University of Padova

Department of Mathematics "Tullio Levi-Civita"

Master Thesis in Data Science

Inference and Community Detection in Hypergraphs: Incorporating Node Attributes

Supervisor
Prof. Wolfgang Erb
University of Padova

Co-supervisor
Dr. Caterina De Bacco
Max Planck Institute for Intelligent Systems

Master Candidate
Anna Badalyan

Student ID
2041296

Academic Year
2023-2024

Time is the only constraint. Give me unlimited GPUs and I will move the earth.

Anna Badalyan

This thesis is dedicated to my dear parents: my dad for the endless support and always being a phone call apart to explain how to take an integral; my mum for being the most hardworking person I have ever seen and always encouraging me to succeed.

Abstract

In this work, we consider the community detection problem on hypergraph networks. It often occurs that network information comes with additional attributes on nodes, which could be used to improve our understanding of the network structure. We thus propose a probabilistic generative model that is able to use the information about higher-order interactions as well as the node attributes to infer the structure of the network. We demonstrate a variety of cases where using our model provides a significant advantage compared to the methods that do not use any attribute information or the methods that infer network structure from attributes alone. The proposed method is able to identify automatically if the attributes are informative and discard them otherwise. We show the benefits of using our model on the link prediction task when the given attribute is informative. The model comes with an efficient implementation that allows it to generalize to hypergraphs of large size both in terms of the number of nodes and number of edges.

Contents

Listing of figures

Listing of tables

Listing of acronyms

AUC . . . . . Area under the curve as defined in subsection 4.2.1

Hy-MMSBM . . . Hypergraph Mixed Membership Stochastic Block Model

HyCoSBM . . . . . Hypergraph Covariates Stochastic Block Model

1 Introduction

The notion of a hypergraph was first introduced in 1973 by Berge [1] as an extension to graph approaches [2]. While a dyadic graph is a combinatorial structure that consists of vertices (or nodes) and edges (or links) between these nodes, a hypergraph consists of vertices and hyperedges. The edges of a dyadic graph can connect only two nodes, whilst the hyperedges can connect an arbitrary number of nodes of the hypergraph. The hypergraphs are also called higher-order networks due to their ability to describe interactions of more than two nodes. In this work, we use the words hypergraph, higher-order network, and network interchangeably unless otherwise specified.

The ability of hypergraphs to represent systems where group interactions are observed has increased the interest of the machine learning community. Such systems include cellular networks [3], ecological systems [4], brain networks [5], human interactions [6], and drug recombination [7]. The research in this area often focuses on studying the structure of higher-order networks. These networks, however, often come with additional information about node attributes that describe the properties of the nodes. Such properties could be the age, education, or job title in human interaction networks or features of genes in biological networks. Therefore, this work is dedicated to analyzing how to use the information about node attributes to improve our understanding of higher-order networks.

Community detection is a powerful tool to study the mechanism driving the formation of edges in the network. This is also the main focus considered in this thesis to address our goal of modeling hypergraphs with the use of node attributes.

We propose a probabilistic generative model that is capable of inferring the structure of hypergraphs guided by the node attributes. We demonstrate that our approach is superior to using only the hypergraph network information or only the attribute information in various settings. Our proposed model can handle categorical or binary attributes and can be applied to both weighted and unweighted hypergraphs. In addition, we demonstrate that the model can handle overlapping communities as well as various network structures (assortative, disassortative, and core-periphery). The computational complexity of our implementation is linear in terms of the number of nodes and hyperedges of the hypergraph, which makes it efficient in handling large hypergraph networks.

The code is publicly available at github.com/badalyananna/HyCoSBM. The main results of our method are available as a preprint at [8].

The thesis is structured as follows:

- chapter 2 provides a review of the commonly used methods for community detection on hypergraphs and the methods to incorporate node attributes into community detection algorithms;
- chapter 3 describes in detail the models used in this thesis. The chapter starts with the notation and mathematical definition of a hypergraph. The subsequent sections describe the Hy-MMSBM model used as a baseline and the HyCoSBM model, which is our method to incorporate node attributes. The last section covers alternative methods that were implemented but resulted in inferior performance;
- chapter 4 presents the main results obtained by our models on synthetically generated data and real hypergraph networks. The last section shows the results obtained by alternative models;
- chapter 5 summarizes the main aspects of the work that have been carried out and gives the outline of the possible future research directions.

2 Related works

Mathematical frameworks that allow the representation of higher-order interaction, such as hypergraphs, are an emerging topic in network science [9]. A variety of tools have been developed for the analysis of higher-order networks, the most popular of them being community detection. The first part of this review is dedicated to the current methods available for community detection on hypergraphs. In the second part, we describe the most common methods to incorporate node attributes in such community detection models. In this part, we keep the notation similar to the one used in the original paper, while starting from chapter 3, we introduce the notation for the proposed method.

2.1 Community detection methods on hypergraphs

Community detection is one of the most popular tools for network analysis. There are several types of network structures that a community detection algorithm can recover, such as assortative, disassortative, and core-periphery. An *assortative structure* assumes that nodes within one community interact mainly with the other nodes in the same community. A *disassortative structure* assumes that nodes interact predominantly with the nodes from other communities. A *core-periphery* structure presumes the presence of core nodes that have many connections with the other nodes and periphery nodes that are only connected to a small number of core nodes.

A plethora of methods have been developed to solve the community detection problem on

dyadic graphs [10]. While there exist techniques to project a hypergraph into a dyadic graph with the aim of applying established graph methods for community detection, they may induce unwanted information loss [11]. Thus, a number of methods have been introduced to solve the community detection or clustering problem directly on hypergraphs. Such methods include non-parametric methods with hypergraphons, latent space distance models, latent class models, tensor decompositions, flow-based models, spectral clustering, spectral embedding, and statistical inference [12]. In this section, we discuss the most popular of these methods, namely tensor decomposition, spectral clustering, and statistical inference.

2.1.1 Spectral clustering and tensor decomposition

One of the best-known algorithms for clustering on dyadic networks is the Normalized cut initially introduced for image segmentation by Shi and Malik [13]. It was shown that eigenvectors of the normalized graph Laplacian corresponding to k smallest eigenvalues can be used to represent a graph in a k-dimensional Euclidean space. Algorithms similar to k-means can be applied to partition these points. Zhou et al. [14] extended the classical spectral clustering approach to hypergraphs. The authors approximate hypergraph normalized cut using real-valued relaxations. In line with the graph Laplacian, the authors introduced the hypergraph Laplacian, which is based on the hypergraph normalized cut criterion. The hypergraph Laplacian is defined as follows

$$L = I - \frac{1}{2} D^{-1/2} H W H^T D^{-1/2}, \tag{2.1}$$

where D is a diagonal matrix containing the node degrees, $H \in \mathbb{R}^{|V| \times |E|}$ is a binary incidence matrix containing 1 if node v belongs to hyperedge e and 0 otherwise, and W is a diagonal matrix containing hyperedge weights. Taking k smallest eigenvectors of the hypergraph Laplacian allowed us to develop additional hypergraph embedding and transductive inference algorithms.

Ghoshdastidar et al. [15] propose a community detection method for uniform hypergraphs by introducing the notion of associativity maximization and formulating a problem as a tensor trace maximization problem. The authors show that other methods, such as normalized spectral clustering, non-negative tensor factorization, and hypergraph reduction by clique averaging, become a special case of the proposed formulation. In addition, the authors provide theoretical guarantees for the proposed algorithm under a planted partition model, which is a specific type of a stochastic block model.

Another community detection method on hypergraphs based on spectral clustering was proposed by Angelini et al. [16]. The method is based on the generalization of the non-backtracking

matrix, which is defined for hypergraphs as

$$B_{(i\to\mu)(j\to\nu)} = \begin{cases} 1 & \text{if } j \in \partial\mu \setminus i, \nu \neq \mu \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

Here, $i, j = \{1, \ldots, N\}$ are node indices, $\mu, \nu = \{1, \ldots, M\}$ are hyperedges, $\partial\mu$ denotes a set of nodes in a hyperedge μ. Thus, the B matrix shows the connections between each group of hyperedges, and $i \to \mu$ denotes the indices of nodes belonging to the hyperedge μ. The size of the B matrix is $\hat{k}M \times \hat{k}M$, where $\hat{k}$ is the average number of nodes per hyperedge. The eigenvectors corresponding to the largest q eigenvalues of the B matrix after the first one are used to partition the nodes in q groups. This algorithm performs well only on extremely sparse hypergraphs where the number of nodes is of the same order as the number of hyperedges.

Ke et al. [11] propose a community detection algorithm on hypergraphs based on the approximation of the low-rank tensor decomposition of the hypergraph adjacency tensor. The approximation is performed via regularized higher-order orthogonal iteration (reg-HOOI) algorithm, which performs better compared to other tensor decomposition methods like HOSVD. The results of the tensor decomposition are then used to perform k-means clustering to the rows of the normalized factor matrix. In addition, the authors provide a theory for the degree-corrected block model for hypergraphs that is used to generate hypergraphs and show consistency in community detection of the clustering algorithms.

One of the limitations of these methods is that spectral clustering can be unreliable, especially when the difference between eigenvalues related to different communities is small, which is true even on dyadic networks [10]. With hypergraphs, in addition to these issues, higher-order decomposition of adjacency tensors can become prohibitively expensive, especially for large networks. Another limitation of the methods based on spectral clustering is the assumption of the assortative structure of the network. Lastly, these methods cannot detect overlapping communities.

2.1.2 Statistical inference

Statistical inference uses the observation and a hypothesis of an underlying structure to discover properties of the graph, such as how the nodes are connected to each other [17]. It has become a standard approach to use generative models that fit the data by maximizing the likelihood of observing the graph given some model [10]. The simplest type of a model on graphs is the

Erdős–Rényi random graph model, which assumes that any 2 nodes in the graph are connected with probability p [18]. A simple but powerful extension to the random graph model is the stochastic block model that has become the most popular generative model on networks [10]. Given a number of communities k the stochastic block model assumes that the probability of the nodes in different communities being connected with each other is represented by a symmetric matrix w. In the case of an assortative structure where nodes interact only with other nodes in the same community, the matrix becomes an identity matrix. Such representation can describe various types of structures, such as disassortative or core-periphery.

An example of such an approach on dyadic networks is a MULTITENSOR algorithm presented in [19]. In this work, De Bacco et al. proposed a mixed membership stochastic block model for the inference of overlapping communities in directed multilayer networks. The model assumed that outgoing and incoming memberships of each node i are described by vectors u_i and v_i. An affinity matrix $w^{(\alpha)} \in \mathbb{R}^{k \times k}$ shows the density of connections between communities k for each layer α. Finally, the entries i, j of the adjacency matrix A for each layer α are assumed to be extracted from a Poisson distribution with mean $M^{(\alpha)}$ defined as

$$M_{i,j}^{(\alpha)} = \sum_{k,q=1}^{K} u_{ik} v_{jq} w_{kq}^{(\alpha)} \quad . \tag{2.3}$$

To maximize the likelihood of observing all edges an efficient Expectation Maximization algorithm is applied.

Contisciani et al. extended the MULTITENSOR approach to hypergraphs in [20]. In this work, a hypergraph is represented as an adjacency tensor A with entries $A_{i1,\ldots,id}$ being the weights of d-dimensional interactions. The membership vector u_i and affinity tensor w control the Poisson distribution of hyperedge weights with a mean

$$\lambda_{i_1,\ldots,i_d} = \sum_{k_1,\ldots,k_d} u_{i_1 k_1} \ldots u_{i_d k_d} w_{k_1,\ldots,k_d} \quad . \tag{2.4}$$

With this formulation, we can see that the size of the affinity tensor w can become exponentially large with the increase in the hyperedge size d. Thus, to reduce the dimensions of the w parameter, the authors assume the assortative structure of the network. This makes it possible to reduce the dimensions of w to $D \times K$ where D is the maximum hyperedge size and K is the number of communities. However, even with this simplification the maximum number of hyperedges has to be limited in practice due to the increasing computation complexity.

Chodrow et al. [21] presented a probabilistic generative model for community detection on hypergraphs based on a degree-corrected stochastic block model. An important feature of the model is that hypergraphs are heterogeneous in hyperedge size and node degree. To fit the model to the data, the authors introduce an approximate coordinate ascent scheme for maximum likelihood estimation and formulate a modularity-like objective. The modularity objectives are solved by adopting Louvain heuristics for graphs to hypergraphs. The Louvain-like algorithms consist of 2 phases. In Phase 1, each node is initialized as a single cluster, and at each iteration the node is moved to the adjacent one if it maximizes the modularity objective. In Phase 2, the "supernode" that unites all the nodes sharing the same label is formed. The phases are repeated until no improvement can be reached. Similarly to other maximum likelihood approaches, the resulting algorithm is not guaranteed to reach a global maximum.

Brusa and Matias [22] propose a stochastic block model for simple hypergraphs, that is hypergraphs where the set of hyperedges consists of unique nodes. In contrast to the previously discussed works, the authors assume the Bernoulli distribution of hyperedges, which can accommodate only unweighted hypergraphs. The parameters of the model are inferred using a variational Expectation-Maximization algorithm. However, due to computational considerations, the introduced approach is applicable only to small hypergraphs.

To solve the computation complexity issues Ruggeri et al. [12] introduce the Hy-MMSBM model. Similarly to MULTITENSOR, the distribution of hyperedge weights is controlled by parameters $u \in \mathbb{R}^{N\times K}$ and $w \in \mathbb{R}^{K\times K}$. This significantly reduces the computation complexity compared to Hypergraph-MT and allows inference of various network structures such as core-periphery and disassortative. The computational efficiency and flexibility of the method make it a viable option to use as a baseline for incorporating attributes. Thus, we describe this model in detail in section 3.2.

2.2 Methods to incorporate node attributes in community detection

There are works that focus on providing tools for the exploratory analysis of annotated hypergraphs, such as [23], where the authors generalize the common methods used for dyadic network analysis including centrality, assortativity, and modularity scores. However, to the best of our knowledge, there have been few works in the literature that develop a community detection model on annotated hypergraphs. Thus, we first review the important works on the

methods that incorporate attributes in community detection models on dyadic networks and then discuss the existing methods on hypergraphs.

There exist a number of works that provide a comprehensive survey on the methods used to cluster attributed graphs [24, 25]. Chunaev [25] focuses on attributed social networks and classifies the existing methods of incorporating node attributes into community detection into 3 main categories namely *early fusion methods, simultaneous fusion methods* and *late fusion methods*. *Early fusion methods* join network information with the attributes before applying a community detection procedure. An example of such an approach is a modification of the graph adjacency matrix entries with weights that are obtained from the information about nodes. The main benefit of this approach is that after obtaining a modified network any known algorithm for the community detection can be applied. *Late fusion methods* carry out a separate community detection process on the network part and attributes part and then merge the obtained partitions. Similarly to the first method, this method can use the existing algorithms for the main task. On the contrary, *simultaneous fusion methods* use network and attributes together in the community detection process and thus, require a separate implementation of the algorithm. These methods usually modify the objective function of the existing community detection algorithms to include the attributes term and in some sense they find a qualitatively optimal solution while the methods described previously simply merge the attributes and network together and the optimality of the merging procedure cannot be assessed properly.

Simultaneous fusion methods showed a significant improvement in the quality of detected communities when the attributes are used on dyadic networks [26, 27]. Yang et al. [26] proposed a method based on a generative model for networks. The network part is modeled by assuming a Bernoulli distribution of the entries of the adjacency matrix of the graph. The probability of observing the entry i, j of the adjacency matrix is

$$\mathbb{P}\{A_{i,j} = 1\} = 1 - \exp(-\sum_{c \in C} u_{ic} \cdot u_{jc}) \quad ,$$

where C is a set of all communities and u is a community membership parameter to be learned. The attribute part is modeled by assuming a Bernoulli distribution of the attributes. The probability of observing the attribute k with the node i is

$$\mathbb{P}\{X_{ik} = 1\} = \frac{1}{1 + \exp(-\sum_{c \in C} \beta_{kc} \cdot u_{ic})} \quad . \tag{2.5}$$

To join the 2 parts the authors assume conditional independence of X and A given u and β and

apply l_1 regularization to the β parameter. The objective function then becomes

$$\underset{u \geq 0, \beta}{\operatorname{argmax}} \mathcal{L}_A + \mathcal{L}_X - \lambda |\beta|_1 .$$

The function is maximized using the block coordinate ascent approach, that is updating u_i while keeping u_j and β fixed and updating β while u is fixed. This method, however, assumes an assortative structure of the network, i.e. the nodes tend to interact only with the nodes from the same community.

A similar approach is adopted by Contisciani et al. [27] for multilayer networks. The entries of the adjacency matrix for each layer are assumed to be extracted from a Poisson distribution that depends on a community membership matrix u. The entries of the attributes matrix are categorical and one-hot encoded; therefore, they are assumed to be extracted from a Multinomial distribution, which also depends on the u matrix. The probabilities are combined in line with [26] and the final optimization problem is solved using an efficient EM algorithm. This approach was shown to be flexible in predicting missing links and attributes as well as discovering interpretable community divisions. In contrast to Yang et al. [26], the model can discover the community structures other than assortative (i.e. disassortative, core-periphery). Therefore, we adopt the main ideas from this approach in order to incorporate node attributes in the community detection algorithm on hypergraphs.

Fanseu et al. [28] developed a community detection method on attributed hypergraphs based on hypergraph convolution. This is an early fusion method that constructs a nonlinear hypergraph adjacency matrix A_s using the information about attributes in addition to the hypergraph structure. The convolution filter constructed using A_s is then applied to the node attributes matrix X. The resulting matrix is normalized and the top k eigenvectors are used to carry out k-means clustering. Although the definition of hypergraph Laplacian developed by the authors is computationally more efficient than the one proposed by Zhou et al. [14], it still suffers from the general limitation of the methods based on spectral clustering discussed previously.

A simultaneous fusion method developed for text analysis was proposed by Du et al. [29]. The method is based on non-negative matrix factorization that joins the objectives for text classification and hypergraph clustering. The authors propose a block coordinate descent scheme to minimize the joint objective function. Similarly to spectral clustering based methods, this method allows only node clustering without considering alternative structures of the hypergraph network.

The most recent work published by Li et al. [30] clusters attributed hypergraphs using k-nearest neighbor augmentation to include the attributes information into the hypergraph. The resulting augmented hypergraph is then partitioned using a random walk based model. This approach is subject to the same limitations as other early fusion methods.

3 Models

We have developed a model for community detection and inference in hypergraphs with the possibility of incorporating information about node attributes. Our approach is based on statistical inference and uses a generative stochastic block model for hypergraph structure. We add node attributes by encoding them into binary variables that are distributed according to Bernoulli distributions. We then use a simultaneous fusion approach to maximize the joint likelihood of both network structure and attributes. We model the network structure using the Hy-MMSBM model [12], which we are going to describe in this section. Then, we propose a model for the attributes and combine it with the one for the network structure to finally describe our model, which we refer to as HyCoSBM. In the section 3.4, we present alternative formulations of the model to incorporate attributes that were not as successful as HyCoSBM.

3.1 Notation

The hypergraph is represented as a $H = (V, E, A)$, where

- $V = \{1, \ldots, N\}$ is a set of nodes,
- E is a set of observed hyperedges with a hyperedge $e \in E$ representing an arbitrary set of two or more nodes in V,
- A is a vector containing the weights of edges, which are assumed to be positive integers.

The set of all possible hyperedges is denoted as Ω. We further denote A_e as the weight of the hyperedge $e \in E$ with $A_e = 0$ if $e \in \Omega \setminus E$.

To incorporate attribute information we represent the attributes on nodes as a matrix $X \in \mathbb{R}^{N \times Z}$, where Z is the number of attributes, with entries equal to 1 if the node i has attribute z and 0 otherwise. In principle, each attribute (i.e. job title) can have several discrete values which are then one-hot encoded and stacked together.

3.2 Hy-MMSBM

The Hypergraph Mixed Membership Stochastic Block Model [12] (Hy-MMSBM) is a model for detecting mixed-membership communities in hypergraphs. By using only the set of observed hyperedges and their weights in input, it assigns membership vectors to nodes that describe how nodes are partitioned into overlapping groups. This means that each node can belong to several communities. The number of possible communities denoted as K is a hyperparameter of the model, which can be selected using model selection criteria. In our experiments, we use cross-validation. The community structure is modeled using two main parameters:

1. u — an $N \times K$ membership matrix showing the extent to which a node i belongs to each of the communities K. The matrix is non-negative.
2. w — a $K \times K$ affinity matrix controlling the likelihood that the nodes within one group are to interact with the nodes from another group. If the nodes interact only with the nodes in the same group, the affinity matrix becomes a diagonal matrix. This matrix is also non-negative.

These two parameters control the Poisson distribution of hyperedge weights, which are positive and discrete quantities. Specifically, the likelihood of observing a hyperedge e with weight A_e is:

$$P(A_e|u, w) = Pois\left(A_e; \frac{\lambda_e}{k_e}\right) \quad , \tag{3.1}$$

where

$$\lambda_e = \sum_{i<j:i,j\in e} u_i^T w u_j = \sum_{i<j:i,j\in e} \sum_{k,q=1}^{K} u_{ik} u_{jq} w_{kq} \quad . \tag{3.2}$$

The parameter k_e is a normalization constant that depends only on the hyperedge size $|e|$. The authors suggest the value of $k_e = \frac{|e|(|e|-1)}{2}\binom{N-2}{|e|-2}$ which intuitively normalizes the number of possible choices of nodes given the two existing nodes i and j.

It is assumed that the hyperedges are conditionally independent, given the parameters u, w. Hence, the probability of observing a hypergraph can be factorized into products of probabilities of individual hyperedges as in Equation 3.1. Taking the logarithms and simplifying the expressions, the authors derived the following expression of the log-likelihood:

$$L(u,w) = -C \sum_{i<j\in V} u_i^T w u_j + \sum_{e\in E} A_e \log \sum_{i<j\in e} u_i^T w u_j \tag{3.3}$$

$$\geq -C \sum_{i<j\in V} u_i^T w u_j + \sum_{e\in E} A_e \sum_{i<j\in e} \sum_{k,q=1}^{K} \rho_{ijkq}^{(e)} \log \left(\frac{u_{ik} u_{jq} w_{kq}}{\rho_{ijkq}^{(e)}} \right), \tag{3.4}$$

where the inequality is obtained by using a standard variational approach via Jensen's inequality $\log \mathbb{E}[X] \geq \mathbb{E}[\log(X)]$ and introducing a probability distribution $\sum_{i<j\in e} \sum_{k,q=1}^{K} \rho_{ijkq}^{(e)} = 1$. The equality is reached when

$$\rho_{ijkq}^{(e)} = \frac{u_{ik} u_{jq} w_{kq}}{\lambda_e} \quad . \tag{3.5}$$

Therefore, maximizing Equation 3.4 is equal to maximizing Equation 3.3.

This can be performed efficiently using an Expectation-Maximization algorithm [32]. The algorithm alternates between updating ρ and parameters (u, w). The following updates for u and w were obtained by setting the derivative with respect to each of the parameters to 0

$$w_{kq} = \frac{\sum_{e\in E} A_e \sum_{i<j\in e} \rho_{ijkq}^{(e)}}{C \sum_{i<j\in V} u_{ik} u_{jq}} \qquad u_{ik} = \frac{\sum_{e\in E: i\in e} A_e \sum_{j\neq i\in e} \sum_{q=1}^{K} \rho_{ijkq}^{(e)}}{C \sum_{j\in V, j\neq i} \sum_{q=1}^{K} u_{jq} w_{kq} + \lambda_{ik}^{(u)}} \quad . \tag{3.6}$$

The authors show that with the efficient implementation, the complexity of the algorithms is linear in terms of the number of nodes N and number of hyperedges $|E|$ of the hypergraph. Ruggeri et al. [12] conducted a number of experiments on synthetic datasets to show that the accuracy of Hy-MMSBM in recovering community structure exceeded the state-of-the-art approaches. The authors showed that the model is flexible in recovering both assortative and disassortative community structures which could not have been previously studied. Moreover, the computational efficiency of the model makes it possible to use it on large hypergraphs. The above-listed advantages of the model make it a good choice for a starting baseline to incorporate node attributes.

3.3 HyCoSBM

We propose HyCoSBM (Hypergraph Covariates Stochastic Block Model), which is capable of using the information on node attributes for community detection and inference on hypergraphs. We call the structure of a hypergraph *structural information* and the node attributes *attribute information*.

We model the two types of information probabilistically assuming the joint probability distribution conditioned on a set of latent variables θ. We denote the set of variables related to the structural information θ_A and the set of variables related to the attribute information θ_X. It is important to note that $\theta_A \cup \theta_X = \theta$, but $\theta_A \cap \theta_X \neq \emptyset$, which means that there should be at least one shared latent variable so that it is possible to learn from both distributions. Finally, we assume that the distributions are conditionally independent given the latent variables

$$P(A, X \mid \theta) = P_A(A \mid \theta_A)\, P_X(X \mid \theta_X)\,. \tag{3.7}$$

The factorization is analogous to the approaches on dyadic graphs presented in [27] and [26]. Such an approach is advantageous as it allows for closed form solutions, as we will show below. In addition, it also allows predicting both missing hyperedges and missing attributes, which can be useful, for example, in case of corrupted data or prediction tasks.

3.3.1 Modeling hypergraph structure

To model the hypergraph structure we follow an approach analogous to the Hy-MMSBM model described above. Thus, the latent variables controlling the distribution of hyperedge weights are $\theta_A = \{u, w\}$, where u is a $N \times K$ community membership matrix and w is a $K \times K$ affinity matrix. The likelihood of the hypergraph is:

$$P_A(A|u, w) = \prod_{e \in \Omega} Pois\left(A_e; \frac{\lambda_e}{k_e}\right) \quad , \tag{3.8}$$

where similarly to the Hy-MMSBM model

$$\lambda_e = \sum_{i<j : i,j \in e} u_i^T w u_j = \sum_{i<j : i,j \in e} \sum_{k,q=1}^{K} u_{ik} u_{jq} w_{kq} \quad . \tag{3.9}$$

The logarithm is up to constant terms:

$$\log P_A(A|u,w) = \sum_{e\in\Omega} -\frac{1}{k_e} \sum_{i<j\in e} u_i^T w u_j + \sum_{e\in E} A_e \log \sum_{i<j\in e} u_i^T w u_j \tag{3.10}$$

The summation over the set of all possible hyperedges Ω is in principle intractable. However, using the same trick as in the Hy-MMSBM model [33], the first term in Equation 3.10 can be rewritten in terms of C, the count of how many times each term $u_i^T w u_j$ appears in all possible hyperedges, weighted by $1/k_e$. Specifically, $C = \sum_{n=2}^{D} \frac{1}{k_n}\binom{N-2}{n-2}$, with $k_n = \frac{n(n-1)}{2}\binom{N-2}{n-2}$ in our case. The log-likelihood is then simplified as:

$$\log P_A(A|u,w) = -C \sum_{i<j\in V} u_i^T w u_j + \sum_{e\in E} A_e \log \sum_{i<j\in e} u_i^T w u_j \quad . \tag{3.11}$$

3.3.2 Modeling attribute information

To model node attributes, we assume that the community memberships u regulate the attributes on nodes. To allow this, we introduce a parameter β, which regulates the contribution of an attribute z to community k. Thus, β is a $K \times Z$ non-negative matrix. Conceptually, the role of the parameter β for the attributes matrix X is similar to the role of the matrix w for the hyperedge weights vector A. We model the probability of observing an attribute z on a node i assuming an underlying Bernoulli distribution

$$\pi_{iz} = \sum_{k=1}^{K} u_{ik}\beta_{kz} \quad . \tag{3.12}$$

It is further assumed that the attributes are conditionally independent given the parameter π_{iz}. Therefore, the likelihood of observing the covariate matrix X is the product of the following probabilities

$$P_X(X|u,\beta) = \prod_{i=1}^{N}\prod_{z=1}^{Z} \pi_{iz}^{x_{iz}}(1-\pi_{iz})^{(1-x_{iz})} \quad . \tag{3.13}$$

To ensure valid values for the probabilities, i.e. $\pi_{iz} \in [0,1]$ we introduce constrains $u_{ik} \in [0,1], \forall i,k$ and $\sum_{k=1}^{K}\beta_{kz} = 1, \forall z$. After introducing these constraints, although the underlying assumptions are the same as in the Hy-MMSBM model, the final model is different, as in our case u_{ik} is constrained and represents the probability of a node i to belong to the commu-

nity k.

This model allows inference using discrete and unordered attributes. Moreover, our definition of X allows stacking several discrete one-hot encoded attributes together, allowing for several attributes per node. Formally, the dimension Z of the matrix X can be represented as $Z = \sum_{i=p}^{P} z_p$, where P is a total number of unique attributes and z_p is the number of discrete values an attribute p can take. One can also consider modeling categorical attributes using only one excluding attribute per node, i.e. each node can only be assigned one attribute value. This formulation is discussed in section 3.4.

Another way to model different attributes would be adding new terms P_X to the total likelihood, with P_X encoding different types of distribution, e.g. Gaussian, Gamma, etc... This, however, may increase the computational complexity of the model and may make the analytical derivations intractable. Thus, we do not explore this approach further.

3.3.3 Inference of latent variables

Given the probabilistic model in Equation 3.7 and having defined both of the underlying distributions in Equation 3.8 and Equation 3.13, we aim at inferring the set of latent variables $\{u, w, \beta\}$ to maximize the probability of observing both hyperedge weights A and attributes matrix X. The standard approach of taking the logarithm of joint likelihood would lead to the sum of the respective parts as follows:

$$\log P(A, X \mid \theta) = \log P_A(A \mid \theta_A) + \log P_X(X \mid \theta_X) \,. \tag{3.14}$$

In practice, however, it has been shown that the performance of the model might improve if the contributions of the parts are properly balanced [27, 26, 31]. For this purpose, we introduce a balancing parameter $\gamma \in [0, 1]$ which balances the contribution of the network part and the attributes part yielding the following log-likelihood

$$L(A, X \mid \theta) = (1 - \gamma) L_A(A \mid \theta_A) + \gamma L_X(X \mid \theta_X) \,. \tag{3.15}$$

As the value of γ cannot be known a priori, it can be learned using the standard techniques for hyperparameter tuning. In our experiments, we use cross-validation, as also done for the selection of K.

Another reason for introducing the γ hyperparameter is that in our case the hypergraph part is relatively larger in scale than the attributes part and the contribution of attributes is small if

γ is not tuned. It is important to highlight that the value of γ can be clearly interpreted only in extreme cases when $\gamma = 1$ meaning only attributes are used and $\gamma = 0$ meaning only the network part is used. As the hypergraph part is generally larger, using cross-validation often yields high values of γ, i.e. $\gamma = 0.99$ to compensate for the difference, but this does not mean that the network part is barely used.

Variational lower bound

Taking the logarithm of the partial likelihoods in Equation 3.8 and Equation 3.13 we get the following expression for the total log-likelihood

$$L(A, X|\theta) = -C \sum_{i<j\in V} u_i^T w u_j + \sum_{e\in E} A_e \log \sum_{i<j\in e} u_i^T w u_j + \sum_{i=1}^{N}\sum_{z=1}^{Z} x_{iz} \log\left(\sum_{k=1}^{K} u_{ik}\beta_{kz}\right) + \sum_{i=1}^{N}\sum_{z=1}^{Z}(1-x_{iz}) \log\left(\sum_{k=1}^{K}(1-u_{ik})\beta_{kz}\right) \quad . \tag{3.16}$$

We use a standard variational approach to derive a lower bound for the summation terms inside the logarithm in Equation 3.16. Introducing the probability distributions $\rho^{(e)}_{ijkl}$, h_{izk} and h'_{izk} and using Jensen's inequality $\log \mathbb{E}[x] \geq \mathbb{E}[\log x]$, we get the following lower bounds:

$$\sum_{e\in E} A_e \sum_{i<j\in e} \log \sum_{k,q=1}^{K} \left(u_{ik}u_{jq}w_{kq}\right) \geq \sum_{e\in E} A_e \sum_{i<j\in e}\sum_{k,q=1}^{K} \rho^{(e)}_{ijkq} \log\left(\frac{u_{ik}u_{jq}w_{kq}}{\rho^{(e)}_{ijkq}}\right) ; \tag{3.17}$$

$$\sum_{i=1}^{N}\sum_{z=1}^{Z} x_{iz} \log\left(\sum_{k=1}^{K} u_{ik}\beta_{kz}\right) \geq \sum_{i=1}^{N}\sum_{z=1}^{Z} x_{iz} \sum_{k=1}^{K} h_{izk} \log\left(\frac{u_{ik}\beta_{kz}}{h_{izk}}\right) ; \tag{3.18}$$

$$\sum_{i=1}^{N}\sum_{z=1}^{Z}(1-x_{iz}) \log\left(\sum_{k=1}^{K}(1-u_{ik})\beta_{kz}\right) \geq \sum_{i=1}^{N}\sum_{z=1}^{Z}(1-x_{iz}) \sum_{k=1}^{K} h'_{izk} \log\left(\frac{(1-u_{ik})\beta_{kz}}{h'_{izk}}\right) ; \tag{3.19}$$

with equality reached when

$$\rho^{(e)}_{ijkq} = \frac{u_{ik}u_{jq}w_{kq}}{\lambda_e} ; \qquad h_{izk} = \frac{\beta_{kz}u_{ik}}{\sum_{k'}\beta_{k'z}u_{ik'}} ; \qquad h'_{izk} = \frac{\beta_{kz}(1-u_{ik})}{\sum_{k'}\beta_{k'z}(1-u_{ik'})} \tag{3.20}$$

respectively. To derive Equation 3.19 we used the fact that $u_{ik} \in [0,1]$ and

$$1 - \pi_{iz} = 1 - \sum_{k=1}^{K} \beta_{kz} u_{ik} = \sum_{k=1}^{K} (1 - u_{ik}) \beta_{kz} \quad .$$

This gives the lower bound on the total log-likelihood as

$$\begin{aligned} \mathcal{L}(A, X|\theta) := -C \sum_{i<j \in V} u_i^T w u_j + \sum_{e \in E} A_e \sum_{i<j \in e} \sum_{k,q=1}^{K} \rho_{ijkq}^{(e)} \log \left(\frac{u_{ik} u_{jq} w_{kq}}{\rho_{ijkq}^{(e)}} \right) \\ + \sum_{i=1}^{N} \sum_{z=1}^{Z} x_{iz} \sum_{k=1}^{K} h_{izk} \log \left(\frac{u_{ik} \beta_{kz}}{h_{izk}} \right) + \sum_{i=1}^{N} \sum_{z=1}^{Z} (1 - x_{iz}) \sum_{k=1}^{K} h'_{izk} \log \left(\frac{(1 - u_{ik}) \beta_{kz}}{h'_{izk}} \right) . \end{aligned} \tag{3.21}$$

Optimization procedure

To ensure the constrains $u_{ik} \in [0,1], \forall i, k$ and $\sum_{k=1}^{K} \beta_{kz} = 1, \forall z$ are satisfied, we introduce Lagrange multipliers $\lambda^{(\beta)}$ and $\lambda^{(u)}$ and obtain the following objective

$$\mathcal{L}_{constr} := \mathcal{L} - \sum_{z=1}^{Z} \lambda_z^{(\beta)} \left(\sum_{k=1}^{K} \beta_{kz} - 1 \right) - \sum_{i=1}^{N} \sum_{k=1}^{K} \lambda_{ik}^{(u)} u_{ik} \, . \tag{3.22}$$

To maximize this function, we use the Expectation-Maximization algorithm [32]. The algorithm alternates between an expectation step updating the values of $\rho_{ijkl}^{(e)}$, h_{izk} and h'_{izk} while keeping θ fixed, and a maximization step that maximizes the lower bound with respect to θ keeping $\rho_{ijkl}^{(e)}$, h_{izk} and h'_{izk} fixed. The procedure is described in detail in Algorithm 3.1. The updates for the variational parameters are given in Equations 3.20.

To derive updates for each parameter in θ, we set the derivative of $\mathcal{L}_{constr}$ with respect to each parameter to 0 and solve for it. The updates for w are straightforward to derive and are the same as the Hy-MMSBM

$$w_{kq} = \frac{\sum_{e \in E} A_e \sum_{i<j \in e} \rho_{ijkq}^{(e)}}{C \sum_{i<j \in V} u_{ik} u_{jq}} \quad , \tag{3.23}$$

which is valid when $\gamma \neq 1$.

Algorithm 3.1 HyCoSBM: EM algorithm

Inputs: hypergraph A, attributes matrix X, hyperparameters γ and K
Outputs: inferred (u, w, β)

$u, w, \beta \leftarrow init(u, w, \beta)$: `Randomly initialize the parameters`
while `convergence not reached`
  $\rho, h, h' \leftarrow update(\rho, h, h')$ ▷ Equation 3.20
  $u \leftarrow update(u)$ ▷ Solving Equation 3.26
  if $\gamma \neq 1$
    $w \leftarrow update(w)$ ▷ Equation 3.23
  end if
  if $\gamma \neq 0$
    $\beta \leftarrow update(\beta)$ ▷ Equation 3.24
  end if
end while

Solving for β updates we get the following expression which includes the Lagrange multiplier

$$\frac{\partial \mathcal{L}}{\partial \beta_{kz}} = \gamma \frac{\sum_{i=1}^{N} x_{iz} h_{izk}}{\beta_{kz}} + \gamma \frac{\sum_{i=1}^{N} (1 - x_{iz}) h'_{izk}}{\beta_{kz}} - \lambda_z^{(\beta)} = 0$$
$$\beta_{kz} = \frac{1}{\lambda_z^{(\beta)}} \gamma \left(\sum_{i=1}^{N} x_{iz} h_{izk} + \sum_{i=1}^{N} (1 - x_{iz}) h'_{izk} \right) \quad .$$

By imposing the constraint $\sum_{k=1}^{K} \beta_{kz} = 1$ we get the following update when $\gamma \neq 0$.

$$\frac{1}{\lambda_z^{(\beta)}} \gamma \sum_{i=1}^{N} \sum_{k=1}^{K} (x_{iz} h_{izk} + (1 - x_{iz}) h'_{izk}) = 1$$
$$\lambda_k^{(\beta)} = \gamma \sum_{i=1}^{N} \sum_{k=1}^{K} (x_{iz} h_{izk} + (1 - x_{iz}) h'_{izk})$$
$$\beta_{kz} = \frac{\sum_i (x_{iz} h_{izk} + (1 - x_{iz}) h'_{izk})}{\sum_{i,k'} (x_{iz} h_{izk'} + (1 - x_{iz}) h'_{izk'})} \quad . \tag{3.24}$$

Finally, we derive the update for u, which is the most complex derivation as it includes both the hypergraph part and the attributes part. Setting the derivative of $\mathcal{L}_{constr}$ with respect to u to

0 we get the following equation

$$\frac{\partial \mathcal{L}}{\partial u_{ik}} = (1-\gamma)\left[-C\sum_{j\in V, j\neq i}\sum_{q=1}^{K} u_{jq}w_{kq} + \sum_{e\in E: i\in e} A_e \sum_{j\neq i\in e}\sum_{q=1}^{K}\rho_{ijkq}^{(e)}\frac{1}{u_{ik}}\right] + \\ + \gamma\left[\frac{\sum_{z=1}^{Z} x_{iz}h_{izk}}{u_{ik}} - \frac{\sum_{z=1}^{Z}(1-x_{iz})h'_{izk}}{1-u_{ik}}\right] - \lambda_{ik}^{(u)} = 0$$

By setting

$$a_{ik} := (1-\gamma)\, C \sum_{j\in V, j\neq i}\sum_{q=1}^{K} u_{jq}w_{kq} + \lambda_{ik}^{(u)} \quad ,$$

$$b_{ik} := (1-\gamma) \sum_{e\in E: i\in e} A_e \sum_{j\neq i\in e}\sum_{q=1}^{K}\rho_{ijkq}^{(e)} + \gamma\sum_{z=1}^{Z} x_{iz}h_{izk} \quad ,$$

$$c_{ik} := \gamma\sum_{z=1}^{Z}(1-x_{iz})h'_{izk} \quad .$$

we get the following equation to find the updates of u

$$\frac{\partial \mathcal{L}}{\partial u_{ik}} = -a_{ik} + \frac{b_{ik}}{u_{ik}} - \frac{c_{ik}}{1-u_{ik}} = 0 \tag{3.25}$$

$$a_{ik}u_{ik}^2 - (a_{ik}+b_{ik}+c_{ik})u_{ik} + b_{ik} = 0, \quad 0 < u_{ik} < 1 \tag{3.26}$$

As all the terms in the quadratic equation in 3.26 are non-negative, it can be easily verified that the equation has two distinct and real solutions. Moreover, we can see that when $\gamma \neq 0$ and $\gamma \neq 1$, the only viable solution to the Equation 3.26 is the smallest root of the equation which is guaranteed to be in the allowed range $(0,1)$. The largest root is always larger than 1, thus we do not consider it here. Therefore, the Lagrange multiplier $\lambda_{ik}^{(u)}$ is equal to 0 in this case.

When $\gamma = 1$ the equation simplifies to

$$u_{ik} = \frac{\sum_{z=1}^{Z} x_{iz}h_{izk}}{\sum_{z=1}^{Z} x_{iz}h_{izk} + \sum_{z=1}^{Z}(1-x_{iz})h'_{izk}} \quad , \tag{3.27}$$

which is also guaranteed to be within $(0,1)$, thus $\lambda_{ik}^{(u)} = 0$ also in this case.

When $\gamma = 0$, the update is the same as with the Hy-MMSBM except for the Lagrange multiplier

$$u_{ik} = \frac{\sum_{e \in E: i \in e} A_e \sum_{j \neq i \in e} \sum_{q=1}^{K} \rho_{ijkq}^{(e)}}{C \sum_{j \in V, j \neq i} \sum_{q=1}^{K} u_{jq} w_{kq} + \lambda_{ik}^{(u)}}, \tag{3.28}$$

which is activated to make $u_{ik} = 1$ whenever the value is exceeded. Therefore, our model can be similarly powerful to the Hy-MMSBM when attributes are not used, but yielding a higher γ with cross-validation would always guarantee improvement.

The EM algorithm is not guaranteed to converge to the global optimum. Thus, in practice, we run the Algorithm 3.1 several times (20 times in our experiments) with different random initializations and choose the result with the highest log-likelihood.

3.3.4 Implementation and complexity

The implementation of HyCoSBM algorithm is based on Hy-MMSBM as the main building blocks of the update formulas for u and w are the same. The algorithm scales favorably with respect to both the number of nodes N and the number of hyperedges $|E|$ due to the efficient implementation using sparse binary incidence matrix $B \in \{0,1\}^{N \times |E|}$ to represent a hypergraph. Overall, the complexity is $O\Big(K(K+Z)(N+|E|)\Big)$ with `numpy` package implementation in Python programming language. The implementation is available at github.com/badalyananna/HyCoSBM.

3.4 Alternative formulations to model attributes

In section 3.3, we described the model that allows multiple values for each type of the attribute assuming their distribution is Bernoulli. Alternatively, one may want to allow exclusive attributes, e.g. age, where only one value can be assigned to each node at a time. While the HyCoSBM still allows for this and is more flexible to allow other formulations, we also considered modeling attributes using a multinomial distribution, thus we call the model *Multinomial attributes model*. Here, we present the main steps of this approach and outline the main reasons why we preferred the modeling approach described in section 3.3.

3.4.1 Multinomial attributes model

In this case, X is a $N \times Z$ matrix consisting of $\{0,1\}$ s.t. $\sum_{z=1}^{Z} X_{iz} = 1,\ \forall i$ where Z is the number of discrete values of the attribute. It is assumed that each entry of the matrix X is extracted from a Multinomial distribution with parameter $\pi_{iz} = \sum_{k=1}^{K} u_{ik}\beta_{kz}$. Thus, the likelihood of observing the matrix X becomes

$$P_X(X|u,\beta) = \prod_{i \in V} Mult\left(X_i; \pi_i\right) \quad , \tag{3.29}$$

yielding the following log-likelihood of the attributes

$$L_X(u,\beta) = \sum_{i=1}^{N}\sum_{z=1}^{Z} x_{iz}\log(\pi_{iz}) = \sum_{i=1}^{N}\sum_{z=1}^{Z} x_{iz}\log\left(\sum_{k=1}^{K}\beta_{kz}u_{ik}\right). \tag{3.30}$$

Introducing a probability distribution $h_{izk} = \frac{\beta_{kz}u_{ik}}{\sum_{k'}\beta_{k'z}u_{k'z}}$ and using a standard variational approach yields that maximizing the log-likelihood in Equation 3.30 is equivalent to maximizing

$$\mathcal{L}_X(u,\beta,h) = \sum_{i,z,k} x_{iz}\left[h_{izk}\log(\beta_{kz}u_{ik}) - h_{izk}\log(h_{izk})\right] \quad . \tag{3.31}$$

We need to enforce the following constraints on the parameters. For the community-attribute interactions we need $\sum_{z=1}^{Z}\beta_{kz} = 1, \forall k$, and for the community assignments $\sum_{k=1}^{K} u_{ik} = 1, \forall i$ and $u_{ik} > 0, \forall i, \forall k$. Therefore, we add Lagrange multipliers $\lambda = (\lambda^{(\beta)}, \lambda^{(u)}, \mu^{(u)})$. Adding the hypergraph part as described in subsection 3.3.2. and the Lagrange multipliers, we get the following objective

$$\begin{aligned}\mathcal{L}(U, W, \beta, \rho, h, \lambda) = (1-\gamma)&\left[-C\sum_{i<j\in V} u_i^T w u_j + \sum_{e\in E} A_e \sum_{i<j\in e}\sum_{k,q=1}^{K}\rho_{ijkq}^{(e)}\log\left(\frac{u_{ik}u_{jq}w_{kq}}{\rho_{ijkq}^{(e)}}\right)\right] \\ &+\gamma\left[\sum_{i,z,k} x_{iz}\left[h_{izk}\log(\beta_{kz}u_{ik}) - h_{izk}\log(h_{izk})\right]\right] \\ &-\sum_{k}\lambda_k^{(\beta)}\left(\sum_{z=1}^{Z}\beta_{kz} - 1\right) - \sum_{i}\lambda_i^{(u)}\left(\sum_{k=1}^{K}u_{ik} - 1\right) + \sum_{i,k}\mu_{ik}^{(u)}u_{ik}.\end{aligned} \tag{3.32}$$

We can see that now the constraint is not on the individual elements of u_{ik} but on the sum over the dimension k, which increases the complexity of subsequent derivations. In addition, there's also a positivity constraint on u_{ik}.

While the updates for w remain unchanged, and the updates for β simplify to

$$\beta_{kz} = \frac{\sum_i x_{iz} h_{izk}}{\sum_{i,z'} x_{iz'} h_{izk'}} \quad , \tag{3.33}$$

the updates of u_{ik} could not have been derived in closed form. Therefore, we experimented with 2 different approaches:

1. Solving for u_{ik} numerically.

2. Substituting w in the update equation for u to simplify subsequent derivations.

We now describe both of these approaches in detail.

Solving for u_{ik} numerically

The derivative of $\mathcal{L}$ in Equation 3.32 with respect to u_{ik} is as follows

$$\frac{\partial \mathcal{L}}{\partial u_{ik}} = \frac{1}{u_{ik}} \left[(1-\gamma) \sum_{e \in E: i \in e} A_e \sum_{j \neq i \in e} \sum_q \rho_{ijkq}^{(e)} + \gamma \sum_{z=1}^{Z} x_{iz} h_{izk} \right] -$$
$$- (1-\gamma) C \sum_{j \in V, j \neq i} \sum_{q=1}^{K} u_{jq} w_{kq} - \lambda_i^{(u)} + \mu_{ik}^{(u)}$$

and setting it to 0 yields

$$u_{ik} = \frac{(1-\gamma) \sum_{e \in E: i \in e} A_e \sum_{j \neq i \in e} \sum_q \rho_{ijkq}^{(e)} + \gamma \sum_z x_{iz} h_{izk}}{\lambda_i^{(u)} - \mu_{ik}^{(u)} + (1-\gamma) C \sum_{j \in V, j \neq i} \sum_{q=1}^{K} u_{jq} w_{kq}} \quad . \tag{3.34}$$

Given the constraints $\lambda_i^{(u)} - \mu_{ik}^{(u)}$ we cannot solve the u_{ik} update in closed from. To simplify the notation, we set

$$a_{ik} := (1-\gamma)C \sum_{j \in V, j \neq i} \sum_{q=1}^{K} u_{jq} w_{kq}$$
$$b_{ik} := (1-\gamma) \sum_{e \in E: i \in e} A_e \sum_{j \neq i \in e} \sum_{q} \rho_{ijkq}^{(e)} + \gamma \sum_{z} x_{iz} h_{izk} \quad ,$$

which gives

$$u_{ik} = \frac{b_{ik}}{\lambda_i^{(u)} - \mu_{ik}^{(u)} + a_{ik}} \tag{3.35}$$

To impose positivity constraint it is convenient to set $u_{ik} \geq \alpha$, where $\alpha \geq 0$ is an arbitrarily small constant. Applying the positivity constraint further, we have

$$\mu_{ik} = \begin{cases} -\lambda_i^{(u)} - a_{ik} + \frac{b_{ik}}{\alpha} & \text{if} \quad \lambda_i^{(u)} + a_{ik} \leq 0 \\ 0 & \text{if} \quad 0 < \lambda_i^{(u)} + a_{ik} \leq \frac{b_{ik}}{\alpha} \\ -\lambda_i^{(u)} - a_{ik} + \frac{b_{ik}}{\alpha} & \text{if} \quad \lambda_i^{(u)} + a_{ik} > \frac{b_{ik}}{\alpha} \quad . \end{cases} \tag{3.36}$$

This leads to the following function of u_{ik} in terms of $\lambda_i^{(u)}$ which we call $f_{ik}(\lambda_i^{(u)})$

$$u_{ik} = f_{ik}(\lambda_i^{(u)}) = \begin{cases} \alpha & \text{if} \quad \lambda_i^{(u)} + a_{ik} \leq 0 \\ \frac{b_{ik}}{\lambda_i^{(u)} + \alpha_{ik}} & \text{if} \quad 0 < \lambda_i^{(u)} + a_{ik} \leq \frac{b_{ik}}{\alpha} \\ \alpha & \text{if} \quad \lambda_i^{(u)} + a_{ik} > \frac{b_{ik}}{\alpha} \quad . \end{cases} \tag{3.37}$$

We can now impose the summation constraint as

$$\sum_{k=1}^{K} f_{ik}(\lambda_i^{(u)}) = 1 \iff \sum_{k=1}^{K} f_{ik}(\lambda_i^{(u)}) - 1 = 0 \tag{3.38}$$

Thus, we can solve for u_{ik} numerically by finding the roots of the Equation 3.38 using root-finding algorithms.

In practice, however, this method was not guaranteed to converge to the solution, which caused significant problems, especially for real datasets. Hence, we considered making some modifications to the objective function, which we discuss below.

Similarly to the ideas presented in [27], we substitute the value of w present in the first part of the total log-likelihood equation 3.32 with its update in Equation 3.23 as follows

$$
\begin{aligned}
-C \sum_{i<j\in V} u_i^T w u_j &= -C \sum_{i<j\in V} \sum_{k,q=1}^{K} u_{ik} u_{jq} w_{kq} \\
&= -C \sum_{i<j\in V} \sum_{k,q=1}^{K} u_{ik} u_{jq} \frac{\sum_{e\in E} A_e \rho_{kq}^{(e)}}{C \sum_{i<j\in V} u_{ik} u_{jq}} \\
&= -\sum_{k,q=1}^{K} \left(C \sum_{i<j\in V} u_{ik} u_{jq} \frac{\sum_{e\in E} A_e \rho_{kq}^{(e)}}{C \sum_{i<j\in V} u_{ik} u_{jq}} \right) \\
&= -\sum_{e\in E} A_e \sum_{k,q=1}^{K} \rho_{kq}^{(e)} \qquad\qquad \sum_{k,q=1}^{K} \rho_{kq}^{(e)} = 1 \\
&= -\sum_{e\in E} A_e
\end{aligned}
$$

The term becomes a sum of all hyperedge weights and can be treated as a constant. The derivative with respect to the u_{ik} simplifies to

$$
\frac{\partial \mathcal{L}}{\partial u_{ik}} = \frac{1}{u_{ik}} \left[(1-\gamma) \sum_{e\in E: i\in e} A_e \sum_{j\neq i\in e} \sum_{q=1}^{K} \rho_{ijkq}^{(e)} + \gamma \sum_{z=1}^{Z} x_{iz} h_{izk} \right] - \lambda_i^{(u)}
$$

Setting the derivative to 0 we get

$$
u_{ik} = \frac{1}{\lambda_i^{(u)}} \left[(1-\gamma) \sum_{e\in E: i\in e} A_e \sum_{j\neq i\in e} \sum_{q=1}^{K} \rho_{ijkq}^{(e)} + \gamma \sum_{z=1}^{Z} x_{iz} h_{izk} \right] \tag{3.39}
$$

By imposing the constraint $\sum_{k=1}^{K} u_{ik} = 1$ we get

$$\sum_{k=1}^{K} u_{ik} = \sum_{k=1}^{K} \frac{1}{\lambda_i^{(u)}} \left[(1-\gamma) \sum_{e \in E: i \in e} A_e \sum_{j \neq i \in e} \sum_{q=1}^{K} \rho_{ijkq}^{(e)} + \gamma \sum_{z=1}^{Z} x_{iz} h_{izk} \right]$$
$$= \frac{1}{\lambda_i^{(u)}} \left[(1-\gamma) \sum_{e \in E: i \in e} A_e \sum_{j \neq i \in e} \sum_{k=1}^{K} \sum_{q=1}^{K} \rho_{ijkq}^{(e)} + \gamma \sum_{z=1}^{Z} x_{iz} \sum_{k=1}^{K} h_{izk} \right] = 1$$

So the value of $\lambda_i^{(u)}$ is:

$$\lambda_i^{(u)} = (1-\gamma) \sum_{e \in E: i \in e} A_e \sum_{j \neq i \in e} \sum_{k=1}^{K} \sum_{q=1}^{K} \rho_{ijkq}^{(e)} + \gamma \sum_{z=1}^{Z} x_{iz} \sum_{k=1}^{K} h_{izk} \tag{3.40}$$

$$= (1-\gamma) \sum_{e \in E: i \in e} A_e \sum_{j \neq i \in e} \sum_{k=1}^{K} \sum_{q=1}^{K} \rho_{ijkq}^{(e)} + \gamma \tag{3.41}$$

as $\sum_{k=1}^{K} h_{izk} = 1$ and $\sum_{z=1}^{Z} x_{iz} = 1$. Plugging $\lambda_i^{(u)}$ back into the Equation 3.39 we get the update for u_{ik}

$$u_{ik} = \frac{(1-\gamma) \sum_{e \in E: i \in e} A_e \sum_{j \neq i \in e} \sum_{q=1}^{K} \rho_{ijkq}^{(e)} + \gamma \sum_{z=1}^{Z} x_{iz} h_{izk}}{(1-\gamma) \sum_{e \in E: i \in e} A_e \sum_{j \neq i \in e} \sum_{k=1}^{K} \sum_{q=1}^{K} \rho_{ijkq}^{(e)} + \gamma} \quad . \tag{3.42}$$

In practice, this method was unstable on synthetic datasets and that is why we have not fully resorted to it. Instead, we applied the updates without substituting w and resorted to the root finding technique as shown in the Equation 3.38 until the method stopped converging and then switched to the updates with substitution as defined in Equation 3.42.

This showed a poor performance on some real datasets, which was inferior to the model that did not use any attributes. The main reason for such behavior could be because the model where the entries of the membership matrix u are normalized to be equal to 1 does not take into account the degree of the nodes. We thus reformulated the model to apply degree correction, which we discuss below.

3.4.2 Multinomial attributes model with degree correction

Some nodes have more connections than other nodes in the network, which is expressed by the node degree. While other models discussed above took this information into account by varying the magnitude of the u_i vector for each node, normalizing the u_i vectors eliminated this benefit. Thus, keeping the community membership matrix u with constraint over the communities axis $\sum_{k=1}^{K} u_{ik} = 1, \forall i$, we introduce a vector $\varphi \in \mathbb{R}^N$, which takes into account the difference in node degree.

This changes only the parameters related to the network part. The parameter $\lambda^{(e)}$ which controls the Poisson distribution of hyperedge weights becomes

$$\lambda_e = \sum_{i<j:i,j\in e} \varphi_i \varphi_j u_i^T w u_j = \sum_{i<j:i,j\in e} \sum_{k,q=1}^{K} \varphi_i \varphi_j u_{ik} u_{jq} w_{kq} \quad . \tag{3.43}$$

We, therefore, have a new parameter φ to update. Doing the derivation as shown previously, we get the following updates for φ, u, w and ρ with β parameters unchanged

$$\varphi_i = \frac{\sum_{e\in E:i\in e} A_e \sum_{j\neq i\in e} \sum_{k,q=1}^{K} \rho_{ijkq}^{(e)}}{C \sum_{j\in V, j\neq i} \varphi_j \sum_{k,q=1}^{K} u_{ik} u_{jq} w_{kq}} \tag{3.44}$$

$$u_{ik} = \frac{(1-\gamma) \sum_{e\in E:i\in e} A_e \sum_{j\neq i\in e} \sum_q \rho_{ijkq}^{(e)} + \gamma \sum_z x_{iz} h_{izk}}{\lambda_i^{(u)} - \mu_{ik}^{(u)} + (1-\gamma) C \varphi_i \sum_{j\in V, j\neq i} \varphi_j \sum_{q=1}^{K} u_{jq} w_{kq}} \tag{3.45}$$

$$w_{kq} = \frac{\sum_{e\in E} A_e \sum_{i<j\in e} \rho_{ijkq}^{(e)}}{C \sum_{i<j\in V} \varphi_i \varphi_j u_{ik} u_{jq}} \tag{3.46}$$

$$\rho_{ijkq}^{(e)} = \frac{\varphi_i \varphi_j u_{ik} u_{jq} w_{kq}}{\sum_{i<j\in e} \sum_{k,q=1}^{K} \varphi_i \varphi_j u_{ik} u_{jq} w_{kq}} = \frac{\varphi_i \varphi_j u_{ik} u_{jq} w_{kq}}{\lambda_e} \tag{3.47}$$

Similarly to the Multinomial attributes model without degree correction, we can try solving for u_{ik} in Equation 3.45 using root finding techniques. However, we are faced with a similar problem of non-convergence and instability. Therefore, adopting the same techniques as for the model without degree correction and substituting w, we can simplify the u_{ik} updates as in Equation 3.42.

This model showed significant improvement over the model that didn't use the degree correction; however, it still remained unstable due to the complicated optimization procedure. Therefore, we resorted to the HyCoSBM because of its simplicity and stable results.

We must also note that our initial experiments with synthetically generated data contained an error in the data generation procedure, which resulted in unstable hypergraphs. Therefore, we discarded the model with degree correction that used the Equation 3.42 for the u_{ik} updates. Having discovered the error, we didn't replicate the experiments with the current model on synthetic data due to the lack of time. Considering the fact that on real datasets the model with degree correction performed similarly well to the HyCoSBM, it could still be a valid choice, and more research needs to be done in this direction. We discuss the experiments with the models presented in this section in detail in section 4.3.

4

Experimental results

To demonstrate the validity of our approach, we have carried out two types of experiments: experiments on synthetically generated data (section 4.1) and experiments on real datasets (section 4.2). In this chapter, we give a detailed description of the datasets used and analyze the performance of the HyCoSBM model comparing it to various baselines including the Hy-MMSBM model described in section 3.2. The section 4.3 discusses the experiments with alternative model formulations described in section 3.4.

4.1 Experiments on synthetic hypergraphs

This section is dedicated to the experiments with synthetically generated hypergraphs. First, we describe the process of data generation to create synthetic networks. We then proceed with comparing our model to the model that did not use any attributes in input as well as considering using only attributes for the community detection task. We show that our model provides a significant advantage compared to both baselines.

4.1.1 Synthetic data generation

We generate synthetic hypergraphs with a pre-defined community structure using the Hy-MMSBM based sampler presented in [33, 34]. We tuned the parameters of the sampling algorithms to generate networks where the inference with the Hy-MMSBM alone is non-trivial

in order to better assess the usage of attributes.

The parameters used for network generation were set as follows:

- Number of nodes $N = 500$
- Number of hyperedges $|E| = 2720$
- Number of communities $K = \{2, 3, 5, 10\}$.

In addition, we set a dimension sequence that specifies the number of hyperedges of each size as `dim_seq = {2: 300, 3: 300, 4: 200, 5: 200, 6: 150, 7: 150, 8: 150, 9: 150, 10: 120, 11: 120, 12: 120, 13: 120, 14: 100, 15: 100, 16: 100, 17: 100, 18: 80, 19: 80, 20: 80}`.

With the given parameters we created a membership matrix $u \in \mathbb{R}^{N \times K}$ and assigned a community to each node i uniformly at random so that $u_{ik} = 1$ if the node i belongs to the community k and $u_{ik} = 0$ otherwise. We generated networks with the assortative structure with the nodes in one group interacting only with each other and thus set an affinity matrix w to the identity matrix $I_K \in \mathbb{R}^{K \times K}$. Finally, we generated 10 random networks of each configuration. As the sampler is based on the Monte Carlo sampling technique, we increased the default number of burn-in steps to 100000 to ensure that the procedure converged. Each next sample was generated from the same generator by applying 1000 more burn-in steps.

We then proceeded to generating attributes matching the community structure. In our experiments, we set the number of communities equal to the number of attributes so that $K = Z$. The matrix X is then generated by first setting it equal to u and then randomly shuffling the proportion $1 - \rho$ of the attributes. The attributes were generated with ρ ranging in $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. As in the case of the network, we generated 10 random samples of attributes of each configuration.

4.1.2 Experimental setup

Our goal is to compare the performance of HyCoSBM with the baseline model Hy-MMSBM in the community detection task. Moreover, we want to show that HyCoSBM performs better than using attributes alone; that is, it is capable of efficiently using both structural and attribute information. For this reason, we aim to compare the attributes matrix X and community membership matrices obtained by HyCoSBM and Hy-MMSBM.

A natural choice in this case is using cosine similarity with the ground truth matrix u_{gt} used to generate the data. It is important to note that the communities in the inferred matrix u might

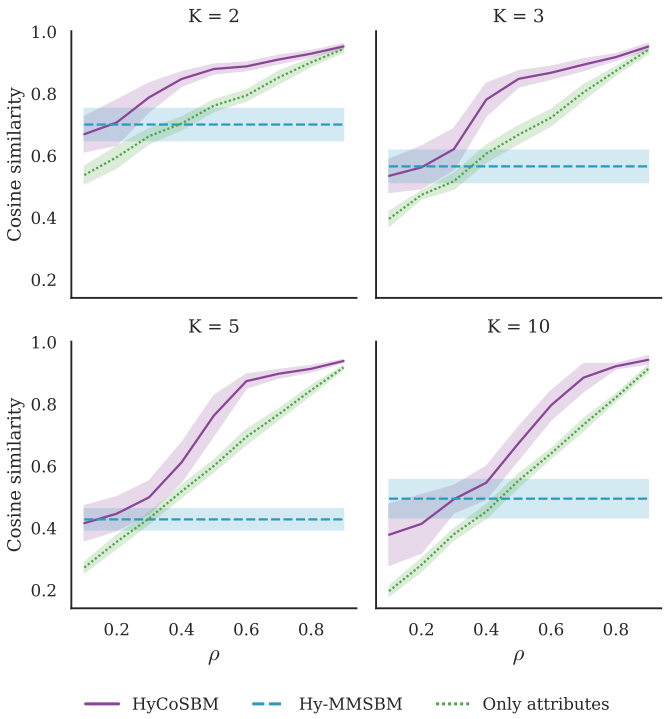

Figure 4.1: Community detection in synthetic hypergraphs. The graphs show a cosine similarity between the ground truth membership matrix and the membership matrices inferred by HyCoSBM and Hy-MMSBM algorithms in synthetic networks with $N = 500$ and $|E| = 2720$. The number of attributes Z is equal to the number of communities K. Hyperparameter γ is set equal to the proportion of non-shuffled attributes. The *Only attributes* line shows the cosine similarity between the attributes matrix X and the ground truth membership matrix u_{gt}.

be permuted not matching the u_{gt}. A simple solution to this problem is permuting the inferred matrix u until we get the best possible match with the u_{gt} and compute the cosine similarity between them.

The Hy-MMSBM model was run with each of the generated hypergraphs. The HyCoSBM model was run with each hypergraph attribute configuration pair, which resulted in a total of 100 runs per each ρ and K. The value of K in each experiment corresponded to the number of ground truth communities. The value of γ hyper-parameter was set equal to the proportion of non-shuffled attributes ρ.

4.1.3 Results

The results are summarized in Figure 4.1. It is clearly seen that when the attributes match the community structure by $\rho > 0.4$, the HyCoSBM model performs better than using only

network information or only attribute information. We can see that the gap is particularly large when the correspondence with communities is between $(0.4, 0.8)$, which shows the strength of our model in leveraging network and attribute information.

In cases when $\rho > 0.8$, we can see that as attributes already provide a good description of community structure, the advantage in using the network part decreases. Thus, while the HyCoSBM still performs better than *Only attributes*, the gap between them decreases.

When $\rho < 0.4$, the HyCoSBM performs slightly worse than the Hy-MMSBM, especially in cases when the gap between the attributes and the Hy-MMSBM is large. This shows that when attributes absolutely don't reflect the community structure, their usage provides no benefits, and in practice we can solve the issue by cross-validating the appropriate value of γ, as we show later with real hypergraphs.

Overall, the performance of HyCoSBM monotonically increases with the increase in the quality of the attribute information and always remains higher than using the attributes alone. It also remains higher than using only the network information when the attributes are informative. Thus, we can conclude that on synthetic datasets, the HyCoSBM model successfully leverages both network and attribute information to improve community detection.

4.2 Experiments on real hypergraphs

We have carried out experiments on various real attributed hypergraphs to show the advantages of using our approach in practice. This section is structured as follows: first, we describe the experimental setup with real datasets and then provide the analysis and results for each of the datasets used. The results are divided into two parts: one for the datasets and attributes where the use of the HyCoSBM model showed positive improvement and the other for those where the usage of the model didn't contribute positively to inference. The real datasets come from social, political, and biological domains to illustrate the wide applicability of our method. These datasets include:

- *Contact datasets* where hyperedges represent groups of people that were in close proximity to each other at some point in time in different settings. The results for these datasets are reported in subsections 4.2.2 and 4.2.3 as some of the attributes were informative while others were not.
- *Political datasets* show co-sponsorship of bills or co-participation in a committee by U.S. Congress members. The node attributes are political parties. As these attributes were uninformative, we report the results in a dedicated subsection 4.2.3

- *Gene Disease associations* where nodes of the hypergraph are genes, and the diseases are represented by hyperedges. The results are reported in subsection 4.2.2
- *Enron Email dataset* shows senders and receivers of emails in a hyperedge. The results are shown in subsection 4.2.2.

4.2.1 Experimental setup

Unlike the case of synthetic data, with real datasets we do not have access to ground truth. Thus, we evaluate the performance of various models on a hyperedge prediction task. We infer the model parameters using only a portion of the hyperedges and use the AUC metrics to evaluate the ability of the model to predict the held-out hyperedges.

Given a set of hyperedges, the AUC metric is computed by comparing the probabilities assigned by the model to a hyperedge present in the set with the probabilities assigned to a randomly generated hyperedge. For each hyperedge of a given size, we generate a hyperedge of the same size uniformly at random and compute their Poisson probabilities. These probabilities are then saved in a vector R_1 for the observed hyperedges and a vector R_0 for the randomly generated ones. To compute the AUC, we then compare these vectors as follows

$$AUC = \frac{\sum(R_1 > R_0) + 0.5\sum(R_1 == R_0)}{|R_1|} \quad ,$$

where $\sum(R_1 > R_0)$ is the number of times the probability of observing an existing hyperedge is higher than the probability of observing a randomly generated one according to our model, and $\sum(R_1 == R_0)$ is the number of times these probabilities are equal. $|R_1|$ is the total number of comparisons made, which is equal to the total number of hyperedges in the set. Therefore, higher values of AUC indicate better predicting capabilities of the model. We also note that due to the way the metric is scaled, the value near 0.5 is the lowest possible value, which means that the probabilities outputted by the model are equal to a random prediction.

We choose the model hyperparameters K and γ with 5-fold cross-validation. In our experiments, we varied the values of K from 2 to 30 and used the values of γ in $\{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.99, 0.995, 1.0\}$. We added high values of γ in order to better balance the contributions of the network and the attributes part, as the former usually significantly exceeds the latter.

The hyperedges are divided into train and test sets with the respective proportions of 80% and 20%. After the splitting, it might happen that some nodes present in the test set are not present in the train set anymore. This often happens with sparse datasets; therefore, in order to get fair results, we eliminate the nodes not present in the train hyperedges from the test set. Otherwise, the model that uses the attributes has a slight advantage compared to other models as it contains the information about these nodes that comes from attributes. For each cross-validation fold, the model is fit on the train set, and the AUC is computed on the test set. The best hyperparameters are chosen based on the average over 5 cross-validation folds. These values and their standard deviation are further reported as the best result.

To assess the improvement in using attributes with HyCoSBM, we compare the model with three baselines:

1. The Hy-MMSBM model defined in section 3.3. The model uses only the hypergraph part and thus serves as a valid baseline.

2. The HyCoSBM model with $\gamma = 0$. This is equivalent to not using any attributes. This case is different from the Hy-MMSBM model due to the constraints on the membership vectors u_i, s.t. $u_{ik} \in [0, 1], \forall k$.

3. Fixing community memberships u to be equal to the attributes matrix X and inferring only the affinity matrix w, which is equivalent to using only the attributes. In this case, the HyCoSBM and the Hy-MMSBM are exactly the same in theory; however, in practice, HyCoSBM is more numerically stable, so we used it in our experiments.*

We assume an assortative network structure and set the model parameter `assortative = True`, which initializes off-diagonal elements of the affinity matrix w to 0. This is done in all experiments except the Enron Email dataset, which presumes a core-periphery structure of the network. Due to the computational and time constraints, the range of K and γ in cross-validation was also reduced for the Enron Email dataset. Based on the previous results reported in [12], we cross validated only with $K = 2$ and $\gamma = \{0.5, 0.9\}$.

In addition, we measure the similarity of community partitions detected either with the attributes or with a partition obtained by another method. For this purpose, we compute the cosine similarity measure by averaging the cosine similarity of the membership vectors for each node $i \in V$. It is important to note that, unlike synthetically generated networks where the number of communities was equal to the number of attributes, in the case of real data, the

*The numerical stability comes from the fact that HyCoSBM was tuned to handle small values of u_{ik}, while Hy-MMSBM implementation does not consider these cases as the entries of u_{ik} do not reach values close to zero.

communities detected by one model may be different from another and also different from the number of attributes. Therefore, in this case, when we compare the cosine similarities, we pad the smaller vector with 0 until the sizes of the vectors are the same.

All the plots that show hypergraphs in this section were obtained by projecting a hypergraph into a dyadic graph using clique expansion. The plots were created by using the HGX library [34].

4.2.2 Performance with informative attributes

This subsection shows the results where the use of attributes with the HyCoSBM model showed improvement in the hyperedge prediction task measured by AUC.

Contact datasets

Contact datasets contain data about human close proximity interactions using the data obtained from wearable sensor devices. The hyperedges represent the people that were within a certain distance from each other at some point in time. We used four datasets of this type:

- *High School* shows the interactions of students in a high school. The students are nodes of the hypergraph and each has 4 attributes: class, sex, has facebook, and has compiled questionnaire;
- *Primary School* shows the interactions of students and teachers in a primary school. The attributes are class and sex;
- *Workplace* shows the interaction of co-workers in a workplace with attributes being the departments;
- *Hospital* shows the interactions of patients and staff in the hospital. The attributes called *status* show the role of the person in the hospital, such as a patient, a doctor, a nurse, etc.

It has been previously shown that the models that use only the structural information perform well on the hyperedge prediction task on these datasets [12, 20, 21]. Therefore, we vary the amount of the structural information available to the algorithm in order to estimate the performance in real-world scenarios when we do not have access to the complete data. This allows us to study how having information about attributes can compensate for the sparse structure of the network. Therefore, we eliminate an increasing fraction of the hyperedges until the hypergraph remains connected and perform 5-fold cross-validation on the remaining network.

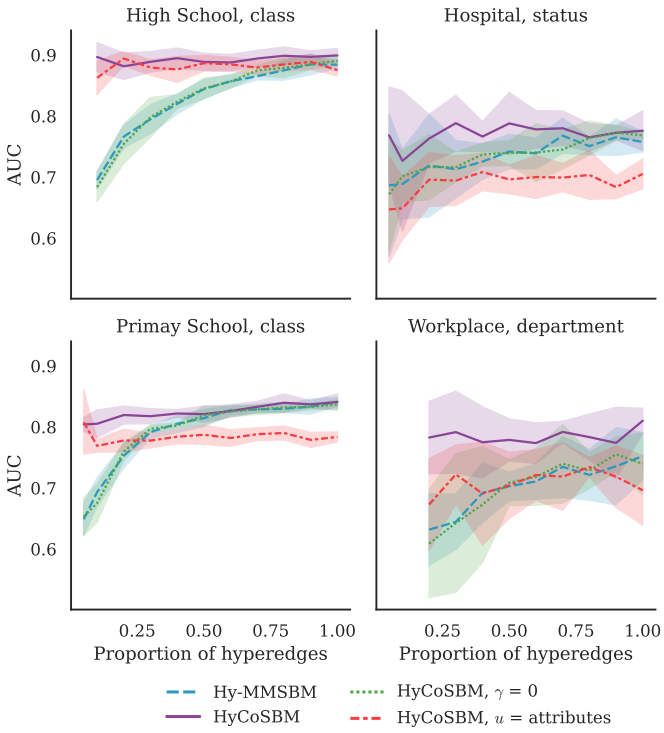

Figure 4.2: Hyperedge prediction in contact datasets with partial hyperedges. The graph charts illustrate the performance of HyCoSBM and three baselines in the hyperedge prediction task measured by the AUC. The performance of HyCoSBM that uses the attributes stays high, while the performance of the methods that do not use attributes drops as the availability of hyperedges declines.

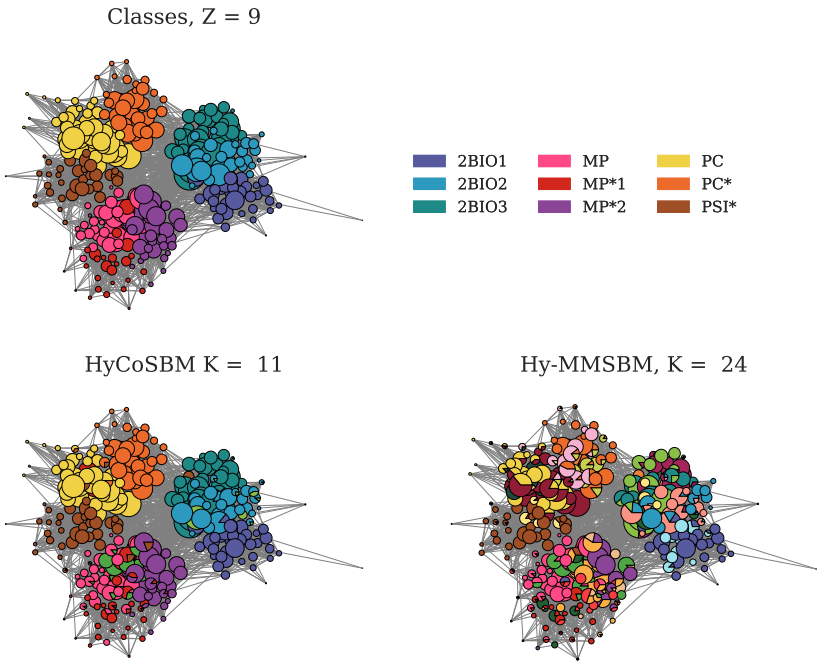

Figure 4.3: Communities detected in High School dataset with 100% of hyperedges used and the attribute Class. In the first row, we can see the classes used as attributes and their labels. In the second raw, the communities detected by HyCoSBM and Hy-MMSBM are shown.

The results of this experiment are illustrated in Figure 4.2. We can clearly see that while the performance of HyCoSBM stays at approximately the same level as the proportion of hyperedges available to the algorithm is being reduced, the performance of Hy-MMSBM and HyCoSBM with $\gamma = 0$ drops significantly. The difference is particularly large for High School and Primary School datasets where having only 10% of hyperedges, HyCoSBM outperforms Hy-MMSBM by approximately 0.17 and 0.11 points of the respective AUC scores.

Even in the case when all hyperedges are available, we can see that the AUC score of the HyCoSBM model is higher than the others, and it remains the highest also when the hyperedges are removed on Hospital and Workplace datasets. On Hospital dataset, the AUC score of HyCoSBM is larger than Hy-MMSBM by 0.02, and on Workplace, the difference is almost 0.06. Moreover, the value of $\gamma = 0.995$ in the later case indicates a strong use of the attributes by the model.

When the performance of HyCoSBM is similar to the models that do not use any attributes, the detected communities are nevertheless different. For example, in the High School dataset, the cosine similarity between the matrix X based on attribute class and the community mem-

bership matrix u inferred by HyCoSBM is 0.95 while the cosine similarity u inferred by Hy-MMSBM with the same attribute matrix is only 0.54. The communities detected by these models with all hyperedges available in the High School dataset are shown in Figure 4.3, where we can see that the communities inferred by HyCoSBM are almost analogous to the classes. In contrast, Hy-MMSBM finds a finer partition with a much larger number of communities K, which still performs similarly well in terms of AUC.

The difference in correlation with attributes between the communities detected by HyCoSBM and Hy-MMSBM together with the same AUC indicates the existence of competing network divisions as similar AUC means similar ability to explain network partitions. This has previously been observed in network datasets [35, 36, 37]. When using HyCoSBM with attributes, the partition is drawn closer to the one defined by the attributes.

It is important to note that although the communities detected by our model correlate with the attributes, they are not identical. As shown in Figure 4.4, HyCoSBM finds 5 communities and using 100% of hyperedges they look very similar to the attributes. We can notice, however, that some nodes have mixed memberships. Looking at the results of the model where community membership matrix u was set exactly to the attributes, we can see that the AUC is significantly smaller, which indicates that mixed memberships discovered by HyCoSBM contribute to the inference of better community structure. In High School dataset Figure 4.3, although the communities detected by HyCoSBM are extremely similar to the attributes, the number of communities is larger, 11 versus 9, which shows the existence of two smaller subgroups.

When the amount of structural information available decreases to 50% in Workplace dataset Figure 4.4, we can see that the communities are not as similar to the attributes, but the difference with Hy-MMSBM that doesn't use attributes is even more pronounced, which shows that HyCoSBM is more robust when data is partially available. Overall, while community detection with HyCoSBM is being guided by attributes, it produces superior results to the models that use only attributes or no attributes at all.

Gene Disease associations dataset

Our next application is to a biological domain. The gene disease associations dataset represents genes as nodes of a hypergraph and diseases as its hyperedges that show a combination of genes observed together with the disease. For each gene (node), we have its Disease Pleiotropy Index (DPI), which shows how likely the node is to be associated with several types of diseases. The index is a number between $(0, 1)$ with a higher index meaning a larger number of disease types associated with the gene. To use the index as an attribute, we consider each index value discrete

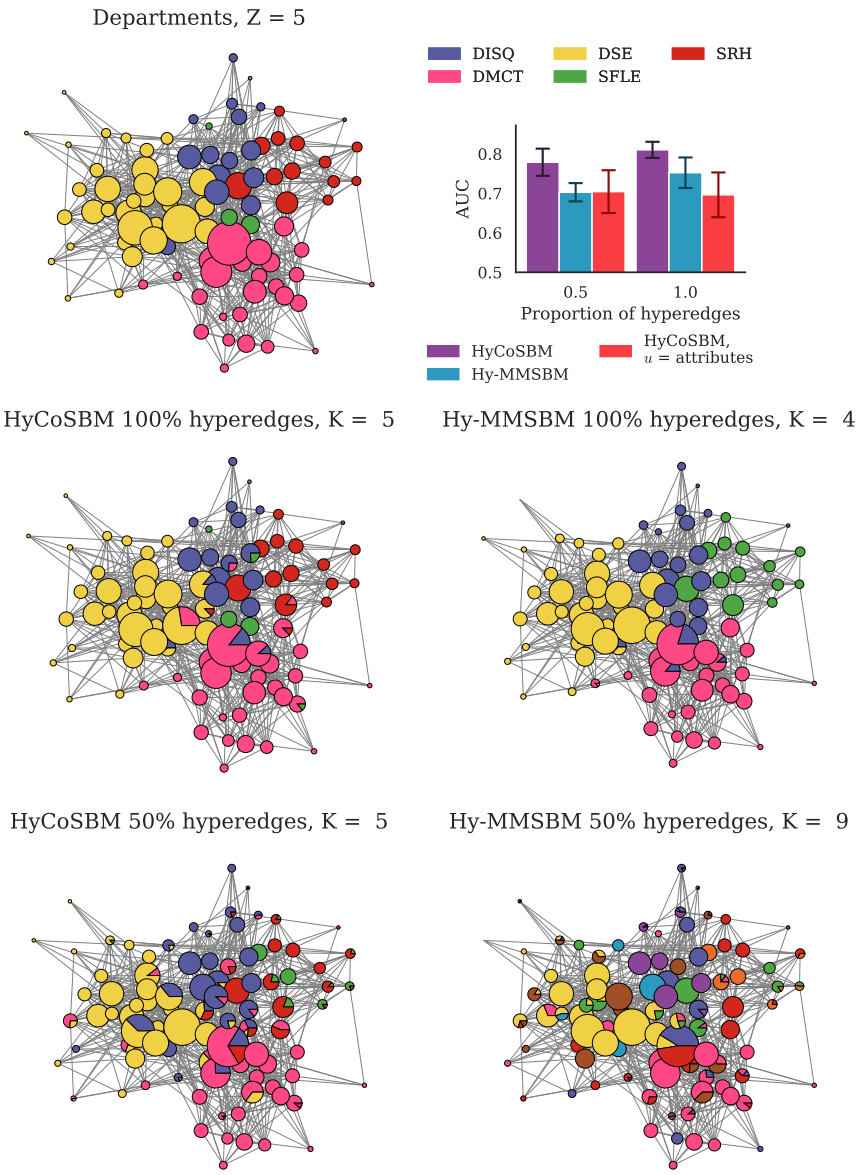

Figure 4.4: Communities detected in the Workplace dataset and respective AUC. The first row shows the attributes and their labels. On the left, we show the labels for each department and the AUC of the models shown below. In the second row, we see the communities detected by HyCoSBM and Hy-MMSBM using 100% of hyperedges, and in the third row, using 50% of the hyperedges.

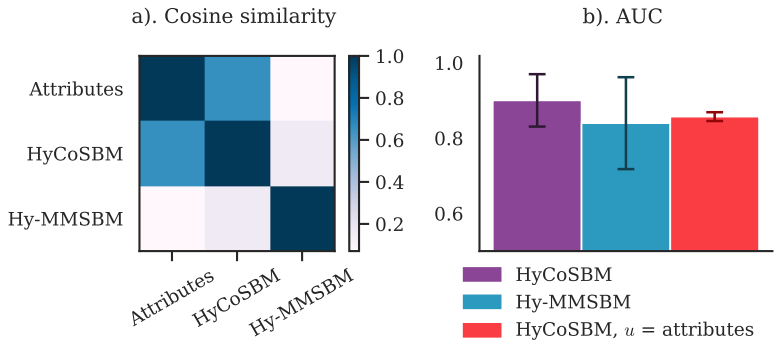

Figure 4.5: Cosine similarity and AUC in the Gene disease associations dataset. Part A shows the cosine similarity between the communities extracted from attributes (DPI) inferred by HyCoSBM and Hy-MMSBM. Part B shows the AUC achieved by HyCoSBM, Hy-MMSBM, and HyCoSBM with membership matrix u fixed to attributes.

and thus obtain $Z = 25$. The structure of the dataset is highly sparse as there are only 3128 hyperedges per 9262 nodes, and many nodes are present only in one hyperedge.

It has been previously shown that the hyperedge prediction on this dataset improves when using hyperedges of all sizes [12] rather than limiting the models to use only hyperedges up to size 25 [20]. Therefore, using all the hyperedges, we further assess if the inference of the network structure can improve with the use of attributes.

The AUC score achieved by HyCoSBM outnumbers the score achieved by Hy-MMSBM showing an increase from 0.84 to 0.90 as illustrated in Figure 4.5. This means that the attribute DPI is informative, and the usage of the attribute with HyCoSBM provides a significant improvement in the link prediction task. The cosine similarity of the community membership matrix u inferred by HyCoSBM with the attributes is also high (0.65). As in the case with the High School dataset, the communities inferred by HyCoSBM do not correspond exactly to the attributes, and the model finds finer divisions than the DPI attribute as $K = 30$ while $Z = 25$. In contrast, the cosine similarity of the communities inferred by Hy-MMSBM with the attribute matrix is near 0, which means the models are finding different partitions.

Enron Email dataset

Enron Email dataset shows employees in a company that send and receive emails. Nodes of the hypergraph are employees and hyperedges show the employees that appeared as a sender or a receiver in an email. This dataset is reported to have a core-periphery structure, which means that there are groups of central nodes called core that tend to interact with many other nodes and have a high degree. On the other hand, periphery nodes interact only with the core nodes.

The attribute of the dataset indicates if the node is a core or a periphery node.

While it may seem evident that giving the presumed network structure improves inference, our experiments have shown that HyCoSBM doesn't simply copy the membership matrix u but is capable of finding an affinity matrix w that matches the structure of the dataset. The AUC score on the hyperedge prediction task reached 0.987, which is a dramatic improvement over Hy-MMSBM with the AUC of 0.915. HyCoSBM with the membership matrix u fixed to attributes performed worse than HyCoSBM with $AUC = 0.951$.

We can see the membership matrices u and the affinity matrices w inferred by HyCoSBM and Hy-MMSBM in Figure 4.6. In the figure, we show HyCoSBM trained with $\gamma = 0.9$. It is clearly seen that while Hy-MMSBM divides the nodes into two groups and recovers the assortative structure of the network, HyCoSBM finds the core-periphery structure. Moreover, while all nodes belong to the community 0, the lighter blue nodes, which correspond to the core nodes, also have membership in the community 1. This is not visible because the respective u_{ik} values for those nodes are of order 10^{-6}. The matrix w shows that all nodes in group 0 interact mostly with this small portion of the group 1 and vice versa which is precisely the definition of a core-periphery structure.

We have also trained the HyCoSBM model with membership matrix u fixed to the attributes. The resulting affinity matrix w, which can be seen in the lower right corner in Figure 4.6, also shows the core-periphery structure. This matrix tells us that nodes in the core group intensively interact with themselves and only marginally with nodes in the periphery group. Periphery nodes do not interact with themselves and interact only with core nodes. While this structure also attains high values of AUC, HyCoSBM performs significantly better. This lets us conclude that even in cases when we have the information on the possible network structure, using this information as attributes helps our model to recover a different structure, which could not have been inferred otherwise.

4.2.3 Performance with uninformative attributes

We have observed in the previous subsection that the attributes contributed positively to the inference of the network structure. However, it cannot be expected that any type of attribute used by the model will help in improving inference. There can be cases when the attributes are weakly correlated with the community structure like in the example with synthetic experiments when the ρ was near 0.1. Therefore, in this section we show that when the attribute is uninformative we are able to detect this via cross-validation. We observed such uninforma-

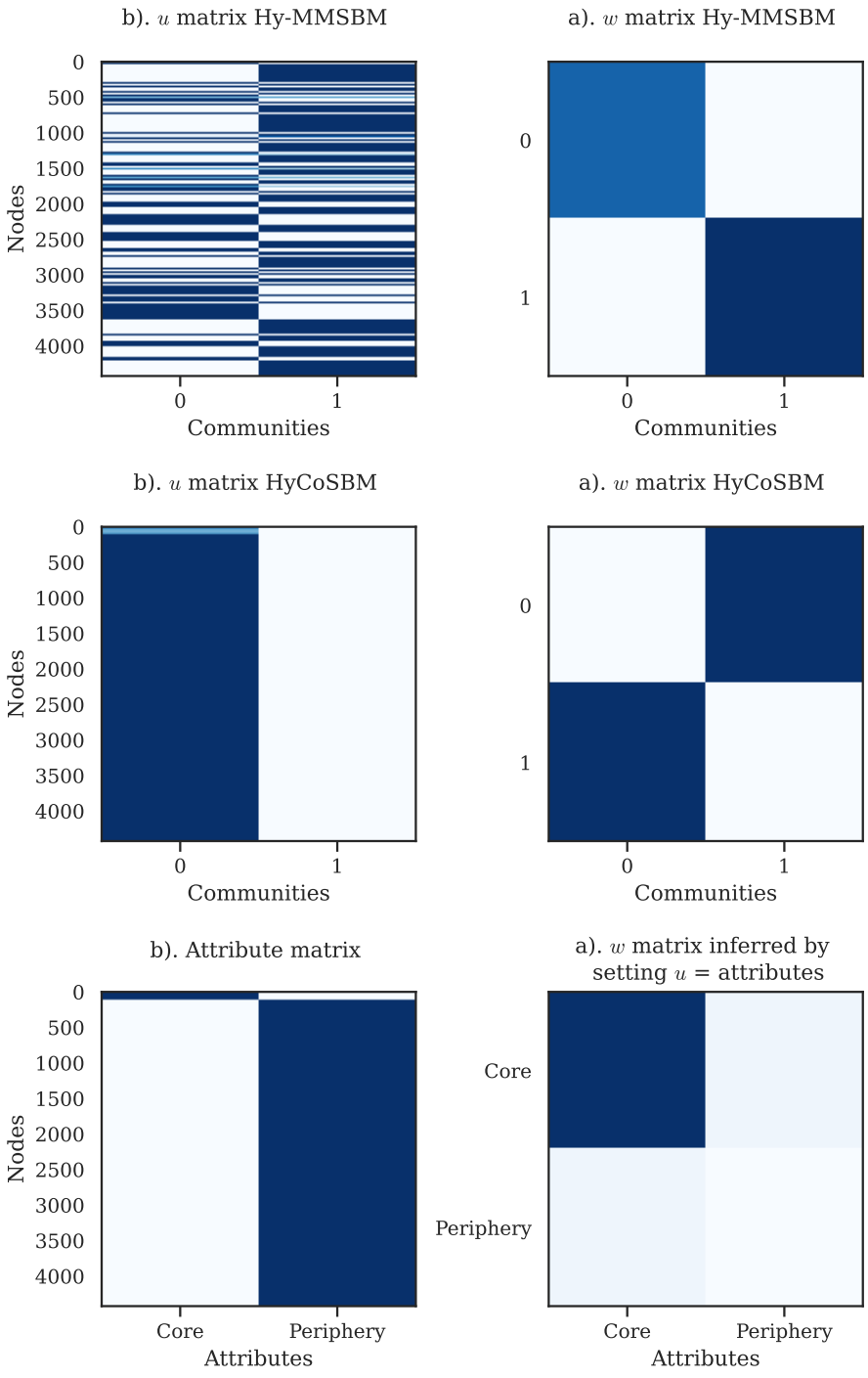

Figure 4.6: Inferred u and w parameters by HyCoSBM, HyMMSBM, and HyCoSBM with u = attributes on the Enron Email dataset. Both models have been trained with $K = 2$. HyCoSBM has been trained with $\gamma = 0.9$. The matrix u inferred by Hy-MMSBM has been normalized to sum to 1. The matrix u inferred by HyCoSBM contains small values for the community 1 which are not visible in the chart.

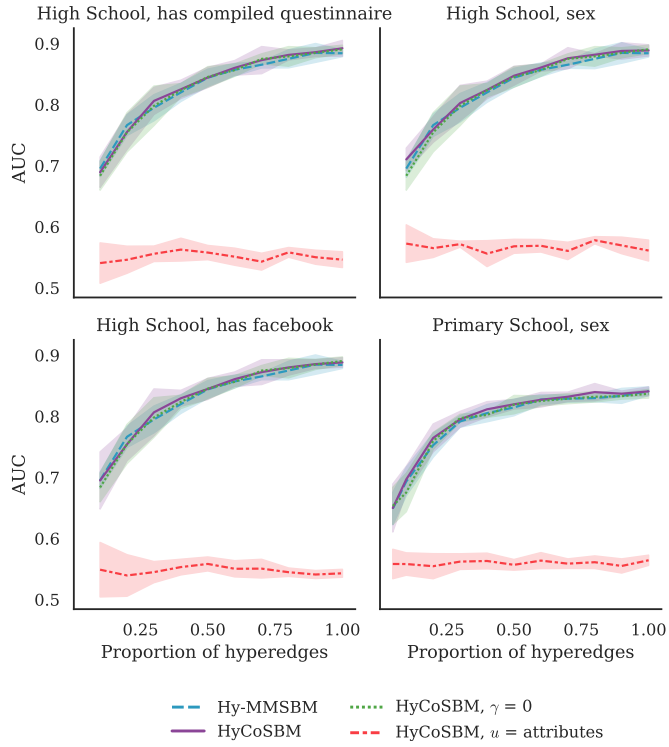

Figure 4.7: Hyperedge prediction in Contact datasets with partial hyperedges and uninformative attributes. The graph chart illustrates the AUC score of HyCoSBM and three baselines using uninformative attributes. The performance of HyCoSBM is analogous to that of other models.

tive attributes in *Contact* datasets of human interaction and *Political* datasets of co-voting and co-participation of the members of U.S. congress.

Contact datasets

We described these datasets in detail in subsection 4.2.2. We note here that the attributes presented in that section (class for High School and Primary School datasets or department for Workplace dataset) already represent a community structure. Thus, the fact that they proved informative was expected. Some of those datasets contain other attributes, and in this section, we demonstrate the behavior of HyCoSBM with those attributes.

The High School dataset contained additional attributes, namely *has compiled questionnaire*, *had facebook* and *sex*, and the Primary school dataset contained the attribute *sex*. All of these attributes are binary and contain only two possible values. Therefore, we did not expect them to be helpful in the community detection task.

The results illustrated in Figure 4.7 prove our initial hypothesis. It is clearly seen that when we use the attributes other than *class* as an input to our model, the performance of HyCoSBM stays the same as that of Hy-MMSBM and HyCoSBM with $\gamma = 0$ with the performance dropping as the number of hyperedges available decreases. Moreover, we can see that fixing the membership matrix u to the attributes matrix X results in the value of $AUC = 0.55$, which is near the random prediction value of 0.5. This further supports our claim that the attributes *has compiled questionnaire, had facebook* and *sex* are uninformative.

Political datasets

Political datasets include the following co-voting and co-participation datasets of U.S. congress members

1. *House Bills* is the dataset that shows co-voting in the House of Representatives. The nodes are the representatives, and the hyperedges show the representatives that voted for a particular initiative.
2. *House Committees* is the dataset that shows the participation of the members of the House of Representatives in the committee. The hyperedges show the representatives that participate together.
3. *Senate Bills* dataset shows the co-voting of senators in the U.S. Senate.
4. *Senate Committees* dataset shows the co-participation of senators in a committee.

The attribute in all four datasets is the political affiliation of a member, which can take two values: a Democratic party or a Republican party.

Dataset	HyCoSBM			Hy-MMSBM	
	K	γ	AUC	K	AUC
House Bills	22	0.0	0.952 ± 0.003	25	0.952 ± 0.001
House Committees	13	0.1	0.985 ± 0.015	24	0.972 ± 0.011
Senate Bills	23	0.0	0.929 ± 0.006	19	0.923 ± 0.003
Senate Committees	23	0.0	0.972 ± 0.01	21	0.963 ± 0.023

Table 4.1: AUC scores on co-voting and co-participation datasets of U.S. congress members by HyCoSBM and Hy-MMSBM. The results show the best average AUC and respective K and γ obtained via 5-fold cross-validation.

The results of performing a link prediction task on these datasets are summarized in the Table 4.1. We can clearly see that HyCoSBM performs almost identically to Hy-MMSBM,

and the best γ chosen by cross-validation is equal to 0 in three datasets out of four. Therefore, we conclude that the attribute is not conducive to explaining the co-voting and co-participation patterns in the U.S. Congress.

4.2.4 Summary

The results for all runs of experiments with real datasets, as well as the dataset statistics, are summarized in Table 4.2. Overall, we can see that, with almost all datasets, when HyCoSBM was used with informative attributes, the best cross-validation value for γ was higher than 0.5. The only exception is the Hospital dataset with the attribute status where the best γ selected via cross-validation was 0.2. However, even with the small γ, the performance on the hyperedge prediction task exceeded Hy-MMSBM.

The improvement achieved by the use of attributes is particularly pronounced on sparse hypergraphs. We have seen this with contact datasets with only 20% of hyperedges, Enron Email, and Gene Disease datasets. On hypergraphs with a large number of hyperedges, the network part is sufficient to achieve good results; however, using attributes helps to find partitions that are closer to the attributes, as shown on contact datasets with 100% of hyperedges.

4.3 Experiments with alternative models

In section section 3.4, we discussed alternative models to incorporate node attributes in hypergraph inference models. In this section, we show the experimental results achieved with those models that illustrate the reasons they were discarded.

In Figure 4.8, we compare the results of the experiments on Contact datasets with partial hyperedges between HyCoSBM, Hy-MMSBM, Multinomial attributes, and Multinomial attributes with degree correction models. We can see that the Multinomial model without degree correction performs worse than other models that use attributes. In particular, on the Hospital dataset it underperforms even compared to the Hy-MMSBM model which does not use any attribute information. We can see that the reason for such behavior is the weak performance of the underlying model for inference on hypergraphs, as in the case when $\gamma = 0$, the Multinomial models performed significantly worse than all other models. Therefore, while introducing attributes helped to substantially improve the performance of the model, it was not enough to compensate for the initial disadvantage. Similar behavior of the Multinomial model can be seen also on other datasets where it performed slightly worse than other methods with $\gamma = 0$.

Dataset	Attribute	N	$\lvert E\rvert$	Z	HyCoSBM			Hy-MMSBM		Source
					K	γ	AUC	K	AUC	
Enron Email	structure	4423	5743	2	3	0.900	0.987 ± 0.002	2	0.915 ± 0.004	[3]
Gene Disease	DPI	9262	3128	25	30	0.500	0.9 ± 0.07	2	0.84 ± 0.122	[38]
High School	class	327	7818	9	11	0.995	0.899 ± 0.011	24	0.884 ± 0.006	[39]
	has compiled questionnaire			2	21	0.800	0.892 ± 0.013			
	has facebook			2	15	0.950	0.888 ± 0.008			
	sex			2	16	0.800	0.889 ± 0.009			
Primary School	class	242	12704	11	10	0.600	0.841 ± 0.013	11	0.841 ± 0.007	[39]
	sex			2	12	0.200	0.841 ± 0.007			
Hospital	status	75	1825	4	2	0.200	0.776 ± 0.032	2	0.758 ± 0.016	[39]
Workplace	department	92	788	5	5	0.995	0.81 ± 0.02	5	0.752 ± 0.039	[39]
House Bills	political party	1494	54933	2	22	0.000	0.952 ± 0.003	25	0.952 ± 0.001	[40, 41]
House Committees	political party	1290	335	2	13	0.100	0.985 ± 0.015	24	0.972 ± 0.011	[42]
Senate Bills	political party	294	21721	2	23	0.000	0.929 ± 0.006	19	0.923 ± 0.003	[40, 41]
Senate Committes	political party	282	301	2	23	0.000	0.972 ± 0.01	21	0.963 ± 0.023	[42]

Table 4.2: AUC scores on real datasets. The reported AUC scores are an average and standard deviation over 5 cross-validation folds. The values of K, γ, and AUC correspond to the best cross-validation fold. In addition, dataset statistics on the number of nodes N, number of hyperedges $|E|$, and number of attributes Z are reported.

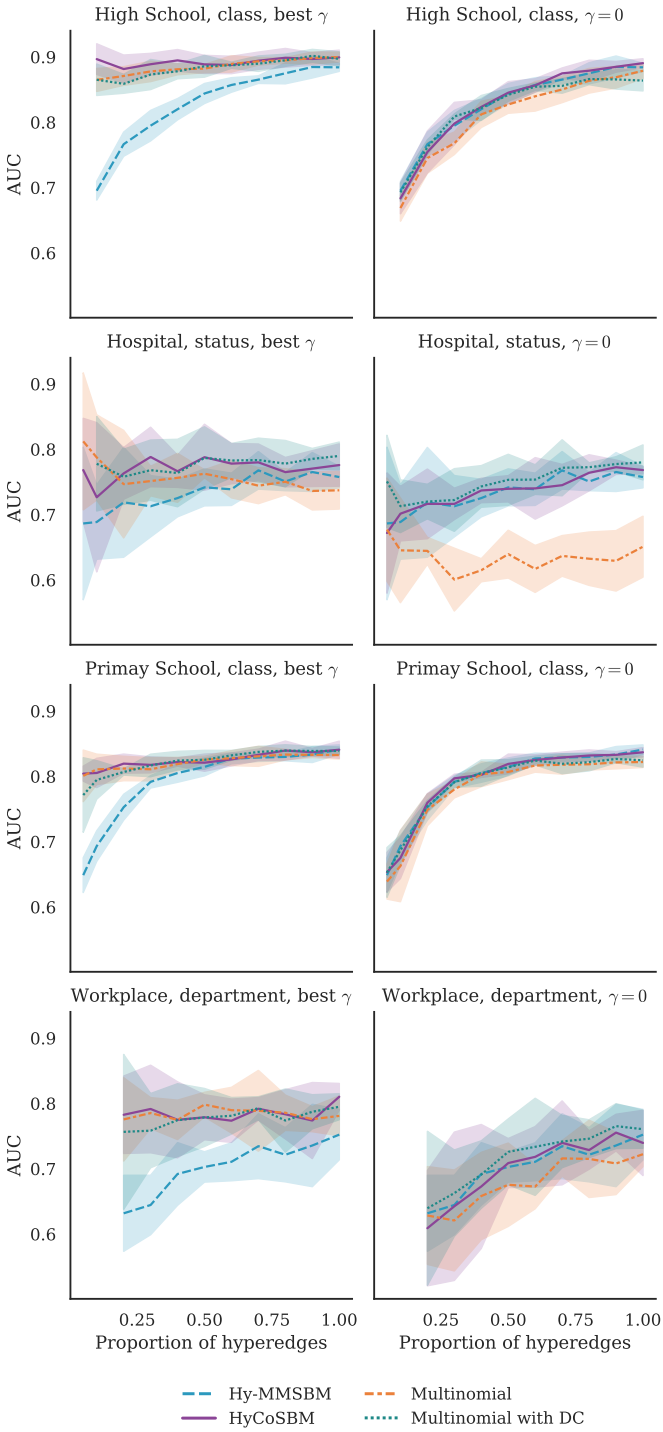

Figure 4.8: Hyperedge prediction in Contact datasets with partial hyperedges: comparison between HyCoSBM, Hy-MMSBM, Multinomial attributes and Multinomial attributes with degree correction models. The right side shows the results with the best gamma. The left side shows the result with $\gamma = 0$.

On the other hand, considering the Multinomial attributes model with degree correction we can see that this model performs similarly well to HyCoSBM both with the attributes and when $\gamma = 0$. The experiments with the Gene Disease dataset, however, uncover more surprising results. We recall that using the HyCoSBM model with DPI attribute we improved the AUC score on this dataset from 0.84 to 0.90. The Multinomial model with degree correction achieved the same AUC score as HyCoSBM, however, the best γ selected via cross-validation was equal to 0. This means that the Multinomial model without degree correction achieved the same improvement as HyCoSBM without using any attributes. This could be partially caused by the fact that conceptually the DPI index and the node degree are meant to represent a similar phenomenon, that is, how likely the gene (node) is to be associated with many types of diseases (be present in many hyperedges). Therefore, we can see that the base model for community detection with degree correction is more powerful than the base model used by HyCoSBM, which may sometimes make it difficult to identify if the improvement comes from the use of attributes or from the introduction of degree correction.

We can conclude that the degree correction can be a powerful extension of the hypergraph inference model. As the optimization procedure with the Multinomial model was not giving stable results, a good solution could be to apply degree correction to the HyCoSBM, but we have not explored this topic further. The main challenge here is that when we add a parameter φ to the membership matrix u, these two parameters compete with each other for the scale, which did not happen with the Multinomial model as the values of u_i were constrained to sum to 1. This may lead to a model that is not identifiable and stable as sometimes the values of the inferred parameters may be pushed to 0 causing numerical instability. Therefore, we leave this topic as a possible future research direction.

5 Conclusion

In this work, we presented the HyCoSBM model, a probabilistic generative model to study and analyze the structure of higher-order networks with the help of node attributes. The model is capable of recovering overlapping communities guided by the available attribute information. We have demonstrated that the model is able to efficiently incorporate both hypergraph and attribute information and find partitions that are more expressive when both types of information are combined.

The algorithm behind the HyCoSBM model is based on the Expectation Maximization method that maximizes the likelihood of observing both a hypergraph and attributes on its nodes. The likelihood of observing a hypergraph was modeled by adopting the ideas from the Hy-MMSBM model [12] and adding constraints on the parameter u. The likelihood of observing node attributes was modeled by assuming the Bernoulli distribution of the entries of the attribute matrix. The contribution of these two terms is regulated by the γ parameter. Other choices can be made for modeling both likelihoods differently from the examples we explored in detail in this thesis. However, these choices may impact analytical tractability and computational efficiency. The examples we considered tackle both these problems effectively and lead to the results discussed in this thesis. Other choices may instead limit the practical ability to implement the model on real datasets.

We have demonstrated that our approach provides an advantage compared to the models that do not use attributes during inference, such as the baseline model Hy-MMSBM and HyCoSBM with $\gamma = 0$. We have also shown that HyCoSBM does not simply copy the attributes

and performs better than using attributes alone with both synthetically generated and real data.

The HyCoSBM model was applied to four datasets in various domains, including social, political, and biological. We have shown that the model provided a particular advantage for social proximity datasets, a gene disease dataset, and a dataset with a clear core-periphery structure (Enron Email). In the cases where the model performed similarly well to the baselines that did not use attributes, we demonstrated that the model can be used to infer the community structure that is more similar to the attributes. It was also shown that as not all attributes are beneficial in improving our understanding of the networks' structure, our model was able to successfully discard such uninformative attributes in co-voting and co-participation political datasets.

In addition, we discussed possible alternative methods that can be used to model attributes. We have shown that restricting a model to have only one covariate and assuming their Multinomial distribution leads to difficulty in obtaining closed-form solutions for the updates. Moreover, even when we manage to obtain approximate solutions to the optimization problem, the inferred community structure is inferior to the models that did not use any attribute. The principal reason for such behavior is that the model that constrains the membership vectors to sum to 1 does not take into consideration node degrees.

We thus developed a degree corrected model that showed similar performance to HyCoSBM on real datasets but was unstable on synthetically generated hypergraphs. An interesting feature of this model is that, in some cases, it performs similarly well to HyCoSBM but without the use of attributes. We, therefore, conclude that degree correction was a powerful modification to the original model, and its usage with and without attributes should be explored further. The reason could be that allowing for more heterogeneity in the node-level parameters via an additional φ may provide similar benefits to those obtained using more information. In fact, there could be multiple local minima for the log-likelihood cost function considered in this probabilistic formulation. Hence, there could be multiple valid community partitions that explain the observed data similarly well. Using the degree information may help in considering valid alternative partitions than those obtained by using node attributes. However, to better substantiate this hypothesis, more experiments are needed. A more extensive investigation of the effects of degree correction is an interesting avenue for future work.

We have shown the case of the Enron Email dataset, where the attributes helped to recover the core-periphery structure of the network. We notice, however, that due to the time constraints, we could not have carried out a full-scale experiment with this dataset. Therefore, completing this experiment, as well as studying the role of attributes in networks with a struc-

ture different from assortative remains a possible future research direction.

Another possible direction of future research could be considering different types of attributes, such as numerical and modeling them using normal distribution. This would require efforts in deriving closed form updates and maintaining low computation complexity. Another possible extension is adding the attributes on hyperedges to the model. In addition, one could consider eliminating some assumptions that were made in the HyCoSBM model, such as the conditional independence of the hypergraph part and attributes part given the latent variables. This approach may be implemented either with the same latent variables or by introducing a different set. However, an efficient implementation of updates still remains a principal issue with this approach. Lastly, another extension to this model could be modeling multilayer hypergraphs with several types of connections represented as layers or dynamic hypergraphs where interactions between nodes change over time. It would be interesting to evaluate the role of attributes in these contexts.

References

[1] C. Berge and E. Minieka, *Graphs and hypergraphs / Claude Berge ; translated by Edward Minieka*, ser. North-Holland mathematical library. Amsterdam etc: North-Holland, 1973.

[2] X. Ouvrard, "Hypergraphs: an introduction and review," *arXiv preprint arXiv:2002.05014*, 2020.

[3] S. Klamt, U.-U. Haus, and F. Theis, "Hypergraphs and cellular networks," *PLoS computational biology*, vol. 5, no. 5, p. e1000385, 2009.

[4] M. M. Mayfield and D. B. Stouffer, "Higher-order interactions capture unexplained complexity in diverse communities," *Nature ecology & evolution*, vol. 1, no. 3, p. 0062, 2017.

[5] C. Giusti, R. Ghrist, and D. S. Bassett, "Two's company, three (or more) is a simplex: Algebraic-topological tools for understanding higher-order structure in neural data," *Journal of computational neuroscience*, vol. 41, pp. 1–14, 2016.

[6] G. Cencetti, F. Battiston, B. Lepri, and M. Karsai, "Temporal properties of higher-order interactions in social networks," *Scientific reports*, vol. 11, no. 1, p. 7028, 2021.

[7] A. Zimmer, I. Katzir, E. Dekel, A. E. Mayo, and U. Alon, "Prediction of multidimensional drug dose responses based on measurements of drug pairs," *Proceedings of the National Academy of Sciences*, vol. 113, no. 37, pp. 10 442–10 447, 2016.

[8] A. Badalyan, N. Ruggeri, and C. De Bacco, "Hypergraphs with node attributes: structure and inference," *arXiv preprint arXiv:2311.03857*, 2023.

[9] F. Battiston, G. Cencetti, I. Iacopini, V. Latora, M. Lucas, A. Patania, J.-G. Young, and G. Petri, "Networks beyond pairwise interactions: structure and dynamics," *Physics Reports*, vol. 874, pp. 1–92, 2020.

[10] S. Fortunato and D. Hric, "Community detection in networks: A user guide," *Physics reports*, vol. 659, pp. 1–44, 2016.

[11] Z. T. Ke, F. Shi, and D. Xia, "Community detection for hypergraph networks via regularized tensor power iteration," *arXiv preprint arXiv:1909.06503*, 2019.

[12] N. Ruggeri, M. Contisciani, F. Battiston, and C. De Bacco, "Community detection in large hypergraphs," *Science Advances*, vol. 9, no. 28, p. eadg9159, 2023.

[13] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[14] D. Zhou, J. Huang, and B. Schölkopf, "Learning with hypergraphs: Clustering, classification, and embedding," *Advances in neural information processing systems*, vol. 19, 2006.

[15] D. Ghoshdastidar and A. Dukkipati, "A provable generalized tensor spectral method for uniform hypergraph partitioning," in *International Conference on Machine Learning*. PMLR, 2015, pp. 400–409.

[16] M. C. Angelini, F. Caltagirone, F. Krzakala, and L. Zdeborová, "Spectral detection on sparse hypergraphs," in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2015, pp. 66–73.

[17] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[18] R. Vershynin, *High-dimensional probability: An introduction with applications in data science*. Cambridge university press, 2018, vol. 47.

[19] C. De Bacco, E. A. Power, D. B. Larremore, and C. Moore, "Community detection, link prediction, and layer interdependence in multilayer networks," *Physical Review E*, vol. 95, no. 4, p. 042317, 2017.

[20] M. Contisciani, F. Battiston, and C. De Bacco, "Inference of hyperedges and overlapping communities in hypergraphs," *Nature communications*, vol. 13, no. 1, p. 7229, 2022.

[21] P. S. Chodrow, N. Veldt, and A. R. Benson, "Generative hypergraph clustering: From blockmodels to modularity," *Science Advances*, vol. 7, no. 28, p. eabh1303, 2021.

[22] L. Brusa and C. Matias, "Model-based clustering in simple hypergraphs through a stochastic blockmodel," *arXiv preprint arXiv:2210.05983*, 2022.

[23] P. Chodrow and A. Mellor, "Annotated hypergraphs: models and applications," *Applied network science*, vol. 5, pp. 1–25, 2020.

[24] C. Bothorel, J. D. Cruz, M. Magnani, and B. Micenkova, "Clustering attributed graphs: models, measures and methods," *Network Science*, vol. 3, no. 3, pp. 408–444, 2015.

[25] P. Chunaev, "Community detection in node-attributed social networks: a survey," *Computer Science Review*, vol. 37, p. 100286, 2020.

[26] J. Yang, J. McAuley, and J. Leskovec, "Community detection in networks with node attributes," in *2013 IEEE 13th international conference on data mining*. IEEE, 2013, pp. 1151–1156.

[27] M. Contisciani, E. A. Power, and C. De Bacco, "Community detection with node attributes in multilayer networks," *Scientific reports*, vol. 10, no. 1, p. 15736, 2020.

[28] B. Fanseu Kamhoua, L. Zhang, K. Ma, J. Cheng, B. Li, and B. Han, "Hypergraph convolution based attributed hypergraph clustering," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 453–463.

[29] R. Du, B. Drake, and H. Park, "Hybrid clustering based on content and connection structure using joint nonnegative matrix factorization," *Journal of Global Optimization*, vol. 74, pp. 861–877, 2019.

[30] Y. Li, R. Yang, and J. Shi, "Efficient and effective attributed hypergraph clustering via k-nearest neighbor augmentation," *Proceedings of the ACM on Management of Data*, vol. 1, no. 2, pp. 1–23, 2023.

[31] O. Fajardo-Fontiveros, R. Guimerà, and M. Sales-Pardo, "Node metadata can produce predictability crossovers in network inference problems," *Physical Review X*, vol. 12, no. 1, p. 011010, 2022.

[32] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society: series B (methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[33] N. Ruggeri, F. Battiston, and C. De Bacco, "A framework to generate hypergraphs with community structure," *arXiv preprint arXiv:2212.08593*, vol. 22, 2023.

[34] Q. F. Lotito, M. Contisciani, C. De Bacco, L. Di Gaetano, L. Gallo, A. Montresor, F. Musciotto, N. Ruggeri, and F. Battiston, "Hypergraphx: a library for higher-order network analysis," *Journal of Complex Networks*, vol. 11, no. 3, p. cnad019, 2023.

[35] B. H. Good, Y.-A. De Montjoye, and A. Clauset, "Performance of modularity maximization in practical contexts," *Physical review E*, vol. 81, no. 4, p. 046106, 2010.

[36] L. Peel, D. B. Larremore, and A. Clauset, "The ground truth about metadata and community detection in networks," *Science advances*, vol. 3, no. 5, p. e1602548, 2017.

[37] M. E. Newman and A. Clauset, "Structure and inference in annotated networks," *Nature communications*, vol. 7, no. 1, p. 11863, 2016.

[38] J. Piñero, J. M. Ramírez-Anguita, J. Saüch-Pitarch, F. Ronzano, E. Centeno, F. Sanz, and L. I. Furlong, "The disgenet knowledge platform for disease genomics: 2019 update," *Nucleic acids research*, vol. 48, no. D1, pp. D845–D855, 2020.

[39] M. Génois and A. Barrat, "Can co-location be used as a proxy for face-to-face contacts?" *EPJ Data Science*, vol. 7, no. 1, pp. 1–18, 2018.

[40] J. H. Fowler, "Connecting the congress: A study of cosponsorship networks," *Political Analysis*, vol. 14, no. 4, pp. 456–487, 2006.

[41] ——, "Legislative cosponsorship networks in the us house and senate," *Social networks*, vol. 28, no. 4, pp. 454–465, 2006.

[42] C. Stewart III and J. Woon, "Congressional committee assignments, 103rd to 114th congresses, 1993–2017: House," MIT mimeo, Tech. Rep., 2008.

Acknowledgments

I would like to express my deepest gratitude to Dr. Caterina de Bacco for giving me the opportunity to experience the real research working on this thesis as well as encouraging me when the things did not work quite as expected. Biggest thanks to both Nicoló Ruggeri and Dr. Caterina de Bacco for working together with me on this problem and being always there to give guidance and support at every step of the research process. Thanks to all Physics for Inference and Optimization group members for the cluster money to run experiments, precious pieces of advice and all the great time at MPI. I consider myself incredibly lucky for having such a great team during my internship.

I would also like to express my gratitude to Professor Wolfgang Erb for supervising this thesis as well as teaching one of the best courses during my whole Data Science journey. Although I had to follow Mathematical Models and Numerical Methods for Big Data course twice, it helped me to fill the void in my mathematical knowledge and this work would certainly not have been possible without it. Special thanks to Professor Rinaldi for all the best experiences and opportunities associated with this Masters Degree.