DCNMF: Dynamic Community Discovery with Improved Convex-NMF in Temporal Networks

Limengzi Yuan[1,2], Yuxian Ke[1], Yujian Xie[1], Qingzhan Zhao[1,2], and Yuchen Zheng[1,2](✉)

[1] College of Information Science and Technology, Shihezi University, Shihezi, China
[2] Geospatial Information Engineering Research Center, Xinjiang Production and Construction Corps, Shihezi, China
{ylmz,zqz_inf}@shzu.edu.cn

Abstract. For its crucial importance in the study of temporal networks, techniques for detecting community structures and tracking evolutionary behaviors have been developed. Among these techniques, evolutionary clustering is an efficient method which unveils substructures in complex networks and models the evolution of a system. Most research works in this domain mainly employ Semi-NMF to discover evolving communities. However, in some cases, it can not jointly maintain the quality of community detection and track the temporal evolution infallibly. In this paper, we present a novel community discovery model based on an evolutionary clustering framework using convex non-negative matrix factorization (Convex-NMF), called DCNMF. It is an improvement of Semi-NMF when applied in temporal networks to detect and track evolutionary communities. The proposed model, with temporal smoothness constraint considering the Convex-NMF results, is more accurate and robust both than the evolutionary clustering method based on Semi-NMF and some other existing methods. Specifically, we adopt the gradient descent algorithm to optimize the objective function and prove the correctness and convergence of the algorithm. Experimental results on several synthetic benchmarks and real-world networks show the effectiveness of the proposed method in discovering communities and tracking evolution in dynamic networks.

Keywords: Dynamic community discovery · Temporal networks · Convex non-negative matrix factorization

1 Introduction

Many social, physical, technological, and biological systems can be modeled as networks composed of numerous interacting parts [19]. As an increasing amount of time-resolved data have become available, it has become increasingly

H. Gao and X. Wang (Eds.): CollaborateCom 2021, LNICST 406, pp. 460–475, 2021.
https://doi.org/10.1007/978-3-030-92635-9_27

important to develop methods to quantify and characterize the dynamic properties of temporal networks [8]. Real-world examples of temporal networks include person-to-person communication (e.g., via mobile phones [20]) and one-to-many information dissemination (such as Twitter networks [6]). Analyzing such temporal networks can uncover the important phenomenon and characterize the properties of networks of dynamical units. Dynamic community discovery (DCD) is considered as a sort of effective tool for discovering the structure of complex networks, and ultimately extracting useful information from them. Recently, it is also applied in some edge computing scenarios [14]. A major problem for DCD is to identify stable network decompositions, which comes from the very nature of communities. It lies in the fact that, generally, it is difficult to know if the differences observed on complex networks between consecutive time slices are due to the algorithm's instability or the communities' evolution. A great variety of methods have been proposed to address this problem.

In this paper, we propose a novel model which detects and tracks dynamic communities with convex non-negative matrix factorization (Convex-NMF) based on an evolutionary clustering framework. In most of existing DCD works using non-negative matrix factorization (NMF), it is more common to apply Semi-NMF to address concrete problems. Considering the Semi-NMF form $\mathbf{X} = \mathbf{F}\mathbf{G}^\top$, Convex-NMF is obtained when the basis vectors of $\mathbf{F}$ are constrained to be convex combinations of the data points [2]. As regards the community membership matrices derived from NMF methods, Convex-NMF solutions are more sparse and orthogonal than Semi-NMF solutions, which is reasonably consistent with the network rules that the node propensities of belonging to communities would not be highly ambiguous. Consequently, we argue that Convex-NMF would be a better option to interpret substructures in evolving real-world networks. Considering the convincing performance of Convex-NMF in clustering problems as well as static community detection, we employ it in reconstructing the network topology task and constraining temporal smoothness for DCD in the proposed model. More concretely, the main contributions of our work are as follows:

- We propose a unified model DCNMF which detects communities and analyzes their evolution. Using probabilistic community membership and temporal smoothness constraints, the proposed model unveils latent community structure and discovers network changes.
- By employing Convex-NMF to obtain community membership matrices, the proposed model achieves community discovery results that are of a better quality than Semi-NMF solutions.
- The proposed model is easily extended to the networks whose number of nodes and communities may change over time, which is a fairly common phenomenon in temporal networks.
- An optimal algorithm is proposed to solve the obtained objective function. We prove that the algorithm is guaranteed to converge and verify the correctness of the algorithm. Analysis of its computational efficiency is also provided.

The rest of the paper is organized as follows. We review related work of dynamic community detection in Sect. 2. In Sect. 3, the proposed gradient descent algorithm and its theoretical analysis are provided. Experimental results performed on synthetic and real-world data are presented in Sect. 4. The conclusion and discussion follow in Sect. 5.

2 Related Work

Recently, considerable research works have been devoted to discover communities in temporal networks. The available methods can be classified into heuristic algorithms and low rank approximation approaches. Since the objective of temporal community detection is typically NP-hard to optimize, some methods employed heuristics to find sets of nodes which can be understood or interpreted as real communities, including two-step methods [21,24–26] and multi-objective optimization algorithms [3,4,10,16–18]. However, two-step methods extract community structure from each time step independently, which often results in community structure with high temporal variation. In addition, the major drawback of the multi-objective approaches is that the random generation of initial population will greatly increase the search space and hence cause high spatial and temporary complexity.

To overcome the aforementioned problems, low rank approximation based methods [11,12,15,22] (e.g. evolutionary clustering [1]) simultaneously optimized the community discovery accuracy and drift based on the temporal smoothness framework. These methods transformed the time slices of network to the counterparts of each node, which can be used to create an alternative to graph drawings for visualization of node dynamics. Among them, matrix factorizations are widely applied for time-varying community exploration and detection in time-evolving graph sequences. Specifically, NMF is well satisfied with networks, for the reason that most of their edges which commonly correspond to flows, capacity, or binary relationships, are non-negative. Capable of extracting inherent patterns and structures in high dimensional data, the NMF-based methods have become one of the hottest research topics in community discovery.

The NMF-based methods is the most relevant type of approaches to our work, with innate interpretability and good performance in practice. Wu et al. [22] introduced hypergraph in NMF model and utilized the higher-order relationship among the points to promote the clustering performance. Li et al. [11] proposed a method which is based on semi-supervised matrix factorization and random walk to execute community partition. However, these two methods both have the same problem that the number of communities needs to be known as a prior information. Hong Lu et al. [15] used an improved density peak clustering to obtain the number of cores as the pre-defined parameter of NMF and adopted non-negative double singular value decomposition initialization, which can rapidly reduce the approximation error of NMF.

The proposed algorithm DCNMF falls into the category of evolutionary clustering based methods, which discovers communities at time t on the basis of both the topology of the network at t and community structures found previously. It introduces a parameter $\alpha \in [0, 1]$ to determine the trade-off between a solution to optimal community detection at t and a solution for maximizing the similarity with the result at $t-1$. DCNMF hence is capable of coping with the instability problem, while not diverges much from the common community discovery. Moreover, it takes advantage of the partition searched for at time step $t-1$ to speed up the community discovery at time step t.

3 Algorithm

In this section, we introduce the proposed model, derive the optimization rules, and analyze the complexity of the algorithm.

3.1 Notation

In this paper, bold uppercase letters will donate matrices, e.g. $\mathbf{X}$, bold lowercase letters will donate column vectors, e.g. $\mathbf{x}$, while operators $(\cdot)^\top$ will stand for matrix transposition, e.g. $\mathbf{X}^\top$. Both x_{ij} and $(X)_{ij}$ represent the Entry (i, j) of the matrix $\mathbf{X}$, and the Frobenius norms will be represented by $\|\cdot\|_F$.

Consider a dynamic N-node network whose time-varying structures are captured by the time-series adjacency matrices $\{\mathbf{X}^t \in \mathbb{R}^{N\times N}\}_{t=1}^T$. $x_{ij}^t = 1$ if there is an edge from node i to node j at time t, and $x_{ij}^t = 0$ otherwise. We assume that the dynamic network is undirected, i.e. $x_{ij}^t = x_{ji}^t$, and there are no self-edges, i.e. $x_{ii}^t = 0$.

3.2 The Unified DCNMF Model Formulation

Considering the observed network at time t, denoted by $\mathbf{X}^t$, the non-negative data matrix $\mathbf{X}^t$ can be factorized into $\mathbf{X}^t\mathbf{W}^t{\mathbf{G}^t}^\top$, i.e. $\mathbf{X}^t \approx \mathbf{X}^t\mathbf{W}^t{\mathbf{G}^t}^\top$, with the constraints that $\mathbf{W}^t$ and $\mathbf{G}^t$ are non-negative. In the factorization, $\mathbf{W}^t$ can be considered as the node weight matrix of all nodes and $\mathbf{G}^t$ can be considered to be a community membership matrix with $\mathbf{G}_{ij}^t$ denoting the probability that the node i belongs to the community j. Specifically, $\mathbf{F}^t \approx \mathbf{X}^t\mathbf{W}^t$ can be considered to be a centroid matrix in which each column represents a community central node. As a result, we derive the following function in matrix formulation at time t:

$$\min_{\mathbf{X}^t \geq 0} \|\mathbf{X}^t - \mathbf{X}^t\mathbf{W}^t{\mathbf{G}^t}^\top\|_F^2. \tag{1}$$

Here, we restrict central node column vectors to convex combinations of the columns of $\mathbf{X}^t$ to achieve good interpretability of obtained matrices. For one thing, the basic matrix columns would capture the notion of central nodes whose movements often influence the drifts of nodes who have close relationship with them. For another, restricted constraints lead to the desired NMF solution that community membership matrix $\mathbf{G}^t$ is more sparse and orthogonal, which gives sharper indicators of the community.

We impose the temporal smoothness constraints on community membership matrices to regularize the community structure, so that it is less likely to change dramatically in terms of the community memberships from time $t-1$ to t. The temporal cost is defined as the difference between the community membership matrices at time $t-1$ and that at time t. Regarding both the snapshot cost of modeling network topologies and the temporal cost of smoothness constraint, the cost function can be defined as the sum of community detection quality and historical cost. To achieve evolutionary clustering for DCD, we solve this by maximizing the community detection quality of current time-stamp and minimizing the historical cost, then the obtained cost function is as follows:

$$\min_{\mathbf{W}^t \geq 0, \mathbf{G}^t \geq 0} \|\mathbf{X}^t - \mathbf{X}^t \mathbf{W}^t {\mathbf{G}^t}^\top\|_F^2 + \alpha \|\mathbf{X}^{t-1} - \mathbf{X}^{t-1} \mathbf{W}^{t-1} {\mathbf{G}^t}^\top\|_F^2, \tag{2}$$

where, α is a temporal smoothness trade-off parameter.

In experiments, for each time step, we obtain a random initial value of $\mathbf{W}_0^t$ and $\mathbf{G}_0^t$ by setting $\alpha = 0$, then we restart the optimization with $\mathbf{W}^t = \mathbf{W}_0^t$ and $\mathbf{G}^t = \mathbf{G}_0^t$.

Extensions. In this subsection, we introduce two extensions to the proposed unified DCNMF model in order to handle the insertion and removal of nodes and communities.

Assume that at time t, n_1 nodes are removed from and n_2 nodes are inserted into the dynamic network. We first handle the n_1 removed nodes by removing the corresponding n_1 rows from $\mathbf{W}^{t-1}$ and $\mathbf{G}^{t-1}$ in Eq. (2) and from $\mathbf{W}^t$ and $\mathbf{G}^t$ in the last item of Eq. (2). After that, to add n_2 nodes, we pad n_2 rows of zeros to $\mathbf{W}^{t-1}$ and $\mathbf{G}^{t-1}$ in the second item of Eq. (2). Finally, we scale the vector $\mathbf{1}_n$ in Eq. (2) to get $\mathbf{1}_{n'}$. The basic idea behind this heuristic is that we assume that these n_2 nodes already exist as isolated nodes at time $t-1$. Moreover, we just preserve the membership of unchangeable nodes between successive time steps.

Assume that at time t, c_1 communities disappear and c_2 communities emerge in a dynamic network. To handle the c_1 disappearing communities, we remove the corresponding c_1 columns from $\mathbf{W}^t$ and $\mathbf{G}^t$ in Eq. (2). The basic idea of the removal is that considering a disappearing community c_1, we assume that there are no nodes belonging to other communities at time t will join it, and also all those nodes belonging to it at $t-1$ will leave and change to other communities. Even more intuitively, it is equivalent to assuming that the disappearing communities at time t split or combine with other communities. In order to add c_2 new communities, we add c_2 columns to $\mathbf{W}^t$ and $\mathbf{G}^t$ in Eq. (2). The purpose of the addition is that the nodes joining to emerging communities at time t come from previous communities.

3.3 Optimization

To solve the objective function in the Eq. (2), we propose an iterative algorithm using following updating rules which are obtained by using auxiliary functions.

The algorithm iteratively updates $\mathbf{W}^t$ with $\mathbf{G}^t$ fixed and then $\mathbf{G}^t$ with $\mathbf{W}^t$ fixed using following updating rules. And in such a way, the objective function defined in Eq. (2) is monotonically decreased and therefore converges to an optimal solution. The updating rules are as follows,

$$\mathbf{w}_{ij}^t \leftarrow \mathbf{w}_{ij}^t\left(\frac{({\mathbf{X}^t}^\top \mathbf{X}^t \mathbf{G}^t)_{ij}}{({\mathbf{X}^t}^\top \mathbf{X}^t \mathbf{W}^t {\mathbf{G}^t}^\top \mathbf{G}^t)_{ij}}\right), \tag{3}$$

then normalize such that $\sum_k \mathbf{w}_{ik} = 1, \forall i$,

$$\mathbf{g}_{ij}^t \leftarrow \mathbf{g}_{ij}^t\left(\frac{({\mathbf{X}^t}^\top \mathbf{X}^t \mathbf{W}^t + \alpha {\mathbf{X}^{t-1}}^\top \mathbf{X}^{t-1} \mathbf{W}^{t-1})_{ij}}{(\mathbf{G}^t {\mathbf{W}^t}^\top {\mathbf{X}^t}^\top \mathbf{X}^t \mathbf{W}^t + \alpha \mathbf{G}^t {\mathbf{W}^{t-1}}^\top {\mathbf{X}^{t-1}}^\top \mathbf{X}^{t-1} \mathbf{W}^{t-1})_{ij}}\right). \tag{4}$$

The overall procedure of DCNMF can be described as Algorithm 1. Algorithm 1 is able to guarantee that the objective function Eq. (2) converges to a local minimum, and the proof will be presented in the next section. After obtaining $\{\mathbf{W}^t\}_{t=1}^T$ and $\{\mathbf{G}^t\}_{t=1}^T$, we can use $\{\mathbf{G}^t\}_{t=1}^T$ to get the final disjoint communities at each time step and analyze the dynamic behaviors of communities between time intervals.

Algorithm 1. DCNMF

Input: $\mathbf{X}^t, \mathbf{X}^{t-1}, K, \alpha$
- $\mathbf{X}^t$: The adjacency matrix of the network at time step t;
- $\mathbf{X}^{t-1}$: The adjacency matrix of the network at time step $t-1$;
- K: The number of communities;
- α: The temporal smoothness trade-off parameter.

Output: $\mathbf{W}^t, \mathbf{G}^t$
- $\mathbf{W}^t$: The weight matrix of nodes at time step t;
- $\mathbf{G}^t$: The community membership matrix at time step t.

1: Initialize $\mathbf{W}^t, \mathbf{G}^t$ with random values.
2: **repeat**
3: Update $\mathbf{W}^t$ using Eq. (3) with $\mathbf{G}^t$ fixed.
4: Update $\mathbf{G}^t$ using Eq. (4) with $\mathbf{W}^t$ fixed.
5: Normalize such that $\sum_k \mathbf{W}_{ik} = 1, \forall i$.
6: **until** Convergence

Besides, the optimal solution to problem Eq. (2) is constructed from optimal solutions to two subproblems of Eq. (2). We prove the correctness of updating rules Eq. (3) and Eq. (4), and provide the time complexity analysis of Algorithm 1.

$\mathbf{W}^t$-subproblem. When update $\mathbf{W}^t$ with $\mathbf{G}^t$ fixed, Eq. (2) is reformulated as:

$$\min_{\mathbf{W}^t \geq 0, \mathbf{G}^t \geq 0} \|\mathbf{X}^t - \mathbf{X}^t \mathbf{W}^t {\mathbf{G}^t}^\top\|_F^2. \tag{5}$$

To solve the optimization problem, we introduce the Lagrangian multiplier matrix λ with non-negative values, which constrains the non-negativity of $\mathbf{X}^t$, and then we obtain the following equivalent Lagrangian function:

$$L(\mathbf{W}^t) = tr(-\mathbf{X}^{t\top}\mathbf{X}^t\mathbf{W}^t\mathbf{G}^{t\top} - \mathbf{G}^t\mathbf{W}^{t\top}\mathbf{X}^{t\top}\mathbf{X}^t + \mathbf{G}^t\mathbf{W}^{t\top}\mathbf{X}^{t\top}\mathbf{X}^t\mathbf{W}^t\mathbf{G}^{t\top}). \tag{6}$$

This function satisfies KKT complementary conditions. By setting the gradient $\frac{\partial L(\mathbf{W}^t)}{\partial \mathbf{W}^t} = 0$, we have the following equation from the complementary conditions:

$$-2\mathbf{X}^{t\top}\mathbf{X}^t\mathbf{G}^t + 2\mathbf{X}^{t\top}\mathbf{X}^t\mathbf{W}^t\mathbf{G}^{t\top}\mathbf{G}^t = \lambda_{ij}\mathbf{W}^t_{ij} = 0. \tag{7}$$

This is the fixed point equation, and the solution must eventually converge to a stationary point. From Eq. (7), we derive another equivalent equation:

$$(-2\mathbf{X}^{t\top}\mathbf{X}^t\mathbf{G}^t + 2\mathbf{X}^{t\top}\mathbf{X}^t\mathbf{W}^t\mathbf{G}^{t\top}\mathbf{G}^t)(\mathbf{W}^t_{ij})^2 = 0. \tag{8}$$

The constrained solution with updating rule in Eq. (3) satisfies Eq. (8), so it satisfies the KKT fixed point condition. The proof of convergence for Eq. (3) is presented in [2].

$\mathbf{G}^t$-subproblem. When update $\mathbf{G}^t$ with $\mathbf{W}^t$ fixed, the goal is to solve the optimization problem Eq. (2). Similar to $\mathbf{W}^t$-subproblem, we introduce the Lagrangian multiplier matrix λ with non-negative values which constrains the non-negativity of $\mathbf{G}^t$, and then we obtain the following equivalent Lagrangian function:

$$\begin{aligned} L(\mathbf{G}^t) = tr(&-\mathbf{X}^{t\top}\mathbf{X}^t\mathbf{W}^t\mathbf{G}^{t\top} - \mathbf{G}^t\mathbf{W}^{t\top}\mathbf{X}^{t\top}\mathbf{X}^t + \mathbf{G}^t\mathbf{W}^{t\top}\mathbf{X}^{t\top}\mathbf{X}^t\mathbf{W}^t\mathbf{G}^{t\top} \\ &- \alpha\mathbf{X}^{t-1\top}\mathbf{X}^{t-1}\mathbf{W}^{t-1}\mathbf{G}^{t\top} - \alpha\mathbf{G}^t\mathbf{W}^{t-1\top}\mathbf{X}^{t-1\top}\mathbf{X}^{t-1} \\ &+ \alpha\mathbf{G}^t\mathbf{W}^{t-1\top}\mathbf{X}^{t-1\top}\mathbf{X}^{t-1}\mathbf{W}^{t-1}\mathbf{G}^{t\top}). \end{aligned} \tag{9}$$

This function satisfies KKT complementary conditions. By setting the gradient $\frac{\partial L(\mathbf{G}^t)}{\partial \mathbf{G}^t} = 0$, we have the following equation from the complementary conditions:

$$\begin{aligned} -2\mathbf{X}^{t\top}\mathbf{X}^t\mathbf{W}^t + 2\mathbf{G}^t\mathbf{W}^{t\top}\mathbf{X}^{t\top}\mathbf{X}^t\mathbf{W}^t - 2\alpha\mathbf{X}^{t-1\top}\mathbf{X}^{t-1}\mathbf{W}^{t-1} \\ +2\alpha\mathbf{G}^t\mathbf{W}^{t-1\top}\mathbf{X}^{t-1\top}\mathbf{X}^{t-1}\mathbf{W}^{t-1} = \lambda_{ij}\mathbf{G}^t_{ij} = 0. \end{aligned} \tag{10}$$

This is the fixed point equation, and the solution must eventually converge to a stationary point. From Eq. (10), we derive another equivalent equation:

$$\begin{aligned} (-2\mathbf{X}^{t\top}\mathbf{X}^t\mathbf{W}^t + 2\mathbf{G}^t\mathbf{W}^{t\top}\mathbf{X}^{t\top}\mathbf{X}^t\mathbf{W}^t - 2\alpha\mathbf{X}^{t-1\top}\mathbf{X}^{t-1}\mathbf{W}^{t-1} \\ +2\alpha\mathbf{G}^t\mathbf{W}^{t-1\top}\mathbf{X}^{t-1\top}\mathbf{X}^{t-1}\mathbf{W}^{t-1})(\mathbf{G}^t_{ij})^2 = 0. \end{aligned} \tag{11}$$

The constrained solution with updating rule in Eq. (4) satisfies Eq. (11), so it satisfies the KKT fixed point condition. The proof of convergence for Eq. (4) is presented in [2].

Time Complexity. In Eq. (3), the time to calculate $\mathbf{X}^{t\top}(\mathbf{X}^t\mathbf{W}^t)(\mathbf{G}^{t\top}\mathbf{G}^t)$ and $\mathbf{X}^{t\top}(\mathbf{X}^t\mathbf{G}^t)$ is $\mathcal{O}(4m_1k + 2nk^2)$ and $\mathcal{O}(4m_1k)$, and in Eq. (4), the time to calculate $\mathbf{G}^t((\mathbf{W}^{t-1\top}\mathbf{X}^{t-1\top})(\mathbf{X}^{t-1}\mathbf{W}^{t-1}))$, $\mathbf{X}^{t\top}(\mathbf{X}^t\mathbf{W}^t)$, $\mathbf{X}^{t-1\top}(\mathbf{X}^{t-1}\mathbf{W}^{t-1})$, and $\mathbf{G}^t((\mathbf{W}^{t\top}\mathbf{X}^{t\top})(\mathbf{X}^t\mathbf{W}^t))$ is $\mathcal{O}(4m_2k + 2nk^2)$, $\mathcal{O}(4m_1k)$, $\mathcal{O}(4m_2k)$, and $\mathcal{O}(4m_1k + 2nk^2)$, respectively, where n is the number of nodes, m_1 and m_2 is the number of edges at time t and $t-1$, and k is the number of communities. Therefore, the time to evaluate Eq. (3) and Eq. (4) once is $\mathcal{O}(m_1k + nk^2)$ and $\mathcal{O}(m_1k + m_2k + nk^2)$, respectively, and then the time complexity of DCNMF is $\mathcal{O}(L(m_1k + m_2k + nk^2))$, where L is the iteration number for convergence. The proposed method is applicable to large-scale network, since the number of communities is much smaller than the network size.

4 Experiments and Results

In experiments, we evaluate the performance of the proposed algorithm DCNMF, and compare the results on different types of synthetic and KIT-email real-life datasets with four popular methods: FacetNet [12], DYNMOGA [3], $AFFECT_{kmeans}$ [23], and $AFEECT_{spectral}$ [23]. Since Convex-NMF derives from Semi-NMF, we compare the results of evolutionary clustering method using Semi-NMF, named Semi-NMF, with other methods and DCNMF additionally. DYNMOGA applies local smoothing based multi-objective optimization method to discover dynamic communities. As Facetnet and AFFECT are two classic algorithms among evolutionary clustering methods, we choose them as baselines. We specifically select three synthetic and real-world temporal networks to assess the performance of the proposed method, where the synthetic networks are to prove the accuracy of the algorithm and the real-world one is to validate the feasibility of the algorithm in practice.

In experiments, we set the parameter α of DCNMF and Semi-NMF both as 0.3, and adopt random initialization to NMF solutions at each step for all datasets. Since the compared algorithms converge to local minima, we run 20 times each pair of compared algorithms and finally report the average results into comparison, which ensures the robustness and accuracy of the proposed algorithm as for the community detecting results. Specifically, for Facetnet, we set the parameter of α as 0.8 for three datasets. DYNMOGA can adjust the parameters by itself to search for the global or local optimal solutions. And AFFECT methods also search for the best parameters both in kmeans and spectral based methods on their own.

In this section, we first introduce evaluation measures we employ and then report experiment results of different datasets. The proposed algorithm is not sensitive to the parameter α, and the experimental results of parameter analysis will be shown in Subsect. 4.2.

4.1 Evaluation Measures

In order to measure the DCD performance, we select three evaluation indexes which are widely employed, i.e. Normal Mutual Information (NMI) [9], error rate (CA) [13], and Fscore [9]. Let C_q be the set of the cluster of dataset (the annotated class) and C_p be the set of the cluster detected by the community discovery algorithm. Let $n_p, n_q, n_{p,q}$ be the numbers of the amount of nodes in community C_p, community C_q, and both the communities C_p and community C_q, respectively. The computational processes of three metrics are briefly illustrated as follows.

Normalised mutual information (NMI) is one of the popular evaluation indexes of clustering quality [9], which can be formulated as,

$$NMI = \frac{\sum_{p=1}^{K} \sum_{q=1}^{K} n_{p,q} log(\frac{n \times n_{p,q}}{n_p \times n_q})}{\sqrt{(\sum_{p=1}^{K} n_p log \frac{n_p}{n})(\sum_{q=1}^{K} n_q log \frac{n_q}{n})}}. \tag{12}$$

NMI is a value between 0 and 1, which equals 1 when two partitions are equivalent.

The error rate (CA) [13] can be formulated as

$$CA = \|\mathbf{Z}\mathbf{Z}^\top - \mathbf{G}\mathbf{G}^\top\|_F^2, \tag{13}$$

where, $\mathbf{Z}$ is the indicator matrix of clustering result which is computed by a given algorithm, where the i-th row of $\mathbf{Z}$ indicates the community membership of the i-th node (i.e., if the i-th node belongs to the k-th community, then $z_{ik} = 1$ and $z_{ik'} = 0$ for $k^{'} \neq k$). A similar indicator matrix $\mathbf{G}$ is constructed for the ground truth. Then the error rate computed by Eq. (13) measures the distance between the community structure represented by $\mathbf{Z}$ and that represented by $\mathbf{G}$.

Fscore integrates the metrics of precision and recall, which is extensively applied in evaluating the community detection performance [9]. The precision and recall are calculated as

$$Precision(C_q, C_p) = \frac{n_{p,q}}{C_p}, \tag{14}$$

$$Recall(C_q, C_p) = \frac{n_{p,q}}{C_q}. \tag{15}$$

Then the Fscore of the detected community C_p and the real community C_q can be computed as

$$F(C_q, C_p) = \frac{2 \times P(C_q, C_p) \times R(C_q, C_p)}{P(C_q, C_p) + R(C_q, C_p)}. \tag{16}$$

4.2 Synthetic Dataset 1: Dynamic-GN Dataset

GN-network benchmark was previously proposed by Girvan and Newman [5] for static community detection, where each network contains 128 nodes and 4 communities. Each community contains fixed 32 nodes, and the average degree of nodes is 16 fixed. To incorporate evolution into the GN-network, Lin et al. [12] developed dynamic GN-network, in which the membership of three vertices

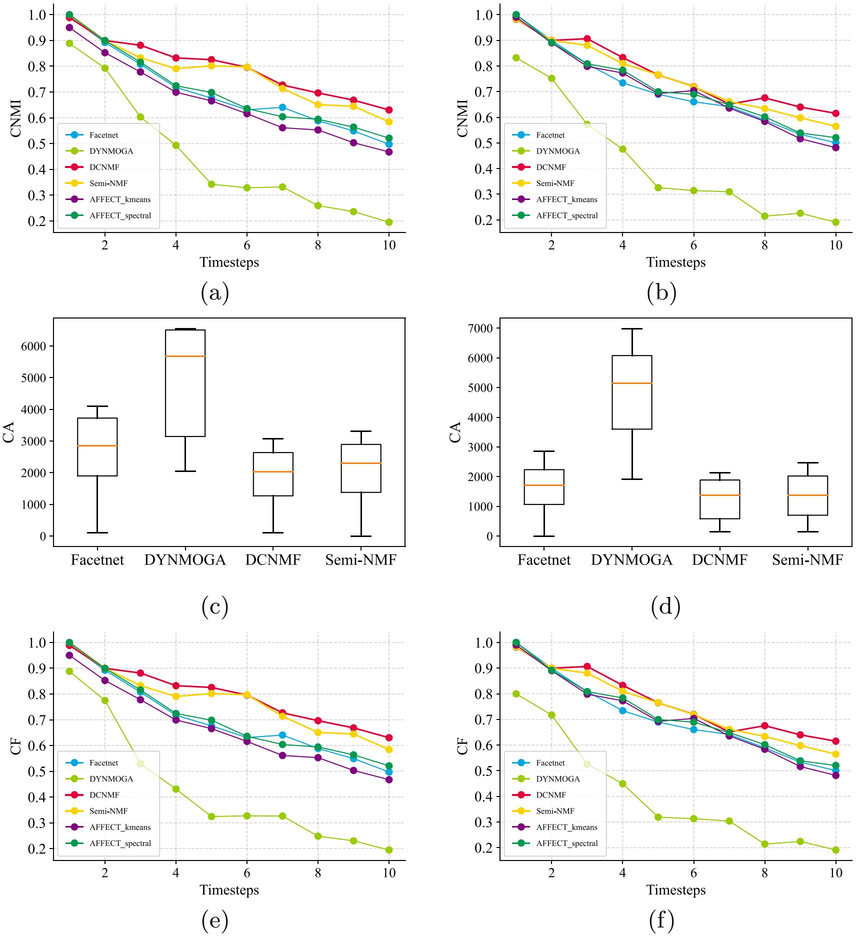
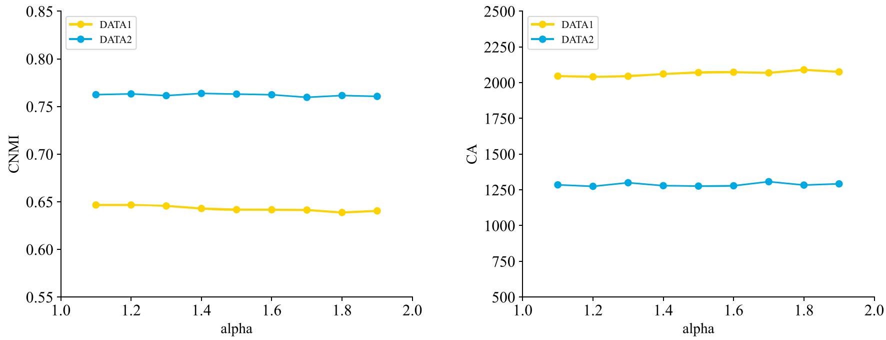


Fig. 1. Average NMI, CA and CF results on the synthetic dataset 1: the network size is 128; (a, c, e): $z = 3, d = 20, nc = 10\%$; (b, d, f): $z = 4, d = 25, nc = 30\%$.

in each community is changed by random assignment to other communities. In experiments, we set the number of time steps $T = 10$, the number of nodes at each timestep $N = 128$, and the number of communities $K = 4$, respectively. The mixing parameter z which controls the noise level of communities, is set to 3 and 4. Moreover, we set the average degree of nodes $d = 20$ and $d = 25$, and the community evolution parameter $nc = 10\%$ and 30%, which is used to control the degree of transition that nodes transfer from their own communities to others randomly between consecutive time slices. For each dataset of this type of temporal network, we run 20 times and take the average NMI, CA, and CF values as the final reported results.



(a) (b)

Fig. 2. Average NMI and CA results of different parameter α on the synthetic dataset 1: the network size is 128; DATA1: $z = 4, d = 25, nc = 10\%$; DATA2: $z = 4, d = 25, nc = 30\%$.

As we can see from Fig. 1, as the network becomes more complicated, all of the methods have much worse performance on networks with a high noise level. As shown in Fig. 1(a, b), for the NMI index which measures the quality of the DCD result, the proposed DCNMF outperforms both Semi-NMF and other four methods at almost each time step nearly for all datesets. Similarly, from Fig. 1(c, d, e, f), DCNMF also performs well on CA and CF and always has relatively small variances as for CA values. Above all, it can be concluded that the proposed algorithm DCNMF is more accurate and robust on various networks of Dynamic-GN dataset aiming at DCD problem.

We have mentioned that the proposed algorithm is almost not sensitive to the parameters, and we test the parameter-sensitivity on two datasets of the network with parameters: the mixing parameter $z = 4$, the average degree of node $d = 25$, and the community evolution parameter nc of Data1 and Data2 is 10% and 30%, respectively. Parameter α is an important factor to tune the smoothness of the evolutionary trends of the dynamic networks. Figure 2 shows the results of NMI and CA over the changes of α from $\alpha = 1.1$ to $\alpha = 1.9$. From the results, we can see that the DCD results of DCNMF are stable as for NMI and CA indexes.

4.3 Synthetic Dataset 2: Dynamic-LFR Dataset

To obtain a synthetic network that is more consistent with a real-world network, we employ the extended LFR [7] model to generate dynamic networks, which is based on the embedding of events into synthetic graphs. We test the proposed algorithm on the synthetic datasets which mainly include the node switch event, i.e., nodes switch community membership between consecutive time slices. To evaluate different methods, we construct different synthetic networks by setting different parameters which cover 1000 nodes over 10 snapshots. In each of the synthetic networks, μ is the mixing parameter which controls the level of edges between communities, and p is the probability of the nodes switching among communities. In experiments, we set the number of the time steps $T = 10$, the number of communities $k = 36$, the average degree of nodes $d = 20$, and the mixing parameter $p = 0.5$, $\mu \in [0.1, 0.8]$.

The results of two examples of generated temporal networks are shown in Fig. 3, which involves 1000 nodes, 36 embedded dynamic communities, $p = 0.5$, and $\mu = 0.1$, 0.3, and 0.6, respectively. According to Fig. 3(a, c, e), we observe that the DCNMF performs better when the network has a lower level of noise than the other compared methods. When the probability of switching is high, i.e. the network is more complex, the proposed model DCNMF still has higher community detection quality than other methods as shown in Fig. 3(e). It is also noticed in Fig. 3(b, d, f) that DCNMF has lower errors in most cases, which succeeds to discover dynamic communities accurately. Consequently, these results demonstrate that DCNMF is more accurate and robust than the other methods.

4.4 KIT-Email Data

To validate the feasibility of the proposed algorithm, we compare DCNMF with other methods on KIT-email data which is a large number of snapshots of the e-mail communication network in the Department of Informatics at KIT[1]. In the network, the vertices represent email contacts of the department of computer science at KIT, which evolves during 48 consecutive months from September 2006 to August 2010. If an email is sent to a recipient, there is an edge from the sender vertex to the recipient vertex. We construct the adjacency matrices among 231 active members. In the E-mail network, the department of computer science at KIT is considered as a community. Since the number of communities increases when taking more months as a snapshot, the number of communities is 14, 23, 25, 25, and 27, for the snapshots of 1, 2, 3, 4, and 6 months, respectively.

Table 1 shows the results of compared algorithms on the real world dataset with different resolutions. Correspondingly, we set 2 months, 3 months, 4 months, and 6 months of the KIT email dataset as the length of snapshots to form four temporal networks, on which the results in terms of NMI and CA are presented in Table 1. From the table, it demonstrates that the proposed DCNMF substantially improves the performance of Semi-NMF based method. Besides, although DCNMF does not always perform the best on datsets, it is quite competitive with the best one (i.e. Facetnet) on CNMI and CA. Overall, the proposed method DCNMF shows superior performance on NMI and CA compared with other methods.

[1] http://i11www.iti.uni-karlsruhe.de/en/projects/spp1307/emaildata.

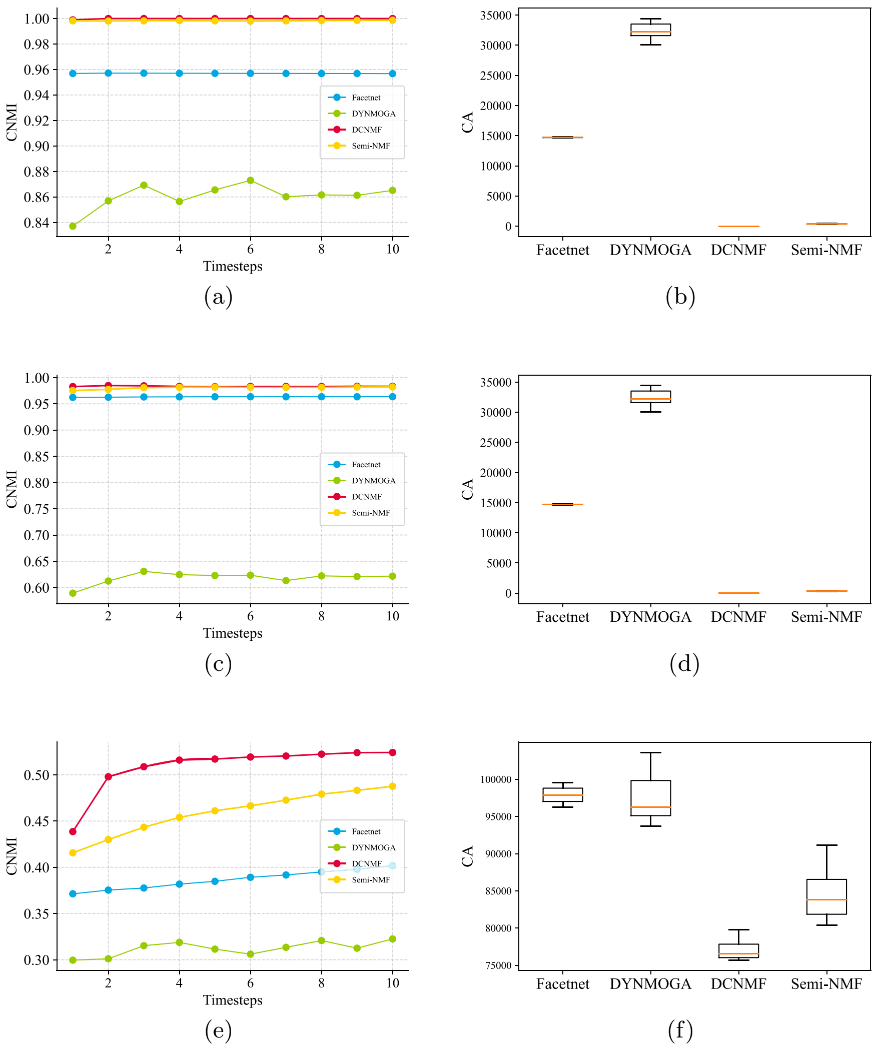


Fig. 3. Average NMI and CA results on the synthetic dataset 2: the network size is 1000, $p = 0.5$; (a, b): $\mu = 0.1$; (c, d): $\mu = 0.3$; (e, f): $\mu = 0.6$.

Table 1. Average NMI and CA values of KIT-email real network data: Data1: taking 2 months as a snapshot; Data2: taking 3 months as a snapshot; Data3: taking 4 months as a snapshot; Data4: taking 6 months as a snapshot.

Method	CNMI			
	$Data_1$	$Data_2$	$Data_3$	$Data_4$
Facetnet	$\mathbf{0.84 \pm 0.019}$	$\mathbf{0.83 \pm 0.018}$	0.80 ± 0.018	$\mathbf{0.81 \pm 0.018}$
DYNMOGA	0.72 ± 0.036	0.68 ± 0.035	0.66 ± 0.029	0.64 ± 0.029
$AFFECT_{kmeans}$	$\mathbf{0.84 \pm 0.027}$	0.81 ± 0.022	0.80 ± 0.023	0.79 ± 0.021
$AFFECT_{spectral}$	$\mathbf{0.84 \pm 0.021}$	0.82 ± 0.020	$\mathbf{0.81 \pm 0.021}$	$\mathbf{0.81 \pm 0.016}$
SemiNMF	0.83 ± 0.03	0.79 ± 0.036	0.76 ± 0.050	0.76 ± 0.043
DCNMF	$\mathbf{0.84 \pm 0.023}$	$\mathbf{0.83 \pm 0.020}$	$\mathbf{0.81 \pm 0.021}$	$\mathbf{0.81 \pm 0.020}$
Method	CA			
	$Data_1$	$Data_2$	$Data_3$	$Data_4$
Facetnet	$\mathbf{814.0 \pm 77.7}$	$\mathbf{1233.4 \pm 123.3}$	$\mathbf{1621.6 \pm 173.2}$	$\mathbf{2007.3 \pm 175.8}$
DYNMOGA	1350.2 ± 251.0	2134.9 ± 369.3	2847.6 ± 404.4	4034.0 ± 585.5
$AFFECT_{kmeans}$	889.0 ± 128.3	1456.7 ± 134.7	1896.6 ± 208.7	2321.9 ± 284.0
$AFFECT_{spectral}$	1017.0 ± 124.8	1431.5 ± 147.3	1708.7 ± 191.0	2239.8 ± 222.6
SemiNMF	911.3 ± 218.0	1473.7 ± 309.5	2131.1 ± 584.9	2778.2 ± 170.1
DCNMF	$\mathbf{847.7 \pm 93.3}$	$\mathbf{1289.9 \pm 103.5}$	$\mathbf{1633.3 \pm 135.8}$	$\mathbf{2117.3 \pm 149.8}$

5 Discussion and Conclusion

In this paper, we present a unified model named DCNMF which is able to detect communities and track their evolution. Because of the employment of Convex-NMF, a constrained version of Semi-NMF, the proposed algorithm DCNMF has achieved the great success on DCD issues. Compared with several typical methods, it is proved that the proposed method utilizes the previous information more effectively in the task of analyzing dynamics. The experimental results on the synthetic and real data show that DCNMF outperforms other typical DCD methods for various evolutionary networks.

In terms of DCD methods based on temporal trade-off smoothness, one of potential issue is that the long-term coherence of dynamic communities. For the reason that at each iteration the detected community depends on previous ones, temporal trade-off approaches are subject to the risk of an avalanche effect: communities can experience rapid changes. Under such circumstance, it is more suitable to directly employ static community discovery algorithms for solutions at a given time, and it will be our future work to capture such substantial drifts in temporal networks. Furthermore, we also plan to extend the proposed model to networks whose number of nodes and communities may change over time.

Acknowledgements. This work is supported by the Innovation and Cultivation Project for Youth Talents of Shihezi University (Program No. CXPY201905), the Scientific Research Project for High-level Talents of Shihezi University (Program No. RCZK202028), and the Financial and Science Technology Plan Project of Xinjiang Production and Construction Corps (Program No. 2017DB005).

References

1. Chi, Y., Song, X., Zhou, D., Hino, K., Tseng, B.L.: On evolutionary spectral clustering. ACM Trans. Knowl. Discov. Data **3**(4), 1–30 (2009)
2. Ding, C., Li, T., Jordan, M.I.: Convex and semi-nonnegative matrix factorizations. IEEE Trans. Pattern Anal. Mach. Intell. **32**(1), 45–55 (2010)
3. Folino, F., Pizzuti, C.: An evolutionary multiobjective approach for community discovery in dynamic networks. IEEE Trans. Knowl. Data Eng. **26**(8), 1838–1852 (2013)
4. Gao, C., Chen, Z., Li, X., Tian, Z., Li, S., Wang, Z.: Multiobjective discrete particle swarm optimization for community detection in dynamic networks. Europhys. Lett. **122**(2), 28001 (2018)
5. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proc. Natl. Acad. Sci. **99**(12), 7821–7826 (2002)
6. González-Bailón, S., Borge-Holthoefer, J., Rivero, A., Moreno, Y.: The dynamics of protest recruitment through an online network. Sci. Rep. **1**, 197 (2011)
7. Greene, D., Doyle, D., Cunningham, P.: Tracking the evolution of communities in dynamic social networks. In: ASONAM, pp. 176–183 (2010)
8. Holme, P., Saramäki, J.: Temporal networks. Phys. Rep. **519**, 97–125 (2012)
9. Jing, L., Ng, M.K., Huang, J.Z.: An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. IEEE Trans. Knowl. Data Eng. **19**(8), 1026–1041 (2007)
10. Li, Q., Cao, Z., Ding, W., Li, Q.: A multi-objective adaptive evolutionary algorithm to extract communities in networks. Swarm Evol. Comput. **52**, 100629 (2020)
11. Li, W., Xie, J., Xin, M., Mo, J.: An overlapping network community partition algorithm based on semi-supervised matrix factorization and random walk. Expert Syst. Appl. **91**, 277–285 (2018)
12. Lin, Y.R., Chi, Y., Zhu, S., Sundaram, H., Tseng, B.L.: FacetNet: a framework for analyzing communities and their evolutions in dynamic networks. In: WWW, pp. 685–694 (2008)
13. Lin, Y.R., Chi, Y., Zhu, S., Sundaram, H., Tseng, B.L.: Analyzing communities and their evolutions in dynamic social networks. ACM Trans. Knowl. Discov. Data **3**(2), 8:1–8:31 (2009)
14. Liu, F., Lv, B., Huang, J., Ali, S.: Towards mobility-aware dynamic service migration in mobile edge computing. In: CollaborateCom (2020)
15. Lu, H., Zhao, Q., Sang, X., Lu, J.: Community detection in complex networks using nonnegative matrix factorization and density-based clustering algorithm. Neural Process. Lett. (12) (2020)
16. Marler, R.T., Arora, J.S.: Survey of multi-objective optimization methods for engineering. Struct. Multidiscip. Optim. **26**(6), 369–395 (2004)
17. Messaoudi, I., Kamel, N.: A multi-objective bat algorithm for community detection on dynamic social networks. Appl. Intell. **49**(6), 2119–2136 (2019)

18. Mu, C., Zhang, J., Liu, Y., Qu, R., Huang, T.: Multi-objective ant colony optimization algorithm based on decomposition for community detection in complex networks. Soft. Comput. **23**(23), 12683–12709 (2019). https://doi.org/10.1007/s00500-019-03820-y
19. Newman, M.: Networks: An Introduction. Oxford University Press, Oxford (2010)
20. Onnela, J.P., et al.: Structure and tie strengths in mobile communication networks. Proc. Natl. Acad. Sci. **104**(18), 7332–7336 (2007)
21. Sarswat, A., Jami, V., Guddeti, R.M.R.: A novel two-step approach for overlapping community detection in social networks. Soc. Netw. Anal. Min. **7**(1), 1–11 (2017). https://doi.org/10.1007/s13278-017-0469-7
22. Wu, W., Kwong, S., Zhou, Y., Jia, Y., Gao, W.: Nonnegative matrix factorization with mixed hypergraph regularization for community detection. Inf. Sci. **435**, 263–281 (2018)
23. Xu, K.S., Kliger, M., Hero III, A.O.: Adaptive evolutionary clustering. Data Min. Knowl. Disc. **28**(2), 304–336 (2013). https://doi.org/10.1007/s10618-012-0302-x
24. Zhang, D., Huang, Y., Wang, Y., Zhu, Y., Zhao, C.: A novel two-step community detection approach based on community tree and the n-players cooperative game in large-scale social networks. J. Comput. Methods Sci. Eng. **18**(4), 1007–1020 (2018)
25. Zhang, Y., Yin, D., Wu, B., Long, F., Cui, Y., Bian, X.: Plinkshrink: a parallel overlapping community detection algorithm with link-graph for large networks. Soc. Netw. Anal. Min. **9**(1), 66 (2019)
26. Zhao, Z., Li, C., Zhang, X., Chiclana, F., Viedma, E.H.: An incremental method to detect communities in dynamic evolving social networks. Knowl.-Based Syst. **163**, 404–415 (2019)