



Fast Community Detection in Dynamic and Heterogeneous Networks

Maoyu Zhang, Jingfei Zhang & Wenlin Dai

To cite this article: Maoyu Zhang, Jingfei Zhang & Wenlin Dai (2024) Fast Community Detection in Dynamic and Heterogeneous Networks, Journal of Computational and Graphical Statistics, 33:2, 487-500, DOI: [10.1080/10618600.2023.2232852](https://doi.org/10.1080/10618600.2023.2232852)

To link to this article: <https://doi.org/10.1080/10618600.2023.2232852>



View supplementary material [↗](#)



Published online: 05 Sep 2023.



Submit your article to this journal [↗](#)



Article views: 491



View related articles [↗](#)



View Crossmark data [↗](#)



Fast Community Detection in Dynamic and Heterogeneous Networks

Maoyu Zhang^{a,b}, Jingfei Zhang^b, and Wenlin Dai^a

^aCenter for Applied Statistics, Institute of Statistics and Big Data, Renmin University of China, Beijing, China; ^bGoizueta Business School, Emory University, Atlanta, GA

ABSTRACT

Dynamic heterogeneous networks describe the temporal evolution of interactions among nodes and edges of different types. While there is a rich literature on finding communities in dynamic networks, the application of these methods to dynamic heterogeneous networks can be inappropriate, due to the involvement of different types of nodes and edges and the need to treat them differently. In this article, we propose a statistical framework for detecting common communities in dynamic and heterogeneous networks. Under this framework, we develop a fast community detection method called DHNet that can efficiently estimate the community label as well as the number of communities. An attractive feature of DHNet is that it does not require the number of communities to be known a priori, a common assumption in community detection methods. While DHNet does not require any parametric assumptions on the underlying network model, we show that the identified label is consistent under a time-varying heterogeneous stochastic block model with a temporal correlation structure and edge sparsity. We further illustrate the utility of DHNet through simulations and an application to review data from Yelp, where DHNet shows improvements both in terms of accuracy and interpretability over alternative solutions. Supplementary materials for this article are available online.

ARTICLE HISTORY

Received October 2022
Accepted June 2023

KEYWORDS

Community detection;
Consistency; Dynamic
heterogeneous network;
Modularity; Null model; Yelp
reviews

1. Introduction

One of the fundamental problems in network data analysis is community detection that aims to divide the network into nonoverlapping groups of nodes such that nodes within the same community are densely connected and nodes from different communities are relatively sparsely connected. Community detection can provide valuable insights on the organization of a network and greatly facilitate the analysis of network characteristics. As such, community detection methods have been applied to numerous scientific fields such as social science (Moody and White 2003), biology (Sørli et al. 2001) and business (Linden, Smith, and York 2003). Over the past few decades, the problem of community detection has been approached from methodological, algorithmic and theoretical perspectives with substantial developments. We refer to Fortunato (2010) and Abbe (2017) for comprehensive reviews on this topic.

While the majority of existing community detection methods are developed for a homogeneous network or a dynamic network, networks that are dynamic and heterogeneous are fast emerging in recent years. For example, in a dynamic healthcare network, nodes can be *patients*, *diseases*, *doctors* and *hospitals* and edges can be in the type of patient–disease (patient treated for disease) and patient–doctor (patient treated by doctor) and doctor–hospital (doctor works at hospital). These edges are expected to evolve with time as patients may develop new diseases that are treated by different doctors at possibly different hospitals. Figure 1 provides an illustration

of a dynamic heterogeneous Yelp review network, which is analyzed in Section 6. In this figure, there are three types of nodes including *users*, *businesses*, and *categories* and three types of edges including user–user (user is friend with user), user–business (business is reviewed by user), and business–category (business is labeled with category). As users review different businesses over time, this network is both heterogeneous and dynamic.

Due to the rich information embedded in a dynamic heterogeneous network, many methods have been developed recently for its analysis, such as network embedding (Wang et al. 2022; Zhang, Huang, and Tan 2022), representation learning (Yin et al. 2019) and link prediction (Xue et al. 2020; Jiang, Koch, and Sun 2021). However, community detection in dynamic heterogeneous networks is less studied. One relevant work is Sun et al. (2010), which provides a mixture model-based generative model for estimating the community structure, which is assumed to be time-varying. Other works on this topic include Sengupta and Chen (2015) and Zhang and Cao (2017), though they only focus on a single heterogeneous network.

In our work, we focus on detecting common communities in a dynamic heterogeneous network, that is, the community assignment does not vary with time but the interactions within and between communities do. Finding common communities are useful in many applications. For example, in genetic studies and brain connectivity studies, the common communities represent functional groups of genes or brain regions that are coordinated in biological processes, and identifying them is of keen

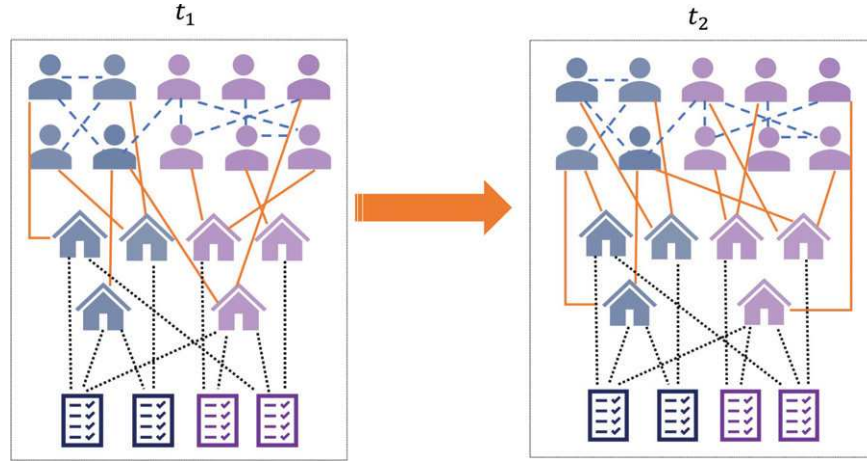


Figure 1. The dynamic heterogeneous Yelp network with two communities, including three types of nodes (user, business, category) and edges (user is friend with user, business is reviewed by user, business is labeled with category).

scientific interests (Zhang and Cao 2017; Zhang, Sun, and Li 2020). In the Yelp review network, it is plausible that businesses, categories and the majority of users have a common community structure over time, as the service offered by a business and the interests of users (e.g., pets, parks, fine dining) are often stable over a period of time. One notable advantage of considering a common community structure is that the networks observed at different time points are allowed to be highly sparse if S , the number of time points, increases. For example, we show in Theorem 1 that consistent community detection is achievable as long as $\lambda S \rightarrow \infty$, where λ is the average degree, while the single network case requires $\lambda \rightarrow \infty$ to achieve community detection consistency. Moreover, our approach allows the community strength to be highly variable over time. For example, a community needs to be active for only a very short period of time for it to be consistently identified; see more discussions after Theorem 1.

In this article, we propose a statistical framework for modularity-based common community detection in the dynamic heterogeneous network, where no parametric assumptions are made on the model underlying the observed networks. Under this framework, we develop a fast community detection method called DHNet that can efficiently estimate the community label as well as the number of communities. An attractive feature of DHNet is that it does not require the number of communities to be known a priori, a common assumption in community detection methods. Although DHNet does not rely on parametric assumptions on the underlying network model, we propose a new dynamic heterogeneous stochastic block model with a temporal correlation structure and edge sparsity, and show that DHNet can consistently estimate the community label under this model. This provides theoretical justifications of the proposed method and also sheds lights on how different network properties (e.g., sparsity, size, community strength) affect its performance. The consistency property of our method when applied to dynamic bi-partite or multi-partite networks follows as special cases.

The remainder of the article is organized as follows. Section 2 describes a community detection framework and proposes a modularity function for finding common communities

in a dynamic heterogeneous network. Section 3 describes a fast community detection method called DHNet that can efficiently estimate the community label as well as the number of communities. Section 4 shows the consistency property of DHNet under a dynamic heterogeneous stochastic block model. Section 5 demonstrates the efficacy of DHNet through simulation studies and Section 6 applies the proposed method to review data from Yelp. The article is concluded with a short discussion in section.

2. Community Detection with Modularity

2.1. Notation

We write $[m] = \{1, \dots, m\}$ for an integer $m > 0$. To ease notation, we start the introduction with a *single* heterogeneous network with L types of nodes. Let $V^{[l]} = (v_1^{[l]}, \dots, v_{n_l}^{[l]})$ be the set containing the l th type of nodes for $l \in [L]$, where n_l is the number of l th type nodes. Denote the heterogeneous network as $\mathcal{G} = (\cup_{l=1}^L V^{[l]}, \mathcal{E} \cup \mathcal{E}^+)$, where set $\mathcal{E}$ contains edges between nodes of the same type and set $\mathcal{E}^+$ contains edges between nodes of different types. When $\mathcal{E} = \emptyset$, $\mathcal{G}$ forms a multi-partite network, that is, edges are only established between different types of nodes. Let $G^{[l]}$ denote the homogeneous network formed within node set $V^{[l]}$ with an $n_l \times n_l$ adjacency matrix $A^{[l]}$, and $G^{[l_1 l_2]} = (V^{[l_1]} \cup V^{[l_2]}, E^{[l_1 l_2]})$ denote the bi-partite network formed between node sets $V^{[l_1]}$ and $V^{[l_2]}$ with an $n_{l_1} \times n_{l_2}$ bi-adjacency matrix $A^{[l_1 l_2]}$, $l_1, l_2 \in [L]$. See Figure 2 for an example of a heterogeneous network with $L = 2$.

Consider a dynamic heterogeneous network $\{\mathcal{G}(t), t \in \mathcal{T}\}$ with L types of nodes, where $\mathcal{G}(t) = (\cup_{l=1}^L V^{[l]}, \mathcal{E}(t) \cup \mathcal{E}^+(t))$ is a heterogeneous network at time t defined as above. The network $\mathcal{G}(t)$ at time t can be uniquely represented by its adjacency matrix $\mathcal{A}(t)$ defined as

$$\mathcal{A}(t) = \begin{pmatrix} A^{[11]}(t) & \dots & A^{[1L]}(t) \\ \vdots & \ddots & \vdots \\ A^{[L1]}(t) & \dots & A^{[LL]}(t) \end{pmatrix},$$

where $A^{[l_1 l_2]}(t) \in \mathbb{R}^{n_{l_1} \times n_{l_2}}$ is defined as in Figure 2. Define $\mathbf{d}^{[l]}(t) = (d_1^{[l]}(t), \dots, d_{n_l}^{[l]}(t))$, where $d_i^{[l]}(t)$ is the number of

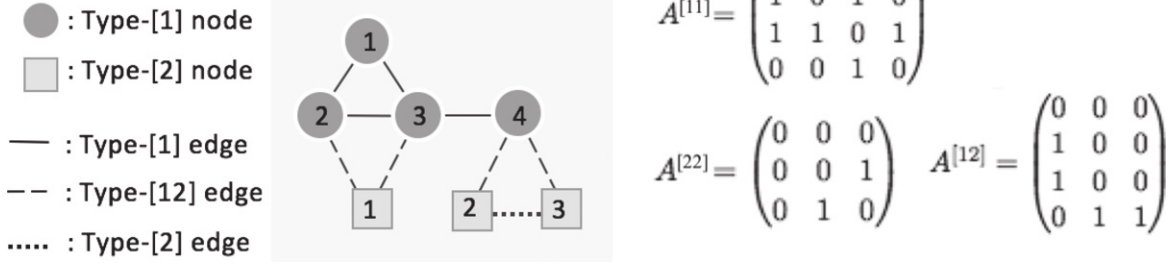


Figure 2. An illustrative example of a heterogeneous network with two types of nodes.

links incident to $v_i^{[l]}$ from $V^{[l]}$ at time t , and $\mathbf{d}^{[l_1 l_2]}(t) = (d_1^{[l_1 l_2]}(t), \dots, d_n^{[l_1 l_2]}(t))$, where $d_i^{[l_1 l_2]}$ is the number of links incident to $v_i^{[l_1]}$ from $V^{[l_2]}$ at time t . Write the number of edges in $A^{[l_1 l_2]}(t)$ as $m^{[l_1 l_2]}(t) = \sum_{i,j} A_{ij}^{[l_1 l_2]}(t)$ for $l_1, l_2 \in [L]$.

2.2. Modularity Function

The modularity function measures the strength of division of a network into communities, and the maximum modularity function value is a metric frequently used for quantifying the strength of community structure within a network (Fortunato 2010). The function was first defined in Newman and Girvan (2004) for a simple network $G(V, E)$ with n nodes, m edges, adjacency matrix $A_{n \times n}$ and a community assignment $\mathbf{e} = (e_1, \dots, e_n)$, where $e_i \in [K]$, as

$$Q(\mathbf{e}, G) = \frac{1}{2m} \sum_{1 \leq i < j \leq n} [A_{ij} - \mathbb{E}(A_{ij})] 1(e_i = e_j), \quad (1)$$

where $1(\cdot)$ is the indicator function. In (1), the expectation $\mathbb{E}(A_{ij})$ is calculated under a null model for random networks with no community structure. A common consideration is to let graphs in the null space share some basic structural properties with the observed graph G (Newman, Strogatz, and Watts 2001; Zhang and Chen 2017). In particular, the distribution of edges in real world networks is often inhomogeneous with global inhomogeneity, where the majority of nodes have low degrees and a few nodes have high degrees, and local inhomogeneity (or community structure), where there is a high concentration of edges within certain groups of nodes and a low concentration of edges between these groups (Fortunato 2010). To study local inhomogeneity (or community structure), it is desirable to preserve the observed degree sequence in the null model. For this purpose, the most common choice for the null model is the Chung-Lu model (Newman and Girvan 2004; Newman 2006). The Chung-Lu model (Chung et al. 2006) is a random graph model that generalizes the Erdos-Renyi model where all edges are placed with a uniform probability p_0 and all nodes have the same expected degree. In a Chung-Lu model, given the expected degrees for a pair of nodes, denoted as d_i and d_j , the probability of having an edge between nodes i and j is

$$\mathbb{E}(A_{ij}) = f(d_i)f(d_j) = \frac{d_i d_j}{2m}, \quad (2)$$

where $m = \sum_i d_i/2$. The Chung-Lu model is a random graph model that allows for general degree distributions, and it has

been widely used to generate null graphs, that is, simple random graphs that follow certain characteristics in their degree distributions, for in network analysis (Milo et al. 2002). In another view, Zhang and Chen (2017) considered a null model defined as,

$$\mathbb{P}(G) = \frac{1}{|\Sigma_{\mathbf{d}}|}, \quad G \in \Sigma_{\mathbf{d}}, \quad (3)$$

where $\Sigma_{\mathbf{d}}$ is the set of all simple graphs with degree sequence $\mathbf{d}$ and $|\Sigma_{\mathbf{d}}|$ is the total number of graphs in $\Sigma_{\mathbf{d}}$. Under the null model (3) with a uniform distribution, every network in the null space occurs with the same probability and there is no preference for any particular graph configuration such as community structures. Interestingly, Zhang and Chen (2017) showed that $\mathbb{E}(A_{ij}) \approx \frac{d_i d_j}{2m}$ under model (3), which gives the Chung-Lu model in (2). Hence, under the Chung-Lu model, networks in the null space $\Sigma_{\mathbf{d}}$ asymptotically occurs with the same probability, which makes it a desirable choice as the null model. It is seen that the modularity function in (1) measures the difference between the observed number of intra-community edges and the expected number of intra-community edges under the null with no community structure. Correspondingly, the community label of a network is identified by maximizing the modularity function with respect to $\mathbf{e}$.

To define the modularity function in a dynamic heterogeneous network, we first describe the corresponding null model that characterizes a dynamic heterogeneous network with no community structure. Consider the heterogeneous network at time t , $\mathcal{G}(t) = (\cup_{l=1}^L V^{[l]}, \mathcal{E}(t) \cup \mathcal{E}^+(t))$ with degree sequence $\mathbf{D}(t) = \{\mathbf{d}^{[l_1 l_2]}(t), l_1, l_2 \in [L]\}$. We define a heterogeneous Chung-Lu model as the null. Under the null, we assume that a heterogeneous network at time t is generated with

$$A_{ij}^{[l_1 l_2]}(t) \sim \text{Bernoulli} \left(\frac{d_i^{[l_1 l_2]}(t) d_j^{[l_2 l_1]}(t)}{m^{[l_1 l_2]}(t)} \right), \quad l_1, l_2 \in [L], \quad (4)$$

where all edges in $\mathcal{G}(t)$ are independent. Under (4), it is easy to show that the expected degree sequence under the null is the same as the observed degree sequence $\mathbf{D}(t)$. Following the same argument as in Zhang and Chen (2017), it can be shown that under (4), every heterogeneous network in the null space $\Sigma_{\mathbf{D}(t)}$ occurs with the same probability.

Next, we move to define the modularity matrix. At time $t \in \mathcal{T}$ and given $\mathcal{A}(t)$, we write the $(n_1 + \dots + n_L) \times (n_1 + \dots + n_L)$

modularity matrix $\mathcal{M}(t)$ as

$$\mathcal{M}(t) = \begin{pmatrix} \frac{M^{[11]}(t)}{\bar{m}^{[11]}(t)} & \cdots & \frac{M^{[1L]}(t)}{\bar{m}^{[1L]}(t)} \\ \vdots & \ddots & \vdots \\ \frac{M^{[L1]}(t)}{\bar{m}^{[L1]}(t)} & \cdots & \frac{M^{[LL]}(t)}{\bar{m}^{[LL]}(t)} \end{pmatrix},$$

where $M^{[l_1 l_2]}(t) = A^{[l_1 l_2]}(t) - \mathbb{E}(A^{[l_1 l_2]}(t))$. The modularity matrix $\mathcal{M}(t)$ measures the distance between the observed network and the expected network under the null model at time t . Given the dynamic heterogeneous networks $\{\mathcal{A}(t), t \in \mathcal{T}\}$, the integrated modularity matrix $\mathcal{M}$ is defined as

$$\mathcal{M} = \begin{pmatrix} \mathcal{M}^{[11]} & \cdots & \mathcal{M}^{[1L]} \\ \vdots & \ddots & \vdots \\ \mathcal{M}^{[L1]} & \cdots & \mathcal{M}^{[LL]} \end{pmatrix},$$

$$\text{where } \mathcal{M}^{[l_1 l_2]} = \frac{\int_{t \in \mathcal{T}} M^{[l_1 l_2]}(t)}{\bar{m}^{[l_1 l_2]}},$$

$\bar{m}^{[l_1 l_2]} = \int_{t \in \mathcal{T}} m^{[l_1 l_2]}(t)$. The integrated modularity matrix $\mathcal{M}$ measures the distance between the observed network and the expected network under the null model over all $t \in \mathcal{T}$.

We are now ready to define the modularity function. Write the community assignment label as $\mathbf{e} = (\mathbf{e}^{[1]}, \dots, \mathbf{e}^{[L]})$ with $\mathbf{e}^{[l]} = (e_1^{[l]}, \dots, e_{n_l}^{[l]})$, $l \in [L]$, the modularity function of the dynamic heterogeneous network is defined as

$$Q(\mathbf{e}, \{\mathcal{G}(t)\}_{t \in \mathcal{T}}) = \frac{1}{L^2} \sum_{1 \leq l_1, l_2 \leq L} \sum_{ij} \mathcal{M}_{ij}^{[l_1 l_2]} 1(e_i^{[l_1]} = e_j^{[l_2]}). \quad (5)$$

From the above definitions, it can be shown that $Q(\mathbf{e}, \{\mathcal{G}(t)\}_{t \in \mathcal{T}}) \in [-1, 1]$. This modularity function measures the overall difference between the observed number of intra-community edges and the expected number of intra-community edges under the null model. When $Q(\mathbf{e}, \{\mathcal{G}(t)\}_{t \in \mathcal{T}})$ approaches 1, the observed number of intra-community edges is greater than the expected values, which indicates a strong community structure. In contrast, when $Q(\mathbf{e}, \{\mathcal{G}(t)\}_{t \in \mathcal{T}})$ approaches 0, the observed number of intra-community edges is close to the expected values under the null, which indicates no or weak community structure.

In practice, the networks are often only observed on a number of time points $\mathcal{T} = \{t_1, t_2, \dots, t_S\}$, where S is the total number of observations or snapshots. In this case, we can define

$$\mathcal{M} = \begin{pmatrix} \sum_{s=1}^S M^{[11]}(t_s)/\bar{m}^{[11]} & \cdots & \sum_{s=1}^S M^{[1L]}(t_s)/\bar{m}^{[1L]} \\ \vdots & \ddots & \vdots \\ \sum_{s=1}^S M^{[L1]}(t_s)/\bar{m}^{[L1]} & \cdots & \sum_{s=1}^S M^{[LL]}(t_s)/\bar{m}^{[LL]} \end{pmatrix}, \quad (6)$$

where $\bar{m}^{[l_1 l_2]} = \sum_{s=1}^S m^{[l_1 l_2]}(t_s)$, $l_1, l_2 \in [L]$, and write the modularity function as

$$Q(\mathbf{e}, \{\mathcal{G}(t_s)\}_{s \in [S]}) = \sum_{1 \leq l_1, l_2 \leq L} Q^{[l_1 l_2]}(\mathbf{e}, \{\mathcal{G}(t_s)\}_{s \in [S]}),$$

and

$$Q^{[l_1 l_2]}(\mathbf{e}, \{\mathcal{G}(t_s)\}_{s \in [S]}) = \frac{\sum_{s=1}^S m^{[l_1 l_2]}(t_s) Q^{[l_1 l_2]}(\mathbf{e}, \mathcal{G}(t_s))}{\sum_{s=1}^S m^{[l_1 l_2]}(t_s)},$$

where $Q^{[l_1 l_2]}(\mathbf{e}, \mathcal{G}(t_s)) = \frac{1}{m^{[l_1 l_2]}(t_s)L^2} \sum_{ij} M_{ij}^{[l_1 l_2]}(t_s) 1(e_i^{[l_1]} = e_j^{[l_2]})$. The above modularity function can be considered as an averaged version of the modularity in each graph $\mathcal{G}(t_s)$, $s \in [S]$.

3. Modularity Maximization

We aim to find the community assignment that maximizes the modularity function (5), that is,

$$\hat{\mathbf{e}} = \arg \max_{\substack{\mathbf{e} = (\mathbf{e}^{[1]}, \dots, \mathbf{e}^{[L]}), \\ e_i^{[l]} \in \{1, \dots, K\}}} Q(\mathbf{e}, \{\mathcal{G}(t)\}_{t \in \mathcal{T}}). \quad (7)$$

Finding the exact maximizer of (5) is challenging due to the combinatorial nature of the problem and the fact that the number of communities K is generally unknown. Brandes et al. (2008) showed that finding the partition that maximizes the modularity function for a simple graph is NP-hard. There are a number of existing heuristic algorithmic solutions to maximizing the modularity function, some of which are fast and hence feasible for very large networks (Clauset, Newman, and Moore 2004; Wakita and Tsurumi 2007; Blondel et al. 2008), while some others could be more precise though restricted to graphs of moderate sizes (Guimera, Sales-Pardo, and Amaral 2004; Massen and Doye 2005).

In our approach, we adopt a fast Louvain-type maximization method. The Louvain method was first proposed by Blondel et al. (2008) for modularity maximization in simple graphs. In the Louvain method, small communities are first identified by optimizing the modularity function locally on all nodes. Then each small community is grouped into one “meta” node and the first step is repeated. The Louvain method is fast to compute and enjoys a good empirical performance. It has been successfully applied to network analyses from various scientific fields, permitting up to 100 million nodes and billions of edges. Notably, the modularity maximum found by the Louvain method often compares favorably with those found by alternative methods such as Clauset, Newman, and Moore (2004) and Wakita and Tsurumi (2007); see Fortunato (2010).

Motivated by the Louvain algorithm, we propose a dynamic heterogeneous network modularity maximization algorithm, referred to DHNet. To do so, we first define a *unit*, which is a group of nodes with at most one from each node type and it can be regarded as a “meta” vertex in a heterogeneous network. For example, a unit may contain one node of any type or L nodes of different types. In our algorithm, a unit serves as the building block and a community is built to contain a group of units. The restriction that each unit can take at most one node of each type is needed to distinguish a unit and a community. The concept of a unit is developed for community detection in heterogeneous networks. In a homogeneous network where there is only one type of nodes, a unit becomes the same as a node. Next, given a heterogeneous $n \times n$ modularity matrix $\mathcal{M}$ as in (6), we define a *modularity network*, which is a network of n nodes and the edge between nodes (i, j) is $\mathcal{M}_{ij}$. From (5) and (7), it is easy to see that our optimization task is to find a partition of the modularity network such that the within-community sum of edges from $\mathcal{M}$ is maximized.

The DHNet has three phases and is summarized as **Algorithm 1. Phase 0:** to start, the algorithm assigns each node to

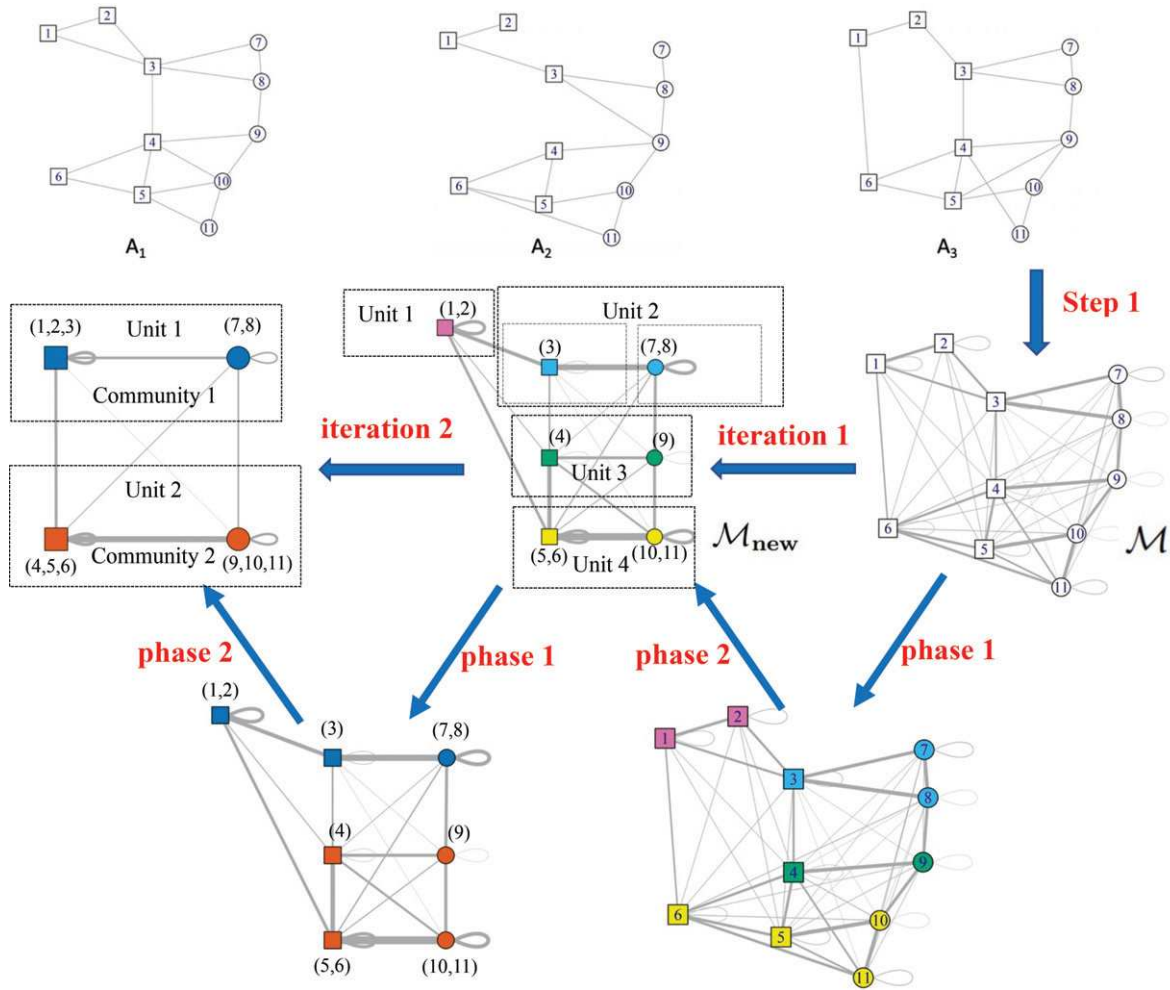


Figure 3. A simple illustration of DHNet. Nodes of the same type are marked using the same shape and nodes of the same color are in the same community.

its own unit and each unit to its own community. Hence, there are n units and n communities initially, where n is the total number of nodes in the network. The algorithm DHNet then seeks to combine units to form larger communities. **Phase 1:** for each unit i , DHNet removes it from its current community and assigns it to its neighboring community (communities to which the unit is linked to), such that it leads to the largest increase of the modularity function. If no move increases the modularity, unit i remains in its current community. This step is repeated to all units until none can be moved and the ordering in which the units are considered are random. The algorithm then moves to the next stage that forms a new “reduced” network. **Phase 2:** taken the communities formed after Phase 1, DHNet merges nodes of the same type in each community, such that each community becomes a unit in the new network and the edge between merged nodes in the new network are given by summing the edge weights connecting per-merging nodes. This is further illustrated via a concrete example below. With this new network, DHNet then returns to Phase 1 and perform community assignment.

Figure 3 shows with an illustrative example of DHNet applied to a dynamic heterogeneous network with $n = 11$ nodes and $L = 2$ node types. There are two iterations in the implementation and each iteration has two phases, including the community assigning phase and merge phase. The algorithm first

computes an 11×11 modularity matrix $\mathcal{M}$, and then starts with 11 units and 11 communities. **In phase 1 of iteration 1**, four communities are formed with nodes $\{1, 2\}$, $\{3, 7, 8\}$, $\{4, 9\}$ and $\{5, 6, 10, 11\}$ and colored in pink, blue, green and yellow, respectively. **In phase 2 of iteration 1**, nodes of the same type in each community are merged. That is, nodes 1 and 2 are merged into $v_{(1,2)}^*$. Similarly, nodes 5 and 6, 7 and 8, 10 and 11 are merged into $v_{(5,6)}^*$, $v_{(7,8)}^*$, $v_{(10,11)}^*$, respectively. Nodes 3, 4, 9, denoted as v_3^* , v_4^* , v_9^* , are not merged with others, as there are no other nodes of the same type in the communities that they are in. Next, each of the four communities is treated as a unit in the new network. After this step, we have a new network with seven nodes $v_{(1,2)}^*$, v_3^* , v_4^* , $v_{(5,6)}^*$, $v_{(7,8)}^*$, v_9^* , $v_{(10,11)}^*$ and four units, where unit 1 is $\{v_{(1,2)}^*\}$, unit 2 is $\{v_3^*, v_{(7,8)}^*\}$, unit 3 is $\{v_4^*, v_9^*\}$, and unit 4 is $\{v_{(5,6)}^*, v_{(10,11)}^*\}$. We then calculate a new modularity matrix $\mathcal{M}_{new}$ of dimension 7×7 . The edges of $\mathcal{M}_{new}$ are calculated by summing weights of the corresponding nodes that are merged. For example, the edge between nodes $v_{(5,6)}^*$ and $v_{(10,11)}^*$ are obtained by summing four edge weights (v_5, v_{10}) , (v_5, v_{11}) , (v_6, v_{10}) , and (v_6, v_{11}) from the original modularity matrix. **In phase 1 of iteration 2**, we compute the change in modularity when unit 1 = $\{v_{(1,2)}^*\}$ is placed with unit 2 = $\{v_3^*, v_{(7,8)}^*\}$ in the same community. The same is calculated for unit 1 with 3 and unit 1 with 4. We then place unit 1 with the one that increases the modularity the most. If no such move is found, then unit 1 remains in its own community. This is

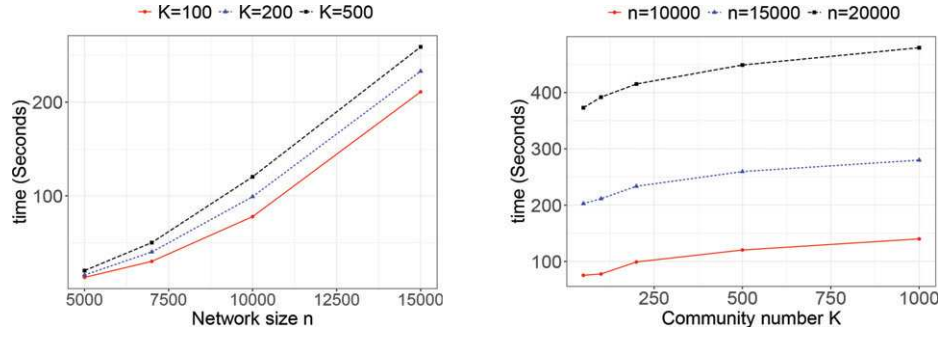


Figure 4. The computing time of DHNet with a varying network size n (left) and number of communities K (right).

Algorithm 1 Dynamic Heterogeneous Network Modularity Maximization (DHNet)

Input: Dynamic heterogeneous networks $A_1, \dots, A_S$.

Step 1: Calculate the modularity matrix using (6).

Step 2: Assign each node to its own unit and assign each unit to its own community.

Step 3: Repeat Steps 3.1–3.4 until the modularity value no longer increase.

Step 3.1: For each unit, place it into the neighboring community that leads to the largest modularity increase in (5). If no such move is possible, then this unit stays in its present community.

Step 3.2: Repeated apply Step 3.1 to all units until none can be moved.

Step 3.3: If the modularity is higher than that from the previous iteration, merge nodes of the same type in each community such that each community is regarded as a unit and go to Step 3.4. If not, exit with the assignment from the previous iteration.

Step 3.4: Calculate the modularity matrix of the merged network.

Output: Community assignment and the corresponding modularity value.

done for units 2, 3 and 4 as well. After this step, two communities are formed as $\{v_{(1,2)}^*, v_3^*, v_{(7,8)}^*\}$ and $\{v_4^*, v_{(5,6)}^*, v_9^*, v_{(10,11)}^*\}$, respectively. In phase 2 of iteration 2, nodes of the same type in each community are merged. That is, nodes $v_{(1,2)}^*$ and v_3^* in community 1 are merged into $v_{(1,2,3)}^*$, nodes v_4^* and $v_{(5,6)}^*$ in community 2 are merged into $v_{(4,5,6)}^*$ and nodes v_9^* and $v_{(10,11)}^*$ in community 2 are merged into $v_{(9,10,11)}^*$. Each community is then treated as a unit in the new network. After this step, we have a new network with four nodes $v_{(1,2,3)}^*, v_{(4,5,6)}^*, v_{(7,8)}^*, v_{(9,10,11)}^*$ and two units, where unit 1 is $\{v_{(1,2,3)}^*, v_{(7,8)}^*\}$ and unit 2 is $\{v_{(4,5,6)}^*, v_{(9,10,11)}^*\}$. We then calculate a new modularity matrix $\mathcal{M}_{\text{new}}$ of dimension 4×4 . Finally, merging these two units cannot further increase the modularity. Thus, DHNet returns two communities with the first community including nodes 1, 2, 3, 7, 8 and the second community including nodes 4, 5, 6, 9, 10, 11.

Remark 1 (initialization). In Step 3.1, if there are multiple communities that lead to the same maximum modularity increase, DHNet randomly selects a community to assign the unit to. Hence, the result of DHNet may differ each time the algorithm

is implemented. Moreover, the result of the algorithm may differ depending on the node ordering in Step 2. That is, a node ordering of $\{1, 2, 3\}$ or $\{3, 1, 2\}$ may give different results. These two sources of randomness in the implementation of DHNet are discussed in Section S2 of the supplementary materials. We recommend applying the Louvain method κ times with random node orderings and using the assignment with the largest modularity function value as the final output. In our simulation studies and real data analysis, we set $\kappa = 100$ and notice that the output from DHNet is not sensitive to node orderings. Generally, it is recommended that κ should increase with the size of the network.

Remark 2 (time complexity). The complexity of computing the modularity matrix input in DHNet is $O(nS)$, where n is the number of nodes and S is the number of snapshots. In Step 3, computing whether and where to move each unit based on modularity changes is of time complexity $O(1)$, assuming node degrees are bounded. The complexity per iteration of DHNet is $O(m)$, where m is the total number of edges in the network. Consequently, the total running time of DHNet can be upper-bounded by $O(\gamma m)$, where γ is the total number of iterations. While no upper bound has been established on the number of iterations in a Louvain-type method, we find DHNet converges within tens of iterations in practice. Figure 4 provides the computation time of DHNet with a varying network size n and number of communities K . Specifically, we set $S = 20$ and generate networks from DHSBM where the inter- and intra-community connecting probabilities are 0.1 and 0.15, respectively, in the homogeneous networks, and the inter- and intra-community connecting probabilities are, respectively, 0.05 and 0.1 in the multi-partite networks. It is seen that the computing time is roughly linear in n and K . All experiments are run on an Intel(R) Xeon(R) with 3.10GHz and 192 GB memory processor.

4. Consistency

In this section, we investigate the theoretical properties of DHNet for finding common communities in a dynamic heterogeneous network. To do so, we first propose a discrete-time heterogeneous stochastic block model with a temporal correlation structure.

Dynamic Heterogeneous Stochastic Block Model (DHSBM)

1. Dynamic heterogeneous network $\{\mathcal{G}(t_s), s \in [S]\}$ with L node types has a latent community label $\mathbf{c} = (c^{[1]}, \dots, c^{[L]})$,

where $\mathbf{c}^{[l]} = (c_1^{[l]}, \dots, c_{n_l}^{[l]})$ and $c_i^{[l]} \in \{1, \dots, K\}$ denotes the community that node i of type- $[l]$ belongs to, $l \in [L]$.

2. The label $\mathbf{c}^{[l]}$ follows a multinomial distribution with n_l trials and probability $\boldsymbol{\pi}^{[l]} = (\pi_1^{[l]}, \dots, \pi_K^{[l]})$, $l \in [L]$.
3. Define the time-varying probability matrix $\boldsymbol{\Theta}(t_s)$ such that

$$\boldsymbol{\Theta}(t_s) = \begin{pmatrix} \Theta^{[11]}(t_s) & \dots & \Theta^{[1L]}(t_s) \\ \vdots & \ddots & \vdots \\ \Theta^{[L1]}(t_s) & \dots & \Theta^{[LL]}(t_s) \end{pmatrix},$$

$$\text{where } \Theta^{[l_1 l_2]}(t_s) = \begin{pmatrix} \theta_{11}^{[l_1 l_2]}(t_s) & \dots & \theta_{1K}^{[l_1 l_2]}(t_s) \\ \vdots & \ddots & \vdots \\ \theta_{K1}^{[l_1 l_2]}(t_s) & \dots & \theta_{KK}^{[l_1 l_2]}(t_s) \end{pmatrix}$$

and $\theta_{k_1 k_2}^{[l_1 l_2]}(t_s)$ is a function of t_s , $l_1, l_2 \in [L]$, and $k_1, k_2 \in [K]$. Specifically, $\Theta^{[l_1 l_2]}(t_s)$ is a $K \times K$ probability matrix that specifies the connecting probabilities between type- $[l_1]$ nodes and type- $[l_2]$ nodes in different communities at time t_s , and each element $\theta_{k_1 k_2}^{[l_1 l_2]}(t_s)$ represents the probability of generating an edge between a type- $[l_1]$ node in community k_1 and a type- $[l_2]$ node in community k_2 at time t_s .

4. Given $\mathbf{c}$, we treat $A_{ij}^{[l_1 l_2]}(t_s)$'s as independent Bernoulli random variables satisfying

$$A_{ij}^{[l_1 l_2]}(t_s) = u A_{ij}^{[l_1 l_2]}(t_{s-1}) + (1 - u) v^{[l_1 l_2]},$$

where $u \stackrel{\text{iid}}{\sim} \text{Bernoulli}(\alpha)$, and given $c_i^{[l_1]} = k_1$ and $c_j^{[l_2]} = k_2$,

$$v^{[l_1 l_2]} \stackrel{\text{iid}}{\sim} \text{Bernoulli} \left(\frac{\theta_{k_1 k_2}^{[l_1 l_2]}(t_s) - \alpha \theta_{k_1 k_2}^{[l_1 l_2]}(t_{s-1})}{1 - \alpha} \right),$$

$$l_1, l_2 \in [L].$$

In Assumption 4, we use a two-step design to impose a temporal correlation structure, with α controlling the strength of correlation. Also, we require that $0 \leq \alpha < 1$, $\alpha \theta_{k_1 k_2}^{[l_1 l_2]}(t_{s-1}) \leq \theta_{k_1 k_2}^{[l_1 l_2]}(t_s)$, and $\alpha (1 - \theta_{k_1 k_2}^{[l_1 l_2]}(t_{s-1})) \leq 1 - \theta_{k_1 k_2}^{[l_1 l_2]}(t_s)$, so that the above Bernoulli distribution is valid with the probability parameter in $[0, 1]$. Here, it is possible to let $u \stackrel{\text{iid}}{\sim} \text{Bernoulli}(\alpha^{[l_1 l_2]})$, though we assume $\alpha^{[l_1 l_2]} = \alpha$ to simplify notation. Based on Assumption 4, some algebra shows that

$$\mathbb{P}(A_{ij}^{[l_1 l_2]}(t_s) = 1) = \theta_{k_1 k_2}^{[l_1 l_2]}(t_s),$$

which shows that the marginal distribution of $A_{ij}^{[l_1 l_2]}(t_s)$ is Bernoulli $(\theta_{k_1 k_2}^{[l_1 l_2]}(t_s))$. Hence, for a fixed t_s , $A^{[l_1 l_2]}(t_s)$ follows a stochastic block model with a probability matrix $\Theta^{[l_1 l_2]}(t_s)$. Additionally, under our DHSBM model, we have

$$\text{corr}(A_{ij}^{[l_1 l_2]}(t_s), A_{ij}^{[l_1 l_2]}(t_{s-1}))$$

$$= \alpha^{[l_1 l_2]} \sqrt{\frac{\theta_{k_1 k_2}^{[l_1 l_2]}(t_{s-1}) (1 - \theta_{k_1 k_2}^{[l_1 l_2]}(t_{s-1}))}{\theta_{k_1 k_2}^{[l_1 l_2]}(t_s) (1 - \theta_{k_1 k_2}^{[l_1 l_2]}(t_s))}}.$$

And for the special case $\alpha^{[l_1 l_2]} = 0$, $A_{ij}(t_s), s = 1, \dots, S$, are independent. If $\Theta(t_s)$ is constant over time, then $\text{corr}(A_{ij}^{[l_1 l_2]}(t_s), A_{ij}^{[l_1 l_2]}(t_{s-k})) = (\alpha^{[l_1 l_2]})^k$ for $k = 1, 2, \dots$.

Next, we show the consistency property of the estimated assignment vector $\hat{\mathbf{c}}$ under the DHSBM model when the network size n and the number of time points increases in that $nS \rightarrow \infty$. This regime is more general and includes the results from Zhang and Cao (2017) and Zhang and Chen (2020) as special cases. We say a label $\mathbf{e} = (\mathbf{e}^{[1]}, \dots, \mathbf{e}^{[L]})$ is consistent if it satisfies

$$\forall \epsilon > 0, \quad P \left[\frac{1}{n} \sum_{l=1}^L \sum_{i=1}^{n_l} I(\mathbf{e}_i^{[l]} \neq \mathbf{c}_i^{[l]}) < \epsilon \right] \rightarrow 1 \text{ as } nS \rightarrow \infty,$$

which stipulates that the misclassification ratio tends to zero. Here $\hat{\mathbf{c}}_i^{[l]} = \mathbf{c}_i^{[l]}$ means that they belong to the same equivalent class of label permutations. To allow sparsity, we reparameterize $\boldsymbol{\Theta}(t_s)$ as $\tilde{\boldsymbol{\Theta}}(t_s) = \rho_{n,S} \boldsymbol{\Theta}(t_s)$, where $\boldsymbol{\Theta}(t_s)$ is fixed as $nS \rightarrow \infty$. This reparameterization allows us to separate $\rho_{n,S}$, the sparsity parameter, from the structure of the network.

Theorem 1. Consider a dynamic heterogeneous network $\mathcal{G}(\bigcup_{i=1}^L V^{[i]}, \mathcal{E}(t_s) \cup \mathcal{E}^+(t_s))$ from the DHSBM with $\mathbf{c}, \boldsymbol{\pi}^{[l]}$'s, α and $\boldsymbol{\Theta}(t_s)$'s, and further assume that the community sizes are balanced, that is, $\min_l n_l/n$ is bounded away from zero. Define a $K \times K$ matrix

$$T_{ab}^{[l_1 l_2]}(t_s) = \frac{\pi_a^{[l_1]} \pi_b^{[l_2]} \theta_{ab}^{[l_1 l_2]}(t_s)}{\sum_{ab} \pi_a^{[l_1]} \pi_b^{[l_2]} \theta_{ab}^{[l_1 l_2]}(t_s)}.$$

Let $W_{[ab]}^{[l_1 l_2]}(t_s) = T_{ab}^{[l_1 l_2]}(t_s) - T_{a.}^{[l_1 l_2]}(t_s) T_{.b}^{[l_1 l_2]}(t_s)$ with $T_{a.}^{[l_1 l_2]}(t_s) = \sum_{q=1}^K T_{aq}^{[l_1 l_2]}(t_s)$. If the following assumptions hold

$$\sum_{s=1}^S \sum_{l_1, l_2} W_{aa}^{[l_1 l_2]}(t_s) > 0 \quad \text{and} \quad \sum_{s=1}^S \sum_{l_1, l_2} W_{ab}^{[l_1 l_2]}(t_s) < 0$$

$$\text{for all } a \neq b \in [K], \quad (8)$$

and $nS\rho_{n,S} \rightarrow \infty$, then we have

$$\forall \epsilon > 0, \quad P \left[\frac{1}{n} \sum_{l=1}^L \sum_{i=1}^{n_l} I(\hat{\mathbf{c}}_i^{[l]} \neq \mathbf{c}_i^{[l]}) < \epsilon \right] \rightarrow 1 \text{ as } nS \rightarrow \infty,$$

where $\hat{\mathbf{c}}$ is the maximizer of (5).

It is seen that the network is allowed to be highly sparse at each time point s , for example, the probability of forming an edge can be $O(\frac{\log(nS)}{nS})$. Denoting the average degree as $\lambda = n\rho_{n,S}$, it is seen that consistency is achievable when $\lambda S \rightarrow \infty$, while the single network case requires $\lambda \rightarrow \infty$ to achieve community detection consistency (Zhang and Chen 2020). When $L = 1$, the above result reduces to that in Zhang and Cao (2017) and when $S = 1$, the above result reduces to that in Zhang and Chen (2020). We note that Zhang and Cao (2017) only considered the case where the network size n is fixed and their results require $\rho_{n,S} = O(1)$. In comparison, our result in Theorem 1 allows n and/or S to diverge and only requires $nS\rho_{n,S} \rightarrow \infty$ as $nS \rightarrow \infty$.

The condition in (8) requires that edges are on average more likely to be established within communities than they

are between communities, though communities may not exist within all types of nodes or at all time points. For example, in the simulation setting in Section 5.2, there is no community structure within type-[1] nodes and/or type-[2] nodes, while exists a community structure between the type-[1] and type-[2] nodes at some time points. This type of assortative condition, requiring more edges within communities than between communities, is often required for algorithm-based community detection such as modularity maximization. For the special case of $L = 1$, $K = 2$, and $\Theta(t_s)$ is time homogeneous, the condition (8) can be simplified as

$$\theta_{11}^{[11]}\theta_{22}^{[11]} > \left(\theta_{12}^{[11]}\right)^2.$$

When $L = 2$, $K = 2$ and $\Theta(t_s)$ is time-varying, the condition (8) is satisfied if

$$\begin{aligned} & \sum_{s=1}^S \left(\theta_{11}^{[11]}(t_s) + \theta_{11}^{[22]}(t_s) + \theta_{11}^{[12]}(t_s) + \theta_{11}^{[21]}(t_s) \right) \\ & > \sum_{s=1}^S \left(\theta_{12}^{[11]}(t_s) + \theta_{12}^{[22]}(t_s) + \theta_{12}^{[12]}(t_s) + \theta_{12}^{[21]}(t_s) \right), \\ & \sum_{s=1}^S \left(\theta_{22}^{[11]}(t_s) + \theta_{22}^{[22]}(t_s) + \theta_{22}^{[12]}(t_s) + \theta_{22}^{[21]}(t_s) \right) \\ & > \sum_{s=1}^S \left(\theta_{12}^{[11]}(t_s) + \theta_{12}^{[22]}(t_s) + \theta_{12}^{[12]}(t_s) + \theta_{12}^{[21]}(t_s) \right), \end{aligned}$$

which indicate that edges are more likely to form within communities than between communities.

5. Simulation

In this section, we evaluate the clustering accuracy of DHNet and compare it with several alternative solutions including:

Method 1: treat the dynamic heterogeneous network as a dynamic homogeneous network without distinguishing the different node and edge types and apply a dynamic network community detection method (Zhang and Cao 2017).

Method 2: apply a heterogeneous community detection method (Zhang and Chen 2020) to an aggregated matrix $\bar{A} = \begin{pmatrix} \bar{A}^{[11]} & \dots & \bar{A}^{[1L]} \\ \vdots & \ddots & \vdots \\ \bar{A}^{[L1]} & \dots & \bar{A}^{[LL]} \end{pmatrix}$, where $\bar{A}_{ij}^{[l_1 l_2]} = \max_t A_{ij}^{[l_1 l_2]}(t)$, that is, detect community based on a static summary heterogeneous graph.

Method 3: infer the community label from $\mathcal{G}(t_s)$ for a randomly selected time point t_s in $\{t_1, \dots, t_S\}$. That is, community detection based on a single snapshot of the dynamic heterogeneous network, which is the same as Zhang and Chen (2020).

Method 4: decompose the dynamic heterogeneous network with L different types of nodes into L dynamic homogeneous networks and apply a dynamic network community detection method (Zhang and Cao 2017) to each separately, that is, discard information from the edges linking different types of nodes.

We generate networks from the DHSBM proposed in Section 4 with L types of nodes, K communities and S equal-spaced observations within the time interval $[0, 1]$. We consider

three different settings in our experiments including a time-homogeneous DHSBM with independently sampled networks in Section 5.1, a DHSBM with independently sampled networks in Section 5.2 and a DHSBM with temporally correlated networks in Section 5.3. In each setting, we consider dense and sparse networks. We set $L = 2$, $K = 3$, $n_1 = 300$, $n_2 = 150$ and $\pi^{[1]} = \pi^{[2]} = (1/3, 1/3, 1/3)$. To evaluate the clustering accuracy, we adopt the normalized mutual information (NMI) (Danon et al. 2005), a commonly used metric in community detection experiments to quantify the difference between two clustering labels. Some additional simulation results for $L = 3$ and $K = 4$ are illustrated in Section S3.2 of the supplementary materials.

5.1. Simulation Setting 1

We consider networks independently sampled from a DHSBM with a time-homogeneous probability matrix defined as

$$\Theta(t) = \begin{pmatrix} \theta_1 + r_1 & \theta_1 & \theta_1 & \theta_3 + r_3 & \theta_3 & \theta_3 \\ \theta_1 & \theta_1 + r_1 & \theta_1 & \theta_3 & \theta_3 + r_3 & \theta_3 \\ \theta_1 & \theta_1 & \theta_1 + r_1 & \theta_3 & \theta_3 & \theta_3 + r_3 \\ \theta_3 + r_3 & \theta_3 & \theta_3 & \theta_2 + r_2 & \theta_2 & \theta_2 \\ \theta_3 & \theta_3 + r_3 & \theta_3 & \theta_2 & \theta_2 + r_2 & \theta_2 \\ \theta_3 & \theta_3 & \theta_3 + r_3 & \theta_2 & \theta_2 & \theta_2 + r_2 \end{pmatrix}.$$

In $\Theta(t)$ with $L = 2$ and $K = 3$, the parameter θ_1 (θ_2) represents the connecting probability between type-[1] (type-[2]) nodes in different communities, $\theta_1 + r_1$ ($\theta_2 + r_2$) represents the connecting probability between type-[1] (type-[2]) nodes in the same community, θ_3 represents the connecting probability between type-[1] and type-[2] nodes in different communities, and $\theta_3 + r_3$ represents the connecting probability between type-[1] and type-[2] nodes in the same community. Hence, the θ_k 's are the between community connectivity probabilities and $\theta_k + r_k$'s are the within community connectivity probabilities. By varying the values of r_1 , r_2 and r_3 , we can control the signal strength for community structures.

Scenario 1 (Sc-1): $\theta_1 = 0.5, \theta_2 = 0.6, \theta_3 = 0.3, r_1 = 0, r_2 = 0$,

Scenario 2 (Sc-2): $\theta_1 = 0.1, \theta_2 = 0.2, \theta_3 = 0.05, r_1 = 0, r_2 = 0$.

In Scenarios 1 and 2, neither $G^{[1]}$ or $G^{[2]}$ has a community structures. We have also considered the case where $G^{[1]}$ has a weak community structure while $G^{[2]}$ has no community structure. The results are similar to those from Scenarios 1 and 2 and delayed to Section S3.1 of supplementary materials. We set $S = 20$ and vary r_3 , that is, the strength of the community structure in $G^{[12]}$, from 0.05 to 0.15. Figure 5 summarizes the community detection results averaged over 100 data replicates for Scenarios 1-2, respectively.

For dense networks in Scenario 1, it is seen from the left panel in Figure 5 that DHNet outperforms the other methods on all values of r_3 . The NMIs from Methods 1–4 are below 0.25 for both types of nodes. For Method 1, the clustering output places nodes of the same type in the same community, leading to an NMI close to zero. For Method 2, the aggregated network becomes very dense and the number of inter-community edges is very similar to that of the intra-community edges for each edge

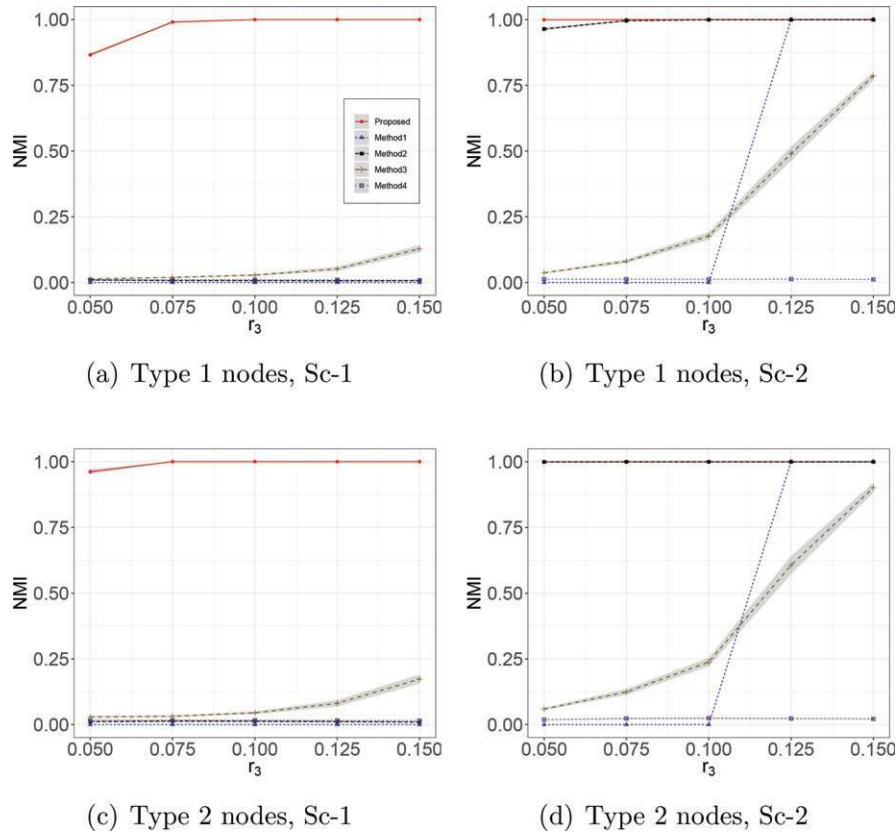


Figure 5. Average NMIs against the value of r_3 for different methods in Setting 1 under Scenarios 1-2. The gray bands are the 95% confidence intervals for the mean NMI.

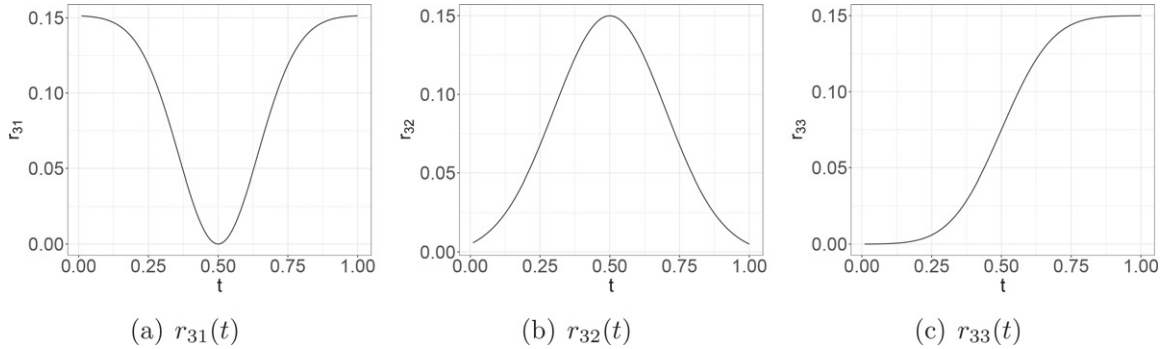


Figure 6. The three time-varying functions $r_{31}(t)$, $r_{32}(t)$ and $r_{33}(t)$.

type, leading to an NMI close to zero. Method 3 detects community based on a random snapshot of network, which contains relatively weak structural information, leading to a lower NMI. In addition, Method 4 ignores the edges linking different types of nodes and hence performs worse when no community structure exists among the investigated type of nodes.

For sparse networks in Scenario 2, it is seen from the right panel in Figure 5 that the performance of Methods 1 and 3 increases notably with r_3 , as the community structure strength (i.e., the difference between the inter- and intra- community connecting probability) is high in this scenario. Due to this reason, Method 2 also performs better in the sparse case as the community structure signal is strong in the aggregated network, with many more inter-community edges than intra-community edges. Our method still outperforms most of the other methods when the signal is weak, for example, $r_3 \leq 0.1$.

5.2. Simulation Setting 2

We consider networks independently sampled from a DHSBM with a time-varying probability matrix defined as

$$\Theta(t) = \begin{pmatrix} \theta_1 + r_1 & \theta_1 & \theta_1 & \theta_3 + r_{31}(t) & \theta_3 & \theta_3 \\ \theta_1 & \theta_1 + r_1 & \theta_1 & \theta_3 & \theta_3 + r_{32}(t) & \theta_3 \\ \theta_1 & \theta_1 & \theta_1 + r_1 & \theta_3 & \theta_3 & \theta_3 + r_{33}(t) \\ \theta_3 + r_{31}(t) & \theta_3 & \theta_3 & \theta_2 + r_2 & \theta_2 & \theta_2 \\ \theta_3 & \theta_3 + r_{32}(t) & \theta_3 & \theta_2 & \theta_2 + r_2 & \theta_2 \\ \theta_3 & \theta_3 & \theta_3 + r_{33}(t) & \theta_2 & \theta_2 & \theta_2 + r_2 \end{pmatrix}.$$

We set θ_1 , θ_2 , θ_3 , r_1 , and r_2 the same as those in the two scenarios in Simulation setting 1 and $r_{31}(t)$, $r_{32}(t)$, and $r_{33}(t)$ as plotted in Figure 6. In this setting, at time $t = 0$, community 1 in $G^{[12]}$ is active while communities 2–3 are inactive; at time $t = 0.5$, community 1 in $G^{[12]}$ becomes inactive while communities 2–3 are active; at time $t = 1$, community 2 in $G^{[12]}$

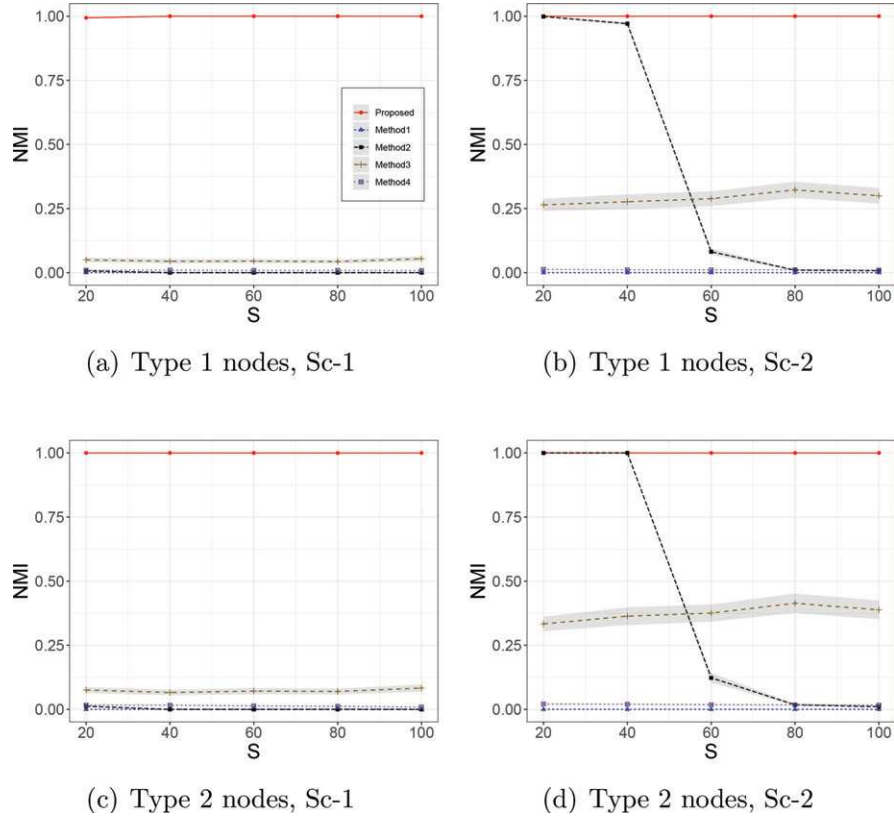


Figure 7. Average NMIs against the value of S for different methods in Setting 2 under Scenarios 1-2.

becomes inactive while communities 1 and 3 are both active. We consider S in $[20, 100]$, and Figure 7 summarizes the community detection results averaged over 100 data replicates for Scenarios 1-2, respectively.

It is seen that DHNet performs better than Methods 1-4 for all values of S , regardless the sparsity of the networks. Interestingly, the performance of Method 2 in Figure 7 is much worse than that in Figure 5 from Simulation 1 when S is large. This is because the connecting probability is time-varying in Simulation 2, and the signal from communities that are active at different time points may get ablated in an aggregated picture when S is large.

5.3. Simulation Setting 3

We consider temporally correlated network samples from a DHSBM with a time-varying connecting probability matrix. Specifically, we adopt the time-varying connecting probability $\Theta(t)$ from Simulation setting 2. At time t_s , the edge $A_{ij}^{[l_1 l_2]}(t_s)$ is a Bernoulli random variable with

$$A_{ij}^{[l_1 l_2]}(t_s) = uA_{ij}^{[l_1 l_2]}(t_{s-1}) + (1-u)v^{[l_1 l_2]},$$

where $u \sim \text{textiid Bernoulli}(\alpha)$ and

$$v^{[l_1 l_2]} \stackrel{\text{textiid}}{\sim} \text{Bernoulli} \left(\frac{\theta_{k_1 k_2}^{[l_1 l_2]}(t_s) - \alpha \theta_{k_1 k_2}^{[l_1 l_2]}(t_{s-1})}{1 - \alpha} \right),$$

$$l_1, l_2 \in [L].$$

Given $\Theta(t)$, a larger α leads to a higher correlation between the networks at two adjacent time points. We set $S = 100$,

and $\alpha \in [0, 0.4]$, as to keep the probability parameter $\frac{\theta_{k_1 k_2}^{[l_1 l_2]}(t_s) - \alpha \theta_{k_1 k_2}^{[l_1 l_2]}(t_{s-1})}{1 - \alpha} > 0$. Figure 8 summarizes the community detection results averaged over 100 data replicates for Scenarios 1-2, respectively.

Similar conclusions as before can be drawn for all methods shown in Figure 8. DHNet has the best performance out of the five methods. When $\alpha = 0$, $A_{ij}(t)$ is uncorrelated with the past observations, and the model is equivalent to the model used in Simulation setting 2. In fact, when the sample size S is small, the effective sample size decreases as α increases, which may lead to deterioration in the performance of DHNet. This deterioration is demonstrated by the green star line in Figure 8(a), which represents the case of $S = 20$. However, when the sample size S is large enough, as in the case of $S = 100$ considered in this part, such a tendency of deterioration can be mitigated. Method 3 relies only on a random snapshot of network and as such, it is insensitive to changes in α .

6. Yelp Review Network

Yelp is a well-known review website, founded in 2004 in the United States. It collects reviews on a wide range of businesses such as restaurants, bars and shops from many countries. On the Yelp platform, users can rate businesses, submit reviews, and share experiences. We analyze the review data from the Yelp Challenge (<https://www.kaggle.com/yelp-dataset/yelp-dataset>) during the period from January 1st, 2006 to December 31, 2017. This dataset contains a set of businesses and the category labels of each business (a business usually has

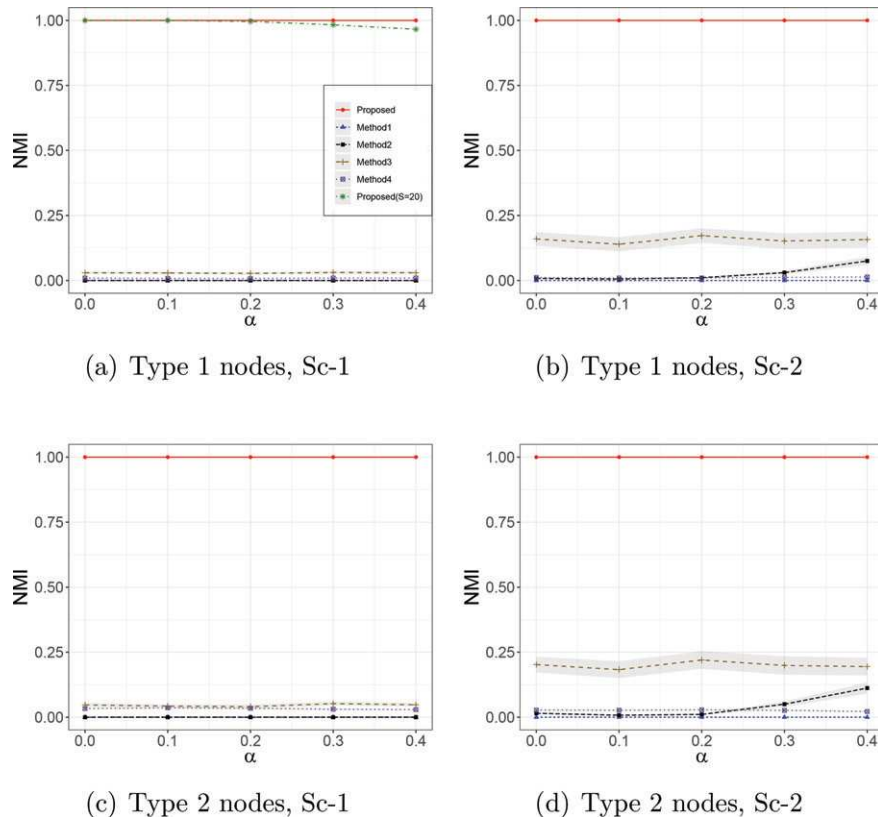


Figure 8. Average NMIs against the value of α for different methods in Setting 3 under Scenarios 1-2.

several labels), a set of users and the friendship information among these users, and the reviews of these businesses by these users. The businesses and users are anonymized and labeled with numerical identifiers. In this dataset, the business-category (business is labeled with category) and user-user (user is friend with user) information are not labeled by time (i.e., not time-varying) while the user-business (business reviewed by user) interactions are labeled by time, and a user may review a business several times.

Our analysis focuses on finding heterogeneous communities in the Yelp review network and predicting interests for new users. Our results show improvements both in terms of accuracy and interpretability over alternative solutions, and demonstrate the need to consider network heterogeneity and dynamics in community detection.

6.1. Finding Heterogeneous Communities

To get a comprehensive view of the Yelp review network, we consider a heterogeneous network with three types of nodes including business, user and category, connected via three types of edges, user-user (user is friend with user), user-business (business is reviewed by user), and business-category (business is labeled with category); see Figure 1 for a simple illustration. As discussed earlier, the user-user and business-category edges are not time-varying but the user-business edges are. We focus on businesses that operated continuously in the study period, business categories that had at least 10 occurrences and users that reviewed at least 20 times in the study period. This gives a total of 3566 businesses, 207 categories and 5116 users, with

Table 1. Summary of the 11 communities identified by DHNet.

	categories	# of users	# of businesses	theme
1	<i>Animal Shelters, Pet Groomers</i> <i>Pet Services, Veterinarians</i>	719	331	Pets
2	<i>Tex-Mex, Southern</i>	939	408	Tex-Mex
3	<i>Tea, Fast Food</i> <i>Diners, Pizza, Restaurants</i>	2629	1036	Casual Dining
4	<i>French, Pasta Shops, Steak House</i> <i>Professional Services, Seafood</i>	679	337	Fine Dining
5	<i>Candy Stores, Farmers Market</i> <i>Chocolatiers & Shops, Grocery</i>	13	239	Stores & Markets
6	<i>Adult Entertainment, Bars</i> <i>Dance Clubs, Beer Bar</i>	46	225	Bars, Entertainment
7	<i>Beauty & Spas, Doctors</i> <i>Hair Salons, Health & Medical</i>	79	441	Beauty & Medical
8	<i>Home Services, Laundry Services</i> <i>Music & Video, Shopping Centers</i>	5	289	Shopping & Life
9	<i>Hotels & Travel, Venues & Event Spaces</i> <i>Landmarks & Historical Buildings, Tours</i>	4	123	Leisure & Travel
10	<i>Buffets, Indian, Pakistani</i>	3	103	Asian Fusion
11	<i>Auto Parts & Supplies, Auto Repair</i> <i>Automotive, Gas Stations, Tires</i>	0	34	Auto

141,744 user-user and 17,280 business-category and 194,712 user-business edges. Due to the high sparsity of user-business edges, we use *year* as the time unit when constructing the dynamic network, that is, the network at time t summarizes the review activity between users and businesses in the t th year of the study period, and correspondingly $S = 12$.

We applied DHNet to the constructed dynamic heterogeneous network with $\kappa = 200$ and identified 11 communities with a maximized modularity value of 0.237. Table 1 shows the representative categories, number of users and number of

businesses in each identified community, along with a summarizing theme. The complete list of categories in each community can be found in the supplement. We found that each community identified by DHNet contains a distinctive type of businesses. For example, Communities 3–4 are on dining and Community 7 is mostly on Beauty & Medical. Users in Community 1 prefer activities related to pets, users in Community 6 prefer bars and entertainment and users in Community 9 show interests in traveling and sports. Community 11 is mostly on Auto and we did not identify users whose main review activity and interests are in this type of businesses. These insights can help us understand the life styles and interests of users in each community. The numbers of users for categories 8–11 detected by DHNet are low, but these communities are still meaningful in summarizing businesses and business categories. Using our method, each user is classified into one community based on their most frequently reviewed businesses and friendship information. As a future research direction, we plan to investigate heterogeneous network clustering with overlapping communities, in which case, a user can belong to more than one communities.

We had also applied Methods 1–3 from Section 5, though we did not implement Method 4, which considers each homogeneous networks separately and discards information from the edges linking different types of nodes, as there are no business-business or category-category edges and the user-user edges are not time-varying. The results from Method 1, which does not distinguish the different node and edge types, are very difficult to interpret. For example, one community contains only businesses and one community contains only users. Method 2, which considers an aggregated heterogeneous network over time, also identified 11 communities (see details of the communities in the supplement). The community detection results from Method 2 are less interpretable compared to DHNet and several communities contain mixed businesses themes. For example, *Hobby Shops* is placed into Community 4 that is on fine dining, and *Colleges & Universities* and *Education* are placed into Community 9 that is on leisure and travel. Method 3, which considers a snapshot of the dynamic work, does not perform well, as the network at each time point is highly sparse with a large number of isolated nodes.

6.2. Prediction Interests for New Users

In this section, we aim to predict the interests, in terms of business categories, for a new Yelp user based on Yelp activities of his/her friends, a practically useful task in making recommendations and placing advertisements. We focus on predicting interests in business categories as opposed to individual businesses, as the number of businesses is large and user-business interactions are highly sparse. For a new Yelp user, the platform can often collect his/her friendship information with other existing Yelp users, by accessing phone contacts, email contacts and Facebook friendship. In terms of make recommendations, the Yelp activities of friends of a new user can help to ease the “cold start” problem, the issue where personalized recommendations cannot be made before a user interacts with the system (e.g., reviewing businesses).

Consider training and testing datasets taken from two different time periods (e.g., data from years 2006 to 2015 as training and years 2015–2017 as testing). We are interested in making predictions for the new users in the testing set, which are user accounts that did not exist in the training data. Specifically, for a new user in the testing set, based on Yelp activities of his/her friends in the training data, we predict the his/her interests over the business categories and compare the prediction with the “true” measure calculated from the testing set.

We compare two different prediction strategies. The first strategy uses community detection results from DHNet in making the prediction and the second strategy is a simple alternative that directly averages interests from the new user’s friends without using any community information, referred to as the naive strategy. Specifically, let g_i denote the interest measure of the i th new user in the testing set, which is a probability distribution over all categories; it is calculated using the appearance frequency of each category in the businesses reviewed by this user. In the first strategy, we apply DHNet to the training network data and find the category distribution of each community, denoted as f_j for the j th community, calculated using the appearance frequency of each category from the businesses in this community. We then make prediction g_i^{DHNet} using the weighted average of f_j ’s as below

$$g_i^{\text{DHNet}} = \sum_j \frac{n_{ij}}{\sum_j n_{ij}} f_j,$$

where n_{ij} denotes the number of friends that the i th user has in the j th community. In this strategy, the prediction is a weighted average of measures from all communities where the weight reflect the number of connections the new user has to each community. In the second strategy, we directly calculate the category distribution g_i^{Naive} based on the businesses that i th user’s friends visited during the training period. This strategy only focuses on the ego-centric network of the new user and does not take into the rich information in the network communities. To assess the prediction accuracy, we use the Jensen–Shannon divergence (JSD) to compare the estimated and observed category distributions, that is,

$$\text{JSD}(\hat{g}_i \| g_i) = \frac{1}{2} D(\hat{g}_i \| m) + \frac{1}{2} D(g_i \| m),$$

where $\hat{g}_i$ refers to the estimated category distribution, $m = \frac{1}{2}(\hat{g}_i + g_i)$ and $D(\hat{g}_i \| m)$ is the Kullback–Leibler divergence between distributions $\hat{g}_i$ and m .

Table 2 compares the performance of the two strategies in 6 different sets of training and testing periods, where

$$\begin{aligned} \text{JSD}^{\text{DHNet}} &= \frac{1}{n_0} \sum_{i=1}^{n_0} \text{JSD}(g_i^{\text{DHNet}} \| g_i), \\ \text{JSD}^{\text{Naive}} &= \frac{1}{n_0} \sum_{i=1}^{n_0} \text{JSD}(g_i^{\text{Naive}} \| g_i) \end{aligned}$$

and n_0 denotes the number of new users in the testing period. It is seen that DHNet outperforms the naive strategy in terms of predicting accuracy in all training and testing datasets, demonstrating the advantage of using community structures when predicting user interests.

Table 2. Jensen–Shannon divergence of the DHNet and naive methods across the 6 moving windows.

training years testing years	2006-2010 2011-2012	2007-2011 2012-2013	2008-2012 2013-2014	2009-2013 2014-2015	2010-2014 2015-2016	2011-2015 2016-2017
JSD ^{DHNet}	0.122	0.133	0.132	0.136	0.135	0.138
JSD ^{Naive}	0.252	0.196	0.199	0.196	0.182	0.196

7. Discussion

Maximizing the modularity function as in (7) is not limited to the Louvain-type method considered in DHNet. Other modularity maximization techniques developed for a homogeneous network may be applied to (7) with some modifications, such as the spectral method based on the eigen decomposition of the modularity matrix or the stochastic optimization method in Massen and Doye (2005). As noted in modularity maximization for other types of networks (Fortunato 2010; Zhang and Chen 2020), we find that the Louvain-type method is computationally much more efficient and yields a good performance in our setting.

While the modularity function value increases at each step of DHNet and the algorithm is guaranteed to converge, there is no guarantee that it will converge to the global optimum. Since the modularity maximization problem is NP-hard, most existing methods are heuristic methods that may only find local optima and are not guaranteed to find the global optimum. A thorough theoretical investigation of the local convergence of DHNet can be helpful. Our proposed method is not designed to identify dynamic or overlapping communities in heterogeneous time-varying networks, that is, when communities vary with time or a node may belong to multiple communities. Finally, our proposed method can be extended to weighted and/or directed networks. To incorporate weighted and/or directed edges into our framework, we need to define a null model for a weighted and/or directed heterogeneous dynamic network, followed by calculating the expectations under the null model. This is an interesting topic to investigate next.

Supplementary Materials

Our narrative supplement provides proofs of theoretical results and additional numerical results for both synthetic and real data. Besides, we provide codes to conduct DHNet and an example to reproduce our simulations results.

Acknowledgement

We are very grateful to three anonymous referees, an associate editor, and the Editor for their valuable comments that have greatly improved the manuscript.

Disclosure Statement

No potential conflict of interest was reported by the author(s).

Funding

Dr. Zhang's research is supported by NSF DMS-2015190 and DMS-2326893. Dr. Dai's research is supported by the National Natural Science

Foundation of China (Grant No. NSFC 11901573, 12171033) and the Beijing Natural Science Foundation (Z200001).

References

- Abbe, E. (2017), "Community Detection and Stochastic Block Models: Recent Developments," *The Journal of Machine Learning Research*, 18, 6446–6531. [487]
- Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008), "Fast Unfolding of Communities in Large Networks," *Journal of Statistical Mechanics: Theory and Experiment*, 2008, P10008. [490]
- Brandes, U., Delling, D., Gaertler, M., Gorke, R., Hoefer, M., Nikoloski, Z., and Wagner, D. (2008), "On Modularity Clustering," *IEEE Transactions on Knowledge and Data Engineering*, 20, 172–188. [490]
- Chung, F., Fan, R., Chung, F. R., Graham, F. C., Lu, L., Chung, K. F., et al. (2006), *Complex Graphs and Networks*, no. 107. American Mathematical Society: Providence, RI. [489]
- Clauset, A., Newman, M. E., and Moore, C. (2004), "Finding Community Structure in Very Large Networks," *Physical Review E*, 70, 066111. [490]
- Danon, L., Diaz-Guilera, A., Duch, J., and Arenas, A. (2005), "Comparing Community Structure Identification," *Journal of Statistical Mechanics: Theory and Experiment*, 2005, P09008. [494]
- Fortunato, S. (2010), "Community Detection in Graphs," *Physics Reports*, 486, 75–174. [487,489,490,499]
- Guimera, R., Sales-Pardo, M., and Amaral, L. A. N. (2004), "Modularity from Fluctuations in Random Graphs and Complex Networks," *Physical Review E*, 70, 025101. [490]
- Jiang, S., Koch, B., and Sun, Y. (2021), "Hints: Citation Time Sries Prediction for New Publications via Dynamic Heterogeneous Information Network Embedding," in *Proceedings of the Web Conference 2021*, pp. 3158–3167. [487]
- Linden, G., Smith, B., and York, J. (2003), "Amazon. com Recommendations: Item-to-Item Collaborative Filtering," *IEEE Internet Computing*, 7, 76–80. [487]
- Massen, C. P., and Doye, J. P. (2005), "Identifying Communities Within Energy Landscapes," *Physical Review E*, 71, 046101. [490,499]
- Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2002), "Network Motifs: Simple Building Blocks of Complex Networks," *Science*, 298, 824–827. [489]
- Moody, J., and White, D. R. (2003), "Structural Cohesion and Embeddedness: A Hierarchical Concept of Social Groups," *American Sociological Review*, 68, 103–127. [487]
- Newman, M. E. (2006), "Finding Community Structure in Networks using the Eigenvectors of Matrices," *Physical Review E*, 74, 036104. [489]
- Newman, M. E., and Girvan, M. (2004), "Finding and Evaluating Community Structure in Networks," *Physical Review E*, 69, 026113. [489]
- Newman, M. E., Strogatz, S. H., and Watts, D. J. (2001), "Random Graphs with Arbitrary Degree Distributions and their Applications," *Physical Review E*, 64, 026118. [489]
- Sengupta, S., and Chen, Y. (2015), "Spectral Clustering in Heterogeneous Networks," *Statistica Sinica*, 25, 1081–1106. [487]
- Sorlie, T., Perou, C. M., Tibshirani, R., Aas, T., Geisler, S., Johnsen, H., Hastie, T., Eisen, M. B., Van De Rijn, M., Jeffrey, S. S., et al. (2001), "Gene Expression Patterns of Breast Carcinomas Distinguish Tumor Subclasses with Clinical Implications," *Proceedings of the National Academy of Sciences*, 98, 10869–10874. [487]
- Sun, Y., Tang, J., Han, J., Gupta, M., and Zhao, B. (2010), "Community Evolution Detection in Dynamic Heterogeneous Information Networks," in *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*, pp. 137–146. [487]

- Wakita, K., and Tsurumi, T. (2007), “Finding Community Structure in Mega-Scale Social Networks,” in *Proceedings of the 16th International Conference on World Wide Web*, pp. 1275–1276. [490]
- Wang, X., Lu, Y., Shi, C., Wang, R., Cui, P., and Mou, S. (2022), “Dynamic Heterogeneous Information Network Embedding with Meta-Path based Proximity,” *IEEE Transactions on Knowledge and Data Engineering*, 34, 1117 – 1132. [487]
- Xue, H., Yang, L., Jiang, W., Wei, Y., Hu, Y., and Lin, Y. (2020), “Modeling Dynamic Heterogeneous Network for Link Prediction Using Hierarchical Attention with Temporal RNN,” arXiv preprint arXiv:2004.01024. [487]
- Yin, Y., Ji, L.-X., Zhang, J.-P., and Pei, Y.-L. (2019), “Dhne: Network Representation Learning Method for Dynamic Heterogeneous Networks,” *IEEE Access*, 7, 134782–134792. [487]
- Zhang, J., and Cao, J. (2017), “Finding Common Modules in a Time-Varying Network with Application to the *Drosophila Melanogaster* Gene Regulation Network,” *Journal of the American Statistical Association*, 112, 994–1008. [487,488,493,494]
- Zhang, J., and Chen, Y. (2017), “A Hypothesis Testing Framework for Modularity based Network Community Detection,” *Statistica Sinica*, 27, 437–456. [489]
- Zhang, J., and Chen, Y. (2020), “Modularity based Community Detection in Heterogeneous Networks,” *Statistica Sinica*, 30, 601–629. [493,494,499]
- Zhang, J., Sun, W. W., and Li, L. (2020), “Mixed-Effect Time-Varying Network Model and Application in Brain Connectivity Analysis,” *Journal of the American Statistical Association*, 115, 2022–2036. [488]
- Zhang, Z., Huang, J., and Tan, Q. (2022), “Multi-View Dynamic Heterogeneous Information Network Embedding,” *The Computer Journal*, 65, 2016–2033. [487]