1. Introduction

Community detection has attracted increasing attention from both academia and industry in the past few decades, and has a wide spectrum of applications in domains, ranging from social science (Newman, Watts, and Strogatz 2002; Zhao, Levina, and Zhu 2011; Ji et al. 2016; Lee, Magallanes, and Porter 2017), life science (Chen and Yuan 2006; Nepusz, Yu, and Paccanaro 2012) to computer science (Agarwal et al. 2005; Tron and Vidal 2007). The community structure has been widely observed in many real-life networks, which usually means that entities within the same community tend to interact much more often than across communities. In the literature, most conventional network analysis focus on pairwise interactions between two vertices (Zhao et al. 2012; Lei et al. 2015; Sengupta and Chen 2018; Loyal and Chen 2020). However, the complexity of real-life networks is usually beyond pairwise interaction, and multi-way interactions among vertices arise naturally. For example, in an academic collaboration network, it is often the case that several researchers work together on a research project; in a protein-to-protein interaction network, a metabolic reaction usually involves multiple proteins. Under such circumstances, a hypergraph network provides a more faithful representation and retains richer information than a merely vanilla graph network, such as each research project being represented by a hyperedge consisting of multiple vertices being researchers, or each metabolic reaction being a hyperedge consisting of multiple vertices being proteins. In this article, we are interested in detecting community structure in a hypergraph network, where vertices within the same community share more similar connection patterns compared with vertices in different communities.

To detect communities in a hypergraph network, most existing methods convert the hypergraph into a weighted graph, and then existing graph community detection methods can be applied. For instance, Kumar et al. (2021) defined the hypergraph modularity as the modularity of the weighted graph and applied standard modularity maximization algorithms for community detection; Lee, Kim, and Chung (2021) extended the graph-likelihood-based convex relaxation methods (Li, Chen, and Xu 2021) on the adjacency matrix of the weighted graph; and Ghoshdastidar and Dukkipati (2017a) conducted spectral clustering on the weighted graph Laplacian for hypergraph community detection. As pointed out in Ke, Shi, and Xia (2021), such conversion will suffer from information loss and lead to suboptimal community detection performance. To circumvent this disadvantage, spiked tensor model (Kim, Bandeira, and Goemans 2017) and Tensor-SCORE method (Ke, Shi, and Xia 2021) were proposed to conduct community detection on the hypergraph adjacency tensor directly.

Note that most existing hypergraph community detection methods focus on uniform hypergraph only (Ghoshdastidar and Dukkipati 2014, 2015b; Lee, Kim, and Chung 2021), where all the hyperedges consist of exactly the same number of vertices. When it comes to nonuniform hypergraph, they have to decompose a nonuniform hypergraph into a collection of uniform hypergraphs with different orders, due to the difficulty of lacking an appropriate adjacency tensor for nonuniform hypergraph. It was only until recently that Ouvrard, Goff, and Marchand-Maillet (2021) proposed a heuristic method to construct an adjacency tensor for general hypergraph by adding $m - m_0$ null vertices, where $m$ and $m_0$ are the maximum and minimum cardinality of a hyperedge in the hypergraph, respectively.

In this article, we propose a novel method for detecting community structure in general hypergraph networks, where a general hypergraph can be uniform or nonuniform. The proposed method consists of an augmentation step and an embedding step. The augmentation step adds a null vertex to the hyperedges...
with smaller cardinality, and converts the hypergraph into a uniform multi-hypergraph (Pearson and Zhang 2014, 2015), which allows vertices to appear multiple times in one hyperedge. The embedding step is formulated in a regularization form with tensor decomposition, which represents each vertex as a low-dimensional numerical vector and encourages vertices within the same community to be close in the embedding space. The proposed hypergraph embedding model is flexible and general, which includes the hypergraph stochastic block model (hSBM), also known as hypergraph planted partition model (Ghoshdastidar and Dukkipati 2017a), as its special case. It also accommodates heterogeneity among vertices by allowing vertices within a community to fluctuate in all directions. This is in sharp contrast to hSBM, which assumes vertices within a community share the same spectral embedding vector. It is also more general than the hypergraph degree-corrected block model (hDCBM; Ke, Shi, and Xia 2021; Yuan et al. 2021), under which spectral embeddings of vertices within a community are along the same direction and only differ by their magnitudes. The advantage of the proposed method is supported in both asymptotic theories and numerical experiments on a number of synthetic and real hypergraph networks.

The main contribution of this article is three-fold. First, we propose a novel hypergraph embedding model (HEM), which consists of an augmentation step and an embedding step. The augmentation step introduces only one null vertex rather than multiple ones as suggested in Ouvrard, Goff, and Marchand-Maillet (2021), which is more efficient and greatly facilitates the subsequent community detection. The embedding step, to the best of our knowledge, is the first statistical framework that extends the latent space model for graph network to general hypergraph network with theoretical guarantees. Second, a joint modeling framework is developed for simultaneously conducting hypergraph estimation and community detection. Third, we establish the asymptotic consistencies of the proposed method in terms of both hypergraph estimation and community detection in sparse hypergraph networks. Particularly, the consistencies hold as long as the link probability is of the order $s_n \gg n^{1-\epsilon} \log n$ in an $m$th-order hypergraph with $n$ vertices. This result compares favorably with the existing sparsity results in literature (Ke, Shi, and Xia 2021, Ghoshdastidar and Dukkipati 2017a, 2017b), not to mention that the proposed method also achieves fast convergence rate in hypergraph estimation.

The rest of the article is organized as follows. Section 2 provides a quick review of some preliminaries on hypergraph and tensor. Section 3 presents the details of the proposed hypergraph augmentation and embedding formulations, and an efficient optimization algorithm. Section 4 establishes the asymptotic consistencies of the proposed method. Section 5 examines the numerical performance of the proposed method on both synthetic and real-life hypergraph networks. Section 6 concludes the paper, and technical proofs and necessary lemmas are contained in the supplementary files.

2. Preliminaries

To begin with, we introduce some basic concepts of hypergraph and tensor that will be used extensively in the sequel. A general hypergraph is denoted as $\mathcal{H}(V, E)$, where $V = \{v_1, v_2, ..., v_n\}$ is a vertex set with $n$ vertices, $E$ consists of all hyperedges, and each hyperedge may contain multiple vertices in $V$. Denote $[n] = \{1, 2, ..., n\}$, and we set $V = [n]$ for simplicity, and thus a hyperedge is a non-empty subset of $[n]$. A hypergraph is called $m$-uniform if the cardinality of every hyperedge equals $m$. On the flip side, a nonuniform hypergraph contains hyperedges whose cardinalities can vary from one to another. A simple nonuniform hypergraph with 5 vertices and 5 hyperedges is displayed in the left panel of Figure 1, where hyperedges $e_1$ and $e_2$ consist of 3 vertices, hyperedges $e_3$ and $e_4$ consist of 2 vertices, and hyperedge $e_5$ consists of 4 vertices. Note that the hypergraphs considered in this paper are undirected; that is, no ordering of vertices is necessary within the hyperedges. For definition of directed hypergraph, interested readers may refer to Gallo et al. (1993) and Ouvrard, Goff, and Marchand-Maillet (2021).

A tensor $\mathcal{A} = (a_{i_1...i_m}) \in \mathbb{R}^{I_1 \times \cdots \times I_m}$ is a cubical tensor if $I_1 = \cdots = I_m = n$, for some $n \in \mathbb{Z}^+$. More formally, such a tensor is called an $m$th-order tensor of dimension $n$. A cubical tensor $\mathcal{A}$ is super symmetric if $a_{i_1...i_m} = a_{\pi(i_1)...\pi(i_m)}$, for any possible permutations $\pi \in S_n$, the symmetric group of degree $n$. An $m$th-order super symmetric tensor $\mathcal{I} \in \{0, 1\}^{n \times \cdots \times n}$ is called an identity tensor if $I_{i_1...i_m} = 1$ when $i_1 = \cdots = i_m$ and 0 otherwise.

Let $M^{(0)} \in \mathbb{R}^{Q_1 \times \cdots \times Q_m}$ be matrices, then the mode $j$ product (Kolda and Bader 2009) between $\mathcal{A}$ and $M^{(0)}$ is defined as $\mathcal{A} \times_j M^{(0)} \in \mathbb{R}^{I_1 \times \cdots \times I_{j-1} \times Q_j \times I_{j+1} \times \cdots \times I_m}$ with entries

$$
(\mathcal{A} \times_j M^{(0)})_{i_1...i_{j-1}q_ji_{j+1}...i_m} = \sum_{i_j=1}^{I_j} a_{i_1...i_{j-1}i_j...i_m} M^{(0)}_{q_j i_j}.
$$

Moreover, $\mathcal{A} \times_1 M^{(1)} \times_2 \cdots \times_m M^{(m)} \in \mathbb{R}^{Q_1 \times \cdots \times Q_m}$ is such a tensor with entries

$$
(\mathcal{A} \times_1 M^{(1)} \times_2 \cdots \times_m M^{(m)})_{q_1...q_m} = \sum_{i_1=1}^{I_1} \cdots \sum_{i_m=1}^{I_m} a_{i_1...i_m} M^{(1)}_{q_1 i_1} \cdots M^{(m)}_{q_m i_m}.
$$

Recall that a multi-set $\{(i_1, i_2, ..., i_m)\}$ is an extension of set in that some elements may appear multiple times. Herein, to distinguish from a set, we use double curly braces $\{\} \}$ to denote a multi-set as in Bahmanian and Sajna (2015) and Kovačević and Tan (2018). Clearly, if the elements $i_1, \ldots, i_m$ are distinct with one another, then the multi-set $\{(i_1, \ldots, i_m)\}$ reduces to the set $\{i_1, \ldots, i_m\}$. For any $i_1, \ldots, i_m \in [n+1]$, we define an augmented Kronecker delta to be $\delta_{i_1...i_m}^{n+1} = 0$ if $\{i_1, \ldots, i_m\} \setminus \{n+1\}$ is indeed a non-empty set and 1 otherwise, where the multi-set difference with a set $\{i_1, \ldots, i_m\} \setminus \{n+1\}$ is a multi-set containing elements in the multi-set $\{i_1, \ldots, i_m\}$ but not in the set $\{n+1\}$. For example, if $n = 100$ and $m = 3$, we have $\delta_{1,2,101}^{101} = 0$, $\delta_{101,101,101}^{101} = 1$ and $\delta_{1,1,2}^{101} = 1$. Furthermore, the order-specified augmented Kronecker delta $\delta_{i_1...i_m}^{n+1,ord}$ is $0$ if and only if $\delta_{i_1...i_m}^{n+1} = 0$ and $i_1 \leq \cdots \leq i_m$. These two extended Kronecker deltas can greatly simplify notations and formulations for nonuniform hypergraph, and they are slightly different from the vanilla extension in Qi (2005), where the extended Kronecker delta equals 0 if there exist at least 2 distinct elements.
3. Community Detection in Hypergraph

This section proposes a novel community detection method for general hypergraphs. The key idea is to introduce a null vertex so that hyperedges with smaller cardinality can be augmented, and the nonuniform hypergraph is transformed into a special uniform multi-hypergraph, where the null vertex may appear multiple times in some hyperedges. Then the proposed method proceeds to seek for a suitable numeric embedding that embeds all vertices of the hypergraph into a low dimensional Euclidean space, where vertices belong to the same community tend to have shorter distance.

3.1. Hypergraph Augmentation

Let \( H(V, E) \) be a hypergraph, possibly nonuniform, with \( V = [n] \) and range \( m \geq 2 \), which is the maximum cardinality of a hyperedge in \( E \). One of the key challenges of analyzing nonuniform hypergraph is due to the unequal hyperedge cardinalities, leading to the ambiguity of discriminating hyperedges and their proper subsets.

To circumvent this difficulty, we introduce a null vertex, denoted as \( v_{n+1} \), and then any hyperedge with cardinality less than \( m \) can be augmented to a multi-set with \( m \) elements. For instance, when \( m > 3 \), a hyperedge with vertices \( \{1, 2, 3\} \) can be augmented to \( \{1, 2, 3, n+1, \ldots, n+1\} \) with \( m-3 \) null vertices, whereas another hyperedge with vertices \( \{1, \ldots, m\} \) stays the same without any additional null vertex. It is clear that \( H \) is converted to an equivalent \( m \)-uniform multi-hypergraph, where each hyperedge is a multi-set with cardinality \( m \) and only the null vertex \( v_{n+1} \) is allowed to appear multiple times in a hyperedge. With slight abuse of notation, we still denote this \( m \)-uniform multi-hypergraph as \( H \). A simple illustration of this augmentation step is displayed in the right panel of Figure 1.

We remark that when \( H \) is a uniform hypergraph, the augmentation step and the subsequent special treatments of the null vertex are not necessary. For ease of presentation, we assume \( H \) is a nonuniform hypergraph hereafter, and the upcoming proposed model can be adapted to uniform hypergraph with slight modification. As such, we use the \( m \)-order adjacency tensor \( A = (a_{i_1 \ldots i_m}) \in \{0, 1\}^{(n+1) \times \cdots \times (n+1)} \) to represent \( H \) with entries

\[
a_{i_1 \ldots i_m} = \begin{cases} 1 & \text{if } \{i_1, \ldots, i_m\} \in \{n+1\} \in E, \\ 0 & \text{otherwise}. \end{cases}
\]

Apparently, a necessary condition for \( a_{i_1 \ldots i_m} = 1 \) is \( \delta_{i_1 \ldots i_m}^{n+1} = 0 \). It is clear that this augmentation step, converting a nonuniform hypergraph to a uniform multi-hypergraph, is critical to make downstream analyses more statistically and computationally tractable.

Suppose there are \( K \) potential communities within the hypergraph, and let \( \psi : [n] \rightarrow [K] \) be the community assignment function, then \( \psi(i) = s \) if vertex \( v_i \) belongs to the \( s \)th community. We also write \( \psi_i = \psi(i) \) for short. Note that the null vertex \( v_{n+1} \) does not belong to any community, and thus we set \( \psi_{n+1} = K + 1 \) for formality. Denote \( Z = (z_{ij}) \in \mathbb{R}^{(n+1) \times (K+1)} \) to be the corresponding community membership matrix with \( z_{ij} = 1 \) if \( j = \psi_i \) and 0 otherwise. It is obvious that \( z_{i,K+1} = 1 \) only when \( i = n+1 \).

3.2. Hypergraph Embedding

Given the \( m \)-uniform multi-hypergraph \( H \) from the augmentation step, we let \( \mathcal{P} = (p_{i_1 \ldots i_m}) \in (0,1)^{(n+1) \times \cdots \times (n+1)} \) be the \( m \)-order probability tensor, with \( p_{i_1 \ldots i_m} = P(a_{i_1 \ldots i_m} = 1) \) if \( \delta_{i_1 \ldots i_m}^{n+1} = 0 \), and let \( \Theta = (\theta_{i_1 \ldots i_m}) \in \mathbb{R}^{(n+1) \times \cdots \times (n+1)} \) be the entrywise transformation of \( \mathcal{P} \) so that

\[
\theta_{i_1 \ldots i_m} = \log \left( \frac{p_{i_1 \ldots i_m}}{s_n - p_{i_1 \ldots i_m}} \right),
\]

where the sparsity factor \( s_n \) that may vanish with \( n \) is introduced to accommodate sparse networks. The modified logit transformation (1) implies that \( p_{i_1 \ldots i_m} = s_n(1 + e^{-\theta_{i_1 \ldots i_m}})^{-1} \), and the number of hyperedges is of order \( O_{p}(s_n n^m) \) if \( \Theta \) lies in a compact subset of \( \mathbb{R}^{(n+1) \times \cdots \times (n+1)} \). We remark that many frequentist analysis of network models multiplies the network underlying linking probability by a decaying sparsity coefficient to accommodate sparse networks, such as the hypergraph stochastic block model (Ghoshdastidar and Dukkipati 2017a, 2017b) and hypergraph degree-corrected block model (Ke, Shi, ...
and Xia 2021). Yet, this treatment may not be directly applied to latent space model with standard logit transformation, since tiny value of \( p_{i_1 ... i_m} \) implies that \( \theta_{i_1 ... i_m} \) will be pushed toward \(-\infty\), and thus the entries of \( \alpha \) may diverge to \( \pm \infty \), leading to unstable numerical performance in estimating \( \alpha \).

We now turn to embed the hypergraph into an \( r \)-dimensional Euclidean space with \( 2 \leq r \ll n \), where each vertex is represented by an \( r \)-dimensional vector \( \alpha_i \), for any \( i \in [n] \). The embedding dimension \( r \) here is allowed to diverge with \( n \). For the null vertex, we simply set \( \alpha_{n+1} = r^{-1/2} 1r \), the \( r \) dimensional vector with every entry \( r^{-1/2} \). Let \( \alpha = (\alpha_1, ..., \alpha_{n+1})^T \) be the embedding matrix. We consider the following hypergraph embedding model (HEM):

\[
\Theta = (1)\times_1 \alpha \times_2 \cdots \times_m \alpha,
\]

where \( (1) \in \mathbb{R}^{r \times \cdots \times r} \) is the \( m \)-th order identity tensor of dimension \( r \). Clearly, for any \( i_1, ..., i_m \in [n+1] \), model (2) assumes that the information contained in \( \theta_{i_1 ... i_m} \) can be fully captured by the embedding matrix \( \alpha \) in that \( \hat{\theta}_{i_1 ... i_m} = (1)\times_1 \alpha_{i_1} \times_2 \cdots \times_m \alpha_i \). When \( \hat{\theta}_{i_1 ... i_m} = 0 \), the probability that \( \{|i_1, ..., i_m| \} \setminus \{n+1\} \) forms a hyperedge is \( s_{\theta}(1 + e^{-\hat{\theta}_{i_1 ... i_m}})^{-1} \), whereas \( \hat{\theta}_{i_1 ... i_m} \) or \( \hat{p}_{i_1 ... i_m} \) defined in (2) does not have any probability interpretation when \( \hat{\theta}_{i_1 ... i_m} = 1 \), and is thus inconsequential.

The proposed HEM is flexible and general, and includes the celebrated hSBM (Ghoshdastidar and Dukkipati 2017a) as its special case. Particularly, taking \( \alpha = ZC \), HEM reduces to \( \Theta = B \otimes_{1 \times 1} X \otimes_{2 \times \cdots \times m} Z \) with \( B = (1) \times_1 X \otimes_{2 \times \cdots \times m} C \), which becomes an hSBM with membership matrix \( Z \in \{0, 1\}^{(n+1) \times (K+1)} \) and a transformed core probability tensor \( B \in \mathbb{R}^{(K+1) \times \cdots \times (K+1)} \).

Yet, HEM is more flexible as it naturally accommodates heterogeneity among vertices by allowing the embeddings or spectral embeddings of vertex within a community to fluctuate in all directions. It is worth pointing out that such accommodation of heterogeneity is more general than that in hDCBM (Ke, Shi, and Xia 2021; Yuan et al. 2021), which requires spectral embeddings of vertices within a community to lie along the same direction and only differ by magnitudes.

It is also interesting to remark that HEM is equivalent to assuming that the super symmetry tensor \( \Theta \) has a symmetric CP decomposition (Kolda and Bader 2009), \( \Theta = \sum_{j=1}^r \alpha_j \otimes_{1} \cdots \otimes_{m} \alpha_j \),

where \( \alpha_j \) is the \( j \)-th column of \( \alpha \) and \( \otimes \) stands for the vector outer product. Furthermore, we define the symmetric rank of an \( m \)-th order tensor \( \Theta \) of dimension \( n+1 \) over \( \mathbb{R} \) as rank(\( \Theta \)) = min\{rank(\( R \)) = \sum_{j=1}^r \| u_r \|_2 \otimes_{1} \cdots \otimes_{m} u_r \in \mathbb{R}^{n+1} \}. Therefore, it is assumed that \( \Theta \) has symmetric rank at most \( r \). We also remark our definition of symmetric rank is slightly different from the one that defined in literature (Kolda and Bader 2009; Kobeva 2016) by requiring the embedding vectors to be real.

More importantly, HEM is identifiable if \( \Theta \) has symmetric rank \( r \) but \( m \geq 3 \), as the factorization (2) is unique up to column permutations of \( \alpha \) (Sidiroopoulos and Bro 2000). When \( \Theta \) has symmetric rank \( r \) but \( m = 2 \), HEM reduces to the latent space model for graph networks (Hoff, Raftery, and Handcock 2002) and the factorization (2) is unique up to any orthogonal transformation of \( \alpha \). Note that column permutations or general orthogonal transformations are isometric linear transformation of the rows of \( \alpha \), the community structure encoded in \( \alpha \) always remains unchanged.

### 3.3. Penalized Log-Likelihood Objective

With the factorization of \( \Theta \) in (2), the negative log-likelihood function of \( \mathcal{H} \) becomes

\[
L(\alpha; \mathcal{A}) = \frac{1}{\psi(n, m)} \sum_{\delta_{i_1 ... i_m}^{n+1} = 0} L(\theta_{i_1 ... i_m}; a_{i_1 ... i_m}),
\]

where \( \psi(n, m) = \sum_{k=1}^m \binom{n}{k} \) is the number of potential hyperedges with \( \delta_{i_1 ... i_m}^{n+1} = 0 \), \( \theta_{i_1 ... i_m} = (1)\times_1 \alpha_{i_1} \times_2 \cdots \times_m \alpha_i \), and

\[
L(\theta_{i_1 ... i_m}; a_{i_1 ... i_m}) = \log \left( \frac{1}{1 - s_n e^{-\theta_{i_1 ... i_m}}} + \frac{s_n}{1 - s_n e^{-\theta_{i_1 ... i_m}}} \right).
\]

We next equip \( L(\alpha; \mathcal{A}) \) with a novel penalty term to enhance the feasibility of computation and hypergraph community detection. This leads to the proposed regularized cost function,

\[
L_{\lambda}(\alpha; \mathcal{A}) = L(\alpha; \mathcal{A}) + \lambda_n ||\alpha - ZC||^2_F,
\]

where \( \lambda_n \) is a positive tuning parameter and

\[
J(\alpha) = \min_{Z \in \Gamma, C \in \mathbb{R}^{(n+1) \times K+1}} ||\alpha - ZC||^2_F, C_{K+1} = r^{-1/2} 1r,
\]

is introduced to encourage the community structure encoded in \( \alpha \). Herein, \( \Gamma \) is the set of all possible community membership matrix. That is, for any \( Z \in \Gamma \subset \mathbb{R}^{(n+1) \times (K+1)} \), each row of \( Z \) contains exactly one 1 with all other entries being zeros, and \( Z_{i(K+1)} = 1 \) if and only if \( i = n + 1 \). It is clear that the embeddings of vertices with similar linking patterns will be pushed toward the same center, and thus close to each other in the embedding space, leading to the desired community structure in \( \mathcal{H} \). A similar regularization term has been employed in Tang, Xue, and Qu (2021) for individualized variable selection and in Zhang, He, and Wang (2021) for community detection in directed networks. In the sequel, with \( \Theta = (1)\times_1 \alpha \times_2 \cdots \times_m \alpha \), we use \( L_{\lambda}(\Theta; \mathcal{A}) \) and \( L_{\lambda}(\alpha; \mathcal{A}) \) interchangeably, as convenience dictates.

We develop an alternative updating scheme to minimize (3). Particularly, given \( Z(t) \) and \( C(t) \) at step \( t \), \( \alpha \) can be updated by solving

\[
\min_{\alpha} \frac{1}{\psi(n, m)} \sum_{\delta_{i_1 ... i_m}^{n+1} = 0} L(\theta_{i_1 ... i_m}; a_{i_1 ... i_m}) + \frac{\lambda_n}{n} ||\alpha - Z(t)C(t)||^2_F.
\]

Denote \( T = \frac{\partial L(\Theta; \mathcal{A})}{\partial \alpha} \), and let \( \Delta = \{0, 1\}^{(n+1) \times \cdots \times (n+1)} \) such that \( \Delta_{i_1 ... i_m} = 1 \) if \( \delta_{i_1 ... i_m}^{n+1} = 0 \) with \( \delta_{i_1 ... i_m}^{n+1, ord(n+1)} = 0 \) or \( \delta_{i_1 ... i_m}^{n+1} = 0 \), and 0 otherwise. We then update the first \( n \) rows of \( \alpha \), which is denoted as \( \alpha_{1, n} \), along its gradient,
\[ \alpha^{(t+1)} = \alpha^{(t)} - \eta_t \nabla_{\alpha^{(t)}} L_{\lambda}^{(i)}(\alpha^{(t)}) , \]
where \( \eta_t > 0 \) is the learning rate at step \( t + 1 \), and
\[
\nabla_{\alpha^{(t)}} L_{\lambda}^{(i)}(\alpha^{(t)}) = \frac{1}{\psi(n,m)} \langle T \star \Delta, I \times 2 \alpha \times 3 \cdots \times m \alpha \rangle_{1 \times n} \cdots \times m + \frac{2\Lambda_{nr}}{n} (\alpha - Z^{(i)})_{1 \times n}.
\]
Herein, \( \star \) is the Hadamard product (entry-wise product) between two tensors, and \( (T \star \Delta, I \times 2 \alpha \times 3 \cdots \times m \alpha)_{[2, \cdots, m]} \) is the tensor inner product over \( (T \star \Delta) \) and \( I \times 2 \alpha \times 3 \cdots \times m \alpha \) with respect to the second, \( \cdots, m \)-th modes. Specifically, the \((i, j)\)-th entry of \((T \star \Delta, I \times 2 \alpha \times 3 \cdots \times m \alpha)_{[2, \cdots, m]} \) is \( \sum_{i_2, \cdots, i_m} (T \star \Delta)_{i_2 \cdots i_m} (I \times 2 \alpha \times 3 \cdots \times m \alpha)_{j i_2 \cdots i_m} \).

Next, given \( \alpha^{(t+1)} \), the sub-optimization task now becomes
\[
\min_{Z \in \mathbb{R}, C \in \mathbb{R}^{1 \times 1 \times m}} \frac{1}{Z} ||\alpha^{(t+1)} - ZC||^2_{\psi}, \text{ subject to } C_{K+1} = r^{-1/2}I.
\]

Clearly, it resembles the K-means formulation for \( \alpha^{(t+1)} \), and thus a standard K-means algorithm can be employed to solve for \( Z \) and \( C \).

As computational remarks, the alternative updating algorithm is guaranteed to converge to a stationary point, and its computational complexity is of order \( O(k_2^n r + k_1 K m r) \), where \( k_1 \) is the number of iterations for the K-means algorithm and \( k_2 \) is the number of iterations for the gradient descent step. Note that low rank tensor approximation tends to be computationally expensive and easy to get trapped in noninformative local minima (Arous et al. 2019). Based on our limited numerical experience, a warm initialization of \( \alpha^{(0)} \) can greatly help with the numerical convergence. In all the numerical examples, we initialize \( \alpha^{(0)} \) with a higher-order singular value decomposition algorithm (HOSVD; De Lathauwer, De Moor, and Vandewalle 2000). We also suggest to set the embedding dimension \( r = K \) in practice, similar suggestions were also made for some spectral-clustering-based algorithms (Ghoshdastidar and Dukkipati 2017a). When \( K \) is unknown, we can follow the procedure in Ke, Shi, and Xia (2021) to investigate the "eigen-gap" of the network adjacency tensor. Specifically, one can first obtain a spectral embedding \( \hat{U} \) with sufficiently large dimension, and then investigate the eigen-gap of the matrix \( M_{\hat{U}}(A \times 3 \hat{U} \times 4 \cdots \times m \hat{U}) \). Other data adaptive selection criteria, such as network cross-validation (Chen and Lei 2018; Li, Levina, and Zhu 2020) may be employed as well, at the cost of increased computational burden.

4. Asymptotic Theory

This section establishes some theoretical results to quantify the asymptotic behavior of the proposed HEM method in estimating the underlying transformed probability tensor \( \Theta \) as well as detecting community structure in a general hypergraph.

4.1. Consistency in estimating \( \Theta^* \)

Let \( \Omega = \{ \Theta : \Theta = I \times 1 \alpha \times 2 \cdots \times m \alpha, \max_{i \in [n]} |\alpha_i| \leq c_0, \alpha_{n+1} = r^{-1/2}I \} \subset [1 \times (r+1)^{1 \times (n+1)}] \) be the domain of the problem, for a positive constant \( c_0 \geq 1 \). It is clear that \( \Omega \) is a compact subset of \( [1 \times (r+1)^{1 \times (n+1)}] \), and for any \( \Theta \in \Omega \), it has symmetric rank at most \( r \). Denote \( \alpha^* \) as the underlying true hypergraph embedding with \( \Theta^* = I \times 1 \alpha^* \times 2 \cdots \times m \alpha^* \in \Omega \). In addition, for any \( \Theta \in \Omega \), we define \( c_l(\Theta, \Theta^*) = \psi^{-1}(n, m) \sum_{i_1, \cdots, i_m} E(L(\theta_{i_1, \cdots, i_m}; a_{i_1, \cdots, i_m} - L(\theta^*_{i_1, \cdots, i_m}; a_{i_1, \cdots, i_m})) \), which is the average of certain Kullback-Leibler divergences and guaranteed to be nonnegative.

The following large deviation inequality is derived to quantify the asymptotic behavior of \( \hat{L}_{\lambda}(\Theta; A) \) in the neighborhood of \( \Theta^* \) defined by \( c_l(\Theta, \Theta^*) \).

**Proposition 1.** Suppose \( \lambda_n(\alpha^*) < \frac{1}{2} \epsilon_n \), then there exists some absolute constants \( c_1, c_2 \) such that \( n \epsilon_n^{-1} (m) \epsilon_n^{-1} \log \epsilon_n^{-1} \leq c_1 \), we have
\[
P \left( \sup_{\Theta \in \Omega |\Theta(\Theta^*; A)} - \hat{L}_{\lambda}(\Theta; A) \geq 0 \right) \leq 2 \exp \left(-c_2 \psi(n, m) \epsilon_n \right).
\]

**Proposition 1** gives an intermediate result for establishing estimation consistency in **Theorem 1**, and it assures that any \( \Theta \) such that \( \hat{L}_{\lambda}(\Theta; A) \leq \hat{L}_{\lambda}(\Theta^*; A) \) shall lie in the neighborhood of \( \Theta^* \) with high probability. By the condition that \( n \epsilon_n^{-1} (m) \epsilon_n^{-1} \log \epsilon_n^{-1} \leq c_1 \), the fastest order of \( \epsilon_n \) can be set as \( \epsilon_n = \frac{\epsilon_n}{\psi(n, m)} \), which is governed by the network size \( \psi(n, m) \) and the number of parameters \( nr \) in HEM. It is interesting to remark that \( \lambda_n \) appears to have no effect on the order of \( \epsilon_n \), as long as it satisfies the condition \( \lambda_n(\alpha^*) < \frac{1}{2} \epsilon_n \).

We are now ready to establish the consistency of \( \hat{\Theta} = I \times 1 \hat{\alpha} \times 2 \cdots \times m \hat{\alpha} \), with \( \hat{\alpha} \) being the estimate from **Section 3.3**. Let \( p(\gamma; \theta) \) be the density of a Bernoulli random variable with parameter \( p = s_n(1 + \exp(-\theta))^{-1} \). Then the discrete Hellinger distance between \( p(\gamma; \theta) \) and \( p(\gamma; \theta^*) \) is defined as
\[
d(\theta, \theta^*) = \left( \frac{p^2(1 - p^2)^2}{1 - (1 - p)^2 - (1 - p^2)^2} \right)^{1/2},
\]
and the deviation between \( \Theta \) from \( \Theta^* \) can be evaluated by the averaged squared Hellinger distance,
\[
D^2(\Theta, \Theta^*) = \frac{1}{\psi(n, m)} \sum_{i_1, \cdots, i_m} d^2(\theta_{i_1, \cdots, i_m}; \theta^*_{i_1, \cdots, i_m}).
\]

**Theorem 1.** Under the assumptions in **Proposition 1**, for any \( \hat{\Theta} \) with \( \hat{L}_{\lambda}(\hat{\Theta}; A) \leq \hat{L}_{\lambda}(\alpha^*; A) \), we have
\[
P(D^2(\hat{\Theta}, \Theta^*) \geq \epsilon_n) \leq 2 \exp \left(-c_2 \psi(n, m) \epsilon_n \right).
\]

Moreover, \( D^2(\hat{\Theta}, \Theta^*) = O_p(\epsilon_n) \) and \( n^{-m/2} ||\hat{\Theta} - \Theta^*||_F = O_p(\sqrt{\epsilon_n / s_n}) \).

**Theorem 1** shows that a reasonably good solution \( \hat{\Theta} \) is guaranteed to converge to \( \Theta^* \) at a fast rate, which depends on the centrality of the community structure encoded in \( \alpha^* \), the network size and number of parameters via \( \epsilon_n \), and the network sparsity factor \( s_n \). The condition \( \hat{L}_{\lambda}(\hat{\Theta}; A) \leq \hat{L}_{\lambda}(\alpha^*; A) \) shall be satisfied by the solution obtained in **Section 3.3**, when the estimation algorithm is initialized by some value in a small neighborhood of \( \Theta^* \). The consistency result in **Theorem 1** holds true with \( O(m^{-1} n \log n) \leq \epsilon_n \ll s_n \). If we further set \( r < \log n \), this yields a slightly weaker sparsity assumption than that
in Ke, Shi, and Xia (2021) and Ghoshdastidar and Dukkipati (2017a, 2017b), where the smallest sparsity factor is of the order \(n^{-m/2} \log n \gamma_n\) for some fixed \(K\).

### 4.2. Consistency in Community Detection

We now turn to establish the consistency of community detection for general hypergraphs. Let \(\psi^* : [n] \rightarrow [K]\) be the true community assignment function, and \(\hat{\psi}\) be the estimation counterpart induced by \(\hat{\alpha}\). Formally, \(\hat{\psi} = \arg\min_{k \in [K]} ||\alpha_i - \hat{C}_k||_2\), where \(\hat{C} = \arg\min_C \min_{Z \in F/\alpha \rightarrow ZC)||_F\), subject to \(C_{K+1} = r^{-1/2}1_r\). The community detection error of \(\hat{\psi}\) can be evaluated by the minimum scaled Hamming distance between \(\hat{\psi}\) and \(\psi^*\) under permutations, which is defined as

\[
\text{err}(\psi^*, \hat{\psi}) = \min_{\pi \in S_K} \frac{1}{n} \sum_{i=1}^{n} I\{\psi^*_i \neq \pi(\hat{\psi}_i)\}, \tag{6}
\]

where \(I\{\cdot\}\) is the indicator function and \(S_K\) is the symmetric group of degree \(K\). Clearly, it measures the minimum fraction of vertices that are mismatched by \(\hat{\psi}\) under permutation, and \(\hat{\psi}\) is a consistent estimator of \(\psi^*\) if \(\text{err}(\psi^*, \hat{\psi})\) goes to zero with probability tending to 1. Such a scaled or unscaled Hamming distance has become a popular metric in quantifying the performance of community detection (Ghoshdastidar and Dukkipati 2017a, 2017b, Ke, Shi, and Xia 2021; Jin et al. 2021; Lee, Kim, and Chung 2021).

Let \(N^*_K = \{i : \psi^*_i = k\}\) be a true network community with cardinality \(n_k\) and \(C^* \in \mathbb{R}^{(K+1) \times K}\) be the associated community embedding center matrix in the sense that \(C^*_k = \frac{1}{n_k} \sum_{i | \psi^*_i = k} \alpha^*_i\), for \(k \in [K]\), and \(C^*_{K+1} = r^{-1/2}1_r\). Denote \(B^* = \mathcal{I} \times_1 C^*_1 \times_2 \cdots \times_m C^*_m\). The following assumptions are made to ensure that communities within the hypergraph networks are asymptotically identifiable.

**Assumption A.** There exists a constant \(c_3 > 0\) such that

\[
\min_{k, k' \in [K], k \neq k'} K^{1-m/2}||B^*_k - B^*_{k'}||_F \geq c_3 \gamma_n,
\]

where \(B^*_k\) is the \(k\)th sub-tensor of \(B^*\) by fixing the first index as \(k\), and \(\gamma_n\) may converge to 0 with \(n\).

**Assumption B.** There exists a constant \(c_4\) such that \(\max_k n_k \leq c_4 \min_k n_k\).

Assumption A is an identifiability assumption, which assumes that the true communities are well separated as \(n\) grows, and is crucial to the feasibility of community detection. It is interesting to remark that a signal-to-noise ratio is used in Yuan et al. (2021) to characterize the separability among communities under hSBM. A similar community separation assumption can be found in Lei, Chen, and Lynch (2019) for multi-layer network model. Assumption B assures the communities are well defined and will not degenerate asymptotically. This assumption is mild and satisfied when the vertex community memberships come from a multinomial distribution. The same assumption can be found in Ke, Shi, and Xia (2021), and a relatively stronger assumption can be found in Chien, Lin, and Wang (2019) assuming equal community sizes.

**Theorem 2.** Suppose all the assumptions in Theorem 1 as well as Assumptions A and B are satisfied, \(\lim_{n \to +\infty} \lambda_n e_n s_n^{-2} (\log s_n^{-1})^{-1} > 0\) and \(K = o(\gamma_n^{2} s_n^{-1} n^{-2})\), then \(\text{err}(\psi^*, \hat{\psi}) = O_P(\epsilon_n^{-1} s_n^{1/2} n^{-1/2})\).

Theorem 2 assures that the community structure in a general hypergraph can be consistently recovered by the proposed HEM method. The consistency result holds true for diverging \(K\) as long as it does not diverge too fast. Furthermore, Theorem 2 requires that \(\lambda_n\) cannot be too small, whereas Theorem 1 requires \(\lambda_n\) to be sufficiently small. Combining these two gives a proper interval for the order of \(\lambda_n\) to assure consistency in both network estimation and community detection.

As a theoretical example, consider a hypergraph network with \(J(\alpha^*) \leq e_n^{2}(s_n^{2} \log(s_n^{-1}))^{-1}\) and \(r\), \(\gamma_n\) and \(K\) are of the constant order. With \(\lambda_n = \frac{1}{2} e_n^{1} s_n^{2} \log(s_n^{-1})\), Theorem 1 implies that \(\epsilon_n\) is of the order \(n^{-1-m} \log n\), and Theorem 2 further implies that \(\text{err}(\psi^*, \hat{\psi}) = O_P(\log n/(n^{1-m} s_n))\), which matches up with the error rates in Ghoshdastidar and Dukkipati (2017a, 2017b) and Ke, Shi, and Xia (2021). To ensure the community detection consistency, Theorem 2 requires that \(s_n \gg \log n/n^{1-m}\), which is slightly weaker than the sparsity requirement in Ghoshdastidar and Dukkipati (2017a, 2017b) and Ke, Shi, and Xia (2021). We also remark that under homogeneous hSBM, where the probability of any \(m\) vertices forming a hypergraph is \(p\) if they are from the same community and \(q\) otherwise, both error rates and sparsity requirement may be improved (Ahn, Lee, and Suh 2018). However, such results highly rely on the restrictive homogeneous hSBM, and it remains unclear whether they can be extended to heterogeneous hSBM or the even more general HEM.

### 5. Numerical Experiments

We evaluate the performance of the proposed HEM method on a variety of synthetic and real-life nonuniform hypergraph network data. We compare its performance with some existing nonuniform hypergraph community detection methods in literature, including Tensor-SCORE (Ke, Shi, and Xia 2021), spectral hypergraph partitioning (SHP; Ghoshdastidar and Dukkipati 2017a), and weighted projection to graph method (WPTG; Kumar et al. 2021; Ghoshdastidar and Dukkipati 2015a). Tensor-SCORE is designed for uniform hypergraph community detection and can be extended to nonuniform hypergraph by representing nonuniform hypergraph as a collection of uniform hypergraphs or by adding multiple null vertices to convert the nonuniform hypergraph to a uniform one as in Ouvrard, Goff, and Marchand-Maillet (2021). We abbreviate the former extension as TS-1, while the latter extension as TS-2. SHP converts a nonuniform hypergraph to an incident network data. We compare its performance with some existing network estimation and community detection methods such as the spectral clustering, SCORE (Jin 2015) and modularity maximization algorithms can be employed.

Both HEM and Tensor-SCORE involve some tuning parameters, which can be optimally determined by some data-adaptive selection criteria, including the stability criteria (Wang 2010).
Table 1. The averaged community detection errors and their standard errors over 50 independent replications of different methods on the nonuniform hypergraph networks generated from Scenario 1.

| n  | sn | HEM  | TS-1          | TS-2          | SHP           | WPTG          |
|----|----|------|---------------|---------------|---------------|---------------|
| 300| 0.4| 0.1052 (0.0152) | 0.1623 (0.0200) | 0.1556 (0.0170) | 0.2781 (0.0270) | 0.2901 (0.0200) |
|    | 0.2| 0.0968 (0.0134) | 0.1788 (0.0206) | 0.1561 (0.0172) | 0.2823 (0.0210) | 0.2874 (0.0202) |
|    | 0.1| 0.1026 (0.0149) | 0.1777 (0.0197) | 0.1587 (0.0172) | 0.2855 (0.0206) | 0.2925 (0.0200) |
|    | 0.05| 0.1161 (0.0173) | 0.1891 (0.0216) | 0.1627 (0.0184) | 0.2906 (0.0202) | 0.2969 (0.0192) |
|    | 0.025| 0.1505 (0.0229) | 0.1797 (0.0208) | 0.1659 (0.0189) | 0.3018 (0.0199) | 0.3077 (0.0197) |
| 400| 0.4| 0.1094 (0.0159) | 0.1592 (0.0200) | 0.1556 (0.0167) | 0.2814 (0.0204) | 0.2968 (0.0197) |
|    | 0.2| 0.1003 (0.0145) | 0.1780 (0.0209) | 0.1566 (0.0168) | 0.2825 (0.0204) | 0.2976 (0.0197) |
|    | 0.1| 0.1120 (0.0155) | 0.1811 (0.0196) | 0.1557 (0.0167) | 0.2868 (0.0201) | 0.2987 (0.0194) |
|    | 0.05| 0.1251 (0.0184) | 0.1802 (0.0200) | 0.1615 (0.0175) | 0.2918 (0.0206) | 0.2996 (0.0198) |
|    | 0.025| 0.1342 (0.0213) | 0.1874 (0.0215) | 0.1736 (0.0197) | 0.2964 (0.0201) | 0.3051 (0.0196) |
| 500| 0.4| 0.1038 (0.0142) | 0.1407 (0.0173) | 0.1483 (0.0155) | 0.2768 (0.0205) | 0.2848 (0.0198) |
|    | 0.2| 0.0927 (0.0125) | 0.1632 (0.0195) | 0.1524 (0.0163) | 0.2783 (0.0205) | 0.2834 (0.0197) |
|    | 0.1| 0.0998 (0.0138) | 0.1688 (0.0198) | 0.1524 (0.0162) | 0.2827 (0.0209) | 0.2873 (0.0203) |
|    | 0.05| 0.1112 (0.0153) | 0.1649 (0.0182) | 0.1542 (0.0167) | 0.2848 (0.0208) | 0.2904 (0.0203) |
|    | 0.025| 0.1216 (0.0185) | 0.1683 (0.0192) | 0.1561 (0.0176) | 0.2916 (0.0203) | 0.2931 (0.0198) |

Table 2. The averaged community detection errors and their corresponding standard errors over 50 independent replications of different methods on the nonuniform hypergraph networks generated from Scenario 2.

| κ₁/κ₂ | HEM  | TS-1          | TS-2          | SHP           | WPTG          |
|------|------|---------------|---------------|---------------|---------------|
| 0.1  | 0.2364 (0.0233) | 0.3930 (0.0179) | 0.2236 (0.0246) | 0.3467 (0.0184) | 0.3050 (0.0225) |
| 0.3  | 0.1169 (0.0166) | 0.2091 (0.0259) | 0.1639 (0.0216) | 0.3207 (0.0211) | 0.3066 (0.0218) |
| 0.5  | 0.1165 (0.0164) | 0.1834 (0.0226) | 0.1541 (0.0177) | 0.3107 (0.0217) | 0.3021 (0.0213) |
| 0.7  | 0.1012 (0.0147) | 0.1711 (0.0218) | 0.1551 (0.0172) | 0.3050 (0.0216) | 0.3054 (0.0209) |
| 0.9  | 0.1064 (0.0152) | 0.1725 (0.0204) | 0.1595 (0.0171) | 0.2927 (0.0205) | 0.2907 (0.0198) |

or the network cross validation (Li, Levina, and Zhu 2020). Yet such data-adaptive selection schemes can be computationally expensive. Alternatively, in our numerical experiments, we follow the treatment in Ke, Shi, and Xia (2021) for the tuning parameters in Tensor-SCORE, and set λᵣ = 10⁻⁴/n for HEM to prevent J(α) from vanishing too fast. We scale λᵣ properly in order to ensure more information can be learned from the network likelihood at the early iterations in the computing algorithm. The numerical performance of all the methods is assessed by the average scaled Hamming error in (6).

5.1. Synthetic Networks

We consider two scenarios of synthetic networks for numerical comparison.

Scenario 1: The nonuniform hypergraph networks are generated from the HEM model in (1) and (2). We vary the number of vertices n ∈ {300, 400, 500}, the sparsity factor sn ∈ {0.4, 0.2, 0.1, 0.05, 0.025}, and set the range of all the hypergraphs to be m = 3 with r = K = 2. First, we generate the community centers Cᵦ ∈ ℝ⁹(K⁺1) × 1 with Cₖ⁺ ≈ Nᵦ(0ᵦ, Iᵦ) for k ∈ [K] and the last row Cₖ⁺⁺ = r⁻¹/²I₁ᵦ, where Iᵦ is the r-dimentional identity matrix. Next, the vertex community memberships ψᵦ*, i ∈ [n], are generated from the multinomial distribution with parameters 1/1ₖ for k ∈ [K] indicating that the communities are approximately equal sizes. The community membership matrix Z* ∈ {0, 1}ⁿ⁺(n⁺1) × (K⁺1) can be constructed accordingly. After that, the hyperedge embedding matrix is generated as α⁺ = Z⁺C⁺ + ε. Herein, ε is a noise matrix with εᵦ₁⁺⁺ ~ N(0, 0.5²) if i ∈ [n] and εᵦ₁⁺⁺ = 0 if i = n + 1. Finally, P⁺ can be computed based on α⁺. For i₁, i₂, i₃ such that n⁺⁺₁⁺⁺ = 0, the hyperedge among vertices [{i₁, i₂, i₃}] \ {n + 1} is generated with probability p⁺⁺₁⁺⁺ independently.

Scenario 2: The hypergraph generation process is the same as in Scenario 1, except that community sizes can be unbalanced. Specifically, we fix n = 300, m = 3, sn = 0.1 and r = K = 2, and vary the parameters of the multinomial distribution (κ₁, κ₂) such that κ₁ + κ₂ = 1 and κ₁/κ₂ ∈ {0.1, 0.3, 0.5, 0.7, 0.9}. Clearly, as κ₁/κ₂ gets smaller, the communities become more unbalanced.

For both scenarios, the averaged scaled Hamming errors and their corresponding standard errors of various community detection methods over 50 independent replications are reported in Tables 1 and 2.

It is evident that HEM yields the smallest community detection error among all the methods in both scenarios. The performance of HEM and Tensor-SCORE is much better than that of SHP and WPTG, mainly due to the fact that SHP and WPTG need to convert the hypergraph adjacency tensor to matrix and thus suffer from information loss. Further, the performance of TS-2 appears to be better than TS-1, suggesting that converting a nonuniform hypergraph to a uniform one by adding null vertices can be a better data processing approach than decomposing a nonuniform hypergraph into a collection of uniform hypergraphs of different range. It is also interesting to note that HEM is fairly robust to the hypergraph network sparsity and the community imbalance, whereas the performance of other competing methods can be substantially affected.

To examine the network estimation accuracy, the averaged estimation errors of HEM, measured by n⁻ᵐ⁻⁻¹/₂||Θ − Θ⁺||₁, and their corresponding standard errors are reported in Table 3. Note that other competing methods solely focus on community detection, and thus do not produce estimate of Θ⁺. Clearly, the estimation error of HEM becomes smaller as the number of ver-
tices increases, the network becomes denser or the communities become more balanced.

Finally, Figure 2 displays the first 15 eigenvalues of two randomly generated hypergraphs from Scenarios 1 and 2. It is clear that the first 2 leading singular values are substantially larger than the remaining singular values, confirming the choice of $K = 2$ in the synthetic networks and the effectiveness of the eigen-gap approach.

5.2. Two Real-Life Hypergraph Networks

We now apply the proposed HEM method to analyze two real-life hypergraph networks, including the Medical Subject Headings (MeSH) hypergraph network (Ke, Shi, and Xia 2021) and the cardiac Single Proton Emission Computed Tomography (SPECT) network (Dua and Graff 2019).

The MeSH network is extracted from the MEDLINE database, which is available at https://www.nlm.nih.gov/bsd/medline.html. It consists of 318 MeSH terms of two diseases: Neoplasms (C04) and Nerve System Diseases (C10), which are represented as the vertices in the hypergraph network. The hyperedges are 10,472 papers published in 1960 where one or more of the above Mesh terms are annotated. After deleting the duplication hyperedges, there are a total of 1,483 hyperedges with sizes varying from 1 to 36. We only consider those hyperedges of size ranging from 2 to 6, and for hyperedges with sizes greater than 3, we convert them to all possible 3-cliques, leading to multiple hyperedges with size 3. Replication hyperedges and isolated vertices are further deleted. After the preprocessing, we obtain a hypergraph with 264 vertices and 2,950 hyperedges, where 211 vertices come from C04 and the other 103 vertices come from C10.

The SPECT network (https://archive.ics.uci.edu/ml/datasets/spect+heart) contains the SPECT images of 267 patients, where each image has been processed to 44 categorical features to discriminate abnormal patients from the normal ones. To construct the hypergraph network, each patient is represented as a vertex, and for a possible value of each feature, we construct a hyperedge that contains all the patients sharing the particular feature value. Such a hypergraph construction method has been studied by Schölkopf et al. (2007) and Ghoshdastidar and Dukkipati (2017a). After deleting the replication hyperedges, there are a total of 1,483 hyperedges with sizes varying from 1 to 36. We only consider those hyperedges of size ranging from 2 to 6, and for hyperedges with sizes greater than 3, we convert them to all possible 3-cliques, leading to multiple hyperedges with size 3. Replication hyperedges and isolated vertices are further deleted. After the preprocessing, we obtain a hypergraph with 264 vertices and 2,950 hyperedges, where 211 vertices come from the abnormal patients and the other 53 ones come from the normal patients.

We perform different community detection methods on both hypergraph networks. As suggested in Ke, Shi, and Xia (2021), we set $K = 6$ for the reg-HOOI algorithm in Tensor-SCORE, and thus we also set the embedding dimension $r = 6$ in HEM for fair comparison. The network sparsity factor in HEM is estimated by hyperedge density; that is, $s_n = \frac{1057 \times \left( \binom{281}{2} \right) + \left( \binom{281}{3} \right)^{-1}}{281} = 0.0427$ in MeSH network and $s_n = 2950 \times \left( \frac{264}{2} \right) + \left( \frac{264}{3} \right)^{-1}$ in SPECT network. The scaled Hamming errors of all the hypergraph community detection methods are reported in Table 4.

It is evident that the scaled hamming errors of HEM are smaller than the other three competitors in both hypergraph networks.
networks, demonstrating its advantage in terms of community detection. We further visualize the estimated communities in both hypergraph networks in Figure 3 by multidimensional scaling, where the community structures detected by HEM are very clear in the embedding space.

6. Conclusion

This article proposes a novel community detection method on general hypergraph networks. The proposed method is built upon a tensor-based hypergraph embedding model, which consists of a network augmentation step and an network embedding step. The resultant negative log-likelihood function is equipped with a new regularization term to encourage community structures among the embedding vectors. The proposed method is supported by various numerical experiments and asymptotic consistency in terms of community detection. Particularly, the theoretical results can be established for very sparse hypergraph network with link probability of order \( sn \gg n^{1-m}\log n \). It is worth pointing out that the proposed community detection method can be extended to various scenarios, such as multi-layer hypergraph networks or communities with mixed memberships, which is under further investigation.

Supplementary Material

The supplementary materials contain proofs of the theoretical results and python codes to run the numerical experiments of the paper.

Acknowledgment

We thank to the associate editor and three anonymous referees, whose constructive comments and suggestions have led to significant improvements of the article. We also thank to Dr Dong Xia for sharing the MeSH dataset.

Funding

This research is supported in part by HK RGC Grants GRF–11303918, GRF–11300919, and GRF–11304520.

ORCID

Yaoming Zhen https://orcid.org/0000-0002-3724-9200

References

Agarwal, S., Lim, J., Zelnik-Manor, L., Perona, P., Kriegman, D., and Belongie, S. (2005), “Beyond Pairwise Clustering,” in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05) (Vol. 2), IEEE, pp. 838–845. [620]

Ahn, K., Lee, K., and Suh, C. (2018), “Hypergraph Spectral Clustering in the Weighted Stochastic Block Model,” IEEE Journal of Selected Topics in Signal Processing, 12, 959–974. [625]

Arous, G. B., Mei, S., Montanari, A., and Nica, M. (2019), “The Landscape of the Spiked Tensor Model,” Communications on Pure and Applied Mathematics, 72, 2282–2330. [624]

Bahmanian, M. A., and Sajna, M. (2015), “Connection and Separation in Hypergraphs,” Theory and Applications of Graphs, 2, 5. [621]

Chen, J., and Yuan, B. (2006), “Detecting Functional Modules in the Yeast Protein–Protein Interaction Network,” Bioinformatics, 22, 2283–2290. [620]

Chen, K., and Lei, J. (2018), “Network Cross-Validation for Determining the Number of Communities in Network Data,” Journal of the American Statistical Association, 113, 241–251. [624]

Chien, I. E., Lin, C.-Y., and Wang, I.-H. (2019), “On the Minimax Misclassification Ratio of Hypergraph Community Detection,” IEEE Transactions on Information Theory, 65, 8095–8118. [625]

De Lathauwer, L., De Moor, B., and Vandewalle, J. (2000), “A Multilinear Singular Value Decomposition,” SIAM Journal on Matrix Analysis and Applications, 21, 1253–1278. [624]

Dua, D., and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science. [627]

Gallo, G., Longo, G., Pallottino, S., and Nguyen, S. (1993), “Directed Hypergraphs and Applications,” Discrete Applied Mathematics, 42, 177–201. [621]

Ghoshdastidar, D., and Dukkipati, A. (2014), “Consistency of Spectral Partitioning of Uniform Hypergraphs Under Planted Partition Model,” in Advances in Neural Information Processing Systems 27, eds. Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., pages 397–405. Curran Associates, Inc. [620]

Ghoshdastidar, D., and Dukkipati, A. (2014a), “Consistency of Spectral Partitioning of Uniform Hypergraphs Under Planted Partition Model,” in Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37 (ICML’15), pp. 400–409. [625]

Ghoshdastidar, D., and Dukkipati, A. (2014b), “A Provable Generalized Tensor Spectral Method for Uniform Hypergraph Partitioning,” in Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI’15), pp. 2610–2616. [620]

Ghoshdastidar, D., and Dukkipati, A. (2015a), “Spectral Clustering Using Multilinear svd: Analysis, Approximations and Applications,” in Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI’15), pp. 2610–2616. [620]

Ghoshdastidar, D., and Dukkipati, A. (2015b), “Uniform Hypergraph Partitioning: Provable Tensor Methods and Sampling Techniques,” Journal of Machine Learning Research, 18, 1–41. [621,622,625]
