t + (1-\alpha)\widetilde{Z}_{t-1}\widetilde{Z}_{t-1}')\widetilde{Z}_t]. \end{aligned} \tag{20}$$

Note that the first two terms are constants. Thus, minimizing $Cost_{NA}$ under the PCM framework is equivalent to maximizing the last term, i.e.

$$\min Cost_{NA} \propto \max Tr[\widetilde{Z}_t'(\alpha A_t + (1-\alpha)\widetilde{Z}_{t-1}\widetilde{Z}_{t-1}')\widetilde{Z}_t]. \tag{21}$$

4.2.2 Evolutionary modularity density

The overall cost of the evolutionary modularity density in Eq.(8) under the PCM framework can be reformulated as

$$\begin{aligned} Cost_{Q_D} &= \alpha(Q_D)_t|_{Z_t} + (1-\alpha)(Q_D)_{t-1}|_{Z_t} \\ &= \alpha Tr(\widetilde{Z}_t'(2A_t - D_t)\widetilde{Z}_t) \\ &\quad + \frac{1-\alpha}{2}\|\widetilde{Z}_t\widetilde{Z}_t' - \widetilde{Z}_{t-1}\widetilde{Z}_{t-1}'\|^2. \end{aligned} \tag{22}$$

The above equation is rewritten as

$$\begin{aligned} Cost_{Q_D} &= \alpha Tr[\widetilde{Z}_t'(2A_t - D_t)\widetilde{Z}_t] + \frac{1-\alpha}{2}(Tr(\widetilde{Z}_t\widetilde{Z}_t'\widetilde{Z}_t\widetilde{Z}_t') \\ &+ Tr(\widetilde{Z}_{t-1}\widetilde{Z}_{t-1}'\widetilde{Z}_{t-1}\widetilde{Z}_{t-1}') - 2Tr(\widetilde{Z}_{t-1}'\widetilde{Z}_t\widetilde{Z}_t'\widetilde{Z}_{t-1})) \\ &= \alpha \cdot Tr[\widetilde{Z}_t'(2A_t - D_t)\widetilde{Z}_t] + (1-\alpha)k \\ &\quad -(1-\alpha)Tr(\widetilde{Z}_t'\widetilde{Z}_{t-1}\widetilde{Z}_{t-1}'\widetilde{Z}_t) \\ &= (1-\alpha)k + Tr[\widetilde{Z}_t'(\alpha(2A_t - D_t) - \\ &\quad (1-\alpha)\widetilde{Z}_{t-1}\widetilde{Z}_{t-1}')\widetilde{Z}_t] \end{aligned} \tag{23}$$

Comparing Eqs.(23) and (21), we conclude that evolutionary spectral clustering is also equivalent with the evolutionary modularity density Q_D under the PCM framework, i.e.

$$\max Cost_{Q_D} \propto \min Cost_{NA}. \tag{24}$$

4.2.3 Evolutionary nonnegative matrix algorithm

The overall cost function for ENMF in Eq.(18) under the PCM framework is transformed into

$$\begin{aligned} Cost_{ENMF} &= \alpha CS_{SNMF} + (1-\alpha)CT_{SNMF} \\ &= \alpha Tr(\widetilde{Z}_t' A_t \widetilde{Z}_t) + \frac{1-\alpha}{2}\|\widetilde{Z}_t\widetilde{Z}_t' - \widetilde{Z}_{t-1}\widetilde{Z}_{t-1}'\|^2 \\ &= \alpha Tr(\widetilde{Z}_t' A_t \widetilde{Z}_t) + (1-\alpha)(k - Tr(\widetilde{Z}_t'\widetilde{Z}_{t-1}\widetilde{Z}_{t-1}'\widetilde{Z}_t)) \\ &= (1-\alpha)k + Tr[\widetilde{Z}_t'(A_t - (1-\alpha)\widetilde{Z}_{t-1}\widetilde{Z}_{t-1}')\widetilde{Z}_t]. \end{aligned} \tag{25}$$

Comparing Eqs.(25), (21) and (23), we conclude that ENMF, evolutionary modularity density and evolutionary spectral clustering are equivalent under the PCM framework, i.e.

$$\begin{cases} \min Cost_{ENMF} \propto \min Cost_{NA}, \\ \min Cost_{ENMF} \propto \max Cost_{Q_D}. \end{cases}$$

As for equivalence, what we want to point out is summarized as follows:

- Even though we only prove the equivalence among evolutionary algorithms for unweighted dynamic networks, the equivalence also holds in weighted dynamic networks because the equations hold if we replace the unweighted adjacency matrices with weighted ones.
- The proved equivalence here extends the theory in Ref. [37], which provides the theoretical foundation for ENMF and topology based evolutionary clustering algorithms.

Next, we attack the second problem by making use of the equivalence to improve the accuracy of algorithms for dynamic community detection.

5 Semi-supervised evolutionary nonnegative matrix factorization (sE-NMF)

The procedures of the sE-NMF algorithm, parameter selection and complexity analysis are proposed.

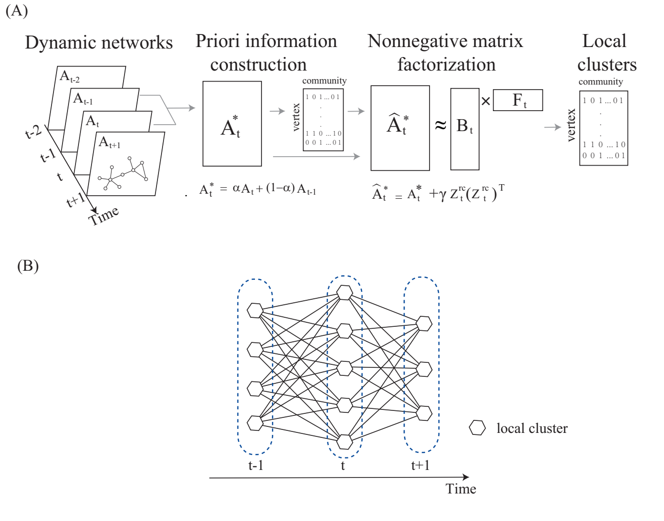

Fig. 2. Overview of the sE-ENMF algorithm. It consists of two major components: (A) discovering local clusters based on the temporal smoothness framework using a priori information, (B) mapping local clusters to identify dynamic communities using a T-partite graph.

5.1 Algorithm

The sE-NMF algorithm is under the PCQ framework (the extension of sE-NMF under the PCM framework is introduced in Section 7), which consists of local cluster discovery for the network at each time step, as well as the dynamic communities discovery (Fig.2).

5.1.1 Semi-supervised local cluster discovery

To discover the local clusters at time t based on the temporal smoothness framework, we take into account both G_t and G_{t-1} via a linear function, which is defined as

$$A_t^* = \alpha A_t + (1-\alpha)(A_t - A_{t-1}), \tag{26}$$

where parameter α controls the relevant importance of CS and CT. The underlying assumption is that if a group of vertices whose connectivity is strong in both G_{t-1} and G_t, then they are very likely to be a local cluster at time t. Usually, α=0.8 [37], [41].

Many algorithms have shown that incorporating a priori information into clustering algorithms significantly improves the accuracy of algorithms, such as the Pagerank [4], semi-supervised NMF [28] and semi-supervised spectral clustering [53]. However, there is no supervised or semi-supervised evolutionary clustering algorithm for dynamic community detection. Here, we present the sE-NMF algorithm that combines t ENMF and spectral clustering. There are two reasons why spectral clustering is employed: first of all, spectral clustering usually achieves excellent performance [55]; second, the equivalence between evolutionary spectral clustering and ENMF lays a solid theoretical foundation for combining them.

The strategy to construct a priori information consists of two steps: obtaining high quality groups of vertices as well as constructing the partial information. For the first issue, we first obtain the local clusters at time t using the spectral clustering algorithm, which is denoted by an indicator matrix Z_t^{sc} where Z_{ijt}^{sc}=1 if the i-th vertex belongs to the j-th local community, 0 otherwise. To ensure the quality of the priori information, for each local community we keep removing vertices within the community until the density of the induced subnetwork of the community reaches a threshold β. The underlying hypothesis is that the strongly connected vertices are very likely to be grouped as a community. The subgraphs with at least 5 vertices are kept for the priori information because small dense subgraphs are ubiquitous in random networks [69]. The refined local community is denoted by Z_t^{rc}. Similar to the strategy [28], [53], we incorporate the priori information into A_t^* as

$$\widehat{A}_t^* = A_t^* + \gamma Z_t^{rc}(Z_t^{rc})', \tag{27}$$

where γ is a parameter controlling the relevant weight of the priori information. How parameter γ affects the performance is discussed in the experiments.

To obtain the local clusters at time t, sE-NMF factorizes matrix $\widehat{A}_t^*$ via minimizing the squared error, i.e.

$$J_t = \|\widehat{A}_t^* - B_t F_t\|^2, s.t. B_t \geq 0, F_t \geq 0. \tag{28}$$

To solve the above equation, we set the derivatives of J_t as

$$\begin{cases} \frac{\partial J_t}{\partial B_t} = 0, \\ \frac{\partial J_t}{\partial F_t} = 0. \end{cases}$$

Then, we obtain the following updating rules

$$B_t = B_t \frac{\widehat{A}_t^* F_t'}{B_t F_t F_t'}, \tag{29}$$

$$F_t = F_t \frac{B_t' \widehat{A}_t^*}{B_t' B_t F_t}. \tag{30}$$

The indicator matrix Z_t for the local clusters of G_t is constructed based on matrix F_t, i.e. $Z_{ij^*t} = 1$ where $j^* = \arg max_j F_{ji}$, 0 otherwise. The procedure is depicted in Fig.2 A.

5.1.2 Dynamic community discovery

Given the local clusters at time step t-1 and t, we discover the dynamic communities by mapping the local clusters. Let $\mathcal{C}_t$ be the set of local clusters at time t and C_{it} be the i-th local cluster in $\mathcal{C}_t$. We construct a bipartite graph between $\mathcal{C}_{t-1}$ and $\mathcal{C}_t$ where each local cluster is treated as a vertex and the link weight between $C_{i,t-1}$ and C_{jt} is defined as the number of overlapped vertices, i.e.

$$l_{ijt} = |C_{i,t-1} \cap C_{jt}|. \tag{31}$$

The underlying assumption is that the more vertices overlap, the more likely the local clusters are to be mapped. Analogously, we construct a T-partite graph from $\mathcal{C}_1$ to $\mathcal{C}_T$ (Fig.2 B).

We define the topological similarity for the T-partite graph using the mutual information [41]

$$MI(C_{i,t-1}, C_{jt}) = \frac{l_{ijt}}{\sum_j l_{ijt}} \log \frac{l_{ijt} \sum_{i,j} l_{ijt}}{\sum_i l_{ijt} \sum_j l_{ijt}}. \tag{32}$$

The assumption is that the higher the mutual information is, the more likely the two local clusters are related.

A greedy search procedure is used to map local clusters from $\mathcal{C}_{t-1}$ to $\mathcal{C}_t$: for each iteration, the local cluster pair with a maximal MI value is selected; once a mapping between $C_{i,t-1}$ and C_{jt} is done, other local clusters within $\mathcal{C}_{t-1}$ cannot be mapped to C_{jt}; this procedure continues until no mapping is available. The dynamic communities are these mapped local clusters. The sE-NMF algorithm is presented in Algorithm 1.

5.2 Parameter selection

The sE-NMF algorithm involves three parameters k_t, β and γ, where k_t is the number of local clusters in network G_t, β is the density threshold for a priori information and γ controls the importance of a priori information.

Determining the number of clusters is challenging in clustering and there are many ways for attacking this issue. For instance, in the spectral clustering algorithm the gap between two successive eigenvalues is used to select the appropriate k_t. When the network size is small, it is easy to obtain a large gap. When the size of the network is large, the gap is difficult to find since the threshold is hard to define. The objective function guided strategy is also commonly used, where k_t is the value corresponding to the maximum (minimum) value of a given objective function such as modularity Q. In this paper, we select k_t based on the matrix spectrum theory. Given network G_t, the spectrum of the adjacency matrix A_t is the set of the eigenvalues

Algorithm 1 sE-NMF under the PCQ framework

```
Input:
    $\mathcal{G}$: Dynamic networks;
    $\{k_i\}_{i=1}^{T}$: number of local clusters;
    $\beta$: threshold for the density of local clusters;
    $\gamma$: weight for the priori information;
Output:
    $\{Z_t\}_{t=1}^{T}$: local clusters and evolutionary community.
    Part I: Local cluster discovery
  For each time step $t$, construct $A_t^*$ according to Eq.(26);
  Use the spectral clustering algorithm to obtain the partial information $Z_t^{rc}$;
  Construct the partial matrix $\widehat{A}_t^*$ according to Eq.(27);
  Use NMF to obtain the local clusters discovery $\{Z_t\}_{t=1}^{T}$ according to Eqs.(29,30);
    Part II: Evolutionary community discovery
  Construct the $T$-partite graph using the local clusters $\{Z_t\}_{t=1}^{T}$;
  Discover the evolution communities using the greedy search algorithm;
  return
```

$\lambda_{1t}, \lambda_{2t}, \ldots, \lambda_{nt}$ as well as the corresponding eigenvectors $x_{1t}, x_{2t}, \ldots, x_{nt}$. Without loss of generality, we assume that $\lambda_{1t} \geq \lambda_{2t} \geq \cdots \geq \lambda_{nt}$. Matrix A_t can be reconstructed based on the eigenvalues and eigenvectors, i.e.

$$A_t = \sum_{i=1}^{n} \lambda_{it} x_{it} x_{it}^{'}. \tag{33}$$

We choose k_t as the minimum value k such that

$$k_t = \arg_{k^*} \min_{k} \sqrt{\|\sum_{i=1}^{k} \lambda_{it} x_{it} x_{it}^{'}\| / \|A_t\|} > \delta$$

where δ is a parameter controlling the approximation.

Let us intuitively explain why the strategy can yield a better parameter selection. As the number of eigenvectors increases, the error between A_t and eigen-decomposition (i.e. $\|A_t - \sum_{i=1}^{k} \lambda_{it} x_{it} x_{it}^{'}\|^2$) decreases. On the other hand, as the number of eigenvalues increases, the features are redundant. As a result, an appropriate value for δ that achieves a reasonable approximation with a reasonably small number of eigenvalues tends to have a good trade-off. The strategy used can be regarded as justifying the use of Occam's razor in evolutionary clustering. How the strategy outperforms others is discussed (Supplementary).

We set δ, γ and β as the tuning parameters, which are selected empirically.

5.3 Algorithm analysis

The space complexity of the sE-NMF algorithm is first discussed. Given a dynamic network $\mathcal{G}$, the 3-dimensional adjacency matrix $A_{n \times n \times T}$ requires space $O(n^2 T)$. Given an adjacency matrix A_t, the space complexity for the indicator matrix is $O(nk_t)$. The total complexity for factorizing A is $O(n(k_1 + k_2 + \ldots + k_n)) = O(nT \max\{k_1, \ldots, k_n\})$, which is $O(n^2 T)$ since $k_i \leq n (1 \leq i \leq T)$. The space for the T-partite graph in the community mapping procedure is $O(T(\max_i k_i)^2)$. Therefore, the overall space complexity is $O(n^2 T)$, demonstrating that the proposed method is efficient in space complexity.

Then, the time complexity is analyzed. For each time t, sE-NMF consists of three major components: the priori information construction using the spectral clustering algorithm, matrix factorization for local clusters, as well as the module mapping procedure. The time complexity of spectral clustering is $O(n^3)$ [62]. The time complexity of NMF for local clusters at time t is $O(rn^2 k_t)$, where r is the number of iterations [52]. The time complexity of the local cluster mapping procedure is $O(k_{t-1} k_t)$. Thus, the time complexity for each time step is $O(n^3 + rn^2 k_t + k_{t-1} k_t) = O(n^3 + rn^2 k_t)$ since $k_i < n$. The overall time complexity of the proposed algorithm is $O(T(n^3 + rn^2 \max\{k_t\}_{t=1}^{T}))$, showing that the proposed method does not increase the time complexity compared to ENMF.

6 Experiments

In order to evaluate the performance of the sE-NMF algorithm, three well-known algorithms are adapted for comparison, including FacetNet [2], Kim-Han [41] and DYNMOGA [38]. The reason why these algorithms are selected is that: both FacetNet and the Kim-Han method are the most well-known algorithms for evolutionary clustering, while DYNMOGA is the method with the best performance currently. Evolutionary spectral clustering [37] is excluded because the priori information is based on spectral clustering.

Five datasets are used to validate the performance, including three artificial and two real world dynamic networks. The artificial networks have various evolution events, which testify whether the compared algorithms can accurately discover different evolution communities. The two real world networks, including social and cancer networks, are used to test whether the algorithms can discover dynamic communities with a particular background.

6.1 Settings

Three criteria are used for measuring the performance of algorithms, which are normalized mutual information (N-MI) [54], error rate [2] as well as the rand index [65]. Given the standard partition $\mathcal{C}^*$and obtained partition $\mathcal{C}$, we construct a confusion matrix N whose rows corresponding to the communities in $\mathcal{C}^*$ and columns corresponding to the communities in $\mathcal{C}$. Element N_{ij} is the number of vertices overlapped by the i-th real and j-th obtained communities. NMI is defined as

$$NMI(\mathcal{C}, \mathcal{C}^*) = \frac{-2 \sum_{i=1}^{|\mathcal{C}|} \sum_{j=1}^{|\mathcal{C}^*|} N_{ij} \log(\frac{N_{ij} N}{N_{i.} N_{.j}})}{\sum_{i=1}^{|\mathcal{C}|} N_{i.} \log(\frac{N_{i.}}{N}) + \sum_{j=1}^{|\mathcal{C}^*|} N_{.j} \log(\frac{N_{.j}}{N})}$$

where $|\mathcal{C}|$ is the number of communities in $\mathcal{C}$ and $N_{i.}$ is the sum of the i-th row of N.

Let Z and Z^* be the indicator matrix for $\mathcal{C}$ and $\mathcal{C}^*$, respectively. The error rate is defined as

$$error(Z, Z^*) = \sqrt{\|Z^*(Z^*)' - ZZ'\|}.$$

Let a be the number of pairs of vertices that are in the same community in $\mathcal{C}$ and in the same community in $\mathcal{C}^*$, b be the number of pairs of vertices that are in different

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2017.2657752, IEEE Transactions on Knowledge and Data Engineering

communities in $\mathcal{C}$ and in different communities in $\mathcal{C}^*$, c be the number of pairs of vertices that are in the same community in $\mathcal{C}$ and in different communities in $\mathcal{C}^*$, and d be the number of pairs of vertices in V that are in different communities in $\mathcal{C}$ and in the same community in $\mathcal{C}^*$. The rand index is defined as

$$RI = \frac{a+b}{a+b+c+d}.$$

6.2 Synthetic Dataset #1

The first artificial dataset [41] contains two types of dynamic networks: SYN-FIX and SYN-VAR. The SYN-FIX network consists of 128 vertices grouped into 4 communities where each of them contains 32 vertices. Every vertex has an average degree 16 and shares z edges connecting vertices outside of the community. To order to introduce dynamics in $\mathcal{G}$, 3 vertices are randomly selected for each community in G_{t-1} and assigned to others in G_t. Thus, the number of communities in the SYN-FIX network is 4 for all of the time steps. SYN-VAR is a modified version of SYN-FIX by forming and dissolving communities. The initial network has 256 vertices, classified into 4 communities with 64 vertices in each. We randomly select 8 vertices from each community at time t-1 and group them into a novel community at time t. This procedure is repeated for 5 time steps, then the vertices are returned to the original communities. Thus, the number of communities for the 10 time steps is 4,5,6,7,8,8,7,6,5,4.

We first investigate how the parameters influence the accuracy of the sE-NMF algorithm because the communities are known. We focus on parameter α, β and γ, where α controls the relevant importance of CS and CT, β controls the density of the local community for the priori information, and γ is the relevant importance of the priori information. For each parameter, we analyze how it affects the performance of sE-NMF by fixing the remaining parameters. We first investigate the effect of parameter α by adjusting it from 0.3 to 0.8 with β=0.95 and γ=0.1. Fig.3 A1-A4 show how the NMIs change as α increases from 0.3 to 0.8, where Fig.3 A1,A2 are for the SYN-FIX networks and A3,A4 are for the SYN-VAR networks. From these we conclude that the accuracy of sE-NMF increases as α increases and it achieves the best performance when α=0.8. The reason is that when α is small, the overall cost is dominated by CT, decreasing the accuracy of the sE-NMF. As α increases, the relevant importance of CS increases, improving the accuracy. When α = 0.8, there is a good trade-off between CS and CT. This result is consistent with the analysis in FacetNet [2], evolutionary spectral clustering [37] and Kim-Han [41]. To further validate the effect of parameter α, we also investigate how it affects the accuracy of sE-NMF in terms of error rate (Fig.3 B1-B4) as well as the rand index (Fig.3 C1-C4). From them, we conclude that there is a similar tendency, indicating that parameter α=0.8 is not sensitive to the selected criteria. Therefore, we amend α=0.8 in the forthcoming experiments.

Then, we ask how parameter β influences the performance of the sE-NMF algorithm by adjusting α=0.8 and γ=0.1. Fig.4 shows how the NMI changes as β increases from 0.75 to 0.95, where Fig.4 A,B are NMIs for SYN-FIX and C,D are for SYN-VAR. It is easy to summarize that as β increases, the accuracy also increases dramatically. There is a good reason to explain this tendency: when β is small, the connectivity within the local communities obtained by spectral clustering is weak, introducing noise in the priori information. Increasing β filters noise in the priori information, improving the accuracy of the algorithm. In the following experiments, we set β=0.95.

The impact of parameter γ on the performance of the algorithm is shown in Fig.5 with α=0.8 and β=0.95. It is easy to summarize that as γ increases, the accuracy decreases dramatically. The reason is that if γ is small, a priori information is auxiliary. When γ increases, the contribution of a priori information to the accuracy increases, deviating the local clusters from the original networks. Fig.5 suggests that γ=0.1 reaches a good balance between a priori information and the original data, which is consistent with the assertion in [28], [53]. In the following experiments, we set γ=0.1.

The comparison among DYNMOGA, Kim-Han as well as sE-NMF on dataset #1 is summarized in Fig.6. We assert that both sE-NMF and DYNMOGA significantly outperform the Kim-Han algorithm on all four types of dynamic networks. Furthermore, sE-NMF and DYNMOGA have similar performance in the SYN-FIX dynamic networks, while sE-NMF is much better than DYNMOGA on the SYN-VAR networks. Overall, sE-NMF achieves the best performance. There are two reasons why the proposed method outperforms others: first of all, NMF is powerful for dynamic community detection; second, the semi-supervised strategy improves the accuracy by escaping the local optimal solution.

6.3 Synthetic Dataset #2

The second artificial dataset is introduced in Ref. [2], where each network consists of 128 vertices divided into 4 communities with 32 vertices each. Every vertex has a fixed average degree, and connects z vertices in other communities. We set the average degree of networks as 16, and generate the dynamic networks by varying parameter z. In order to introduce dynamics in the networks, C% of vertices are moved among communities. To this end, two different cases have been generated. In the former, 10% of vertices in each community are randomly selected and assigned to the others. In the second case, 30% of vertices are selected to change their community membership at each time step. We consider 50 time steps and the accuracy of algorithm is the average NMIs over 50 runs.

With parameter α=0.8, β=0.95, and γ=0.1, we compare sE-NMF with other algorithms in terms of NMI. Fig.7 reports the NMIs of various algorithms for different dynamic networks, where (A) is for networks with z=5 and C=10%, (B) for networks with z=5 and C=30%, (C) for networks with z=6 and C=10%, and (D) for networks with z=6 and C=30%. It demonstrates that sE-NMF achieves the best performance in all four types of dynamic networks. Comparison between Fig.7 A and C shows that increasing the noise level affects the performance dramatically for FacetNet and the DYNMOGA algorithm, but it does not affect the sE-NMF algorithm, indicating that the sE-NMF algorithm is more robust than others. Similar tendency also occurs in Fig.7 B and D. Comparison between 10% and 30% perturbation shows that,

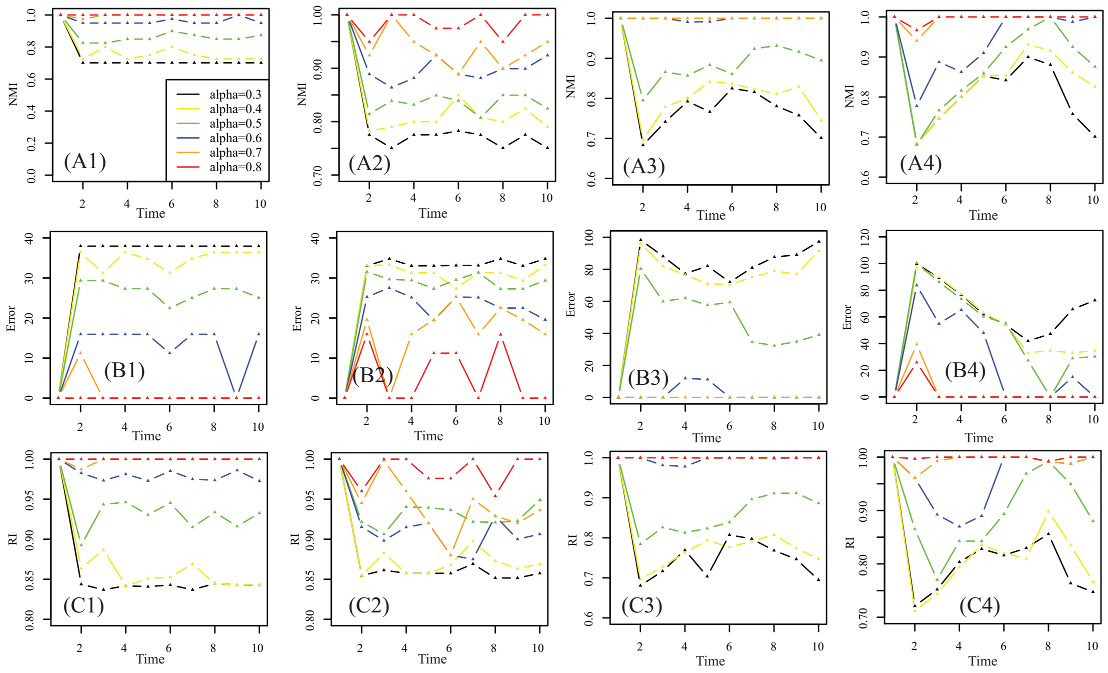

Fig. 3. Parameter effects of α in terms of NMI: (A1-A4) for SYN-FIX with z=3, SYN-FIX with z=5, SYN-VAR with z=3, and SYN-VAR with z=5, respectively; Error rate: (B1-B4) for SYN-FIX with z=3, SYN-FIX with z=5, SYN-VAR with z=3, and SYN-VAR with z=5, respectively; Rand index: (C1-C4) for SYN-FIX with z=3, SYN-FIX with z=5, SYN-VAR with z=3, and SYN-VAR with z=5, respectively.

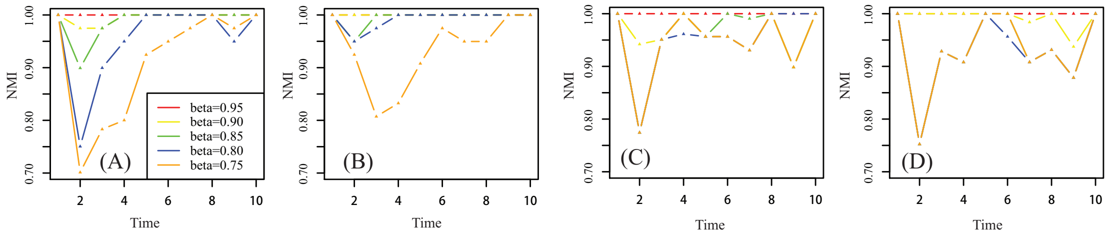

Fig. 4. Parameter effects of β: (A) SYN-FIX when z=3, (B)SYN-FIX when z=5, (C) SYN-VAR when z=3, and (D) SYN-VAR when z=5.

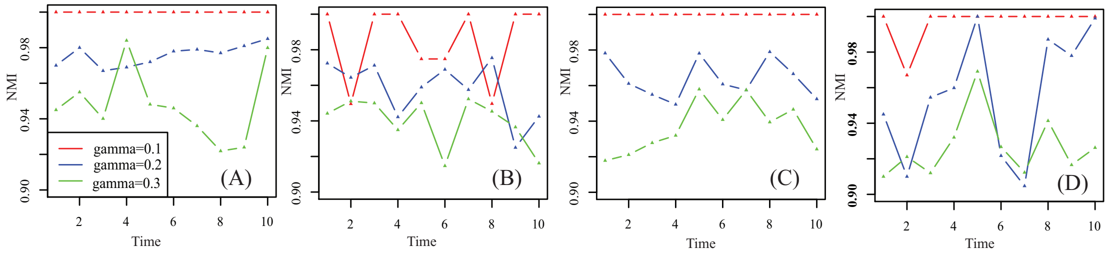

Fig. 5. Parameter effects of γ: (A) SYN-FIX with z=3, (B)SYN-FIX with z=5, (C) SYN-VAR with z=3 and (D) SYN-VAR with z=5.

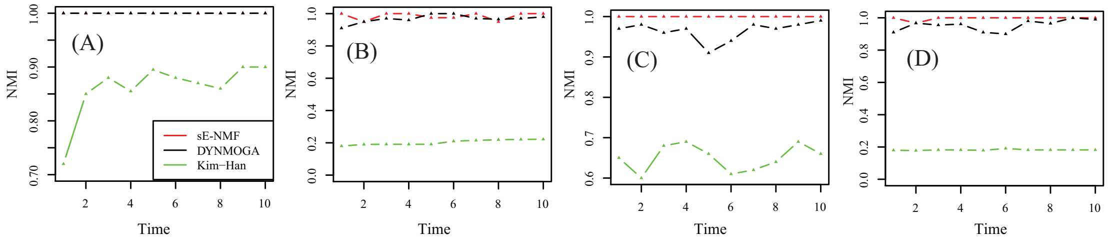

Fig. 6. Comparison among various algorithms: (A) SYN-FIX with z=3, (B)SYN-FIX with z=5, (C) SYN-VAR with z=3 and (D) SYN-VAR with z=5.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2017.2657752, IEEE Transactions on Knowledge and Data Engineering

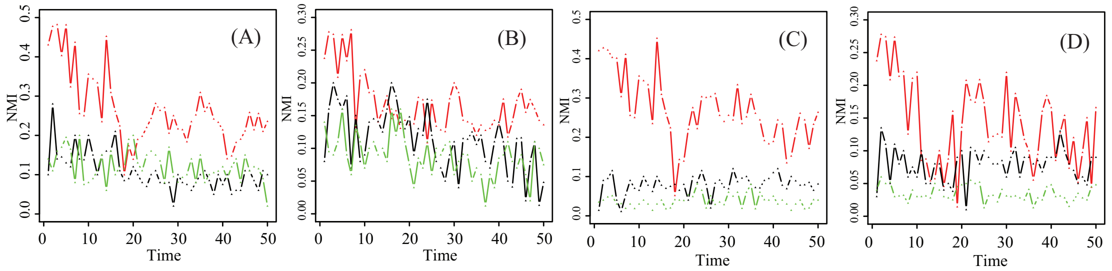

Fig. 7. Comparison among DYNMOGA, FacetNet and sE-NMF on dataset #2 with average degree = 16: (A) z=5 and C=10%, (B) z=5 and C=30%, (C) z=6 and C=10% and (D) z=6 and C=30%.

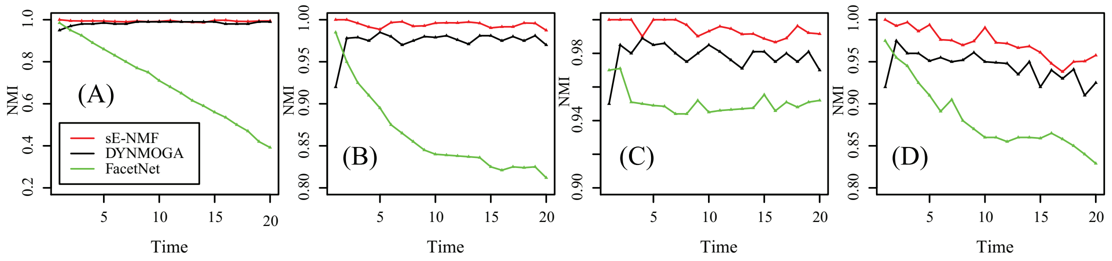

Fig. 8. Comparison among DYNMOGA, FacetNet and sE-NMF on synthetic dataset #3: (A) Birth and death, (B)Expansion and contraction, (C) Intermittent communities and (D) Merging and splitting.

even though the perturbation percentage dramatically decreases the accuracy, sE-NMF still outperforms the FacetNet and DYNMOGA algorithm. The result illustrates that the semi-supervision strategy not only improves the accuracy of algorithms, but also increases the robustness of algorithms, showing that the semi-supervision strategy is promising for clustering in dynamic networks.

6.4 Synthetic Dataset #3

The Greene dataset is adapted with four types of evolution events [63]: *Birth and death*: from the second time step on, 10% of new communities are created by removing vertices from other communities, and randomly removing 10% of the existing communities. *Expansion and contraction*: 10% of communities are randomly selected and expanded or contracted by 25% of size. The new vertices are chosen at random from the other communities for expansion. *Intermittent communities*: 10% of communities from the first time point step are hidden. *Merging and splitting*: at each time step, 10% of the communities are split, 10% of the communities are chosen, and coupled communities are merged.

sE-NMF takes dense subgraphs to construct the priori information. However, the density is insufficient to guarantee the quality of priori information because small subgraphs usually are dense, for example, the density for 2/3/4-cliques is 1. The reason why the small dense subgraphs affect the performance of accuracy because they are ubiquitous in random networks [69]. Therefore, we remove all these subgraphs whose size is ≤ 4 to ensure the quality of priori information, which affects the performance of sE-NMF.

We generate dynamic networks for each evolution event with the parameters settings: average degree 15, number of vertices in each network 1000, number of communities between 20 and 50, and mixing parameter is 0.2. Fig.8 depicts the NMIs of the compared algorithms on the four types of dynamic networks, which clearly show that both DYNMOGA and sE-NMF outperform FacetNet in all four evolution events, and our algorithm achieves the best performance in the four evolution events. We notice that the accuracy of FacetNet decreases dramatically as time increases, while DYNMOGA and sE-NMF are stable. The reason is that the FacetNet algorithm is based on the FCM framework, where the error in communities at the previous time steps affects the communities in the forthcoming time steps. However, both sE-NMF and DYNMOGA are based on the FCQ framework, which avoid the problem.

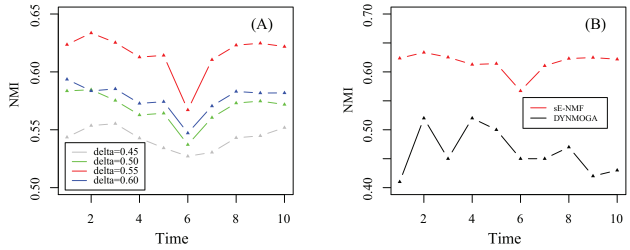

Fig. 9. Performance on cell phone networks: (A) effects of parameter δ in terms of accuracy, (B) comparison between the sE-NMF and DYNMOGA algorithms on the cell phone networks.

6.5 Cell phone calls networks

The previous three datasets are artificial networks, where the membership of the vertices is known. We next check how these algorithms discover the dynamic communities in real world dynamic networks. The cell phone calls[1] among the members of the fictitious Paraiso movement are used, covering ten days in June 2006. The cell phone network is constructed by regarding each individual as a vertex while

[1] http://www.cs.umd.edu/hcil/VASTchange08/

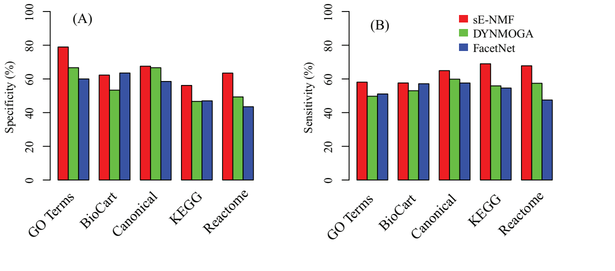

Fig. 10. Sensitivity and specificity of evolution modules obtained by various algorithms in the pathway enrichment analysis: (A) specificity, (B) sensitivity.

a call between a pair of members as an edge. For each day, a cell phone network is constructed. The total number of cell phones is 400.

Because the true community structure is unknown, we follow the same approach of *Lin et al.* [2], which is also adapted by the DYNOMGA algorithm [38]. We first discover the community structure by applying only the local cluster discovery procedure of the sE-NMF algorithm on the aggregated network. The obtained community structure is considered as the ground truth division.

Based on the ground truth division, we check how the sE-NMF algorithm selects the number of clusters k_t using spectral decomposition. Fig.9 A depicts how NMI changes as parameter δ increases from 0.45 to 0.60. It is easy to conclude that sE-NMF achieves the best performance when δ=0.55. The reason is that if δ is small, the spectrum of the matrix cannot capture the information of the original matrix, and if δ is large, the spectrum information is redundant. Thus, δ=0.55 reaches a good balance.

The NMI between the ground truth division and the communities obtained at each time step is considered as the accuracy, which is shown in Fig.9 B. It depicts the NMIs for the sE-NMF and DYNMOGA algorithms. It can be observed that the NMIs of the sE-NMF algorithm are significantly higher than those of the DYNMOGA algorithm. Specifically, the average NMI of the sE-NMF approach is 0.62, while that for the DYNMOGA method is 0.47. The results demonstrate that the proposed sE-NMF algorithm is also effective for discovering dynamic communities in social networks.

6.6 Large-scale breast cancer networks

The sizes of all previous dynamic networks are too small ($\leq$ 1000) to investigate the effectiveness of the algorithms. We employ the dynamic networks associated with breast cancer progression based on the gene expression data [6]. There are 4 time steps and the number of genes in the network at each time point is 9879.

Similar to cell phone call networks, there is no ground truth division of the dynamic communities. We evaluate the obtained dynamic communities using multiple reference annotations, including Gene Ontology [5], KEGG [7], Biocarta [8], Canonical pathways [9] and reactome pathway[2], with an immediate purpose to check how well the discovered dynamic modules enriched these known pathways.

To evaluate the statistical and biological significance of the dynamic communities, the P-value [56] is employed. In detail, given a dynamic community C with k proteins in a functional group F, the P-value is defined as

$$P - value = 1 - \sum_{i=0}^{k-1} \frac{\binom{|F|}{i}\binom{|V|-|F|}{|C|-i}}{\binom{|V|}{|C|}}, \tag{34}$$

where $|V|$ denotes the number of genes in the network. The functional homogeneity P-value is the probability that a given set of genes is enriched by a function merely by chance. All P-values are corrected by the Benjamini-Hochberg method [57] with a cutoff of 0.05.

Following the strategy in Ref. [6], we employ the specificity and sensitivity as criteria to evaluate the performance of the algorithms. Specificity is defined as the fraction of the predicted dynamic communities that are significantly enriched with at least one reference pathway, while the sensitivity is defined as the fraction of the reference pathways that are significantly enriched with at least one predicted dynamic community. Fig.10 reports the results of the specificity and sensitivity of the dynamic communities obtained by various methods. Our algorithm is better than others in both specificity and sensitivity, indicating that the proposed algorithm discovers the biological dynamic communities. For example, the specificity of sE-NMF is 78.9% for GO terms, while DYNMOGA is 64.7% and FacetNet is 59.7%. Generally speaking, DYNMOGA is better than FacetNet and sE-NMF is the best. sE-NMF provides an effective way to explore time-dependent networks associated with disease progression.

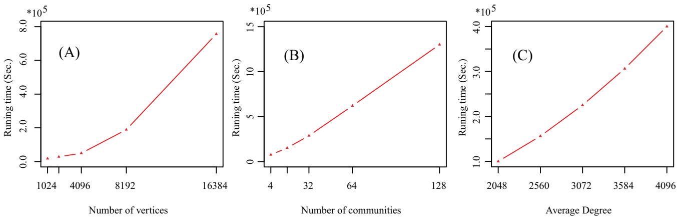

Fig. 11. Running time of the sE-NMF algorithm in seconds against: (A) number of vertices in the networks, (B) number of communities and (C) density of the networks.

6.7 Scalability analysis

Even though the NMF algorithm has been widely studied because of its excellent performance, it is criticized for its time complexity. There are three major factors related to the running time of the NMF algorithm: the size of the network, number of communities as well as sparsity of the network.

To evaluate the scalability of the sE-NMF algorithm, the synthetic dataset #2 is used. First of all, we analyze how the size of the network influences the running time of sE-NMF by increasing the number of vertices as 1024, 2048, 4096, 8192 and 16384 and the corresponding number of edges as 8327, 16600, 33340, 66624, and 133142 respectively. Fig.11 A reports the running time for 10 time steps, where the running time of sE-NMF is cubic in terms of the network size. For example, when the number of vertices is 8912 with 4 communities, the running time is about 52.5 hours. To study the influence of the number of communities on running time, we set the size of the network as 16384 and

[2] www.reactome.org

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2017.2657752, IEEE Transactions on Knowledge and Data Engineering

vary the number of communities as $\{4, 16, 32, 64, 128\}$. The running time for one step time is shown in Fig.11 B, where the running time for sE-NMF is linear in terms of the number communities. Finally, we want to study the influence of network sparsity on the accuracy of the sE-NMF algorithm. To generate the networks with various sparsity, we fix the size of the network as 16384 and increase the average degree of each vertex as $\{2048, 2560, 3072, 3584, 4096\}$. Fig.11 C demonstrates that running time of sE-NMF is quadratic in terms of sparsity.

It is worth to point out that the capabilities of sE-NMF for large networks could be degraded because there are many strategies for network sparsity without destroying the community structure [66]. Also, parallel computation is an alternative since evolutionary clustering is very suitable for the parallel platform. We also propose two strategies to accelerate the sE-NMF algorithm without sacrificing accuracy (supplementary).

7 Extension of sE-NMF

In this section, we investigate the possibility to extend sE-NMF under the PCM framework and discuss the situations where one of the frameworks is preferred.

7.1 sE-NMF under PCM

As shown in Fig.1, the PCM and PCQ differ on how to define the temporal cost CT. At time step t, the PCQ framework quantifies the temporal cost CT as the distance between networks at two subsequent time steps, while the PCM framework measures it as the distance between local clusters at two successive time steps. Therefore, we reformulate Eq.(26) as

$$A_t^* = \begin{cases} A_t + \alpha(A_t - Z_{t-1}Z_{t-1}^{'}) & \text{if } t > 1, \\ A_1 & \text{if } t = 1. \end{cases} \quad (35)$$

Comparison between Eq.(26) and (35) shows the only difference between the PCM and PCM sE-NMF is the strategy to define CT. Therefore, they have the same complexity. The algorithm for sE-NMF under the PCM framework is presented in Algorithm 2.

Algorithm 2 sE-NMF the under PCM framework

```
Input:
    $\mathcal{G}$: Dynamic network;
    $\{k_i\}_{i=1}^{T}$: number of the local clusters;
    $\beta$: weight for the priori information;
Output:
    $\{Z_t\}_{t=1}^{T}$: local clusters and evolutionary community.
  Part I: Local cluster discovery
  For each time step $t$, construct the $A_t^*$ according to Eq.(35);
  Use spectral clustering algorithm to obtain the partial information $Z_t^{rc}$;
  Construct the partial matrix $\widehat{A}_t^*$ according to Eq.(27);
  Use NMF to obtain the local clusters discovery $\{Z_t\}_{t=1}^{T}$ according to Eqs.(29,30);
  Part II: Evolutionary community discovery
  Construct the $T$-partite graph using the local clusters $\{Z_t\}_{t=1}^{T}$;
  Discover the evolution communities using the greedy search algorithm;
  return
```

7.2 Comparing PCQ and PCM sE-NMF

There are two intuitive questions for sE-NMF under two frameworks: 1) do they have similar performance, and 2) if they do not have similar performance, shall we know which situations one of the frameworks is preferred. Chi *et al.* [37] demonstrated that the evolutionary spectral clustering algorithms under PCM and PCQ have similar performance. Here, we check whether sE-NMF under two frameworks are similar in terms of accuracy. Fig.12 contains NMIs for the sE-NMF algorithms under PCM and PCQ on dataset #1, demonstrating that the two algorithms differ greatly in terms of NMI. More specifically, the sE-NMF under PCQ is much better than the one under PCM on both the SYN-FIX and SYN-VAR networks. The reason is that the PCQ framework is based on the similarity between two subsequent networks, while the PCM one is based on the local clusters at the previous time step, introducing noise into the communities at the current time. Therefore, the accuracy of the sE-NMF algorithm under PCM is partially determined by the algorithms generating the local clusters at previous time steps.

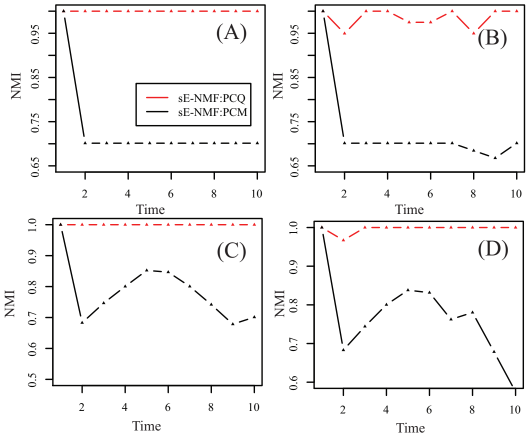

Fig. 12. Comparison of sE-NMF under the PCQ and PCM frameworks on synthetic dataset #1: (A) SYN-FIX when z=3, (B)SYN-FIX when z=5, (C) SYN-VAR when z=3, and (D) SYN-VAR when z=5.

Then, we ask under what situations were the sE-NMF algorithms under the PCQ and PCM framework so that they have similar performance. To check whether the network dynamics contribute to the performance, we use dataset #3 by decreasing C from 10 to 6. Fig.13 A depicts ΔNMI between PCQ and PCM sE-NMF with z=5, where ΔNMI = $|NMI_{PCQ} - NMI_{PCM}|$ is the difference in accuracy. It is easy to conclude that as C decreases, ΔNMI decreases, implying that the two algorithms get closer in terms of accuracy. When C=6, ΔNMI is less than 0.05, demonstrating that they have similar performance. A similar tendency occurs for dynamic networks with z=6 (Fig.13 B). Therefore, we suggest users choose the sE-NMF algorithm under the PCQ framework when the dynamics of the networks are dramatic, otherwise choose either of them.

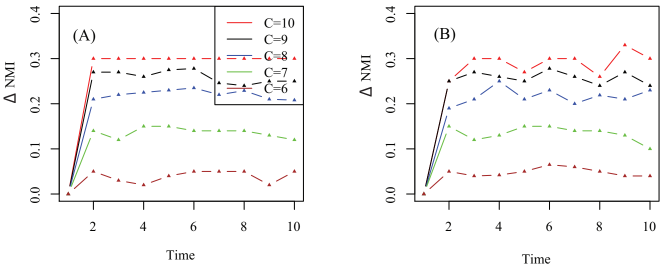

Fig. 13. Comparison between PCQ and PCM sE-NMF on synthetic dataset #3 with various dynamics: (A) z=5, (B) z=6.

8 Conclusion

We have recently witnessed the emergence of dynamic networks in the real world, which provides great opportunities to identify the evolving patterns. Although various clustering algorithms for dynamic communities have been proposed, few attempts have been made to study the relationship among them. Indeed, it has considerable merits to investigate this relationship since it provides an insight into understanding the dynamics of complex networks as well as algorithms themselves.

In this paper, we proved the equivalence among the evolutionary modularity density, sE-NMF and evolution spectral clustering algorithm, which extends the theory of evolutionary spectral clustering [37]. It also further explained why the topology structure could be used for dynamic community detection. Based on the equivalence, a semi-supervised nonnegative matrix factorization for dynamic community detection was proposed by incorporating a partial information into NMF. The algorithm is a general framework for semi-supervised evolutionary clustering.

We would like to close this paper by listing some possible further research directions: First of all, the algorithm analysis implies that it is still unacceptable for large and dense networks, particularly in the big data era [58]. Even though great efforts have been devoted to the acceleration of NMF [59], [60], no attempt has been made to accelerate the algorithms via network reduction, i.e. reducing the number of vertices in the network hence preserving the evolution patterns. Secondly, Morone *et al.* [61] pointed out that the whole frame of interconnections in complex networks hinges on a specific set of structural vertices whose size is much less than the total size. Making the best use of the percolation theory to characterize dynamic communities is also very promising. Extending the evolutionary algorithm to discover dynamic communities in the directed networks is also interesting. Designing efficient methods to solve these issues would be promising.

Acknowledgment

This work was supported by the NSFC (Grant No. 61502363, 81271569), Natural Science Funding of Shaanxi Province (Grant No. 2016JQ6044), Fundamental Research Funding of Central Universities (Grant no. JB160306) and Natural Science Basic Research Plan in Ningbo City (Program No. 2016A610034). The authors wish to thank professor K.V. Deneen for proofreading the paper and appreciate the reviewers for their valuable suggestions and comments.

References

[1] G. Palla, A. Barabási, T. Vicsek, Quantifying social group of evolution, Nature, 466 (2007) 664-667.

[2] Y.R. Lin, S. Zhu, H. Sundaram, B.L. Tseng, Analyzing communities and their evolutions in dynamic social networks, ACM Transactions on Knowledge Discovery from Data, 3 (2009) 18.

[3] M. Girvan, M.E.J. Newman, Community structure in social and biological networks, Proc. Nat. Acad. Sci. USA, 99 (2002) 7821-7826.

[4] P. Lawrence, B. Sergey, M. Rajeev, W. Terry, The PageRank citation ranking: bringing order to the Web, Technical Report, (1998).

[5] M. Ashburner, C.A. Ball, J.A. Blake, D. Botstein et al. Gene ontology: tool for the unification of biology, Nature Genetics, 25 (2000) 25-29.

[6] X.K. Ma, L. Gao, K. Tan, Modeling disease progression using dynamics of pathway connectivity, Bioinformatics, 36 (2014) 2343-2350.

[7] M. Kanehisa, S. Goto, Y. Sato, M. Furumichi, M. Tanabe, KEGG for integration and interpretation of large-scale molecular data sets, Nucleic Acids Research, (2012) D109-D114.

[8] D. Nishimura, BioCarta, Biotech. Softw. Internet Rep., 2 (2001) 117–120.

[9] A. Subramanian, P. Tamayo, V.K. Mootha, S. Mukherjee, B.L. Ebert et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles, Proc. Natl Acad. Sci. USA, 102 (2005) 15545-15550.

[10] A.H. Tong, B. Drees, G. Nardelli, G.D. Bader, B. Brannetti, L. Castagnoli, M. Evangelista, S. Ferracuti, B. Nelson, S. Paoluzi, M. Quondam, A. Zucconi, C.W. Hogue, S. Fields, C. Boone, G. Cesareni, A combined experimental and computational strategy to define protein interaction networks for peptide recognition modules, Science, 295 (2002) 321-324.

[11] A. Arenas, A. Fernández, S. Fortunato, Motif-based communities in complex networks, *Journal of Physic A Mathematical and Theoretical* 41(2008) 224001.

[12] M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, Physical Review E, 69 (2004) 026113.

[13] Z.P. Li, S.H. Zhang, R.S. Wang, Quantative function for community detection, Physical Review E, 77 (2008) 036109.

[14] M.E.J. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. USA, 103 (2006): 8577C8696.

[15] S. Fortunato, M. Barthélemy, Resolution limit in community detection, Proc. Nat. Acad. Sci. USA, 104 (2007) 36-41.

[16] J.M. Pujol, J. Béjar, J. Delgado, Clustering algorithms for determining community in large networks *Phys. Rev. E* 74 (2006)016107.

[17] S.H. Zhang, R.S. Wang, X.S. Zhang, Uncovering fuzzy community structure in complex networks, *Phys. Rev. E* 76(2007)046103.

[18] S.H. Zhang, R.S. Wang, X.S. Zhang, Identification of overlapping community structure in complex networks using fuzzy c-means clustering, *Phys. A* 374(2007)483–490.

[19] M.E.J. Newmann, Detecting community structure in networks, *Eur. Phys.J.B*,38 (2008) 321–330.

[20] A. Medus, G. Acuña, C.O. Dorso, Detection of community structures in networks via global optimization, *Phys. A* 358 (2005) 593–604.

[21] J. Reichardt, S. Bornholdt, Detecting fuzzy community structures in complex networks with a Potts model *Phys. Rev. Lett.* 93 (2004) 218701.

[22] J. Duch, A. Arenas, Community identification using extremal optimization *Phys. Rev. E*,72 (2005) 027104.

[23] Y. Yang, J.X. Yu, H. Gao, J. Pei, J. Li, Mining most frequently changing component in evolving graphs, World Wide Web Journal, 17(3) (2014) 351-376.

[24] C. Yang, X. Zhang, C. Zhong, C. Liu, J. Pei, K. Ramamohanarao, J. Chen. A spatiotemporal compression based approach for efficient big data processing on Cloud. Journal of Computer and System Sciences, 80(8) (2014) 1563-1583.

[25] Z. Xing, J. Pei, P.S. Yu. Early classification on time Series. Knowledge and Information Systems, 31(1) (2012) 105-127.

[26] Y.F. Lin, H.H. Chen, V.S. Tseng, J. Pei. Reliable Early classification on multivariate time series with numerical and categorical attributes. In Proceedings of the 19th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'15), 2015, pp.199C211.

[27] M.E.J. Newman, The structure of scientific collaboration networks, Proc. Natl Acad. Sci. USA, 98 (2001) 404-409.

[28] X. Ma, L. Gao, X. Yong, L. Fu, Semi-supervised clustering algorithm for community structure detection in complex networks, Physica A, 389 (2010) 187-197.

[29] B.D. Craene, G. Berx, Regulatory networks defining EMT during cancer initiation and progression, Nature Review Cancer, 13 (2013) 97-110.

[30] T. Jindal, P. Giridhar, L.A. Tang, J. Li, J. Han, Spatio temporal periodical pattern mining in traffic data, Proceedings of 2013 ACM SIGKDD int. Workshop on Urban computing, Chicago, IL (2013) pp.1-8.

[31] E. Lieberman, C. Hauert, M.A. Nowak, Evolutionary dynamics on graphs, Nature, 433 (2005) 312-316.

[32] J.G. Lee, J. Han, X. Li, A unifying framework of mining trajectory patterns of various temporal tightness, IEEE Transaction on Knowledge and Data Engineering, 27 (2015) 1478-1490.

[33] M. Gupta, J. Gao, C.C. Aggarwal, J. Han, Outlier detection for temporal data: a survey, IEEE Transaction on Knowledge and Data Engineering, 26 (2014) 2250-2267.

[34] P. Holme, J. Saram$\ddot{a}$ki, Temporal networks, Physics Reports, 519 (2012) 97-125.

[35] P.J. Mucha, T. Richardson, K. Macon, M.A. Porter, J. Onnela, Community structure in time-dependent, multiscale and multiple networks, Science, 328 (2010) 876-878.

[36] W. Zhang, R. Li, D. Feng, A. Chernikov, N. Chrisochoides, C. Osgood, S. Ji, Evolutionary soft co-clustering: formulations, algorithms, and applications, Data Mining and Knowledge Discovery, 95 (2015) 765-791.

[37] Y. Chi, X.D. Song, D.Y. Zhou, K.J. Hino, B.L. Tseng, On evolutionary spectral clustering, ACM Transactions on Knowledge Discovery from Data, 3 (2009) 17.

[38] F. Folino, C. Pizzuti, An evolutionary multiobjective approach for community discovery in dynamic networks, IEEE Transactions on Knowledge and Data Engineering, 26 (2014) 1838-1852.

[39] S. Asur, S. Parthasarathy, D. Ucar, An event-based framework for characterizing the evolutionary behavior of interaction graph, ACM Transactions on Knowledge Discovery from Data, 3 (2009) 16.

[40] D. Chakrabarti, R. Kumar, A. Tomkins,Evolutionary clustering, Proceedings of 12th ACM Int. Conf. KDD, (2006) pp.554-560.

[41] M. Kim, J. Han, A particle-and-density based evolutionary clustering method for dynamic networks, Proceedings of International Conference Very Large Database, (2009) pp.622-633.

[42] R. Kumar, J. Novak, A. Tomkins, Structure and evolution of online social networks, Proceedings of 12th ACM Int. Conf. KDD, (2006) pp.611-617.

[43] J. Sun, C. Faloutsos, S. Papadiitriou, P. Yu. GraphScope: parameter-free of large time evolving-graph, Proceedings of 12th ACM Int. Conf. KDD, (2007) pp.687-696.

[44] L. Tang, H. Liu, J. Zhang, Z. Nazeri, Identifying evolving groups in dynamic multimode networks, IEEE Transactions on Knowledge and Data Engineering, 24 (2012) 72-85.

[45] C.C. Aggarwal, J. Han, P.S. Yu, A framework for clustering evolving data streams, Proceedings of the 12th VLDB Conference, (2003) pp.81–92.

[46] C. Gupta, R. Grossman, GenIc: a single pass generalized incremental algorithm for clustering, Proceedings of the SIAM International Conference on Data Mining, (2004) pp.147–153.

[47] Y. Li, J. Han, Clustering moving objects, Proceedings of the 10th ACM SIGKDD Conference, (2004) pp.617–622.

[48] H. Ning, W. Xu, Y. Gong, T. Huang, Incremental spectral clustering with application to monitoring of evolving blog communities, Proceedings of the SIAM International Conference on Data Mining, (2007) pp.261-672.

[49] C. Aggarwal, K. Subbian, Evolving network analysis: A survey, ACM Computer Survey, 47 (2014) 10.

[50] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature, 401 (1999) 788-791.

[51] P. Paatero, U. Tapper, Positive matrix factorization: A non-negative factor model with optimal utilization of error, Environmetrics, 5 (1994) 111-126.

[52] C.J. Lin, Projected gradient methods for non-negative matrix factorization, Neural Computation, 19 (2007) 2756-2779.

[53] D. Mavroeidis, Accelerating spectral clustering with partial supervision, Data Mining and Knowledge Discovery, 21 (2010) 241-258.

[54] L. Duch, J. Diaz-GuileramA, A. Arenas, Comparing community structure identification, Journal of Statistical Mechanics: Theory and Experiment, (2005) P09008.

[55] X.K. Ma, L. Gao, Non-traditional spectral clustering algorithms for the detection of community structure in complex networks: a comparative analysis, Journal of Statistical Mechanics: Theory and Experiment, (2011) P05012.

[56] A.D. King, N. Prulj, I. Jurisica, Protein complex prediction via cost-based clustering, Bioinformatics, 20 (2004) 3013-3020.

[57] Y. Benjamini, Y. Hochberg, Controlling the false discovery rateCa practical and powerful approach to multiple testing, J. R. Stat. Soc. B Methadol., 57 (1995) 289-300.

[58] X. Wu, X. Zhu, G. Wu, W. Ding, Data mining with big data, IEEE Transaction on Knowledge and Data Engineering, 26 (2014) 97-107.

[59] A. Pascual-Montano, J.M. Carazo, K. Kochi, D. Lehmann, R.D. Pascual-Marqui, Nonsmooth nonnegative matrix factorization(nsNMF), IEEE Transaction on Pattern Analysis and Machine Learning, 28 (2006) 403-415.

[60] P.O. Hoyer, Non-negative matrix factorization with sparseness constraits, The Journal of Machine Learning, 5 (2004) 1457-1469.

[61] F. Morone, H.A. Makse, Influence maximization in complex networks through optimal percolation, Nature, 524 (2015) 65-68.

[62] G.H. Golub, C.F. Van Loan, Matrix Computation, Johns Hopkins University Press, 1996.

[63] D. Greene, D. Doyle, P. Cunningham, A framework for clustering evolving data streams, Proceedings of International Conference Advanced Social Network Analysis Mining, (2010) pp.176-183.

[64] C. Ding, X. He, H.D. Simon, On the equivalence of nonnegative matrix factorization and spectral clustering, Proceedings of SIAM International Conference on Data Mining, (2005) pp.606-610.

[65] W.M. Rand, Objective criteria for the evaluation of clustering methods, Journal of the American Statistical Association, (1971) 66(336):846C850.

[66] I. Streinu, L. Theran. Sparsity-certifying Graph Decompositions, Graphs & Combinatorics, (2009) 25(2):219C238.

[67] K. Yu, S. YU, V. Tresp. Soft clustering on graphs, Advances in Neural Information Processing Systems ser. NIPS2005.

[68] S. Ji, W. Zhang, J. Liu. A sparsity-inducing formulation for evolutionary co-clustering, Proc. of 18th ACM SIGKDD KDD, (2012) pp.334-342.

[69] B. Bollob$\acute{A}$s, Random graphs, Academic Press, 1985.

Xiaoke Ma received his Ph.D. degree in computer science from Xidian University in 2012. He was a post-doctor at the University of Iowa (USA) during 2012-2015. He is an associate professor of School of computer science and technology, Xidian University (P.R.China). His research interests include machine learning, data mining and bioinformatics. He is an ad hoc reviewer for many international journals and publishes more than 20 papers in the peer-reviewed international journals, such as IEEE Trans. Knowledge and Data Engineering, Pattern Recognition, Information Sciences, JSTAT, Physica A, Bioinformatics, PLoS Computational Biology, Nuclear Acids Research, IEEE Trans. Computational Biology and Bioinformatics, IEEE Trans. NanoBioScience.

Di Dong received Ph.D. degree in Computer Science from the Chinese Academy of Sciences. At present, he is an assistant professor in the Institute of Automation, Chinese Academy of Sciences. His research interests include data mining and medical image processing. He has published 10 papers in international journals.