DISS. ETH NO. 30480

Probabilistic Models and Approximate Inference Methods for Structured Data

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES
(Dr. Sc. ETH Zurich)

presented by

NICOLÒ RUGGERI

MSc. Università degli Studi di Padova,
born on 25 March 1994

Accepted on the recommendation of
Prof. Dr. F. Yang, examiner
Dr. C. De Bacco, co-examiner
Prof. Dr. J. Young, co-examiner

2024

Abstract

The language of probability is pervasive in almost all of the fields modeling real or abstract systems. Despite presenting a variety of desirable features, such as accounting for uncertainty, allowing reasoning over marginalized variables of a global systems, and permitting predictions, probabilistic models are often challenging to utilize in practice due to the difficulties of inference. For this reason, a variety of approximate inference approaches have been developed by the research community in order to deal with more and more complex scenarios.

Despite the impressive advances that the fields of probabilistic modeling and approximate inference have seen in the last 30 years, most approaches encountered in the inference literature are developed with general utility in mind, aiming at finding solutions to broad classes of probabilistic problems. For this reason, many times probabilistic models tailored to specific applications require equally tailored inference solutions that are not found in the literature.

In this thesis we deal with *structured* scenarios where, compared to traditional approaches, the problems at hand come with more complex generative assumptions or additional information in the data. Accounting for structure usually brings constraints that may be harder to treat, as relations in the data need to be considered at both modeling and inference time. When the difficulties of more restricted inference are overcome, however, such additional information can be exploited to draw more precise conclusions on the observed data itself.

In the works presented in the following chapters we consider the joint problem of developing effective probabilistic models for structured data while performing efficient and informed approximate inference. In particular, we explore some of the most prominent inference approaches, namely Markov Chain Monte Carlo, message passing, variational inference and expectation maximization, in the context of applications to hypergraphs and variational autoencoders. In the former case, we show how to devise models and inference approaches the take into account the higher-order structure of the data. In the latter, we show how to devise theoretically-grounded deep learning models to perform inference under additional supervision and detailed generative assumptions.

In summary, the overarching theme of this thesis is how to exploit additional structure in the data effectively, both theoretically and computationally. As a result of aligning the inductive bias of the algorithms to the data at hand, structured approaches outperform generic ones on specific tasks, allowing to derive improved conclusions from observations.

Sommario

Il linguaggio della probabilità è pervasivo in quasi tutti i campi che modellano sistemi reali o astratti. Nonostante presentino una serie di caratteristiche favorevoli, come la capacità di tenere conto dell'incertezza, di ragionare su variabili marginali di un sistema globale e di consentire previsioni, i modelli probabilistici sono spesso difficili da utilizzare in pratica a causa delle difficoltà di inferenza. Per questo motivo, la comunità di ricerca ha sviluppato una serie di approcci di inferenza approssimata per affrontare scenari sempre più complessi.

Nonostante gli impressionanti progressi che i campi della modellazione probabilistica e dell'inferenza approssimata hanno visto negli ultimi 30 anni, la maggior parte degli approcci incontrati nella letteratura sull'inferenza sono stati sviluppati con uno scopo generale, con l'obiettivo di trovare soluzioni ad ampie classi di problemi probabilistici. Per questo motivo, molte volte i modelli probabilistici adattati a specifiche applicazioni richiedono soluzioni di inferenza altrettanto personalizzate che non si incontrano in letteratura.

In questa tesi ci occupiamo di scenari *strutturati* in cui, rispetto agli approcci tradizionali, i problemi da affrontare sono caratterizzati da ipotesi generative più complesse o da informazioni aggiuntive nei dati. La considerazione della struttura di solito comporta vincoli che possono essere più difficili da trattare, poiché le relazioni nei dati devono essere considerate sia al momento della modellazione che dell'inferenza. Tuttavia, quando si superano le difficoltà di un'inferenza più ristretta, queste informazioni aggiuntive possono essere sfruttate per trarre conclusioni più precise sui dati osservati.

Nei lavori presentati nei capitoli seguenti consideriamo il problema congiunto di sviluppare modelli probabilistici efficaci per i dati strutturati e di eseguire un'inferenza approssimata efficiente e informata. In particolare, esploriamo alcuni dei principali approcci all'inferenza, ovvero Markov Chain Monte Carlo, message passing, inferenza variazionale e massimizzazione dell'aspettativa, nel contesto di applicazioni a ipergrafi e autoencoder variazionali. Nel primo caso, mostriamo come concepire modelli e approcci di inferenza che tengano conto della struttura di ordine superiore dei dati. Nel secondo caso, mostriamo come sviluppare modelli di deep learning teoricamente fondati per eseguire l'inferenza sotto una supervisione aggiuntiva e ipotesi generative dettagliate.

In sintesi, il tema principale di questa tesi è come sfruttare efficacemente la struttura aggiuntiva dei dati, sia dal punto di vista teorico che computazionale. Grazie all'allineamento del bias induttivo degli algoritmi ai dati, gli approcci strutturati superano quelli generici su compiti specifici, consentendo di trarre conclusioni migliori dalle osservazioni.

To Alessandra, for
"The dunes are changed by the wind,
but the desert never changes."[1]

[1]Paulo Coelho, The Alchemist.

Acknowledgements

To begin, I want to acknowledge the contribution and support of my supervisors, Fanny and Caterina. To Caterina, for bringing me into the world of science, providing human support that goes beyond duty, and guiding more than my research choices on the tortuous path of a doctorate. To Fanny, for the valuable inputs during my time at ETH, welcoming me into the group and embroidering a critical spirit that will keep serving me well in the years to come. Additionally, I acknowledge the lessons from Federico Battiston, that changed my perspective on scientific writing and resulted in a fulfilling and enriching series of publications.

As the saying goes "*it takes a village to grow a doctoral student*". Perhaps more than the research network, what it takes is the human support in an adventure full of unforeseen turns. It falls beyond words describing how grateful I am for these connections, which resulted in friendships I deeply cherish and that impressed everlasting influences on my life.

In particular, I want to extend a hug to my peers in Tübingen, for the constant warmth they provided along the years, in no particular order: Martina, Diego, Daniela, Alessandro, Laura, Nicolò, Kibidi, Emilia, Pablo, Jack, and Felix. I can't express how invaluable your support was, and hope I was able to return even a fraction of what I received.

Likewise, I want to thank the colleagues in Zürich, for I couldn't have imagined a more incredible welcome in a new city: Matteo, Luca, Laura, An-phi, Konstantin, and Alexandru. I owe you many dear memories in the beautiful Switzerland, and a period of my life I will treasure in time.

Naturally, that of a doctorate is more than an academic path, enriched by the many people one encounters steering through the choices of life. I would love to extend an acknowledgement to all of the people that contributed making these years an unparalleled process of growth and self-improvement. If you were part of to it, you know who you are, and I thank you deeply.

Contents

Chapter 1

Introduction

In the realm of probabilistic modeling, whether rooted in frequentist or Bayesian principles, inference serves as the keystone guiding the process that allows drawing conclusions from data. Bayesian inference, in particular, offers a rigorous framework for updating beliefs about parameters or hypotheses based on observed data, leveraging prior knowledge to derive posterior distributions. On a high level, performing inference in a frequentist framework entails finding the maximum likelihood estimates of a model's parameters, while Bayesian inference reduces to deriving posterior distributions. While theoretically well-grounded, however, these inference procedures are often hard to accomplish in practice. For this reason, many approximate inference approaches have been developed by the research community during the years. Among these are Monte Carlo approaches, expectation-maximization, variational inference, and message passing, to name a few.

While approximate inference algorithms are developed in full generality, many times specific problems require adjustments for inference to be performed correctly and efficiently. This issue becomes especially apparent when the data at hand is highly *structured*, that is, it comes with precise generative assumptions or specific relations to be exploited and taken into account. In this thesis, we provide examples of how commonly utilized inference frameworks can be adjusted to accommodate for such structured scenarios. We tackle such a problem in a variety of both data and inference regimes, and show how the union of general inference tools with problem-specific adjustments can yield algorithms that are both efficient and effective in practice.

1.1 Inference for structured problems

Historically, examples of hard-to-perform inference can be found in early statistical physics, notably the exploration of systems such as the Ising model

[9, 10, 11]. The Ising model describes the interactions between magnetic spins, which govern the system's energy. The connection between the energy of the system and the probability of observing a specific spin configuration is described by the Boltzmann distribution. Given the spin values $\sigma \in \{-1, +1\}^N$ of the N particles in the system, the Boltzmann distribution is given by

$$p(\sigma) = \frac{e^{-H(\sigma)}}{Z},$$

where Z is a normalizing constant, also known as the free energy of the system. The Hamiltonian H is described by the external field h and the interaction strengths J between spins, amounting to

$$H(\sigma) = -\sum_{i=1}^{N} h_i \sigma_i - \sum_{i<j} J_{ij} \sigma_i \sigma_j .$$

Such relatively simple systems, where variables are discrete and interactions can be clearly decomposed inside the Hamiltonian, already showcase various difficulties which can be encountered in other, possibly more complex, probabilistic scenarios. First, sampling from the Boltzmann distribution poses a significant challenge, particularly as the size and complexity of the system increase. In fact, sampling from the Boltzmann distribution requires knowing the value of the free energy, which has explicit form

$$Z = \sum_{\sigma \in \{-1,+1\}^N} e^{-H(\sigma)} .$$

The combinatorial explosion of the 2^N possible spin configurations renders exhaustive enumeration infeasible, and requires resorting to approximate sampling techniques. Second, when the h_i, J_{ij} parameters are not known, the task of inferring their values from system's observations, also referred to as the inverse problem in the Ising model literature [12], is similarly rendered difficult by the combinatorial explosion of configurations. As a result, the Ising inverse problem gave rise to the first variational methods [11], which are at the basis of numerous modern approaches in approximate inference.

That of the Ising model serves as an illustration of a challenging inference problem where closed-form solutions are not available. More broadly, such examples are encountered in many fields of probabilistic inference, ranging from machine learning to detailed Bayesian modelling of complex systems [13, 14]. In response to the inherent difficulties highlighted by the previous example, researchers have developed a spectrum of modern approximate inference techniques. These approaches offer pragmatic solutions in different inference settings, often representing a trade-off between computational complexity and generative assumptions, together with the higher or lower levels of approximation they provide. In Section 2.2, we provide an overview of modern approaches for approximate inference.

1.2 Accounting for structure

While modern approximate inference methods represent significant advancements in tackling complex probabilistic inference problems, their general applicability may pose limitations when confronted with more structured or specialized scenarios. In fact, these methods offer efficiency and versatility across a range of inference settings, but they are often designed with broad utility in mind. Yet, when faced with more structured problems or when incorporating detailed modeling assumptions, the need for tailored approaches becomes apparent. In such cases, existing tools may require modifications or extensions to accommodate the specific requirements of the problem at hand. In Section 2.1, we delve into hypegraphs, which constitute a prime example of a field where classical approaches need to be adjusted to more structured data. In such a case, previous approaches for community detection on graphs, briefly introduced in Section 2.1.2, are not readily extended, and additional effort is required both in devising viable probabilistic models as well as efficient inference procedures. In Chapters 3–6 we show how to devise probabilistic models for hypergraphs, and perform efficient inference for these models based on data observations. In Chapter 7, we show an example of how to define variational autoencoders respecting specific generative assumptions, and how this results in both theoretical guarantees and practical utility.

1.2.1 Inference in theory and in practice

The study of approximate inference encompasses both theoretical and practical considerations, each playing a vital role in the development and application of inference methodologies. Theoretically, formal guarantees examining the conditions under which these methods yield good approximate solutions, and ideally bound their error, provide valuable insights into the validity of the conclusions drawn from the data. Therefore, such theoretical analyses serve to establish confidence for end-users and the broader community, offering assurances regarding the accuracy and robustness of the inference outcomes. Conversely, in practice, the emphasis lies on devising inference procedures that exhibit computational scalability. Practical considerations in this direction encompass algorithmic optimizations, parallelization and sparsification techniques, dynamic programming for solving intermediate problems, and implementation strategies aimed at ensuring that inference procedures can be implemented efficiently in real-world settings. In Chapters 3 and 6, we show how a variety of such techniques are vital to the practical success of theoretically-grounded algorithms.

1.3 Contributions

This thesis is written with the purpose of presenting the contributions of the author to the scientific community, specifically the following works:

- "*Community detection in large hypergraphs*" [1], presented in Chapter 3.
- "*Framework to generate hypergraphs with community structure*" [2], presented in Chapter 4.
- "*Hypergraphs with node attributes: Structure and inference*" [3], presented in Chapter 5.
- "*Message-passing on hypergraphs: Detectability, phase transitions and higher-order information*" [4], presented in Chapter 6.
- "*Provable concept learning for interpretable predictions using variational inference*" [5], presented in Chapter 7.

Other contributions of the author, which are not presented in this thesis are:

- "*Hypergraphx: a library for higher-order network analysis*" [6]
- "*Fast rates for noisy interpolation require rethinking the effect of inductive bias*" [15]
- "*Sampling on networks: estimating spectral centrality measures and their impact in evaluating other relevant network measures*" [7]
- "*Sampling on networks: estimating eigenvector centrality on incomplete networks*" [8]

Chapter 2

Background

2.1 Graphs and hypergraphs

In many fields, ranging from commercial transportation to social relationships to ecology, the naturally occurring phenomena observed on global systems stem from the interactions of numerous microscopic components. Crucially, the low-level interactions that give rise to macro-observations combine in a highly non-linear and complex fashion. Indeed, such systems are often referred to as complex systems, whose main characteristic is that "*it is difficult to derive their collective behavior from a knowledge of the system's components*" [1]

Within such a context, networks have emerged as invaluable tools for modeling complex systems. Networks serve as maps delineating the physical or virtual spaces wherein interactions unfold. Owing to the integration of graph theory and statistical mechanics, networks have paved the way for a multidisciplinary field that spans fundamental physics to the social sciences.

Yet, traditional network representations have their limitations [16, 17]. While networks capture pairwise interactions effectively, many real-world systems exhibit collective behaviors that cannot be explained only by dyadic connections. Social systems, neuroscience, ecology, and biology often feature interactions among groups of nodes rather than pairs [18, 19, 20, 21]. Such higher-order interactions are critical for understanding phenomena with a lesser level of approximation, sometimes uncovering dynamics that would be fundamentally misunderstood under the lens of traditional network science.

To address this gap, the spotlight has turned to simplicial complexes and hypergraphs [22] as promising frameworks for capturing group interactions. Unlike traditional networks, hypergraphs accommodate interactions among any number of units, making them ideal for modeling real-world systems

[1] Quote from the excellent and comprehensive introduction to complex systems, Prof.Albert-László Barabási's online book http://networksciencebook.com/.

characterized by higher-order relationships. In this section we give an introduction to selected topics in the hypergraph literature, with a special emphasis on community detection methods and on the comparison with traditional dyadic methods.

2.1.1 Hypergraphs: An introduction

Consider a system of units $V = \{1, \dots, N\}$, denoted as nodes. In classical networks, the interactions between these units are represented via a graph $\mathcal{G} = (V, E)$. Here, the interactions between nodes are represented via edges $(i, j) \in E$, where both i, j are nodes, i.e. $E \subseteq V \times V$.

Hypergraphs generalize this representation to systems where interactions happen among an arbitrary number of nodes. A hypergraph $\mathcal{H} = (V, E)$ maintains the definition of nodes as atomic parts of the systems, and allows the interactions to be an arbitrary subset of nodes $E \subseteq \mathcal{P}(V)$, where is $\mathcal{P}(V)$ is the set of all possible subsets of V. The elements of E are called hyperedges. A hyperedge $e = \{i_1, \dots, i_m\}$ is said to have size, or dimension m, and represents a single interaction among the nodes it contains.

A hypergraph whose edges all have size 2 can be equivalently represented as an undirected graph, which explains how hypergraphs are a direct generalization of graphs. The representational power of hypergraphs, however, comes from mixing interactions of various sizes. Such complexity results in the emergence of phenomena that span different magnitudes and increase the non-linearity of the system at hand, at the same time providing the possibility for more nuanced explanations of such phenomena.

Graphs are generally represented via their adjacency matrix $A \in \mathbb{R}^{N \times N}$, defined as:

$$A_{ij} = \begin{cases} 1 & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases}.$$

In the case of weighted graphs, A_{ij} can take on real values for edges present in E, defining a notion of edge weight. The representation of hypergraphs, is rendered more complicated by their high dimensionality. One way to computationally and theoretically represent hypergraphs is directly via it hyperedge list, i.e. as a set of node sets. Theoretically, many works utilize adjacency tensors: For every possible node tuple $(i_1, \dots, i_m)$ of every possible dimension $m = 2, \dots, N$, the adjacency tensor A of a hypergraph is defined by

$$A_{i_1, \dots, i_m} = \begin{cases} 1 & \text{if } \{i_1, \dots, i_m\} \in E \\ 0 & \text{otherwise} \end{cases}.$$

Another computationally advantageous representation of a hypergraphs is given by the incidence matrix $B \in \{0,1\}^{N\times|E|}$, defined by

$$B_{ie} = \begin{cases} 1 & \text{if } i \in e \\ 0 & \text{otherwise} \end{cases}.$$

In both the adjancency tensor and incidence matrix representations, weighted hyperedges can be represented similarly to the case of the adjacency matrix for graphs.

2.1.1.1 Extensions

In many cases the extensions that have been made to graphs in the past are being made now in parallel for hypergraphs.

Among these, notable ones include directed [23], temporal [24], and multilayer [25] hypergraphs. Other extensions comprise the incorporation of covariates, possibly both on nodes and hyperedges [3].

2.1.1.2 Other representations

The representation of a hypergraph as a collection of hyperedges $\mathcal{H} = (V, E)$ is intuitive in terms of interpretation and provides a natural parallel with pairwise networks. However, hypergraphs can be represented via a variety of mathematical objects, which can at times be advantageous for specific tasks or problems. Among these, we mention the representation as factor graphs, or equivalently as bipartite graphs [22]. As introduced in Section 2.2.3.1, factor graphs contain two types of nodes: variable nodes and function nodes. In representing hypergraphs, it is useful to map elements of V, i.e. nodes in the hypergraph, to variable nodes. Consequently, function nodes represent hyperedges, and are connected to the variables nodes the hyperedge contains.

Equivalently, one can map hypergraphs to bipartite graphs, which factor graphs are a special case of. While the mathematical representation is the same, bipartite graphs are usually represented as a set of homogeneous nodes partitioned in two groups, and where interactions can only connect two groups, but never happen between nodes in the same group.

Often times, mapping hypergraphs to known structures can be helpful as classical algorithms can be directly applied on higher-order problems. Examples of such cases involve graphical representations [26], configuration models [27], and synchronization [28].

Yet, many problems arise from the treatment of hypergraphs directly as complex systems of their own. This renewed focus sometimes does not allow a direct mapping to known algorithms for classical representations, and requires solutions tailored to higher-order systems. One such case is that of community detection, which we introduce next.

2.1.2 Community detection

Community detection [29], also known as clustering or node partitioning, involves the task of grouping the nodes within a system into clusters, known as communities, according to observed interactions. Initially applied to social and interaction-based networks, the concept of communities stemmed from the analysis of social structures and relationships [30]. Over time, community detection has expanded its application to various fields, emerging as a prominent method for modeling complex systems across disciplines.

With the development of community detection as a defined branch of complex systems, also the number of definitions of communities evolved to be applied to specific cases where different modeling assumptions are required. To date, there is no one-size-fits-all definition of community. On the contrary, different works advocate for the usage of different generative models for community detection that best match the generative process of the data [31, 32].

As with other fields, the approaches to community detection on networks have been expanded to hypergraphs, with some contributions reported in later chapters of this thesis.

Here, we introduce the topic of community detection with one of the most popular algorithms in the literature, namely the Stochastic Block Model, to then focus on the variational techniques which are commonly employed for their inference.

2.1.2.1 Stochastic block models

Stochastic Block Models (SBMs) [33] are a class of generative models for community detection on networks. In their most basic form, SBMs on networks comprise two parameter sets that are the objective of inference: an affinity matrix $w \in \mathbb{R}^{K\times K}$, where K is the number of communities, and a membership matrix $u \in \mathbb{R}^{N\times K}$, where N is the number of nodes in the network.

A Bernoulli SBM Depending on the type of network at hand, these parameters take on different ranges to accomodate for different types of data. Consider the case of hard communities and unweighted graphs. In the case of hard communities, where a node can only belong to a given community, every row u_i of the membership matrix is a one-hot vector representing such membership. The probability of the data can be represented by a product of Bernoulli probabilities:

$$p(A\,;\,u,w) = \prod_{i,j} Be(A_{ij}\,;\,u_i^T w u_j)\,. \tag{2.1}$$

Notice that for this to be a valid overall Bernoulli probability, the affinity matrix w needs to have entries constrained to the $[0,1]$ interval. It is crucial

to notice that different adjacency matrix structures encode different types of interactions: assortative structures are represented by diagonal matrices w, while the disassortative case of inter-community interaction are represented by higher off-diagonal entries of w. Furthermore, in principle it is possible to infer which case best describes the data at inference time, as opposed to assuming it a priori.

Different SBM flavours have been developed for a variety of cases, including weighted data [34], soft community assignments [35], multilayer networks [36], node and edge covariates [37], temporal data [38], making SBMs one of the most prominent approaches in modern-day community detection.

Extension to hypergraphs The extension of SBMs to hypergraphs is not straightforward, as the mathematical definition of community and assortative (and disassortative) interactions lack a canonical form. As an example, consider two recent models: Hypergraph-MT [39] and Hy-MMSBM [1], the latter being presented in Chapter 3. Both models deal with weighted hypergraphs, and assume a factorized Poisson likelihood. In both cases, $u \in \mathbb{R}_{\geq 0}^{N \times K}$ contains soft memberships, as every entry u_{ik} can take on any non-negative value. For every hyperedge $e = i_1, \ldots, i_m$, the Poisson mean is given by

$$\lambda_e := \sum_{k=1}^{K} w_{|e|k} \prod_{i \in e} u_{ik}$$

in the case of Hypergraph-MT, where w_d is a diagonal tensor for every possibly hyperedge dimension d, and w_{dk} its k-th diagonal entry. For Hy-MMSBM, $w \in \mathbb{R}_{\geq 0}^{K \times K}$ resembles instead the affinity matrix of the graph SBM, and the Poisson mean is modeled as:

$$\lambda_e \propto \sum_{i<j} u_i^T w u_j \,.$$

These different possibilities have both been shown to effectively model a variety of higher-order dataset, and represent only two of a variety of modeling choices that can be made. In general, extending SBMs to hypergraphs needs to be tackled from a variety of angles. Theoretically, the models need to be apt to the data at hand. Computationally however, there is a trade-off between the generality of the probabilistic model, the number of parameters to be estimated, and the scalability of inference [40, 41, 42]. Similarly, notice that the number of interactions to be considered, which we call the configuration space, goes from $O(N^2)$ in the case of graphs to $O(2^N)$ in the case of hypergraphs, making computational considerations yet more urgent. Finally, due to their low inductive bias, very general models will require more data for inference to be performed effectively. Oppositely, more restricted models will work on restricted data when this aligns to the generative assumptions. Similar arguments could be made also in the case of networks.

However, the canonical choice of the SBM as a maximum entropy model [43], together with the restriction to dyadic interactions, imply that the modeling choices are much more restricted than for hypergraphs, where theoretical and computational issues are exacerbated by the exploding configuration space.

Variational approaches for SBMs To understand the need for efficient inference routines, even in the case of networks, consider the example model in Equation 2.1. In this case, inferring the maximum likelihood values of w, u entails possibly iterating over all the possible N^K configurations of community assignment u, and obtaining closed-form formulas for the affinity matrix w, which are hard to obtain in practice. For this reason, most scalable approaches focus on greedy, hierarchical, or approximate solutions that can drift away from the optimal solution. As pointed out in the previous paragraphs, this problem is made even more severe in the case of hypergraphs, where the sheer size of the configuration space makes most naive approaches inapplicable.

Much of the success of SBMs in modeling community structure in networks derives not only from their modeling flexibility, but also from the efficient inference methods that have been recently developed in the field. Among these, variational approaches have quickly gained success, and have proven effective in a variety of contexts [35, 36, 37]. Considered the effectiveness of variational approaches to inference in a dyadic context, the works in Chapters 3 and 5 show how to expand this mathematical framework to the case of hypergraphs. Together with careful probabilistic and implementation choices, these works overcome the issue of scalability, while providing solid predictive performances across a variety of datasets. In Section 2.2.4 we introduce the technical background needed for such approaches, and in Section 2.2.4.5 show examples of how to apply them to hypergraphs, with reference to the work presented in Chapter 3.

2.2 Approximate inference in Bayesian modeling

2.2.1 Bayesian probabilistic models

Bayesian modeling provides a probabilistic framework for updating beliefs based on new evidence. It revolves around the idea of using prior knowledge and observed data to compute the probabilities of different hypotheses or parameters. This approach allows quantifying uncertainty, making informed decisions, and drawing meaningful conclusions across various fields in machine learning and complexity science, among others.

On a high level, the goal of Bayesian inference is that of finding a posterior distribution. Given a probabilistic model comprising a likelihood $p(x|z)$ of

the observed data x given latent variables z, and a prior distribution $p(z)$, Bayes' theorem yields the posterior as distribution as:

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}\,. \tag{2.2}$$

In the scope of this thesis we are interested in finding good solutions or approximations of the posterior for a given probabilistic model. However, devising the probabilistic model, both considering the likelihood and prior, constitutes a problem in-and-of-itself. In many cases this comprises choosing the right prior distribution to avoid biasing inference [44], as well as choosing the right likelihood function to describe the data. While not explicitly mentioned, such tasks were implicit in many of the works presented in this thesis.

Bayesian inference is often challenging to perform analytically. However, there are instances where closed-form solutions are attainable. A notable example is that of conjugate families, pairs of distributions where prior and posterior belong to the same family. For example, a Poisson likelihood and a Gamma prior

$$\begin{aligned} p(x_i|\lambda) &= \text{Poisson}(x;\lambda) \\ p(\lambda) &= \text{Gamma}(\lambda;k,\theta) \end{aligned}$$

yield a posterior which is also Gamma distributed and with close-form updates for its parameters:

$$p(\lambda|\{x_i\}_i) = \text{Gamma}\left(\lambda; k + \sum_i x_i, \frac{\theta}{n\theta+1}\right)$$

where $x_1, \ldots, x_n$ is a sample of n data observations.

In more general cases, closed-form expressions for the posterior are hard to obtain. This is due to the marginal likelihood of the data, also known as evidence, being hard to compute or even approximate in higher dimensions:

$$p(x) = \int p(x,z)\,dz = \int p(x|z)\,p(z)\,dz\,.$$

For this reason, a variety of techniques for the approximation of posterior distributions have been devised. In the following sections we introduce some classical approximation methods, with a particular attention to variational methods in Section 2.2.4.

2.2.2 Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) methods constitute the golden standard for sampling from complex and possibly high-dimensional target distributions. In general, the goal of MCMC methods is to devise a Markov Chain (MC) whose stationary distribution corresponds to the target one. As a result, simulating the Markov Chain for long enough provides samples from a distribution which is guaranteed to asymptotically match the target. Crucially, most MCMC methods rely on having access to quantities only proportional to the probability density, bypassing the estimation of complex normalization constants.

2.2.2.1 Basics of Markov Chains

A Markov Chain is a stochastic process with time dependencies, or autocorrelations, only between consecutive time steps. Formally:

Definition 2.1 *Given a sequence of random variables (r.v.) $\{y_t\}_t$, these satisfy the **Markov property** if*

$$p(y_t|y_{t-1},\dots,y_0) = p(y_t|y_{t-1})\,.$$

Given this definition, it is also straightforward to factorize the probability of the full sequence as:

$$p(y_0,y_1,\dots,y_T) = p(y_0)\prod_{t=1}^{T} p(y_t|y_{t-1})\,.$$

Assume that all of the r.v. belong to the same discrete probability space $y_t \in \mathcal{Y}$. Then, the transition probabilities $p(y_t = i|y_{t-1} = j)$ are described by a stochastic transition matrix P_{ij}, which also defines the stationary distribution of the Markov chains

Definition 2.2 *Consider a discrete MC on a probability space $\mathcal{Y}$ with transition matrix P. A probability distribution π on $\mathcal{Y}$ is said to be **stationary** for the MC if*

$$P\pi = \pi$$

i.e. if it left unchanged after a one step of the Markov process.

In the context of MCMC methods, the stationary distribution of a chain is the central object of interest. In fact, most MCMC techniques aim at constructing chains with a stationary distribution corresponding to a distribution of interest. When the stationary distribution exists, this is reached and asymptotically sampled from by running the Markov chain for a large number of steps. For this reason, the existence of the stationary distribution is crucial to the development of the sampling algorithms we present next.

2.2.2.2 Metropolis-Hastings algorithm

The Metropolis-Hastings algorithm [45] can be utilized in a Bayesian context to draw samples from the posterior distribution of $p(z|x)$. The algorithm is presented in Algorithm 1

Algorithm 1: Metropolis-Hastings algorithm

Data: observations x_i
Result: samples z_t
$t \leftarrow 0$
sample z_0 from $p(z_0)$
while *True* **do**
    perturb z_t to obtain $\tilde{z}_t$ according to a proposal $g(\tilde{z}_t|z_t)$
    compute acceptance threshold $\tau = \frac{g(z_t|\tilde{z}_t)p(\tilde{z}_t|\{x_i\})}{g(\tilde{z}_t|z_t)p(z_t|\{x_i\})}$
    accept $z_{t+1} = \tilde{z}_t$ with probability $\min(1, \tau)$, else $z_{t+1} = z_t$
end

The key to the theoretical and computational utility of this procedure lies in the definition of the transition probabilities: since the threshold $\tau = \frac{p(\tilde{z}_t|\{x_i\}))}{p(z_t|\{x_i\}))}$ is defined as the ratio of the density between the new proposal z_t' and the current state z_t, there is no need to know any normalizing constant. For example, τ can be computed via the typically easier-to-access joint distribution $\tau = \frac{p(\tilde{z}_t,\{x_i\}))}{p(z_t\{x_i\}))}$, or in the case of the Ising models by only accessing the Hamiltonian values without the need to estimate the high-dimensional free energy.

2.2.2.3 Gibbs sampling algorithm

The Gibbs algorithm constitutes a special case of Metropolis-Hastings. It is particularly well-suited for scenarios where sampling from the target distribution is more efficiently achieved by sampling from its conditionals rather than directly from the joint distribution. The observation is that many hierarchical distributions, or more in general graphical models, can be easily sampled knowing the probabilistic dependencies describing the single random variables. More specifically, if we consider a set of n random variables $z_1, \ldots, z_n$, the Gibbs sampling algorithm samples from the conditionals

$$p(z_j|z_{\setminus j})$$

where $z_{\setminus j}$ is the set of all variables but z_j. By iterating over all variables $j = 1, \ldots, n$ and repeating such procedure indefinitely, Gibbs sampling approximates the joint distribution, and can be computationally advantageous

when the conditional ones are known in closed-form or cheap to sample from.

In a Bayesian context, the goal is to obtain samples of the latent variables z from the posterior distribution in Equation 2.2.

2.2.2.4 Limitations of MCMC

Markov Chain Monte Carlo methods stand as the gold standard for sampling from complex distributions, owing to their theoretical foundations and asymptotic guarantees of convergence. However, their applicability in real-life scenarios is often hindered by many practical drawbacks.

First of all, most MCMC techniques require a burn-in period to reach equilibrium, produce samples from the target distribution, and not be biased by the choice of the starting point. Additionally, MCMC samples often exhibit self-correlation between samples, leading to inefficiencies in the exploration of the probability space. Perhaps the most significant issue is the challenge of mixing, where the Markov chain struggles to efficiently explore the high-dimensional space and tends to get stuck in local optima or valleys of the probability distribution. Various techniques have been proposed to mitigate these challenges, such as employing multiple chains in parallel or employing techniques like replica with chain exchanges [46, 47]. Marginalization techniques can also aid in improving mixing by integrating out certain variables from the model, as in Gibbs sampling.

Despite these improvements, MCMC remains computationally intensive, particularly in high-dimensional spaces where the curse of dimensionality exacerbates the challenges of exploration. Consequently, while MCMC offers a powerful framework for sampling from complex distributions, its practical utility is often limited by its computational demands and challenges in achieving efficient mixing.

2.2.2.5 Related techniques

Aside from the MCMC techniques presented above, there exist a variety of other methods to sample from complex target distribution. Among these, we mention acceptance-rejection sampling [48] and importance sampling [49], both based on proposals distributions that approximate the target and provide a reduction in variance in a variety of cases. Alternatively, justified by asymptotic results like the Central Limit Theorem, the Laplace method [50] provides a second-moment Gaussian approximation to a target distribution. More recently, Langevin dynamics have been proposed to simulate sampling from a distribution via its score function, overcoming the need for hard-to-compute normalization constants [51].

2.2.2.6 Application: sampling from the Hy-MMSBM model

In this section, we introduce an application of the MCMC methods explained above to hypergraph sampling. In particular, we focus on the Hy-MMSBM probabilistic model, presented in more detail in Chapters 3 and 4. We show how such methods can be tuned to take into account the structure of the problem at hand.

The Hy-MMSBM model is a probabilistic model for community configurations of hypergraphs, and extends the classical SBM model to higher-order data. Consider a weighted hypergraph $\mathcal{H} = (V, E)$, which can be represented via a vector of natural weights $\{A_e\}_{e\in\Omega} = A \in \mathbb{N}^{|\Omega|}$, for every hyperedge $e \in \Omega$. Here, Ω is the space of all possible hyperedges, comprising both the observed E and unobserved ones.

The Hy-MMSBM model assumes a factorized probability model defined as:

$$p(\mathcal{H}\,;\, w, u) = \prod_{e\in\Omega} p(A_e\,;\, w, u)\,, \tag{2.3}$$

where the single edges are Poisson-distributed according to

$$p(A_e\,;\, w, u) = Pois\left(A_e\,;\, \lambda_e := \frac{\sum_{i<j\in e} u_i^T\, w\, u_j}{\kappa_e}\right)\,, \tag{2.4}$$

and κ_e a normalization constant. The symmetric affinity matrix $w \in \mathbb{R}_{\geq 0}^{K\times K}$ and community assignments $u \in \mathbb{R}_{\geq 0}^{N\times K}$ encode the community structure of the hypergraph. While the problem of inferring such parameters from the data can be tackled via variational techniques, here we focus on the problem of directly sampling from a configuration of the model with given w, u parameters.

The problem of sampling hypergraphs is relatively novel, as sampling dyadic networks from a given SBM model can be done by directly sampling the $O(N^2)$ edges from their marginal distributions. In hypergraphs, however, the configuration space Ω is of size $O(2^N)$, therefore it is not possible to directly sample from the marginals in Equation 2.4 for all the hyperedges.

Similarly, it is also not possible to apply vanilla MCMC techniques to directly sample hypergraphs. For such an approach to be possible, in fact, we would need to start from a random initial configuration of hyperedges and then mix until convergence to the stationary distribution corresponding to that presented in Equation 2.3. As explained in Section 2.2.2.4, the slow mixing time would not allow performing the sampling efficiently, with the possibility of the Markov chain getting stuck in local probability optima.

The solution proposed in [2] is to utilize the structure of the probabilistic model to prompt the Markov chain with more structure, thus effectively

reducing the space of configurations to be explored. Specifically, the sampling procedure is divide in three consecutive steps.

The first step is to sample two global statistics of the hypergraph samples to be produced. These two statistics are the degree sequence $d = (d_i)_{i \in V}$, containing the (unweighted) degree of the single nodes, and size sequence $k = (k_j)_{j=2}^{N}$, containing the number of hyperedges k_j for every possible size j. As proven in Chapter 4, a Central Limit Theorem allows the cheap sampling of such statistics, since their Gaussian approximation is available in closed form.

The second step is to then combine the degree and size sequences into a first unweighted hypergraph proposal, to then be recombined via the MCMC procedure from [52]. Such an MCMC procedure is termed "configurational", since it preserves the degree and size sequences during mixing.

Combining the sequences sampled at the previous stage to only explore hypergraphs that respect such sequences is the crucial step that renders this whole procedure more efficient than a vanilla application of MCMC: the space of configurations to be explored at this stage is reduced to the space of hypergraphs with a given configuration, and can be sampled more efficiently via Monte Carlo techniques.

The final step of the sampling procedure is to obtain the weights of the binary hyperedges obtained from stage two. This can be performed cheaply, as the expected number of hyperedges scales linearly for sparse hypergraphs, and their weights come from a truncated Poisson distribution.

In summary, this approach to sampling hypergraphs from the Hy-MMSBM probabilistic model shows how combining classical MCMC techniques with more structured sampling approaches can allow overcoming the problem of the exploding hypergaph configuration. As a result, it is one of the first scalable approaches to sampling synthetic hypergraphs.

2.2.3 Message passing and belief propagation

Message passing (MP), and its widespread instance called belief propagation (BP) stand out as particularly useful techniques for approximate inference in models with high dimensionality but sparse probabilistic dependencies. First introduced in the context of statistical physics and problems akin to the Ising model presented in Section 1.1, BP operates on factor graphs. Factor graphs, in turn, are useful tools for representing probabilistic dependencies among random variables. In the following sections, we present the building blocks of BP, and outline how and when this can be used in practical inference problems.

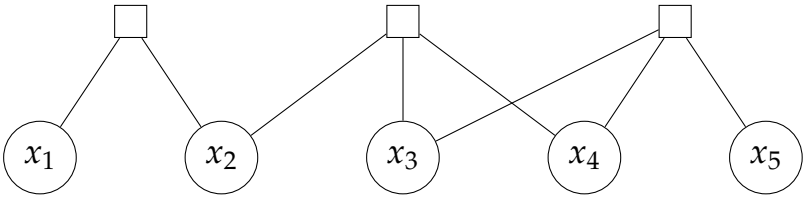


Figure 2.1: Factor graph representation of a probability distribution with factorization $p(x_1, x_2, x_3, x_4, x_5) = f_{1,2}(x_1, x_2) f_{2,3,4}(x_2, x_3, x_4) f_{3,4,5}(x_3, x_4, x_5)$ for some functions $f_{1,2}, f_{2,3,4}, f_{3,4,5}$.
Variable nodes (circles) represent random variables, function nodes (squares) represent probabilistic dependencies between variables, i.e. factors in the probability distribution.

2.2.3.1 Probabilistic models as factor graphs

For any given probabilistic model and its factorization, which illustrates the probabilistic relationships between its variables, it is possible to depict the model as a factor graph. Factor graphs comprise two types of nodes: variable nodes and factor nodes. Variable nodes correspond to the original random variables within the probabilistic model, while factor nodes symbolize the factorization of the probability distributions. Put differently, factor nodes encapsulate the factors involved in the distribution's factorization. In Figure 2.1, we show the factor graph representation of a probabilistic model with factorization $p(x_1, x_2, x_3, x_4, x_5) = f_{1,2}(x_1, x_2) f_{2,3,4}(x_2, x_3, x_4) f_{3,4,5}(x_3, x_4, x_5)$ for some functions $f_{1,2}, f_{2,3,4}, f_{3,4,5}$.

As is intuitive from its representation, a factor graph is as useful as the factorization of the probability distribution it represents. When numerous variables are conditionally independent and the factorization yields sparse representations in the graphs, using such a representation makes it easier to perform inference on the data. The BP algorithm builds on this intuition to devise fast, and in certain cases exact, inference procedures for a given probabilistic model.

2.2.3.2 The belief propagation algorithm

The BP algorithm is based on the idea of cavity distribution in statistical physics [53, 54], and yields useful and simplified formulas for the free energy of a systems. Similar to the case of the Ising model presented in Section 2.2, any probabilistic model on can be reformulated in the form of a Gibbs-Boltzmann distribution, defined as

$$p(x) = \frac{e^{H(x)}}{Z},$$

where H is called the Hamiltonian of the system, and Z is a normalization constant. We notice that, given a probabilistic model with distribution p over all its variables, it is perhaps more familiar in the context of statistics

and probabilistic machine learning to utilize the log-likelihood $\mathcal{L}(x) := \log p(x)$. The connection between the Gibbs-Boltzmann distribution and the original probabilistic model is specified by $H(x) \propto \mathcal{L}(x)$, where any constant independent of x can be absorbed into Z. Additionally, it is possible to read the edges of the deriving factor graph directly from the Hamiltonian: every addend of the Hamiltonian specifies a function node, and each variable x_i appearing in that term an edge between the relative function and variable nodes. Going back to the example of Figure 2.1, the Hamiltonian function is given by:

$$H(x) = \log f_{1,2}(x_1, x_2) + \log f_{2,3,4}(x_2, x_3, x_4) + \log f_{3,4,5}(x_3, x_4, x_5)\,.$$

From the first addend, we obtain a function node that is connected to the variable nodes x_1 and x_2, the second function node is connected to x_2, x_3, and x_4, and similarly for all the addends appearing in $H(x)$.

We now define the formal iterative updates at the core of the BP procedure. In general, consider a Hamiltonian function with general form:

$$H(x) = \sum_{i=1}^{n} H_i(x_i) + \sum_{a=1}^{M} H_a(x_{\partial a})\,. \tag{2.5}$$

Here a are factor nodes, and ∂a defines the set of all variables nodes x_i which are connected to a in the factor graph. We can then define the external fields and interactions as

$$\psi_i(x_i) = e^{H_i(x_i)}$$
$$\psi_a(x_{\partial a}) = e^{H_a(x_{\partial a})}\,,$$

and the messages between nodes as

$$q_{i\to a}(t_i)$$
$$\widehat{q}_{a\to i}(t_i)\,,$$

represented in Figure 2.2. The BP equations define fixed-point updates as follows:

$$q_{i\to a}(t_i) \propto \psi_j(t_j) \prod_{b\in\partial i\setminus a} \widehat{q}_{b\to i}(t_i) \tag{2.6}$$

$$\widehat{q}_{b\to j}(t_j) = \sum_{t_{\partial b}\setminus t_j} \psi_b(t_{\partial b}) \prod_{k\in\partial b\setminus j} q_{k\to b}(t_k)\,. \tag{2.7}$$

In turn, the messages define the marginal beliefs of the variable nodes

$$q_i(t_i) \propto \psi_i(t_i) \prod_{a\in\partial i} \left[\sum_{t_{\partial a}\setminus t_i} \psi_a(t_{\partial a}) \prod_{j\in\partial a\setminus i} q_{j\to a}(t_j)\,, \right]$$

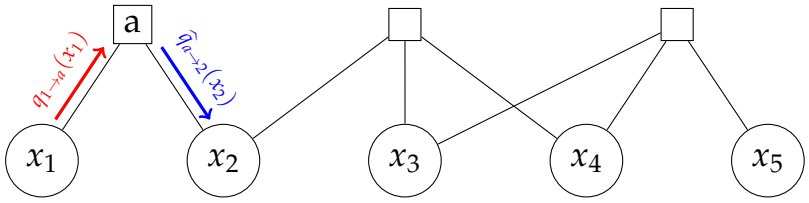


Figure 2.2: Message propagation in a factor graph. In red, the message $q_{1\to a}(x_1)$ from the variable node x_1 to the function node a. In blue, the message $\widehat{q}_{a\to 2}(x_2)$ from node a to node x_2.

which correspond to their marginal distributions. Practically, Equations 2.6 and 2.7 can be alternated until convergence of the messages and of the estimated marginal beliefs, which is attained due to the spread of the messages through the factor graph.

2.2.3.3 Loopy graphs and extensions

Crucially, the BP procedure is theoretically justified and yields exact inference on factors graphs without loops, i.e. on tree-like structures. However, most of its utility comes from its application on complex scenarios where loops are present. For this reason, many recent works tried to explain the empirical success of BP also on loopy graphs in a variety of scenarios. A classical explanation is that in most sparse graphs, the loops are statistically large enough that the tree assumption is locally respected [11]. In other cases, it is possible to disregard close-to-constant contributes of many neighbors of a node, and collect them into graph-independent external fields [55], additionally lightening the computational burden found when such graphs are especially dense. Other extensions of MP include adjustments to directly account for loops in the graphical structure [56, 57] or reduce the approximations to take into account more local correlations [58, 59].

The work presented in Section 2.2.3.4 and Chapter 6 broadly follows this line of literature, and exploits classical MP arguments to make simplifying assumptions and obtain practical inference protocols on hypergraphs.

2.2.3.4 Application: message passing on hypergraphs

Message passing represented one of the first approaches to obtain both theoretical and empirical results on the recovery of community configurations on graphs [55]. In this section, we show how such methods can be expanded to hypergraphs, similarly yielding the first theoretical results that are also empirically checked for a class of hypergraph SMB-like models, and refer to Chapter 6 for further details.

The HySBM model is an extension of the SMB to hypergraphs, and is based on hard-community configurations $t_i \in \{1, \dots, K\}$ for every $i \in V$, and a

symmetric affinity $p \in [0,1]^{K\times K}$. Similarly to the Hy-MMSBM probabilistic model introduced in Section 2.2.2.6, the HySMB probabilistic model is defined as

$$p(\mathcal{H}\,;\,w,u) = \prod_{e\in\Omega} Be\left(A_e\,;\,\frac{\pi_e}{\kappa_e}\right)$$

with Bernoulli sufficient statistics defined via

$$\pi_e := \sum_{i<j\in e} p_{t_i t_j}\,.$$

Additionally, here we impose a prior distribution $t_i \sim Cat(n)$, where n are the categorical probabilities over the K community assignments.

This probabilistic model yields a factor graph representation with Hamiltonian equal to the log-likelihood:

$$\mathcal{L}(A,t\,|\,p,n) = \sum_{e\in\Omega}\left[A_e \log\left(\sum_{i<j} p_{t_i t_j}\right) + (1-A_e)\log\left(1-\frac{\sum_{i<j} p_{t_i t_j}}{\kappa_e}\right)\right] \quad (2.8)$$

The interactions in the derived factor graph represetation can be read directly from Equation 2.8: the function nodes are one per hyperedge $e \in \Omega$, and the variable nodes connected to each are those that belong to the hyperedge itself. Due to the number of hyperedges $|\Omega|$, such a factor graph is too large to approch computationally, and would yield practically infeasible MP updates.

For this reason, the key insight in [4] is to take inspiration from previous approaches on graphs, and isolate the contributions of different hyperedges. Crucially, the contributions of observed ($A_e = 1$), and unobserved ($A_e = 0$) hyperedges can be separated in Equation 2.8. As proven theoretically in sparse regimes, the messages to be exchanged between function and factor nodes corresponding to unobserved hyperedges are approximately constant, and can be absorbed into an external field $h(t_i) \propto \sum_{j\in V}\sum_{t_j} p_{t_i t_j} q_j(t_j)$ that only depends on the node marginals q_j, and is thus efficiently computed and updated.

This reasoning shows how, in parallel with previous results on dyadic networks, the direct application of the MP techniques is not possible, and incorporation of additional information from the probabilistic model is necessary for the efficient implementation of the inference procedure.

2.2.4 Variational methods

Variational methods for statistical inference represent a powerful and versatile tool kit that has gathered significant attention across disciplines such as statistics and machine learning. First developed to address complex problems in physical systems, these methods seek to approximate difficult distribution

by optimizing simpler ones, transforming the problem of inference into an optimization one. In particular, the core idea behind variational methods lies in finding surrogate posterior distributions that closely mimic the true, often intractable one.

One of the key motivations for the widespread adoption of variational methods lies in their computational efficiency. Unlike traditional Monte Carlo methods, which rely on sampling techniques and can be computationally intensive, variational methods offer scalability to large datasets and high-dimensional models. Moreover, they provide a flexible and general framework applicable to a wide range of probabilistic models, from simple latent variable models to complex hierarchical structures. Their integration with deep learning frameworks further enhances their versatility, enabling the approximation of complex functional objectives via neural networks.

In this section, we give a brief introduction to variational methods for statistical inference, highlighting the main technical tools that lie at the basis of its development.

2.2.4.1 Variational inequality and evidence lower bound

The goal of variational inference (VI) is to approximate the posterior distribution in Equation 2.2 via another surrogate distribution $q(z)$. [2] To this end, VI seeks to choose the distribution $q(z)$ so as to minimize a notion of distance to the exact posterior. While many choices are available [60], it is standard to consider the KL-divergence [61]:

$$\begin{aligned}\mathrm{KL}\left(q(z)\,||\,p(z|x)\right) &:= -\mathbb{E}_{z\sim q(z)}\left[\log\left(\frac{p(z|x)}{q(z)}\right)\right] \\ &= -\int q(z)\log\left(\frac{p(z|x)}{q(z)}\right)dz\,.\end{aligned}$$

By expanding the KL-divergence the following equality can be derived

$$\mathrm{KL}\left(q(z)\,||\,p(z|x)\right) = \mathbb{E}[\log q(z)] - \mathbb{E}[\log p(x,z)] + \log p(x)\,, \tag{2.9}$$

where all the expectations are taken with respect to $q(z)$. This simple derivation has a clear interpretation in terms of optimization: the distribution $q(z)$ minimizing the KL-divergence is the same maximizing the much simpler objective on the right-hand side, called evidence lower bound (ELBO)

$$\mathrm{ELBO}(q) := \mathbb{E}[\log p(x,z)] - \mathbb{E}[\log q(z)]\,.$$

[2] We note here that while classical methods seek to find an approximate posterior $q(z)$, modern approaches in amortized inference use in turn a posterior directly dependent on the observables as $q(z|x)$. We keep the notation simple in this first introduction, and go back to the case of amortized inference in Section 2.2.4.4.

The reason why this objective is simpler to optimize is that it does not contain the posterior $p(z|x)$ found in the KL-divergence expression, but the joint distribution $p(x,z)$, which can be usually computed. In the literature, the ELBO has been the object of intense investigations in terms of its interpretation and as an optimization objective [62, 63, 64].

As a final note, we also highlight that the derivation above also yields the so-called variational inequality

$$\begin{aligned}\log p(x) &= ELBO(q) + \mathrm{KL}\left(q(z) \,||\, p(z|x)\right) \\ &\geq ELBO(q)\,.\end{aligned}$$

This inequality can be alternatively derived directly using Jensen's inequality on the log-evidence $\log p(x)$ [3] , and justifies the name of the evidence lower bound itself.

These derivations provide a clear VI framework: given a target posterior distribution $p(z|x)$, find an approximation $q(z)$ that is as close as possible to it in terms of KL-divergence. Such a distribution is found by optimizing the evidence lower bound, and its choice is the subjection of the next section.

2.2.4.2 Choosing the approximate posterior

The ELBO objective provides a computationally viable proxy for optimization. If the approximate posterior $q(z)$ can take on any distribution in the probability space of z, then the optimal choice is indeed the exact posterior $q(z) = p(z|x)$, which yields a null KL-divergence. However, letting q span the whole probability space is a hard-to-treat problem. For this reason, q is often chosen to belong to a restricted family of distributions. Among these, we mention fully or partially factorized distributions, respectively referred to as mean field and structured mean field approximations [61], hierarchical families [65], and other parameterized families, notably neural networks such as in the case of variational auto-encoders (VAEs) [66] and normalizing flows [67].

In general, the choice of the approximation family depends on the complexity of the posterior at hand: complex and multi-modal distributions may require resorting to larger variational families, with the problem rendered even more difficult by the optimization of the variational objective itself. We deal with such a problem in the next sections.

[3] Using Jensen's inequality makes the difference between the ELBO and the log-evidence less explicit: in the derivations provided above, it is clear that such a difference is given by the KL-divergence. The derivation via Jensen's inequality is as follows:

$$\log p(x) = \log \int \frac{p(x,z)}{q(z)} q(z) dz = \log \mathbb{E}_q \left[\frac{p(x,z)}{q(z)}\right] \geq \mathbb{E}_q \left[\log \frac{p(x,z)}{q(z)}\right]$$

2.2.4.3 Optimization in variational inference

Once the evidence lower bound objective has been defined, variational inference turns the posterior inference problem into that of maximizing said objective. In general, two main approaches are available in the literature.

Coordinate ascent mean-field variational inference (CAVI) When utilizing a fully factorized approximate posterior $q(z) = \prod_{i=1}^{m} q_i(z_i)$, it is possible to show [68, 61] that the optimal update for a single latent variable z_j is given by

$$q_j(z_j) \propto \exp\left[E_{z_{\setminus j} \sim q_{\setminus j}(z_{\setminus j})} p(z_j | x, z_{\setminus j}))\right] , \tag{2.10}$$

where $q_{\setminus j}(z_{\setminus j}) = \prod_{i \neq j} q_i(z_i)$. This local update suggests a simple iterative optimization process, where the q_j distributions are updated one at a time while keeping all the other indices fixed. While such natural and simple algorithm has connections to the message passing and factor graph frameworks in Section 2.2.3 [61], it only yields a viable optimization procedure when the updates in Equation 2.10 can be computed in closed form.

Black box variational inference While the CAVI procedure yields an efficient coordinate ascent algorithm on the ELBO, its analytical updates cannot always be computed in closed form. In such cases, black-box variational inference (BBVI) [69] constitutes a viable alternative for general distributions $p(x, z)$ and parameterized posterior families $q(z)$.

When we consider an approximate posterior family $q_\theta(z)$ with parameters θ, BBVI provides a closed form of the gradient as

$$\nabla_\theta \text{ELBO}(q_\theta) = E_q \left[\nabla_\theta q_\theta(z)(\log p(x, z) - \log q_\theta(z))\right] .$$

In practice, approximations of the gradient can be accessed via Monte Carlo samples of the expectation above, and can be utilized for stochastic optimization [70].

2.2.4.4 Amortized inference and VAEs

The use of VI has led to notable advances in Bayesian inference, both in terms of generality of the problems that can be tackled and of computational scalability. Perhaps, one of the most impactful advances made possible by VI is the introduction of variational autoencoders (VAEs) [66]. VAEs combine the generality of variational inference with the representational power of modern neural networks, which in such models are used to both estimate a generative model for complex data, for example images, and perform posterior inference for such a model.

The probabilistic framework of VAEs is very similar to that outlined in this chapter: we set parameterized likelihood and approximate posterior functions as

$$p_\phi(x|z)$$
$$q_\theta(z|x)\,,$$

where the parameters ϕ, θ are arbitrary neural network weights. Here, such neural networks are combined in an encoder-decoder architecture which, together with a simple prior $q(z)$[4], fully define a probabilistic model and its approximate posterior.

Optimization is then performed in a similar fashion to the BBVI procedure presented in Section 2.2.4.3: noisy gradients of the ELBO are estimated with respect to both ϕ and θ, and optimization is performed via stochastic gradient descent. While similar on a high level, the optimization of VAEs is substantially more nuanced due to their complex structure. First, while in classic Bayesian modeling the generative model is fixed, here it is learned in the form of p_ϕ as well, raising theoretical issues on the validity of the ELBO as an objective for such a task [71, 72]. Second, optimization in VAEs is performed in an amortized manner: at every inference step, a possibly new data point x is passed into the neural network, and the posterior $q(z|x)$ is only estimated based on that single point. This approach allows scaling to modern-sized dataset of thousand or millions of data points, as well as performing inference on data never seen during training. Finally, effective optimization of VAEs requires additional techniques, such as the reparameterization trick [73, 74].

In Section 2.2.4.6, we show how the solid probabilistic basis of VAEs allows designing identifiable architectures and obtaining interpretable predictions in scenarios where additional supervision in the form of labels is available.

2.2.4.5 Application: inference for the Hy-MMSBM model

The Hy-MMSBM model, already mentioned in Section 2.2.2.6, is a probabilistic model for communities in hypergraphs. In this section, we briefly mention how the variational techniques presented above can be utilized to efficiently perform inference of the Hy-MMSBM parameters based on data observation, and refer to Chapter 3 for further details.

[4]While worthy of a deeper study, we only note here that the prior $q(z)$ has two main requirements: it mush be possible to sample from it, and its density must be available for computation. While most vanilla VAEs architecture utilize a standard Gaussian to this end, substantial research has gone into looking for more complex priors and their effects on the quality of inference, with proposals ranging from normalizing flows to Gaussian processes. For simplicity of exposition we assume here that the prior $q(z)$ is fixed and hence parameter-free.

Recall that the probabilistic model is based on soft community assignments $u \in \mathbb{R}_{\geq 0}^{N \times K}$ and an affinity matrix $w \in\in \mathbb{R}_{\geq 0}^{K \times K}$. For every possible hyperedge e based on the node set V, we define the Poisson mean

$$\lambda_e := \frac{\sum_{i<j \in e} u_i^T w u_j}{\kappa_e} . \tag{2.11}$$

A first approach to infer the optimal u, w values based on an observed hypergraph is to perform maximum likelihood optimization. Notice that, based on the parameterization in Equation 2.11, the log-likelihood can be simplified to the following form

$$\log p(A\,;\, u, w) = -C \sum_{i<j \in V} u_i^T w u_j + \sum_{e \in E} A_e \log \sum_{i<j \in e} u_i^T w u_j , \tag{2.12}$$

where $C = \sum_{n=2}^{D} \frac{1}{\kappa_n} \binom{N-2}{n-2}$ is a constant independent from the data. Direct optimization of the log-likelihood is not possible, as closed-form solutions for u, w cannot be promptly derived by simple differentiation. For this reason, the optimization of the Hy-MMSBM model proceed via a variational approach, briefly described next.

First, for every hyperedge e we define a probability distribution over all pairs i, j of nodes in e, and choice of communities indices k, q for i and j respectively. This probability distribution is defined by the values $\rho_{ijkq}^{(e)}$ such that $\sum_{i<j \in e} \sum_{k=1}^{K} \sum_{q=1}^{K} \rho_{ijkq}^{(e)} = 1$. The approximation of $p(A_e\,;u,w)$ by any choice of $\rho^{(e)}$ distribution corresponds to the mean-field factorization mentioned in Section 2.2.4.2, and can be utilized for inference. In fact, utilizing Jensen's inequality we can obtain a variational lower-bound to the right-hand-side of Equation 2.12:

$$\begin{aligned}
\log \sum_{i<j \in e} u_i^T w u_j &= \log \sum_{i<j \in e} u_i^T w u_j \frac{\rho_{ijkq}^{(e)}}{\rho_{ijkq}^{(e)}} \\
&= \log \mathbb{E}_{(i,j,k,q) \sim \rho^{(e)}} \left[\frac{u_i^T w u_j}{\rho_{ijkq}^{(e)}} \right] \\
&\geq \mathbb{E}_{(i,j,k,q) \sim \rho^{(e)}} \left[\log \left(\frac{u_i^T w u_j}{\rho_{ijkq}^{(e)}} \right) \right] \qquad (2.13) \\
&= \sum_{i<j \in e} \rho_{ijkq}^{(e)} \log \left(\frac{u_i^T w u_j}{\rho_{ijkq}^{(e)}} \right) . \qquad (2.14)
\end{aligned}$$

While optimizing the log-likelihood directly is not feasible, it can be checked that differentiating Equation 2.14 with respect to both u and w yields closed

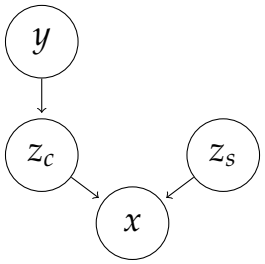


Figure 2.3: Independence assumptions for the dataset of observations x, labels y, and unobserved latent variables z, divided into core and style features z_c and z_s.

form updates. Such updates, in turn, depend on the values of $\rho^{(e)}_{ijkq}$. These can be chosen so as to maximize the tightness of the variational lower bound with respect to the log-likelihood: the passage in Equation 2.13 is tightest when the distribution of the random variable $\frac{u_i^T w u_j}{\rho^{(e)}_{ijkq}}$ has minimum variance. This can be realized by choosing $\rho^{(e)}_{ijkq} \propto u_i^T w u_j$, which yields update values for all the ρ variables as well.

In summary, defining a variational lower-bound for the log-likelihood objective allows recursive optimization that alternates the update of the variational distribution $\rho^{(e)}_{ijkq}$ with that of the model parameters u, w, yielding an efficient and effective approach to inferring community structure from the data.

2.2.4.6 Application: an identifiable and interpretable VAE model

In this section we show an application of the VAE inference framework, where the goal is to develop an identifiable and interpretable VAE model. The technical details are deferred to Chapter 7.

First, we outline the problem setting. We consider the case where we have observational data x, e.g. X-ray images in a medical dataset, and additional labels y attached to each observation, e.g. diseases of the patients, or sex and other physical features. We set out to develop a prediction model for the labels y given an image x, with the additional requirements of the prediction model being interpretable and provably so. Notice that this setting differs substantially from that of vanilla VAEs presented in Section 2.2.4.4, as the additional labels y are seldom considered. Moreover, even when such labels are considered to augment the learned generative model, such models are not able to predict the labels y at test time, since they are required as an input to the VAE itself [75, 76, 77].

Having outlined the problem setting, we make additional generative assumptions to make the problem more treatable. Namely, we assume a Bayesian generative model where the latent variables z are divided in two groups of

"core" and "style" features $z = (z_c, z_s)$, and with the following factorization:

$$p(z \mid y) = p(z_c \mid y)\, p(z_s) \tag{2.15}$$
$$p(x \mid z, y) = p(x \mid z)\,. \tag{2.16}$$

We represent the related graphical model in Figure 2.3. Performing inference on such a model presents different challenges. First, we need to perform inference of the core features z_c in an identifiable manner. By assumption, disentangling such features means recovering some ground-truth, identifiable information that is both interpretable to a human reader (e.g. lung size in a chest X-ray), and informative for prediction of the labels y, as prediction is the goal of the generative model.

To tackle this challenge, we develop a double VAE, structured as follows. We assume two parallel encoder networks $q_{\phi^{cl}}(z \mid x, y)$ and $q_{\phi^{p}}(z \mid x)$, where "cl" stands for "concept learning", and "p" for prediction, parameterized by the relative weights ϕ^{cl}, ϕ^{p}. While the encoders are separate based on the type of input that is given (only the observations x, or both observations and labels (x, y)), the decoder $p_\theta(x \mid z)$ is unique.

This architecture derives from the intuition that the concept learning encoder $q_{\phi^{cl}}$ is provided with all the information to perform posterior inference of the latents z, hence learning the "concepts" that (by assumption) they represent. At test time, however, prediction of the labels y will be performed using the prediction encoder $q_{\phi^{p}}$, that only requires x as input, paired with a classifier $\psi(y \mid z_c)$.

Crucially, and in line with the independence assumptions in Equations 2.15 and 2.16, the shared decoder $p_\theta(x \mid z)$ forces the two encoders to encode an image to the same latent space of core variables z_c, as the generative process from z to x does not depend on the labels. Such an intuition can be mathematically proven, we summarize it in the following informal result.

Theorem 2.3 *Under the generative assumptions in Equations* 2.15 *and* 2.16*, and under mild regularity conditions, the optimal solution of $\phi^{cl}, \phi^{p}, \theta, \psi$ of the ELBO satisfies the following:*

- *the posterior samples z obtained from the two encoders $q_{\phi^{cl}}$ and $q_{\phi^{p}}$ are identical*
- *the posterior samples of z_c are optimally predictive for y, i.e. $p(y|x) = p(y|z_c)$*
- *the estimated posterior samples z_c are equal to the ground truth ones up to scaling and permutation*
- *the classifier $\psi(y \mid z_c)$ is Bayes optimal*

The last two points in the theorem reflect the two requirements for the prediction model: first we recover the correct latents z_c, which are both

informative for prediction and interpretable, second, we utilize the recovered core features z_c for prediction of the labels y, which is an optimal approach under the assumed generative model. Furthermore, despite training the model using the labels y as input, it is possible to directly perform prediction at test time: we recover the latent concepts z_c using the prediction encoder q_{ϕ^p}, and utilize these for prediction. Owing to the first point in the theorem, this approach retains all the information for prediction of y contained in x.

Chapter 3

Community Detection in Large Hypergraphs

Abstract

Hypergraphs, describing networks where interactions take place among any number of units, are a natural tool to model many real-world social and biological systems. In this work we propose a principled framework to model the organization of higher-order data. Our approach recovers community structure with accuracy exceeding that of currently available state-of-the-art algorithms, as tested in synthetic benchmarks with both hard and overlapping ground-truth partitions. Our model is flexible and allows capturing both assortative and disassortative community structures. Moreover, our method scales orders of magnitude faster than competing algorithms, making it suitable for the analysis of very large hypergraphs, containing millions of nodes and interactions among thousands of nodes. Our work constitutes a practical and general tool for hypergraph analysis, broadening our understanding of the organization of real-world higher-order systems.

3.1 Introduction

Over the last decades, most relational data, from biological to social systems, has found a successful representation in terms of networks, where nodes describe the basic units of the system, and links their pairwise interactions [78]. Nevertheless, such a modeling approach cannot properly encode the presence of group interactions, describing associations among three or more system units at a time [22, 79, 16, 80]. Such higher-order interactions have been observed in a wide variety of systems, including collaboration networks [81], cellular networks [21], drug recombination [82], human [83] and animal [84] face-to-face interactions, and structural and functional mapping of the human brain [19, 85, 86]. In addition, the higher-order organization of many

interacting systems is associated with the generation of new phenomena and collective behavior across many different dynamical processes, such as diffusion [87], synchronization [88, 89, 90, 91, 92, 93], spreading [94, 95, 96] and evolutionary games [97, 98, 99].

Networked systems with higher-order interactions are better described by different mathematical frameworks from networks, such as hypergraphs, where hyperedges encode interactions among an arbitrary number of system units [100, 22]. In the last few years several tools have been developed for higher-order network analysis. These include higher-order centrality scores [101, 102], clustering [103] and motif analysis [104, 105], as well as higher-order approaches to network backboning [106, 107], link prediction [39], and methods to reconstruct non-dyadic relationships from pairwise interaction records [108]. A variety of approaches have been suggested to detect communities in hypergraphs, including nonparametric methods with hypergraphons [109], tensor decompositions [110], latent space distance models [111], latent class models [112], flow-based algorithms [113, 114], spectral clustering [115, 116, 117] and spectral embeddings [118]. A different line of works focuses on deriving theoretical detectability limits [119, 120, 121].

Recently, statistical inference frameworks have been proposed to capture in a principled way the mesoscale organization of hypergraphs [41, 39, 122]. Despite their success, current approaches suffer from a number of notable drawbacks. For instance, the method in [122] is restricted to utilizing very small hypergraphs and hyperedges, due to its high computational complexity. Also the approach in [41] suffers from a high computational complexity in the general case, and needs to make strong assumptions to scale to real-life datasets. Finally, the model in [39] is constrained to work only with assortative community structures.

In this work we propose a framework to model the organization of higher-order systems. Our method allows detecting communities in hypergraphs with accuracy exceeding that of state-of-the-art approaches, both in the cases of hard and mixed community assignments, as we show on synthetic benchmarks with known ground-truth partitions. Furthermore, its flexibility allows capturing general configurations that could not be previously studied, such as disassortative community interactions. and core-periphery organization observed in real data.

Finally, overcoming the computational thresholds of previous methods, our model is extremely efficient, making it suitable to study hypergraphs containing millions of nodes and interactions among thousands of system units not accessible to alternative tools. We illustrate the advantages of our approach through a variety of experiments on synthetic and real data. Our results showcase the wide applicability of the proposed method, contributing to broaden our understanding of the organization of higher-order real-world

systems.

3.2 Generative model

A hypergraph consists of a set of nodes $V = \{1, \dots, N\}$ and a set of hyperedges E. Each hyperedge e is a subset of V, representing a higher-order interaction between a number $|e|$ of nodes. We denote by D the maximum possible hyperedge size, which can be arbitrarily imposed up to a maximum value of $D = N$, and Ω the set of all possible hyperedges among nodes in V. We represent the hypergraph via an adjacency vector $A \in \mathbb{N}^{\Omega}$, with entry A_e being the weight of $e \in \Omega$. We assume the weights A_e to be non-negative and discrete. For real-world systems, A is typically sparse. In fact, the number $|E|$ of non-zero entries is typically linear in N, and thus much smaller than the dimension $|\Omega|$.

We model hypergraphs probabilistically, assuming an underlying arbitrary community structure with K overlapping groups, similarly to a mixed-membership stochastic block model. Each node i can potentially belong to multiple groups, as specified by a K-dimensional membership vector u_i with non-negative entries. We collect all the membership assignments in a $N \times K$ matrix u. The density of interactions within and between communities is regulated by a symmetric non-negative $K \times K$ affinity matrix w. These two main parameters, u and w, control the Poisson distributions of the hyperedge weights:

$$p(A_e; u, w) = \text{Pois}\left(A_e; \frac{\lambda_e}{\kappa_e}\right) , \tag{3.1}$$

where

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

Here, $\kappa_e = \kappa_{|e|}$ is a normalization factor that solely depends on the hyperedge size $|e|$. We develop our theory for a general form of κ_n. While in principle any choice $\kappa_n > 0$ is possible, in our experiments we utilize the form $\kappa_n = \frac{n(n-1)}{2}\binom{N-2}{n-2}$, for every hyperedge of size n [2]. Due to the fact that $\kappa_2 = 1$, if the hypergraph contains only pairwise interactions our model is similar to existing mixed-membership block models for dyadic networks [35, 36]. Intuitively, given two nodes i, j, the term $\binom{N-2}{n-2}$ normalizes for the number of possible choices of the remaining $n-2$ nodes in the hyperedge. The term $n(n-1)/2$ averages among the number of possible pairwise interactions among the n nodes in the hyperedge. Note that previous generative

models for hypergraphs were limited to detect only assortative community interactions [39, 41]. By contrast, in our model each entry w_{kq} distinctly specifies the strength of the interactions between each k, q community pair. Hence, for the first time, our method allows encoding more general community structures, without the need to impose a-priori assumptions to ensure computational and theoretical feasibility. In particular, the bilinear form in eq. 3.2 allows for a tractable and scalable inference, regardless of the structure of w. Another relevant feature of the model is that the size of the affinity matrix w does not vary with maximum hyperedge size D nor with the number of hyperedges, making it memory efficient also for hypergraphs with large interactions. We name our model Hy-MMSBM, for Hypergraph Mixed-Membership Stochastic Block Model, and provide an open-source implementation at github.com/nickruggeri/Hy-MMSBM. We have also incorporated our algorithm inside the open-source library Hypergraphx [6].

3.3 Inference

3.3.1 Optimization procedure

In real-life scenarios, practitioners observe a list of hyperedges, encoded in the vector A, and aim to learn the node memberships u and affinity matrix w that best fit the data. To this end, we start by considering the likelihood of A given the parameters $\theta = (u, w)$. Using eqs. 3.1 and 3.2, this is given by

$$p(A|\theta) = \prod_{e \in \Omega} \text{Pois}\left(A_e; \frac{\lambda_e}{\kappa_e}\right) , \tag{3.3}$$

where the hyperedge weights are assumed to be conditionally independent given (u, w). Its logarithm is given by

$$\begin{aligned} \log p(A|\theta) = &\sum_{e \in \Omega} -\frac{1}{\kappa_e} \sum_{i<j \in e} u_i^T \, w \, u_j \\ &+ \sum_{e \in E} A_e \log \sum_{i<j \in e} u_i^T \, w \, u_j \ , \end{aligned} \tag{3.4}$$

where we discarded constant terms not depending on the parameters. The first summation over $|\Omega|$ terms appears intractable due to the exploding size of the configuration space. However, one important feature of our model is that this high dimensionality can be treated analytically, as the likelihood conveniently simplifies. In fact, the summand $\sum_{e \in \Omega} -\frac{1}{\kappa_e} \sum_{i<j \in e} u_i^T \, w \, u_j$ is simply taking the interaction term $u_i^T w u_j$ as many times as it appears in all the possible hyperedges, each weighted by the factor $1/\kappa_e$. This reasoning yields

the count $C = \sum_{n=2}^{D} \frac{1}{\kappa_n} \binom{N-2}{n-2}$ and the following simplified log-likelihood:

$$\log p(A|\theta) = -C \sum_{i<j\in V} u_i^T w\, u_j + \sum_{e\in E} A_e \log \sum_{i<j\in e} u_i^T w\, u_j\,, \tag{3.5}$$

obtaining a tractable sum of terms. To maximize eq. 3.5 with respect to u and w, we use a standard variational approach via Jensen's inequality $\log \mathbb{E}[x] \geq \mathbb{E}[\log x]$, to lower bound the second summand as:

$$\sum_{e\in E} A_e \log \sum_{i<j\in e} u_i^T w\, u_j \geq \sum_{e\in E} A_e \sum_{i<j\in e} \sum_{k,q=1}^{K} \rho_{ijkq}^{(e)} \log \left(\frac{u_{ik}\, u_{jq}\, w_{kq}}{\rho_{ijkq}^{(e)}} \right)\,. \tag{3.6}$$

Here, the variational distribution is specified by the $\rho_{ijkq}^{(e)}$ values, which can be any configuration of strictly positive probabilities such that

$$\sum_{i<j\in e} \sum_{k,q=1}^{K} \rho_{ijkq}^{(e)} = 1\,.$$

The equality in eq. 3.6 is achieved when

$$\rho_{ijkq}^{(e)} = \frac{u_{ik} u_{jq} w_{kq}}{\sum_{i<j\in e} \sum_{k,q=1}^{K} u_{ik} u_{jq} w_{kq}} = \frac{u_{ik} u_{jq} w_{kq}}{\lambda_e}\,. \tag{3.7}$$

Hence, maximizing $\log p(A|\theta)$ is equivalent to maximizing

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

with respect to both (u, w) and ρ. This can be done by alternating between updating ρ and (u, w), as in the Expectation-Maximization (EM) algorithm. The update for $\theta \in \{u, w\}$ is obtained by setting the partial derivative $\partial \mathcal{L}(\theta, \rho)/\partial\theta$ to 0, which yields the following expressions:

$$u_{ik} = \frac{\sum_{e\in E: i\in e} A_e \rho_{ik}^{(e)}}{C \sum_q w_{kq} \sum_{j\neq i\in V} u_{jq}}\,, \tag{3.8}$$

$$w_{kq} = \frac{\sum_{e\in E} A_e \rho_{kq}^{(e)}}{C \sum_{i<j\in V} u_{ik} u_{jq}}\,. \tag{3.9}$$

The terms $\rho_{ik}^{(e)}, \rho_{kq}^{(e)}$ are defined as:

$$\rho_{ik}^{(e)} = \sum_{j \in e: j \neq i} \sum_{q} \rho_{ijkq}^{(e)},$$
$$\rho_{kq}^{(e)} = \sum_{i<j \in e} \rho_{ijkq}^{(e)},$$

and obtained after updating $\rho_{ijkq}^{(e)}$ according to eq. 3.7. These updates presented in this section are based on maximum likelihood estimation, where we do not set any prior for (u, w). However, we can get Maximum-A-Posteriori estimates (MAP) with similar derivations and complexity by arbitrarily setting priors distributions for the parameters, as we show in Supp. Mat. We comment on how to obtain efficient matrix operations that implement the updates in eq. 3.8 and eq. 3.9 in Section Practical implementation and efficiency.

3.3.2 Identifiability, interpretation and theoretical implications

In the following, we make some observations on relevant aspects regarding the identifiability, interpretation and theoretical implications of the proposed generative model. First of all, the log-likelihood in eq. 3.5 is invariant under permutations of the groups and under the rescaling $u \to c\,u$ and $w \to w/c^2$, for any constant $c > 0$. This observation may raise questions about identifiability of the parameters. However, both permutation and rescaling do not change the composition of the communities nor the relative magnitude of the entries of w, thus the mesoscale structure is not impacted by them. Nevertheless, one can easily make the model identifiable by setting a prior probability on w and considering MAP estimates, see Supp. Mat. for details.

Second, for similar invariance reasons, the constant C can be neglected and absorbed after convergence, by either rescaling $u' = \sqrt{C}\,u$ or $w' = C\,w$. While the forms of the rescaling constants κ_e play no role during inference, as they only enter the updates through the C term, they do instead impact the generative process when sampling hypergraphs from it [2]. For instance, calculations similar to those in Supp. Mat., allow getting a closed-form expression for the average weighted degree when only considering interactions of size k. The resulting formula $\mathbb{E}[d_k^w] = \binom{N-2}{k-2} \frac{k}{\kappa_k N} \sum_{i<j \in V} u_i^T\, w\, u_j$ shows that rescaling the constant κ_k translates into a rescaling of the average degree. Similar considerations apply to the expected number of hyperedges of a given size, and show that the normalization constants κ_e play an important role in determining the expected statistics of the model, and hence of the samples it produces. Generally, the sampling procedure from the generative model in eq. 3.3, allows determining the degree sequence (i.e. the degree

array of the single nodes) as well as the size sequence (i.e. the count of hyperedges for every specified size), which depend on the Poisson parameters and hence on the κ_e normalizers. Alternatively the sampling procedure from our generative model can be conditioned to respect such sequences [2].

Third, it is possible to obtain the analytical expressions of the expected degree of a node i, which evaluates to

$$\begin{aligned}\mathbb{E}\left[d_i^w\right] &= \sum_{e\in\Omega: i\in e} \mathbb{E}[A_e] \\ &= Cu_i^T w \sum_{j\in V: j\neq i} u_j + C' \sum_{j<m\in V: j,m\neq i} u_j^T w u_m \quad ,\end{aligned}$$

where $C' = \sum_{d=3}^{D} \frac{\binom{N-3}{d-3}}{\kappa_d}$ is a constant similar to C, see Supp. Mat. This expression has a relevant interpretation, as it reveals a fundamental difference between simple networks and higher-order systems. Since in dyadic systems $C' = 0$, we can think of the rightmost summand as a term contributing only to higher-order interactions, while the leftmost one is a shift of the expected degree coming from binary interactions only. One can also observe an analogy with networks of interactions in physical systems. In this context, the leftmost summand can be seen as a mean-field acting on node i in a cavity system where the node is hypothetically removed, while the rightmost term acts as a background field generated by all interactions involving any pair of nodes that does not include node i. This background term is peculiar to higher-order systems, as remarked above. Its presence has a relevant effect of building higher-order interactions between nodes in different groups. This can be illustrated with a simple example of a system with assortative w and node i belonging to a different community than all the other nodes. While the leftmost summand yields expected degree zero in dyadic systems, the background field allows i to form on average non-zero edges. Intuitively, this difference is due to the bilinear form in eq. 3.2, that allows observing hyperedges that are not completely homogenous, where there could be a minor fraction of nodes that are in different communities than the majority. Notice that such a generation, allowing for mixed hyperedges, is a desirable feature. On the one hand, it is appropriate to model contexts where individuals have multiple preferences and thus are expected to belong to multiple groups. On the other hand, recent work [123] proves the combinatorial unfeasibility of hypergraphs where all nodes exhibit majority homophily–implying rather uniform hyperedges contained in single communities– and encourages the development of more flexible generative models.

3.3.3 Practical implementation and efficiency

From an optimization perspective, the EM algorithm starts by initializing u and w at random and then repeatedly alternating between the eq. 3.8 and

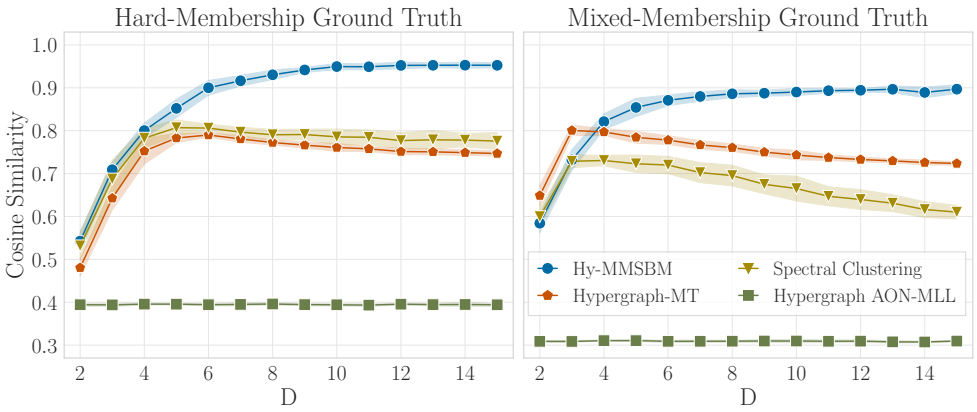


Figure 3.1: Recovery of ground truth community assignments. We measure the cosine similarity between the ground truth and the inferred assignments. We vary the maximum hyperedge size D in synthetic data, and study the cases of hard (left) and mixed (right) ground-truth memberships. When information is scarce, represented by few hyperedges of small maximum size D, our method is comparable to the most efficient approaches currently available. However, as larger hyperedges are considered, our method outperforms competing algorithms, both on hard and mixed-membership planted partitions.

eq. 3.9 updates until convergence of $\mathcal{L}(u, w, \rho)$. This does not guarantee to reach the global optimum, but only a local one. In practice, one runs the algorithm several times, each time from a different random initialization, and outputs the parameters corresponding to the realization with highest log-likelihood $\mathcal{L}(u, w, \rho)$. We provide a pseudocode description of the whole inference procedure in algorithm 2. For all our experiments, we perform MAP inference on the affinity w, setting a factorized exponential prior with rate 1, and maximum likelihood inference on the assignment u. This choice corresponds to the half-Bayesian model presented in Supp. Mat. The updates have linear computational cost, obtained by exploiting the sparsity of most real-world datasets with efficient matrix operations, as we show in Supp. Mat. Overall, the complexity scales as $O(NK + |E|)$, allowing to tackle inference on hypergraphs whose number of nodes and hyperedges was previously prohibitive, see Section Modeling of real data. Another advantage of our inference procedure is that it is stable and reliable for extremely large hyperedges. Due to computational and numerical constraints, previous models were also limited to consider hyperedges with maximal size $D = 25$ [41, 39]. As we illustrate in Section Modeling of real data with an Amazon and a Gene-Disease dataset, large interactions (respectively $D = 9350$ and $D = 1074$) should not be neglected as they provide useful information and substantially boost the quality of inference

Algorithm 2: Hy-MMSBM EM inference

Input: Hypergraph A, training rounds r
Result: Inferred parameters (u, w)

BestLoglik = $-\infty$
BestParams = None
`> Train model r times and choose`
`> realization with best likelihood`
for $t = 1, \dots, r$ **do**
  `> Initialize at random`
  $u, w \leftarrow \text{init}(u, w)$
  `> convergence is attained for a max number of EM steps,`
  `or below a certain change in parameter values`
  while *not converged* **do**
    $u \leftarrow \text{update}(u)$ eq. 3.8
    $w \leftarrow \text{update}(w)$ eq. 3.9
  end
  L = loglik(u, w) eq. 3.5
  if $L >$ *BestLoglik* **then**
    BestLoglik $\leftarrow$ L
    BestParams $\leftarrow (u, w)$
  end
end

3.4 Recovery of ground-truth communities

A standard way to assess the effectiveness of a community detection algorithm is to check if the inferred node memberships match those of a given ground truth. Such ground truth is generally not available for real-world systems [31], whilst it can be imposed as a planted configuration for synthetic data. For this reason, we consider a recently developed sampling method to produce structured synthetic hypergraphs with flexible structures specified in input [2]. For further details, see Supp. Mat.

In fig. 3.1, we generate hypergraphs with an underlying diagonal affinity matrix w (assortative structure) and show the recovery performance for the cases of hard (left) and mixed-membership (right) community assignments. The detailed description of the data generation process is provided in Supp. Mat. We compare our approach with Hypergraph-MT [39], an inference algorithm designed to detect overlapping community assignments and assortative interactions; Spectral Clustering [115], which recovers hard communities via hypergraph cut optimization; and Hypergraph AON-MLL [41], which performs a modularity-like optimization based on a Poisson generative model

with hard memberships. For our comparisons, we compute the cosine similarity between the ground truth and the inferred communities, which is appropriate to measure the similarity for both hard and mixed-membership vectors. A value of zero represents no similarity, while a value of one is attained by completely overlapping vectors. In both cases, we find that our model successfully recovers the ground-truth communities as more information is made available in terms of hyperedges of increasing sizes. This is somehow expected because the generating process of these data reflects the one of our method, and is a sanity check of our maximum likelihood approach. Spectral Clustering and Hypergraph-MT attain comparable cosine similarity scores on hard-membership data (left), while their performances differ when detecting mixed memberships (right), with Hypergraph-MT performing better. This is because Spectral Clustering performs an approximate combinatorial search and can only recover hard communities, while Hypergraph-MT allows for overlapping communities via maximum likelihood inference. The low performance of Hypergraph AON-MLL is explained by its generative assumptions. In fact, AON-MLL assigns the same probability to all the hyperedges containing nodes from more than one community. As most of the hyperedges in this synthetic data are made of nodes from more than one community, the recovery of hypergraph modularity on such systems is close to random. Altogether, such results highlight the effectiveness of the inference procedure, making our model suitable for networked systems with higher-order interactions. Although relevant, the results in fig. 3.1 are just one possible comparison among algorithms with different generative assumptions. Indeed, such assumptions are expected to yield better or worse results depending on the data, and in general, the no-free-lunch theorem implies that no algorithm will consistently outperform all others on all types of data. As a case for this argument, in Supp. Mat. we present additional results on different synthetic data.

3.5 Detectability of community configuration

Previous inference algorithms rely on the strong assumption of assortative community interactions, hampering their ability to model more complex mesoscale patterns observed in the real-world. By contrast, our model allows detecting a variety of different regimes, as it assumes a more flexible w.

Here, we investigate the detection–and detectability–of different assortative and disassortative community structures in hypergraphs, generalizing previous work on pairwise systems [55]. In particular, we generate hypergraphs with hard community assignments, and different community interactions. We take affinity matrices w with diagonal values c_{in} and out-diagonal values c_{out}, and vary both c_{in} and the ratio c_{out}/c_{in}. By fixing the value of c_{out}/c_{in}, we expect higher detectability with increasing c_{in}, as this term regulates the

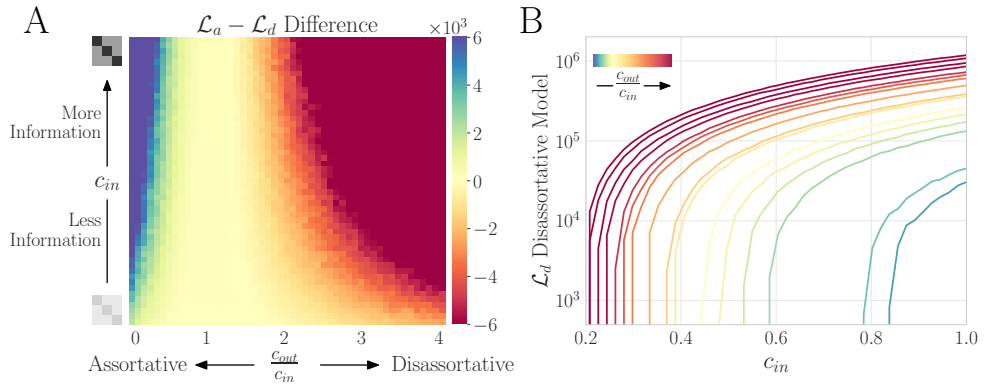


Figure 3.2: Detection of assortative and disassortative community interactions. We generate data where the affinity contains diagonal values c_{in} and out-diagonal c_{out} and measure the ability of our model to detect different assortative and disassortative regimes. (A) Positive (negative) differences in log-likelihood values indicate that the assortative (disassortative) model attains a better fit. An intermediate regime, highlighted in yellow, also emerges. Here, the detectability is compromised due to not having enough structure ($c_{out} \approx c_{in}$) or enough information (low c_{in}). (B) Log-likelihood of the disassortative model. In this case, the model attains better fit for data with marked disassortative structure (darker red).

expected degree and consequently the information contained in the data. On the contrary, for a fixed value of c_{in}, we expect the disassortative model to attain better recovery as the ratio c_{out}/c_{in} increases, due to the stronger inter-community interactions. Details on data generation are provided in Supp. Mat.

We compare the log-likelihoods obtained by the model when the affinity matrix w is initialized as diagonal or full, which we refer to as *assortative* and *disassortative*, respectively. Notice that the multiplicative updates in eq. 3.9 guarantee that, if w is initialized as diagonal, it will remain as such during training. It is also possible that a full matrix will converge to diagonal during inference. Nonetheless, the strong bias of a diagonal initialization restricts the parameter space of the assortative model, facilitating the convergence to better optima for the detection of assortative structures.

Given the log-likelihood of the assortative ($\mathcal{L}_a$) and disassortative ($\mathcal{L}_d$) models, we measure the difference $\mathcal{L}_a - \mathcal{L}_d$ while varying the values of c_{in} and c_{out}/c_{in}. Positive values denote stronger performance of the assortative model, as its likelihood is higher, while negative values favor the disassortative one. We observe that the assortative model attains higher likelihood for low values of c_{out}/c_{in}, when within-community interactions are stronger, as shown in fig. 3.2A. Its performance deteriorates as we increase c_{out}/c_{in} with the disassortative one taking over with higher likelihood values. Furthermore, we can notice an inflexion point at $c_{out}/c_{in} = 1$, where the difference in likelihood between the models is null. While one would expect the disassortative model

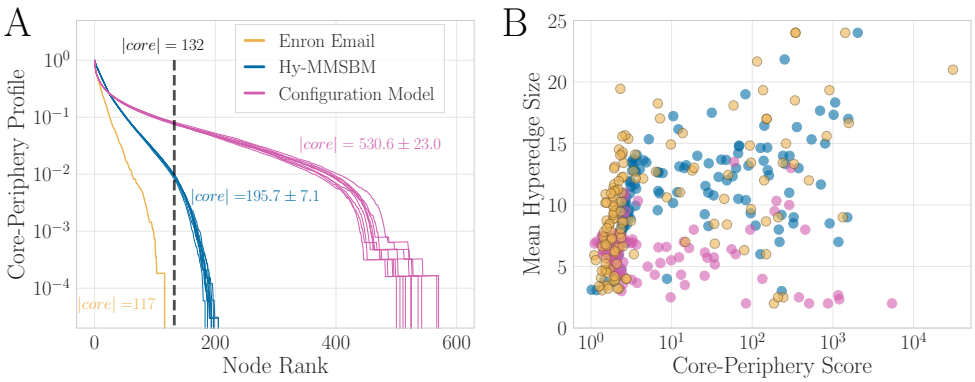


Figure 3.3: Recovery of structural core-periphery information. (A) Core-Periphery profile (eq. 3.10) corresponding to the core-scores computed with HyperNSM on the input Enron email (yellow), ten synthetic samples generated with Hy-MMSBM (blue), and ten synthetic samples generated with a configuration model for hypergraphs (magenta). We plot 600 nodes with the highest core-score in decreasing order, and report the averages and standard deviations of the core dimension for the different datasets. Our method generates samples that closely resemble the property of the input dataset, with an average core dimension close to 132 nodes. (B) Mean size of the hyperedges a node belongs to against its CP score. We observe higher agreement between the data and the inference-based sample generated with Hy-MMSBM. This is also highlighted by the Pearson correlation of the 132 core nodes that is equal to 0.81 ± 0.01 for Hy-MMSBM versus the value of 0.76 ± 0.03 for the samples generated with the configuration model.

to perform better in such a scenario, we highlight that this regime is a challenging and noisy one, as the affinity matrix is the uniform matrix of ones. Hence recovery is difficult and not guaranteed, regardless of the model. We finally notice an increase of $\mathcal{L}_a - \mathcal{L}_d$ with c_{in}, which regulates the strength of the signal and makes it easier to separate the two regimes.

While we expect recovery to improve at more detectable regimes, this may not be observed by only looking at the $\mathcal{L}_a - \mathcal{L}_d$ difference. For this reason, in fig. 3.2B we complement our analysis by plotting only the log-likelihood $\mathcal{L}_d$ attained via the disassortative initialization. In this case we notice that the performance of the disassortative model increases with both c_{out}/c_{in} and c_{in}, as the inter-community interactions get stronger and the expected degree higher. Taken together, our algorithm provides a principled way to extract arbitrary community interactions from higher-order data with varying structural organizations.

3.6 Core-periphery structure

Many real-world systems are characterized by a different mesoscale organization known as core-periphery (CP) structure [124, 125]. Networks characterized by such structure present a group of core of nodes connected among themselves, and often with high degree [126, 127], and a separate

periphery of weakly connected nodes. Recently, methods to study and detect the existence of such patterns in hypergraphs have been proposed [128, 129]. Conceptually, Hy-MMSBM has not been developed with the purpose of core-periphery detection. Nevertheless, we can show its ability in capturing CP structures in hypergraphs through the generation of synthetic data that resemble the core structures of the input dataset.

To measure the recovery of CP structures, we use the method developed by Tudisco et al. [129], HyperNSM, that assigns to each node of a hypergraph a core-score quantifying how close the node is to the core, where higher values denote stronger participation. HyperNSM achieved good performance on synthetic and real-world data and its implementation is extremely efficient.

We analyze the Enron email dataset [130]. Notably, the dataset comes with metadata information identifying a group of core nodes, employees of the organization who send batch emails to the periphery, which in turn only receive emails. This allows us to evaluate the ability of a model to recover a core-periphery structure. In our study, we utilize the dataset used in Tudisco et al. [129] with a planted core set that arises directly from the data collection process, as discussed in Amburg et al. [128] (it is pre-processed by keeping only hyperedges of size $D \leq 25$). The dataset has $N = 4423$ nodes and a core composed by 132 nodes. We apply HyperNSM to quantify the CP structure of the input Enron email dataset, as well as of the samples generated with Hy-MMSBM. To generate the samples, we first run our inference procedure on the Enron email dataset, and then sample hypergraphs distributed according to the obtained u, w parameters. Further details on how to generate the samples are provided in Supp. Mat. For comparison, we also generate samples with a configuration model for hypergraphs [52] and obtain their core-score vectors with HyperNSM as well.

In order to evaluate the quality of the CP assignments in the different samples, we use the CP profile, the metric defined in [129] as:

$$\gamma(S) = \frac{\#\text{ hyperedges with } \textit{all} \text{ nodes in } S}{\#\text{ hyperedges with } \textit{at least} \text{ one node in } S}, S \subseteq V\,. \tag{3.10}$$

For any $k \in \{1, \dots, N\}$ we calculate the value $\gamma(S_k(x))$, where $S_k(x)$ is the set of k nodes with smallest core-score in x. Given its definition, $\gamma(S)$ is small if S is largely contained in the periphery of the hypergraph and it should increase drastically as k crosses some threshold value k_0, which indicates that the nodes in $V \setminus S_{k_0}(x)$ form the core.

In fig. 3.3A we show the CP profiles corresponding to the core-scores computed with HyperNSM on the different datasets, i.e. the input Enron email, the samples generated with Hy-MMSBM, and the samples generated with the configuration model for hypergraphs. We plot 600 nodes with the highest core-score in decreasing order, and for all datasets we notice a sharp drop,

which highlights the existence of a CP structure. The main difference is given by the threshold k_0 at which this drop happens. This determines the dimension of the core. Remember that the data has a core composed by 132 nodes, and when applying HyperNSM on the input data, we obtain a core dimension equal to 117, validating the good core-detection performance of this algorithm. The samples generated with the configuration model present a core with an average of 530.6 nodes, quite far from what observed in the input dataset. On the other hand, Hy-MMSBM generates samples that better resemble the property of the Enron email dataset, with an average core dimension of 195.7 nodes.

To understand the impact of non-pairwise interactions on higher-order CP structure, we also study the connection between hyperedge size and CP score. In fig. 3.3B, we plot the CP score of a given node against the mean size of the hyperedges it belongs to. While we can observe a strong relationship between these two quantities at low CP scores, such regularity disappears in the center of the plot, which contains core nodes and presents a high scattering of hyperedge size values. This unexplained variance is justified by the rich information encoded in the CP score, which jointly depends on different factors related to the topology of the hypergraph. Yet, the scatter plots obtained on the Enron email dataset and the samples generated with Hy-MMSBM have higher similarity than the samples generated with the configuration model. Quantitively, we measure the similarity between the core-scores of the different datasets for the 132 core nodes with the Pearson correlation, a measure $\rho \in [-1, 1]$ of linear correlation between two sets of data. The CP scores of the data have a Pearson correlation equal to 0.81 ± 0.01 with the samples generated with Hy-MMSBM, and of 0.76 ± 0.03 with the samples generated with the configuration model. Similar results are found on the relation between CP score and another structural property, namely the degree of a node, see Supp. Mat.

3.7 Modeling of real data

In this section, we perform an extensive investigation of higher-order real-world systems. As explained in Section Inference and Supp. Mat., the linear-cost EM updates, together with a careful implementation that exploits the sparsity of most datasets, make our method suitable for the analysis of a variety of hypergraphs which were previously inaccessible due to computational constraints. In fact, our method proves to be scalable with respect to both the number of system units and the size of the interactions, improving substantially on competing algorithms currently available in the literature. Moreover, our model is based on a probabilistic formulation, allowing it to perform additional operations and extract information which is not viable via other approaches, such as spectral clustering. First of all, we evaluate

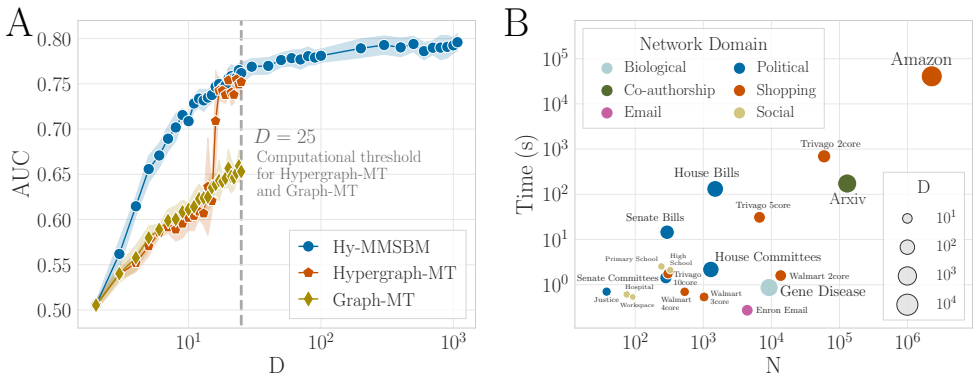


Figure 3.4: Modeling of real data: hyperedge prediction and running time. (A) Quality of hyperedge prediction measured by the AUC score on a Gene-Disease dataset, where nodes are genes, and hyperedges contain genes that are associated with a disease. For Hypergraph-MT and Graph-MT the plot shows a computational threshold at the maximum hyperedge size $D = 25$. Hy-MMSBM attains the highest scores and is able to model the entire hypergraph, up to $D = 1074$. (B) Running time of Hy-MMSBM for a variety of real-world datasets. The node represents the data domain. Both N and D are in log-scale. The corresponding AUC scores are reported in table 3.1.

the quality of fit of various community detection methods based on their hyperedge prediction capabilities on a Gene Disease dataset, where nodes are genes, and interactions contain genes that are associated with a disease. To this end, we utilize the AUC measure, a link prediction metric defined as follows: given a randomly selected observed edge, and a randomly selected non-observed one, the AUC $\in [0, 1]$ computes the number of times that the generative model assigns a higher probability to the observed edge. Here, we split the datasets into train and test subsets, where the train sets are used to estimate the parameters, and we evaluate the prediction performance in terms of AUC on the test sets, see Supp. Mat. for details. Scalability with respect to hyperedge size is a crucial aspect of models for higher-order data. However, due to computational and numerical constraints, previous methods are limited to considering interactions of moderate size only, possibly causing a loss of information and a biased representation of the full system. In contrast, our model is able to efficiently process all the information provided in the dataset, reliably scaling to hyperedges of size of the order of the thousands. In fig. 3.4A we compare our method with other probabilistic approaches with hyperedge prediction capabilities. When only small interactions are considered, our model outperforms the competitive algorithms. At the computational limit of other approaches $D = 25$, Hypergraph-MT and our model attain a similar score, signalling the importance of considering large interactions. Beyond this computational threshold, our method continues to exploit the information provided by interactions among a growing number of units up to the maximum size observed of $D = 1074$, which

results in an AUC score of 0.79.

We then extend our analysis to a variety of datasets from different domains, as described in fig. 3.4B. For each dataset we show the inference running time as a function of the number of nodes N and the size of the largest hyperedge D. The AUC scores, reported in table 3.1 and ranging from 0.74 to 0.98, show that the model generally yields a good fit and predicts the existence of hyperedges reliably. While these scores are on average aligned with those of other existing algorithms [39], the running time of our model is orders of magnitude lower. This allows studying very large hypergraphs such as the Arxiv, Trivago 2core and Amazon datasets, containing up to millions of nodes and hyperedges. Overcoming the resulting computational challenges, our method allows the efficient modeling of a variety of previously unexplored datasets, which, to the best of our knowledge, could not be tackled by competing higher-order community detection algorithms.

Taken all together, these results show the effectiveness of our model in tackling datasets of small and large dimensions, both in terms of quantitative performance and computational scalability, and make Hy-MMSBM a valid tool for the study of complex higher-order systems.

Discussion

In this work we have developed a probabilistic framework to model hypergraphs. Our method allows performing inference on very large hypergraphs, detecting their community structure and reliably predicting the existence of higher-order interactions of arbitrary size. When compared to other available methods on synthetic hypergraphs with known ground truth, for both hard and mixed assignments our model attains the most efficient recovery of the planted partitions. Moreover, compared to previous proposals, Hy-MMSBM relies on less restrictive assumptions on the latent community structure in the data, and is thus able to detect configurations, such as disassortative community interactions, which could not be previously identified. Furthermore, our method is extremely fast. Its efficient numerical implementation exploits optimized closed-form updates and dataset sparsity and has linear cost in the number of nodes and hyperedges. The resulting formulas are also numerically stable, not resulting in under- or overflows during the computations. Such numerical stability carries over to extremely large interactions, a substantial improvement over the computational threshold of previous methods, allowing to explore higher-order datasets with millions of nodes and interactions among thousands of units, that could not be previously tackled.

There are several directions for future work. From a theoretical perspective, our proposed likelihood function is based on a bilinear form for capturing

	N	$\lvert E\rvert$	D	K	AUC
Justice	38	2,826	9	4	0.909 ± 0.008
Hospital	75	1,825	5	2	0.767 ± 0.013
Workspace	92	788	4	5	0.741 ± 0.015
Primary School	242	12,704	5	10	0.832 ± 0.002
Senate Committees	282	301	31	30	0.926 ± 0.023
Senate Bills	294	21,721	99	13	0.921 ± 0.002
Trivago 10core	303	3,162	14	11	0.960 ± 0.005
High School	327	7,818	5	17	0.879 ± 0.007
Walmart 4core	532	2,292	10	4	0.837 ± 0.013
Walmart 3core	1,025	3,553	11	4	0.825 ± 0.010
House Committees	1,290	335	81	25	0.939 ± 0.015
House Bills	1,494	54,933	399	19	0.946 ± 0.001
Enron Email	4,423	5,734	25	2	0.835 ± 0.009
Trivago 5core	6,687	33,963	26	30	0.962 ± 0.001
Gene Disease	9,262	3,128	1,074	2	0.828 ± 0.010
Walmart 2core	13,706	19,869	25	2	0.788 ± 0.004
Trivago 2core	59,536	140,698	52	100	0.863 ± 0.002
Arxiv	130,024	172,173	2,097	10	0.884 ± 0.001
Amazon	2,268,231	4,242,421	9,350	29	0.978 ± 0.002

Table 3.1: AUC scores on real datasets. We report the number of nodes N, number of hyperedges $|E|$, maximum hyperedge size D, number of communities K and AUC scores attained by our method on 19 large-scale real-world hypergraphs. The results are averages and standard deviations over 10 random test sets, and the value of K is chosen via cross-validation, see Supp. Mat.

dependencies within the hyperedges, a key ingredient for ensuring both mixed-membership nodes and fast inference. A possible extension would be to consider alternative likelihood definitions where the probability of the hyperedges is determined by multilinear forms, which would in principle allow capturing more complex interactions within the hyperedges. Similarly, here we have assumed the hyperedges to be independent conditioned on the latent variables. Relaxing this assumption may ameliorate the expressiveness of the model, allowing to capture topological properties that involve more than two hyperedges, as already observed in the case of networks [131, 132, 133]. From an algorithmic perspective, there are different questions that may allow further stabilizing and improving the inference procedure. Among these, the propensity of different initial conditions to be trapped in local optima during EM or MAP inference has not yet been investigated. Devising suitable initialization procedures or parameter priors to favor different membership types, as done in other works [134], offers a promising path in this direction. Finally, we have considered here a standard scenario where the input data is a list of hyperedges, and these are provided all at once. Other approaches may be needed in case of availability of extra information such as node attributes [135, 37] or for dynamic data [136].

Altogether, our work provides an accurate, flexible and scalable tool for the modeling of very large hypergraphs, advancing our ability to tackle and study the organization of real-world higher-order systems.

Chapter 4

Framework to Generate Hypergraphs with Community Structure

Abstract

In recent years hypergraphs have emerged as a powerful tool to study systems with multi-body interactions which cannot be trivially reduced to pairs. While highly structured methods to generate synthetic data have proved fundamental for the standardized evaluation of algorithms and the statistical study of real-world networked data, these are scarcely available in the context of hypergraphs. Here we propose a flexible and efficient framework for the generation of hypergraphs with many nodes and large hyperedges, which allows specifying general community structures and tune different local statistics. We illustrate how to use our model to sample synthetic data with desired features (assortative or disassortative communities, mixed or hard community assignments, etc.), analyze community detection algorithms, and generate hypergraphs structurally similar to real-world data. Overcoming previous limitations on the generation of synthetic hypergraphs, our work constitutes a substantial advancement in the statistical modeling of higher-order systems.

4.1 Introduction

Over the last decades, networks have emerged as a fundamental tool to describe complex relational data in nature, society and technology [78]. Indeed, most real-world systems are nowadays known to be characterized by a highly non-trivial organization, which includes triadic closure and high clustering [137], low diameter and an efficient communication structure [138], and unequal degree distributions [139]. Noticeably, many systems reveal the existence of modules or communities, where nodes are naturally clustered in different groups based on their patterns of connections [29]. Identifying communities is an important task that allows performing various downstream

analysis on networks, describing the roles of nodes and, generally, providing a low dimensional representation of possibly large systems. Since the seminal papers by Newman et al. [140] and Lanchichenetti et al. [141], the problem of generating synthetic data for highly structured graphs with prescribed features has attracted enormous interest in the community. On the one hand, these models have led to tremendous improvements in evaluating which community detection algorithms perform best at a given task [142]. On the other hand, they have allowed the reliable generation of large synthetic data samples, useful to analyze non-trivial statistics from single instances of real networks and systematically investigate the impact of mesoscale structure on dynamical processes on graphs [143, 144]. This methodology has been applied to different domains, including studies on polarization on social media [145], percolation thresholds in brain networks [146], and structural and covariate information [147, 148].

Despite their success, recent evidence suggests that graphs can only provide a limited description of reality, as links are inherently limited to describe pairwise interactions [22, 79, 16, 80]. By contrast, non-dyadic higher-order interactions have been observed across different domains, including the human brain [19, 85, 86], collaboration networks [81], species interactions [20], cellular networks [21], drug recombination [82], and face-to-face human [83] and animal [84] interactions. Interestingly, such higher-order interactions naturally lead to the emergence of new collective phenomena in synchronization [88, 89, 90, 91, 92] and contagion [94, 95, 96] dynamics, diffusive process [149, 87] and evolutionary games [97, 98]. Hypergraphs [100], where hyperedges encode interactions among an arbitrary number of system units, are a natural framework to describe relational data beyond the pair [22]. In the last few years many tools have been developed to characterize the higher-order organization of real-world hypergraphs, including new centrality measures [101, 102], higher-order clustering [103] and motif analysis [104], hypergraph backboning [106], hyperedge prediction [39], methods to infer higher-order interactions from low-order data [108]. In particular, several tools to extract higher-order communities have been proposed, either based on flow distribution [113, 114] or statistical inference frameworks [39, 41].

Nevertheless, how to generate structured hypergraphs is still an open problem. The few currently available models mainly focus on "unstructured" higher-order generalizations of the configuration [150, 151, 52] and the Erdos-Renyi model [152], or on growth models for hypergraphs [153, 154, 155]. A different perspective is that of relational hyperevent models [156], which specify event rates based on hyperedge statistics for hyperedges to exist, similarly to what exponential random graphs do for networks [157, 158]. All these approaches, however, do not account for community structure, hence are of limited usage when it comes to reproducing the complex mesoscale organization of real-world higher-order systems. Recent works introduced latent

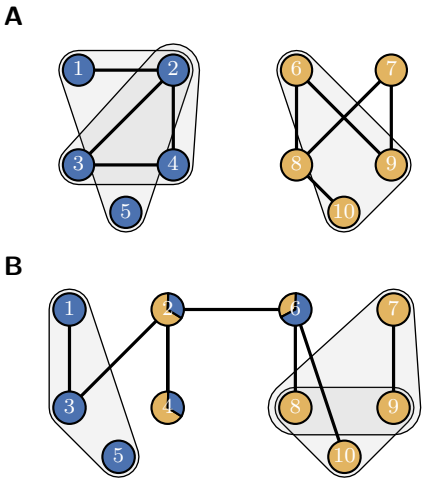


Figure 4.1: Sampling hypergraphs with community structure. A pictorial representation of two small hypergraphs with $N = 10$ nodes, $K = 2$ communities, and (A) hard or (B) overlapping membership assignment. Every node's membership assignment $u_i = (u_{i1}, u_{i2})$ is represented as a pie chart. Nodes with a single color have hard assignments, mixed pie charts represent overlapping assignments. Due to the likelihood in eq. 4.2, nodes with overlapping assignments are more likely to belong to between-community interactions.

variables models to infer community structure in hypergraphs [159, 39, 41], however they do not explain how to sample from the generative model. Indeed, while sampling and inference are often studied jointly in standard networks, these two tasks present distinct computational and theoretical challenges in the case of hypergraphs.

In this work, we provide a principled and general framework to sample hypergraphs. In particular, our method allows flexible sampling of higher-order networks with prescribed microscale and mesoscale features, controlling the distribution of node degrees and hyperedge sizes, as well as specifying arbitrary community structure (e.g. hard vs overlapping membership, assortative vs disassortative, etc.). The method is highly efficient, and scales well with the number of nodes, hyperedges, as well as hyperedge size, making it suitable for the analysis of real-world systems. In the following, we first introduce our generative model and sampling strategy. Then, we extensively characterize the hypergraphs obtained by investigating the phase space associated with the different structural parameters. Finally, we show how to utilize our method to analyze the structural and statistical properties of real-world data.

4.2 Generative model

We consider hypergraphs $\mathcal{H}(V, E)$ consisting of N nodes $V = \{1, \dots, N\}$ and a hyperedge set E, where each hyperedge $e \in E$ describes an interaction

among an arbitrary set of unique nodes, i.e. $e \subseteq V$, and $|e|$ is the hyperedge size. The degree of a node i, i.e. the number of hyperedges it belongs to, is denoted as d_i. Similarly, we define the degree sequence $d = \{d_1, \ldots, d_N\}$ as the vector of node degrees and the size sequence $k = \{k_1, \ldots, k_D\}$ as the count of hyperedges per hyperedge size [52]. We consider hyperedges of arbitrary sizes, up to a maximum of $D \leq N$, and denote the space of all possible such hyperedges with Ω. We assume positive and discrete hyperedge weights, encoded using a vector $A \in \mathbb{N}^{|\Omega|}$, so that $E = \{e \in \Omega : A_e > 0\}$.

Our sampling approach introduces a flexible way to generate highly structured weighted hypergraphs with mesoscale structure, where hyperedges are generated probabilistically and nodes belong to K communities. Specifically, each node $i \in V$ is assigned a K-dimensional membership vector u_i, where we allow $u_{ik} \geq 0$ for the general case of soft membership, where nodes can belong to multiple communities. The particular case of hard membership assignment, where a node can only belong to one community, is recovered by setting only one non-zero entry for u_i. In fig. 4.1 we illustrate these two cases by showing two small hypergraphs with hard or overlapping community structure. The non-negative symmetric $K \times K$-dimensional *affinity* matrix w regulates the interactions between communities. Classic patterns are assortative affinity matrices, with dominant diagonal signaling stronger inter-community interactions, and disassortative ones, where the out-diagonal terms have higher magnitude. For any given hypergraph, we define the following likelihood function:

$$\begin{aligned} p(\mathcal{H}; w, u) &= \prod_{e \in \Omega} p(A_e; u, w) \\ &= \prod_{e \in \Omega} \text{Pois}\left(A_e; \frac{\lambda_e}{\kappa_{|e|}}\right), \end{aligned} \tag{4.1}$$

where

$$\lambda_e := \sum_{i<j \in e} u_i^T \, w \, u_j \; = \sum_{i<j \in e} \sum_{k,q=1}^{K} u_{ik} w_{kq} u_{jq}. \tag{4.2}$$

This parameterization allows generating hypergraphs under different scenarios, e.g. with assortative or disassortative community structures, and is reminiscent of those used in probabilistic models for pairwise networks [36, 160] and in variants of non-negative tensor factorization as used in the machine learning community [161, 162] when $D = 2$. In addition, restricting our model to $D = 2$ and $\kappa_2 = 1$ recovers the canonical Poisson stochastic block model [163]. The parameter $\kappa_{|e|}$ is a normalization factor and is a function of the size $|e|$ of the hyperedge e only (i.e. it only depends on the size of the interaction, and not on the nodes involved in it). These constants regulate the expected statistics of the model, such as expected degree and hyperedge

size distribution. In general, any choice of $\kappa_d > 0$ yields a well-defined probabilistic model. We illustrate sensible values for κ_d in Supp. Mat.

Alternative generative models for hypergraphs have been recently proposed. In particular, the works of Chodrow et al. [41] and Contisciani et al. [39] can be more closely compared to the model in eq. 4.1, since they are both based on factorized Poisson likelihoods based on communities. The former work assumes sufficient statistics only evaluated on hard community assignments and we are not aware of any computationally efficient sampling procedure from the relative generative process. The model of Contisciani et al. [39], instead, bears closer resemblance to the one proposed in this paper. The main difference lies in the specific form of the Poisson means, which, for every hyperedge e, are based on a product of $|e|$ terms, as opposed to the bilinear form in eq. 4.1. Despite the similar generative process, the tools utilized in this work cannot be straightforwardly applied to that model, as closed-form statistics and approximate Central Limit Theorem results cannot be derived in the same manner.
More generally, the primary goal of the aforementioned models is to infer hypergraph structure, leaving the problem of sampling unsolved. While our model is also well suited to efficiently infer hypergraph structure, as we illustrate in Ruggeri et al. [1], the primary objective of this work is to demonstrate how we can effectively sample from its probability distribution. This key model's capability makes it possible to generate highly structured synthetic data with higher-order interactions. This is a key advancement for practitioners handling hypergraph data and follows influential work on such a topic for pairwise networks [140, 141].

4.3 Sampling hypergraphs

We now propose an efficient way to sample hypergraphs from the generative model defined in eq. 4.1. Such a task is far from being straightforward. To see why, let us consider a pairwise network model, where the configuration space is of size $|\Omega| = N^2$, and compare it with our higher-order problem. In the former case, generation is feasible by simply exploring every single edge separately and sampling from the relative Poisson distribution. In the latter case, however, the rapid growth of the Ω space renders both naive sampling techniques and Markov Chain Monte Carlo (MCMC) algorithms inapplicable. Here, we propose a solution to this challenge using approximate sampling. In the following, we focus on the intuition behind our method and illustrate relevant usage examples. For a more technical description, we defer to Supp. Mat.

4.3.1 Sampling algorithm

Our sampling procedure follows three consecutive steps:

Sampling node degrees and hyperedge sizes. The first sampling step consists of approximately sampling the d and k vectors for a given choice of community memberships u and affinity matrix w. Then, we use these two quantities to draw a first proposal of a binary hypergraph defined by the array $A^b \in \{0,1\}^{|\Omega|}$. More in detail, we first approximate $p(d,k;u,w) \approx p(d;u,w)\, p(k;u,w)$ and then use the Central Limit Theorem (CLT) to sample from $p(d;w,u)$ and $p(k;w,u)$ separately. We note that these are the only approximations needed in the whole sampling routine. We elaborate more on their validity in Supp. Mat. After sampling the d,k sequences, we combine them into a first binary hypergraph configuration (i.e. a list of hyperedges) to be passed in input to the next sampling step. Intuitively, we incrementally build a hyperedge list until exhaustion of both sequences, starting by first taking the nodes with the highest degrees. If the two sequences are not compatible, i.e. it does not exist a hypergraph that satisfies both, one can choose which of the two sequences to preserve during the hyperedge list construction. Such sequence will be exactly replicated, while the other will be modified to construct the first list proposal. Notice that the recombination problem has connections with the Havel-Hakimi algorithm [164] and the Erdös-Gallai Theorem [165]. Hence, the algorithm we propose for this task is a technical novelty of independent interest. We explain the algorithm in detail and present a pseudocode for it in Supp. Mat.

Sampling hyperedges. In this second step, we sample the binary hyperedges A^b_e, conditioned on d and k, using a MCMC routine. This works by continuously mixing the hyperedges starting from the initial proposal A^b obtained at step a. The main tool utilized here is the reshuffling operator introduced in Chodrow et al. [52]: given two hyperedges e_1, e_2, reshuffle the nodes not belonging to the intersection $e_1 \cap e_2$ to obtain two new hyperedges e_1', e_2'. Then, accept or reject the new proposal according to the Metropolis-Hastings algorithm [166], whose acceptance rates depend on the Poisson means $\lambda_{e_1}/\kappa_{e_1}, \lambda_{e_2}/\kappa_{e_2}$ and consequently on the u, w parameters. Due to the properties of the reshuffling operator the new hyperedges e_1', e_2' have same sizes as e_1, e_2, hence the sequences d and k are preserved. Intuitively, the Markov chain achieves good mixing owing to conditioning on (d,k), which restricts the space of the possible configurations.

Sampling hyperedge weights. In the third and final step, we sample the weights A_e from $p\left(A_e | A^b_e = 1\right)$. This conditional distribution is a zero-truncated Poisson with mean $\lambda_e/\kappa_{|e|}$. A related efficient sampling procedure based on inverse transform sampling is proposed in Supp. Mat.

Altogether, the three sampling steps described above correspond to the following probabilistic decomposition:

$$p\left(A;u,w\right)=p\left(A|A^{b};u,w\right)\;p\left(A^{b}|d,k;u,w\right)\;p\left(d,k;u,w\right)\;. \tag{4.3}$$

In passing, we note that a single pair (d,k) is uniquely defined by a hypergraph A (or its binary counterpart A^b). For this reason, d and k do not appear on the left-hand side of Equation 4.3.

We provide the pseudocode of the sampling procedure in algorithm 3 and provide an open-source implementation at github.com/nickruggeri/Hy-MMSBM.

Algorithm 3: Sampling algorithm.
a: Lines 1-3; *b*: Lines 4-10; *c*: Line 11.

Input: Number of communities K, memberships u, affinity w, MCMC burn-in steps n_b and intermediate steps n_i, number of samples S.
Result: $\left\{A^{(s)}\right\}_{s=1,\dots,S}$

```
Sample binary degree sequence d ~ p(d; u, w)
Sample size sequence k ~ p(k; u, w)
Create first proposal A^b from d, k
for i = 1, ..., n_b do
    A^b ← reshuffle(A^b), accept according to Metropolis-Hastings,
      depending on (u, w)
end
for s = 1, ..., S do
    for i = 1, ..., n_i do
        A^b ← reshuffle(A^b), accept according to Metropolis-Hastings,
          depending on (u, w)
    end
    sample A^(s) ~ p(A|A^b; u, w)
    yield A^(s)
end
```

4.3.2 Additional user input

The sampling procedure described above only requires the community assignments u and affinity matrix w as generative parameters. However, a practitioner may desire to generate hypergraphs with specific features, such as a given degree or hyperedge size sequence. Our model allows doing so naturally, either by providing such statistics as additional input or by tuning the generative parameters prior to sampling. More precisely, one can skip the initial step and simply fix d or k (or both) as input instead of sampling

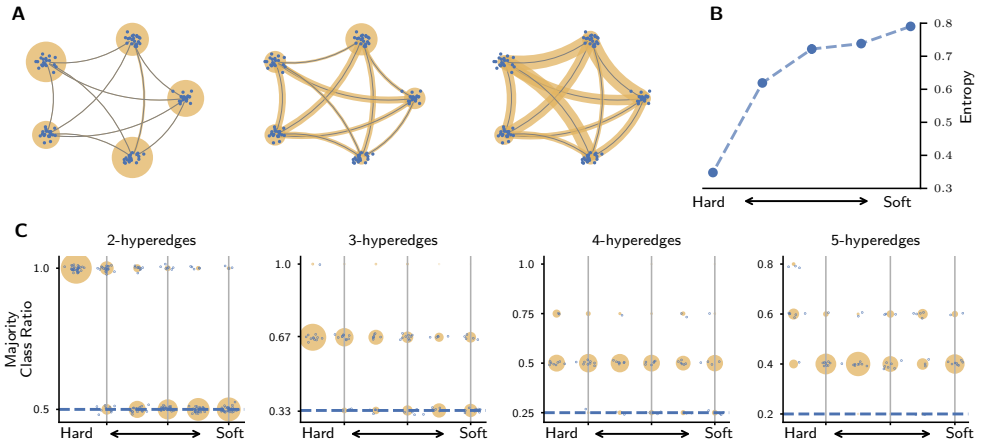


Figure 4.2: Sampling hypergraphs with hard and soft community assignment. (A) We sample hypergraphs from a model with $K = 5$ equally-sized communities, an assortative affinity matrix w, and different node community memberships u (from hard to soft). The five vertices represent different communities, the thicknesses of the edges and circles are proportional to the interaction strength between and within communities, see Supp. Mat. for details. (B) The entropy of community memberships grows as increasingly overlapping configurations are considered. (C) We show the maximum assignment ratio (the relative number of nodes belonging to the majority class for each hyperedge) across hyperedge sizes. The set sizes are proportional to the amount of hyperedges with a given maximum assignment ratio.

them. As explained in section 4.3.1, these quantities are guaranteed to be preserved in the sampled hypergraphs. Algorithmically, this corresponds to starting directly from line line 3 in algorithm 3.

In some cases, one might be interested in replicating the d^{data}, k^{data} sequences observed in a real hypergraph dataset. In such a simplified scenario, one can condition on the (binarized) hyperedges of the data, and proceed by directly mixing them via the MCMC procedure in the second sampling step. Since the hyperedges define the degree and size sequences, these will be preserved and identical to those of the real data, while the samples will come from the model's probability distribution. As per eq. 4.3, the MCMC procedure will yield samples from $p(A^b|d^{data}, k^{data}; u, w)$. Notice that, in general, conditioning on any given sequence d or k might yield samples A outside the high-density areas of the distribution. This is a desirable feature, as it allows the user to further specify constraints and sample hypergraphs that would otherwise be far from the typical samples obtained without conditioning [167].
Finally, with our model we can obtain closed-form expressions for relevant hypergraph properties in terms of u and w, e.g. the expected degree of nodes, as shown in Supp. Mat. This means that, by tuning the u, w parameters, such properties can be specified prior to sampling. We illustrate some examples of this procedure in section 4.4.

4.4 Synthetic Data

In this section, we illustrate how the generative parameters u and w can be tuned to sample hypergraphs with desired structures at a micro (node and hyperedge) and mesoscale (community structure and hypergraph-level statistics) level. We release ready-to-use examples of these synthetic datasets along with the open-source implementation.

4.4.1 Community assignment

We begin by showing how varying the overlap in the membership assignments u leads to different intra and inter-commmunity structure. In fig. 4.2 we tune the assignments from hard (u_i has only one non-zero entry), to soft ($u_i > 0$ for multiple entries), and highlight the strength of the interactions between and within communities by varying the thickness of edges and circles. We include additional details on data and plot generation in Supp. Mat. As memberships vary from hard to soft (left to right), edges become thicker and circles smaller, as inter(intra)-community interactions increase (decrease). Quantitatively, we compute the entropy $-\sum_{k=1}^{K} r_k \log r_k$, where r_k is the ratio of nodes belonging to community k. In mixed-membership settings, one can extract a proxy for a hard assignment for node i by selecting the $k = \arg\max_k u_{ik}$; we use this to compute r_k. Lower entropy denotes hyperedges whose nodes mostly belong to the same communities, higher values denote hyperedges with nodes distributed across different communities. In fig. 4.2B we show how the entropy of the community distribution grows as we sample from increasingly overlapping models. We also study the partition in communities of nodes belonging to hyperedges of different sizes. For each hyperedge we compute the ratio of nodes that belong to the majority class. For example, in a hyperedge of size 5 with two nodes in class 1 and three in class 2, the majority class is 2, yielding a majority class ratio of 3/5. fig. 4.2C shows how this ratio decreases going from hard to soft memberships, illustrating the heterogeneity of the nodes' communities across hyperedges of different sizes.

4.4.2 Affinity matrix and heterogenous community size

While varying u acts on the propensity of individual nodes to participate in groups, the affinity matrix w controls the density of interactions within and between communities. The generative model in eq. 4.1 is well-defined for any non-negative symmetric affinity matrix w, allowing simulating various structures by properly tuning its entries. To illustrate the generation of hypergraphs with different affinity matrices, here we consider a range of matrices that start from diagonal (assortative) to gradually move to the uniform matrix of ones (disassortative), and rescale them to obtain an expected degree of

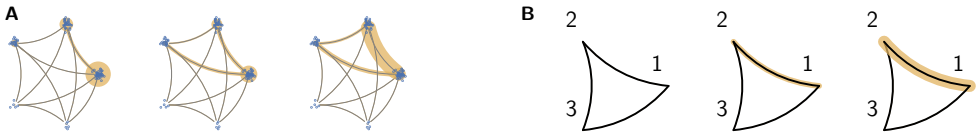


Figure 4.3: Sampling hypergraphs with assortative and disassortative affinity and heterogeneous community size. (A) We sample hypergraphs with five communities of different sizes and hard membership assignments. We vary the affinity matrix w from assortative (left, diagonal) to disassortative (right, uniform matrix filled with ones). The five vertices represent the communities, the thicknesses of the edges and circles are proportional to the interaction strength between and within communities, respectively. (B) We vary the affinity w from diagonal (left) and increase its entries w_{12}, w_{21} (right) for $K = 3$ equally-sized communities. The three vertices represent the communities and the thickness of the edges and circles is proportional to the strength of the interactions between and within communities respectively, see Supp. Mat. for details.

five. For simplicity we set the assignments u to hard membership. The method is well suited to sample not only homogenous hypergraphs, but also higher-order networks with heterogenous distribution of the community size. Here we consider five communities with different sizes. As shown in fig. 4.3A, moving from an assortative to a disassortative configuration, the inter-community interactions strengthen substantially. Further, notice that the strengths of the interactions are influenced by the heterogeneity of the community size, as larger communities are expected to participate in more interactions.

It is also possible to tune individual entries of the affinity matrix w. In particular, in fig. 4.3B we perform an experiment where we start from a diagonal matrix, and gradually increase only the w_{12} (and w_{21}) entries, using three equally-sized communities. In this way, only the expected interactions between communities 1 and 2 are affected, while interactions among other communities are left unchanged.

4.4.3 Analzying community detection

One of the most useful applications of generating synthetic data with a desired underlying structure is the possibility to evaluate how competing algorithms perform on a given task that depends on the structure under control. In fact, when synthetic data with a known structure is available, it is possible to quantitatively compare the outcome of various algorithms and measure their ability to recover ground truth information. In network science, a classical and much investigated problem is assessing the ability of community detection algorithms to extract meaningful partitions of the network [141]. For higher-order networks, the current lack of sampling methods for synthetic data with flexible community structure has led to a variety of custom-built examples, which renders comparison difficult and subject to individual choices [41, 42, 39].

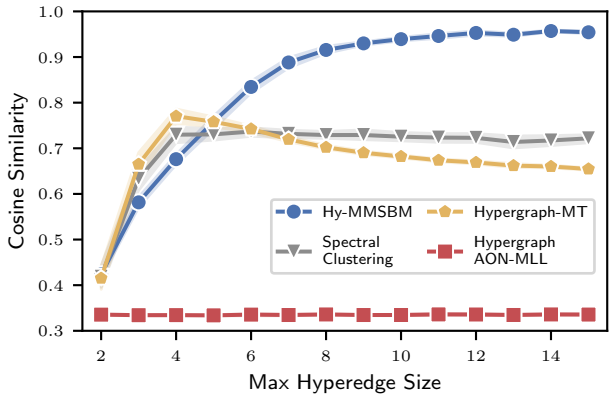


Figure 4.4: Evaluating higher-order community detection algorithms. We sample hypergraphs to test the ability of different higher-order community detection algorithms to recover well-defined planted partitions. We consider hypergraphs with $N = 500$ nodes, $K = 3$ equally sized assortative communities and hard assignments. We plot the cosine similarity between the inferred partitions and the ground truth as a function of the maximum hyperedge size. Additional details on the data generation are given in Supp. Mat.

In this section, we show how our synthetic data can be utilized to analyze the behavior of some of the current algorithms for higher-order community detection. To this end, we generate hypergraphs with assortative structure and hard community assignments, and perform inference with a variety of methods, namely Hy-MMSBM [1], Hypergraph-MT [39], spectral clustering [115] and hypergraph modularity [41]. In fig. 4.4, we show the cosine similarity of the inferred communities with the ground truth as a function of the maximum hyperedge size. As can be observed, Hy-MMSBM attains the best performance when group interactions beyond a critical size are introduced, successfully recovering the ground truth assignments. Hy-MMSBM is a flexible inference tool whose inference procedure is based on the same generative model described in eq. 4.1, and generally able to extract mixed-membership assignments for arbitrary (e.g. assortative or disassortative) community structure. Other algorithms attain varying scores, which might be explained by the different assumptions of the underlying models. For example, Hypergraph-MT is designed to extract overlapping communities, while spectral clustering can only be utilized for the detection of hard assignments. As such, the latter can be expected to perform well only in scenarios where interactions are dictated by hard communities, while the former can be employed when nodes may belong to more than one module. Considerations of this type can be useful to compare the alignment of different algorithms with the data generation procedure, which is expect to correlate with their performance on such data. However, due to the additional optimization and implicit bias of most algorithms, it is sometimes unclear to know beforehand on what types of data each algorithm will perform well empirically. We

believe that synthetic data with known ground-truth structure can provide useful diagnostic and comparison tools in this direction.
Procedures like the one presented in this section can be used to understand the limitations and strengths of different algorithms, allowing researchers to effectively test new proposals in different scenarios by varying the properties of the samples generated with our method, e.g. the degree of assortativity.

4.4.4 Computational cost

Our sampling method is highly efficient and computationally scalable. We analyze the cost of our sampling strategy by discussing the cost of the individual sampling steps. The first step, consisting of sampling the degree and size sequences, can be cheaply performed in $O(N)$ time. In fact, to sample the d, k sequences we need to compute the mean and standard deviations defined in the Central Limit Theorem, and thus draw the sequences from the relative Gaussian distributions. These operations have linear cost, see Supp. Mat. In the second step we first combine the sampled d, k sequences into a first hyperedge configuration, and successively mix the hyperedges via MCMC. Generally, while the number of Markov chain steps needed for mixing is a function of N and $|E|$ [168], it is difficult to specify a pre-defined number. In fig. 4.5, we fix $n_b = 100000$ burn-in steps and $n_i = 20000$ intermediate steps between samples, which is a default value we utilized in most experiments. Nonetheless, the main cost we observe in this case is that prior to MCMC, i.e. the producing the first hyperedge configuration from the sequences. Empirically, such step dominates the computational cost. Finally, the third step consists of sampling the non-zero weights according to $p(A|A^b; u, w)$. The cost of this operation is proportional to the number of hyperedges $|E|$; for sparse hypergraphs—and as often observed in real data—this is comparable to N.

Empirically, we find the CLT approximations to be working well. Nevertheless, one could further improve on the quality of sampling by drawing the pairwise edges from their *exact* Poisson distribution (eqs. 4.1 and 4.2), with cost $O(N^2)$, and resorting to approximations only for interactions of order three or greater. This is of particular help when sampling denser hypergraphs: since the MCMC does not necessarily guarantee non-repeated hyperedges, sampling directly the order-two interactions reduces the probability of repeated edges. For higher-order interactions, the probability of repetitions is negligible, in particular in sparse regimes [52]. Indeed, in all the experiments presented in this paper we sample the order-two interactions directly, and resort to the CLT approximations for hyperedges of order at least three.

In fig. 4.5 we investigate the efficiency of both exact and solely CLT-based sampling and observe the difference to be negligible. As discussed above,

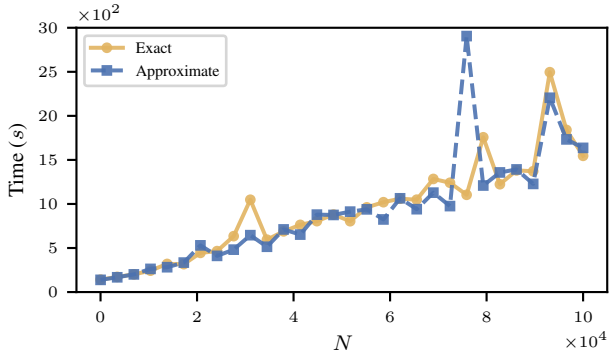


Figure 4.5: Computational complexity and scalability. We plot the computational cost of our sampling model for sparse hypergraphs as a function of the system size N. Our model is highly efficient, as it allows sampling of sparse hypergraphs of dimensions up to $N = 10^5$ nodes in less than one hour. We show results for hypergraphs with fixed expected degree equal to 5, both for an exact (solid line with circles) and an approximate approach (dashed line with squares) based on central limit theorem sampling of dyadic interactions. Here, we utilize $K = 5$ communities and unconstrained maximum hyperedge size $D = N$.

this is a consequence of the higher computational effort required in other sampling steps. Altogether, our model is highly efficient, as it allows sampling sparse hypergraphs of dimensions up to 10^5 nodes in less than one hour.

4.5 Real Data

4.5.1 Modeling real-world systems

In this section we aim at sampling hypergraphs that mimic the community structure of a given dataset. To this end, we proceed as follows. First, we infer the affinity matrix w and community assignments u using the Hypergraph-MT algorithm [39] on the real data. Since this algorithm returns a (diagonal) matrix w_d for every possible hyperedge size d, we take their element-wise geometrical mean to construct the matrix w utilized in eq. 4.2. Notice that a similar approach could have been taken utilizing the Hy-MMSBM algorithm, which employs the same probabilistic model of our sampling method, as explained in section 4.4.3. To highlight the flexibility of our methodology, which can be applied along with any community detection methodology, here we utilize Hypergraph-MT. In fact, our method accepts input parameter w and u regardless how these are obtained; in particular, these can be obtained by using different inference methods applied to the input data. Our method is capable of generating synthetic data conditioning on the desired input communities and affinity matrix. As such, it can be used in a complementary way together with community-based method focusing solely on inference. Second, we condition the degree and size sequences by providing in input

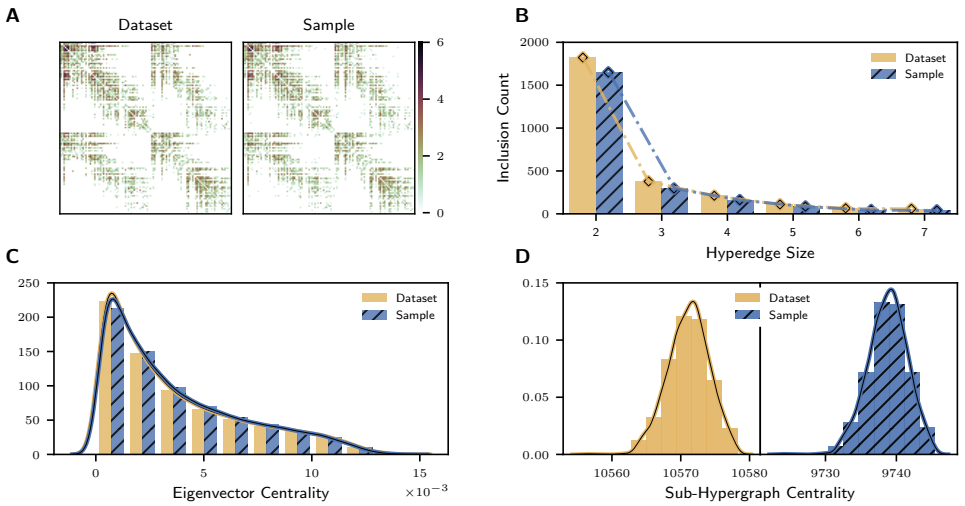


Figure 4.6: Matching statistics of real-world data and samples: the case of the House Bills dataset We plot (A) the adjacency matrices, (B) the hyperedge inclusions occurences, (C) the hyperedge eigenvector centrality distribution and (D) the sub-hypergraph centrality distribution for the House Bills dataset, where nodes represent congresspersons, and hyperedges describe subsets of them that co-sponsor a bill. For all such cases, we observe a good correspondence between the statistics measured on the real data and those obtained from a single sample of our generative model.

the observed hyperedge configuration, i.e. the hyperedges present in the real data. As explained in section 4.3.2, this means skipping the first step of our sampling procedure and moving directly to perform MCMC starting from such configuration. The returned hypergraphs will have a structure similar to that of the data, but will be sampled according to the generative model in section 4.2.

4.5.2 Comparing data and sample statistics

We now apply the proposed methodology on a variety of real datasets. As a representative example, we consider a dataset of co-sponsoring of bills for the U.S. House of Representatives [169, 170]. Nodes correspond to congresspersons, and hyperedges connect subsets of them that co-sponsor a bill. The dataset contains $N = 1494$ nodes, $|E| = 54933$ hyperedges with maximum size $D = 399$, and has been previously analysed via higher-order stochastic block models [39, 41].

As a first sanity check, in Supp. Mat. we verify that the degree and size sequences measured on the samples are identical to those of the data. This is guaranteed by the properties of the reshuffling operator described in section 4.3. We then proceed by comparing additional relevant statistics as measured on the real data and on the samples. Such statistics serve as a test

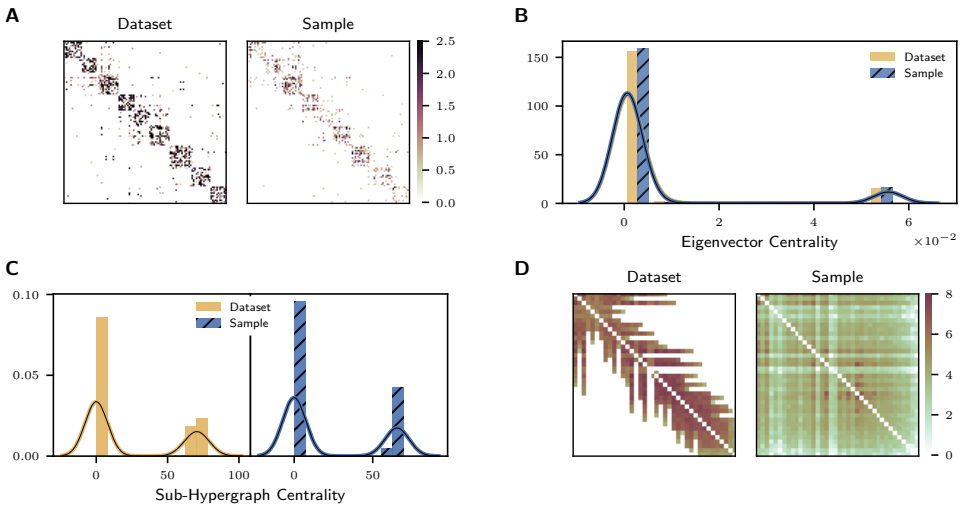


Figure 4.7: Hypergraph sample statistics, null hypothesis and generative assumptions. To illustrate the wide applicability of our model, we compare several statistics on real and sampled data. We plot (A) the adjacency matrix of face-to-face higher-order interactions among students in a High School dataset, (B) the eigenvector centrality distribution of co-purchasing behavior at Walmart, (C) the sub-hypergraph centrality distribution from committees data in the U.S. House. Similarly to the results presented in fig. 4.6, our model correctly reproduces the desired statistics. In (D) we show the adjacency matrix associated with co-voting Justices of the U.S Supreme Court. Such data have a strong temporal structure which is not included in the generative assumption of the model, hence explaining the limited correspondence between real and synthetic statistics.

for the goodness of fit, as they should match if the dataset is well-represented by the model.

We start by performing a visual comparison of the adjacency matrices [171, 22], where the adjacency value X_{ij} of any two nodes i, j is defined as $X_{ij} := \sum_{e \in E: i,j \in e} A_e$. As shown in fig. 4.6A, our samples are well aligned with the real data.

Another relevant structural property of a hypergraph is the inclusion relationships between hyperedges, i.e. which hyperedges are subsets of others [104]. This is of particular interest when comparing a hypergraph with its clique expansion, i.e. the graph obtained by projecting hyperedges onto pairwise interactions, or when comparing with other higher-order representations such as simplicial complexes [172, 173]. In fig. 4.6B, we count the number of hyperedges of size n that are included in hyperedges of size $n+1$. Also in this case, results on our sample match well those measured on the input dataset.

Finally, we explore two centrality measures on hypergraphs. As a first example, in fig. 4.6C we consider a generalization of eigenvector centrality [174] for hyperedges. In particular, we consider the dual representation of the

hypergraph, where nodes represent interactions in the original hypergraph and are connected if they have a non-empty intersection [175]. Moreover, in fig. 4.6D we also compute sub-hypergraph centrality [176, 171], which returns a measure of node importance in hypergraphs. Also in such cases, the quantities measured on our samples behave similarly to those based on the input data.

We highlight that the resemblance between samples and real data is not simply due to the Markov Chain being stuck in a local optimum given by the initial configuration, i.e. the real dataset. To prove this, we further investigate the Markov Chain mixing while producing the samples based on the House-Bills dataset. We observe that 73% of the shuffling steps are accepted by the Metropolis-Hastings algorithm. We leave more formal verifications of mixing via other statistics for future work. As an additional structural confirmation, we measure the Jaccard similarity between the real data and 10 samples, defined as the number of hyperedges in the intersection divided by the number of hyperedges in the union. Also in this case, the resulting score of 0.69 ± 0.11 signals that the microscopic structure of the samples detaches from that of the real data, while the macroscopic statistics in fig. 4.6 are preserved. Finally, we also observe that less structured methods fail to replicate such statistics. In Supp. Mat. we obtain samples utilizing the configuration model from Chodrow [52], which only takes into account the degree and size sequences. In this case, we observe a significant difference between the samples and the data, which could be explained by the lack of additional probabilistic structure in the sampling procedure. In Supp. Mat. we provide additional studies based on synthetic samples, showing how these can be employed for formalizing and carrying out hypothesis testing on complex structural patterns.

To illustrate the wide applicability of our method, we extend this analysis to additional systems. In fig. 4.7A we report the observed adjacency matrix of face-to-face interactions among High School students [177], and the one obtained from a sample of our generative model. In fig. 4.7B we show the distribution of the hyperedge eigenvector centrality computed on co-purchasing customer Walmart data [178]. Finally, in fig. 4.7C we compare the sub-hypergraph centrality on the House Committees dataset [41, 179], where hyperedges connect the members of the U.S. House participating in the same committees. In all such cases, we observe that our sampling method successfully models the desired statistics of the real data.

Synthetic data generated to incorporate a particular structure are often utilized as tests for null hypotheses. Indeed, discrepancies between sampled and real data may arise if some data features are not explicitly taken into account by the generative assumptions of the model [180]. Observing such differences can help unveil some relevant additional structure present in

the data and originally neglected. As an example, we consider a dataset of co-voting patterns of the US Supreme Court Justices, where the nodes are Justices and hyperedges describe co-voting behaviors observed from 1946 to 2019 [181]. Since the number of Justices is fixed to 9 at any point in time, only interactions between Justices working in overlapping years can exist. Such an intrinsic time dependency, however, is not enforced by our model. Hence, we do not expect samples of our model to match the input adjacency matrix well. We illustrate this in fig. 4.7D, where the comparison of the sampled and observed adjacency matrices are distinctively different, with the real data showing a clear time-dependence. Our example illustrates the importance of correctly identifying the existence of particular structures in real-world dataset, showcasing how our sampling method could be used for testing null hypotheses and reproducing real-world statistics.

Discussion

In this paper, we presented a framework for the generation of synthetic hypergraphs with flexible structure. Our model allows specifying different assortative and disassortative mesoscale configurations, tuning the size of the different communities and controlling the strengths of the interactions among them. Moreover, it allows regulating different node-level statistics, including hard or mixed community assignments and expected degrees. Through a variety of experiments, we showed how desired characteristics specified via input parameters are reflected in the generated data. Furthermore, we illustrated how practitioners can use our framework on real systems, both as a computationally efficient sampling tool for the replication of statistical measures, and as a structured null model for hypothesis testing. As an example, our model generates synthetic samples that successfully replicate centrality measures and inclusions relationship between hyperedges in higher-order data from different domains. Similarly, our model can help reveal important missing features in the generative assumptions made by different algorithms, showing clear discrepancies between samples and real data when, for instance, time-dependence is ignored. Finally, our framework allows testing the performance of different higher-order community detection methods.

There are various interesting and relevant avenues for future work. A first one is moving from the likelihood in eq. 4.1, which is based on a bilinear form, to one based on a multilinear form. Examples from the literature include symmetric tensors [39] and affinity functions [41]. While in principle this would allow for more flexible specifications, such as preventing the formation of certain hyperedge configurations, it is currently unclear how to obtain efficient expressions for the expected statistics and compute the moments required in the Central Limit Theorem. On a similar note, it is important to highlight that ours is only one of different possible definitions of community

in hypergraphs. Studying the implications of different probabilistic and optimization procedures on the communities observed in higher-order systems, both theoretically and empirically, is a promising avenue for future work. Moreover, additional information, such as time dependency and attributes on the nodes and hyperedges, could be explicitly incorporated in the probabilistic model. Such an extension could be based on insights from models for dyadic interactions, and result in substantial improvements when this information correlates with the hypergraph structure [37, 135, 182, 133, 3].

Taken together, our methodology provides a principled, scalable and flexible framework to sample structured hypergraphs. To facilitate its usage we provide an open-source implementation at github.com/nickruggeri/HyMMSBM. The method is also implemented as part of the HGX library [6].

Chapter 5

Hypergraphs with Node Attributes: Structure and Inference

Abstract

Many networked datasets with units interacting in groups of two or more, encoded with hypergraphs, are accompanied by extra information about nodes, such as the role of an individual in a workplace. Here we show how these node attributes can be used to improve our understanding of the structure resulting from higher-order interactions. We consider the problem of community detection in hypergraphs and develop a principled model that combines higher-order interactions and node attributes to better represent the observed interactions and to detect communities more accurately than using either of these types of information alone. The method learns automatically from the input data the extent to which structure and attributes contribute to explain the data, down weighing or discarding attributes if not informative. Our algorithmic implementation is efficient and scales to large hypergraphs and interactions of large numbers of units. We apply our method to a variety of systems, showing strong performance in hyperedge prediction tasks and in selecting community divisions that correlate with attributes when these are informative, but discarding them otherwise. Our approach illustrates the advantage of using informative node attributes when available with higher-order data.

5.1 Introduction

Over recent years, systems where units interact in groups of two or more have been increasingly investigated. Such higher-order interactions have been observed in a wide variety of systems, including cellular networks [21], drug recombination [82], ecological communities [183] and functional mapping of the human brain [85].

These systems can be better described by hypergraphs, where hyperedges encode interactions among an arbitrary number of units [22, 16]. Often, research in this area solely considers the topology of hypergraphs, that is, a set of nodes and their higher-order interactions. Many hypergraph datasets, however, include attributes that describe properties of nodes, such as the age of an individual, their job title in the context of workplace interactions, or the political affiliation of a voter. In this work, we consider how to extend the analysis of hypergraphs to incorporate this extra information.

We focus on the relevant task of community detection, where the goal is to cluster nodes in a hypergraph. Community detection algorithms solely based on interactions tend to cluster nodes based on notions of affinity between communities, cluster separation, or other arguments similar to those classically utilized on graphs [29]. However, one can assume that relevant information about the communities and the hyperedge formation mechanism is additionally contained in the attributes accompanying a dataset.

For instance, students in a school have been observed to interact more likely in groups that involve individuals in the same classes [177]. A similar observation was also made for dyadic networks, where incorporating node attributes helped in community detection and other related inference tasks, e.g. prediction of missing information [37, 184, 185, 135, 186].

Several tools have been developed for community detection in higher-order data [114, 113, 187, 115]. Methods based on statistical inference have established themselves as effective tools in this direction, as they are both mathematically principled and have a high computational efficiency [17, 1, 41].

Here, we build on these approaches to incorporate node attributes into a community detection framework for higher-order interactions. More precisely, we follow the principles behind generative models for networks, which incorporate community structure by means of latent variables that are inferred directly from the observed interactions [188, 36, 189] and extend them to incorporate extra information on nodes.

The model we propose has several desirable features. It is flexible, as it can be applied to both weighted and unweighted hypergraphs, it can incorporate different node attributes, categorical or binary, and it outputs overlapping communities, where nodes can belong to multiple groups simultaneously. Furthermore, the model does not assume any a priori correlation structure between the attributes and the communities. Rather, it infers such a connection directly from the data. The extent of this contribution can vary based on the dataset. In the favorable case where attributes are correlated well with the communities, our model exploits such additional information to improve community detection. This is particularly beneficial in situations where data is sparse or when data availability is limited to an incomplete set of observations. In less favorable situations where correlation is low (for instance

when the attributes do not align with the mechanism generating higher-order interactions), the model can nevertheless either discard or downweigh this information.

In some cases, a system can be explained well by different community divisions. Our model allows selecting a particular community structure guided by the desired attribute, provided that it is informative, as measured automatically by fitting the data. This allows a practitioner to focus the analysis of group interactions on some particular node characteristic.

Finally, our model is computationally efficient, as it scales to large hypergraphs and large hyperedge sizes. This feature is particularly relevant in the presence of higher-order interaction, where the increased computational complexity limits the range of models that can be practically implemented into viable algorithms.

Few works are available that investigate community detection in hypergraphs in presence of node attributes [190, 191, 192], but they are limited to clustering nodes without providing additional probabilistic estimates. Furthermore, they can be computationally burdening, or they typically rely on stronger assumptions about the nature of the data (e.g. assume real-valued weights) or the communities (e.g. nodes can only belong to one group).

5.2 Results

5.2.1 The Model

We propose a probabilistic model that incorporates both the structure of a hypergraph, i.e. the interactions observed in the data, and additional attributes (or covariates) on the nodes. These two types of information, which we call *structural* and *attribute* information, have been previously shown to be beneficial to the inference, when correlation to be exploited is present [37, 184, 185, 135].

We denote a hypergraph as $H = (V, E, A)$, where $V = \{1, \dots, N\}$ is a set of nodes, E is a set of observed hyperedges whose elements $e \in E$ are arbitrary sets of two or more nodes in V, and A is a vector containing the weights of edges. In this work, we assume that weights are positive and integer quantities. Denoting Ω as the set of all possible hyperedges, we have that A_e is the weight of edge e when $e \in E$, otherwise $A_e = 0$ if $e \in \Omega \setminus E$. Given these definitions, the observed edge set E can equivalently be represented as $E = \{e \in \Omega \mid A_e > 0\}$. We represent the covariates 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. We note that a node can have several types of covariates, e.g. gender and age, which are then one-hot encoded as attributes.

We model the presence of structural information A and covariate information X probabilistically, assuming a joint probability of these two types of information that is mediated by a set of latent variables $\theta = \{w, \beta, u\}$. Here w, β are specific to each of the two distinct types of information, while the quantity u is a latent variable shared between the two. The presence of a shared u is a key to allow coupling the two types of information and extracting valuable insights about the system. Formally, we assume

$$P(A, X \mid \theta) = P_A(A \mid w, u)\, P(X \mid \beta, u)\,. \tag{5.1}$$

This factorization assumes conditional independence between A and X, given the parameters θ, and is analogous to related approaches on graphs [37, 184]. The factorization in eq. 5.1 presents various advantages. First, the parameters in θ can provide interpretable insights about the mechanism driving hyperedge formation, as we show below. In our case, we focus on community structure, hence we model u to represent the community memberships of nodes. Second, it allows for efficient inference of the model parameters θ, as we show in the Methods section. Third, it allows predicting both A and X, which is relevant for example in the case of corrupted or missing data.

Having introduced the main structure of the model, we now describe the expressions of the two factors of the joint probability distribution in eq. 5.1.

5.2.1.1 Modeling structural information

We model the structural information A by assuming that latent communities control the interactions observed. For this, we utilize the Hy-MMSBM probabilistic model [1], which assumes mixed memberships where nodes can belong to multiple communities. This model flexibly captures various community structures (e.g. assortative, core periphery etc.), scales to large hyperedge sizes and allows incorporating covariates flexibly without compromising the efficiency of its computational complexity, as we explain in the Methods section.

Assuming K overlapping communities, u is an $N \times K$ non-negative membership matrix, which describes the community membership for each node $i = 1, \ldots, N$. A symmetric and non-negative $K \times K$ affinity matrix w controls the density of hyperedges between nodes in different communities. The hypergraph is modeled as a product of Poisson distributions as:

$$P_A(A|u, w) = \prod_{e \in \Omega} Pois\left(A_e; \frac{\lambda_e}{k_e}\right) \quad , \tag{5.2}$$

where

$$\lambda_e = \sum_{i<j:i,j\in e} u_i^T w u_j = \sum_{i<j:i,j\in e} \sum_{k,q=1}^{K} u_{ik} u_{jq} w_{kq} \quad . \tag{5.3}$$

The term k_e is a normalization constant, which can take on any positive value. In all our experiments we set its value to $k_e = \frac{|e|(|e|-1)}{2}\binom{N-2}{|e|-2}$, with $|e|$ being the size of the hyperedge. Other parametrizations of the likelihood $P_A(A|u,w)$ are possible, e.g. using different generative models for hypergraphs with community structure [17, 41], but it is not guaranteed that these would yield closed-form expressions and computationally efficient algorithms when incorporating additional attribute information in the probabilistic model. Similarly, in eq. 5.2 we assumed conditional independence between hyperedges given the latent variables, a standard assumption in these types of models. Such a condition could in principle be relaxed following the approaches of [131, 193, 133]. We do not explore this here.

5.2.1.2 Modeling attribute information

We model the covariates X assuming that the community memberships u regulate how these are assigned to nodes. We then assume that a $K \times Z$ matrix β with entries β_{kz} regulates the contribution of attribute z to the community k. This parameter plays a similar role for the matrix X as the matrix w does for the vector A. We combine the matrix β with the community assignment u via a matrix product that yields the following Bernoulli probabilities:

$$\pi_{iz} = \sum_{k=1}^{K} u_{ik}\,\beta_{kz} \quad . \tag{5.4}$$

We assume that attributes are conditionally independent given the parameters π, which allows flexibly modeling several discrete attributes at a time. This is implemented by assuming that each entry X_{iz} is extracted from a Bernoulli distribution with parameter π_{iz} as:

$$P_X(X|u,\beta) = \prod_{i=1}^{N}\prod_{z=1}^{Z} \pi_{iz}^{x_{iz}}(1-\pi_{iz})^{(1-x_{iz})} \quad . \tag{5.5}$$

To ensure $\pi_{iz} \in [0,1]$, we constraint $u_{ik} \in [0,1]$ and $\sum_{k=1}^{K}\beta_{kz} = 1$, $\forall z$.

We focus here on discrete and unordered attributes. This covers many relevant scenarios, including the ones we study in the several real datasets below, e.g. roles of employees in a company or classes of students. Other specific cases could be treated using similar ideas and techniques as the one we propose by suitably modifying the distribution in eq. 5.5. We give an example of imposing categorical attributes, when we want to explicitly force that having an attribute of one value does exclude any other possible value, in Supplementary Note C.

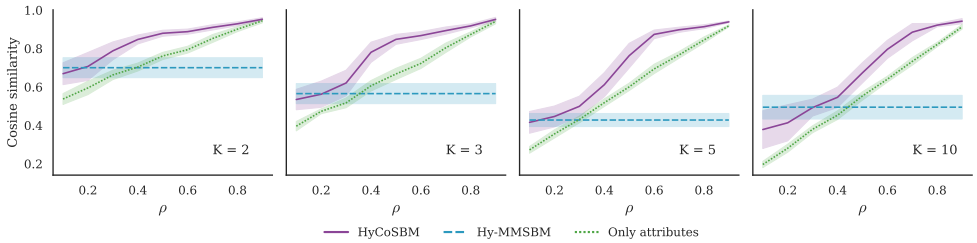


Figure 5.1: Community detection in synthetic hypegraphs. We show the cosine similarity between the communities inferred by the various algorithms and the ground truth communities in synthetic hypergraphs, with $N = 500$ and $E = 2720$. We show results for different numbers of communities K (from left to right). The number of attributes Z is selected to be equal to K, and the parameter γ is set equal to the fraction ρ of unshuffled attributes. We compare HyCoSBM with Hy-MMSBM, which servevs as a baseline that only employs structural information. We also measure the cosine similarity of the attribute matrix X and the ground truth membership matrix u Only attributes).

5.2.1.3 Inference of latent variables

Having defined the probabilistic model eq. 5.1 and the two distributions eqs. 5.2 and 5.5, our goal is to now infer the latent variables u, w and β, given the observed hypergraph A and the attributes X. To infer these values we consider maximum likelihood estimation and use an efficient expectation-maximization (EM) algorithm that exploits the sparsity of the dataset, as detailed in the Methods section. We combine the log-likelihoods of the two sources of information with a parameter γ that tunes their relative contribution, with extreme values $\gamma = 0$ ignoring the attributes and $\gamma = 1$ ignoring the structure, similarly to what has been done in attributed network models [37, 184, 185], or in models for information retrieval from text [194, 195]. In our experiments, we learn the γ hyperparameter from data via cross-validation.

Overall, the inference routine scales favorably with both the system size and the size of the hyperedges, as each EM iteration has a complexity of $O\Big(K(K+Z)(N+|E|)\Big)$, which is linear in the number of nodes and hyperedges. We refer to our model as HyCoSBM and make the code available online at github.com/badalyananna/HyCoSBM.

5.2.2 Detecting communities in synthetic networks

Our first experiments are tests on synthetic networks with known ground-truth community structure and attributes. We generate synthetic hypergraphs using Hy-MMSBM [2] as implemented in the library `HGX` [6]. We select parameter settings where inference with Hy-MMSBM is not trivial, to better assess the influence of using attributes, see details in Supplementary Note A. After the networks are created, we generate discrete attributes that match

the community membership a fraction ρ of the time, while the remaining fraction $1-\rho$ are randomly generated. This allows to vary the extent to which attributes correlate with communities and hence the difficulty of inferring the ground truth memberships. We varied $\rho \in [0.1, 0.9]$, with higher values implying that inference of communities is aided by more informative attributes.

As a performance metric, we measure the cosine similarity between the membership vectors recovered by our model and the ground truth ones. In fig. 5.1 we can see that, when the attributes are correlated with ground truth communities, HyCoSBM performs better than using either of the two types of information alone. In addition, the performance of HyCoSBM increases monotonically with increasing correlation between attributes and ground truth. Although this is observed also when using attributes alone, the performance of HyCoSBM in recovering the ground truth communities is always higher.

This behavior is consistent across different values of K, with larger performance gap between results at low and high ρ at larger K, where there are more choices to select from. We observe small decreases of performances of HyCoSBM at extreme regimes of ρ and K.

In short, these results demonstrate that the model is successfully using both attribute and structural information to improve community detection.

5.2.2.1 Results on empirical data

We analyze hypergraphs derived from empirical data drawn from social, political and biological domains, as detailed in the Methods section. For each hypergraph we describe a different experiment, to illustrate various applications of our method. We select the number of communities K and the hyperparameter γ using 5-fold cross-validation. To assess the impact of using attributes, we compare HyCoSBM with three baselines: i) Hy-MMSBM, that only utilizes the structural information in the hyperedges to detect mixed-membership communities; ii) HyCoSBM with $\gamma = 0$, which is equivalent to not utilizing the attributes; iii) HyCoSBM with community assignments u fixed to match the attributes, and only infer the w parameters, which tests how attributes alone perform. Notice that i) and ii) differ in that the membership vectors u are unconstrained in Hy-MMSBM, while they are restricted to $u_{ik} \in [0, 1]$ in our model. In iii) utilizing HyCoSBM and Hy-MMSBM is equivalent, since the two models coincide in the updates for w.

Recovering interactions on contact dataset In our first experiment we study human contact interactions, using the data obtained from wearable sensor devices in four settings: students in a high school (High School) and

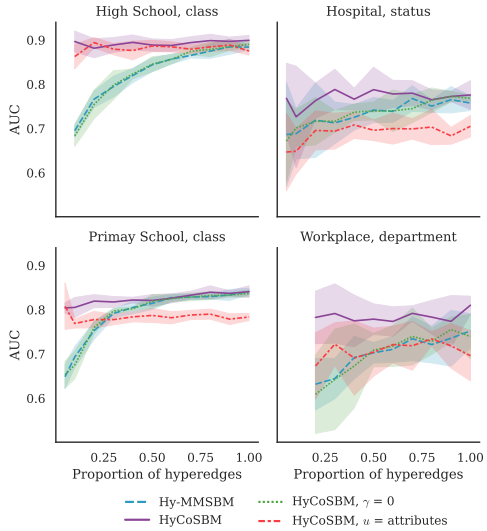


Figure 5.2: Predicting interactions in close-proximity datasets with partial observations. We show the performance of various methods in hyperedge prediction tasks, measured by AUC, as we vary the fraction of hyperedges made available to the algorithms. This plot shows that the performance of HyCoSBM remains high when fewer hyperedges are available in input, while that of the algorithms which do not use any attribute drops.

a primary school (Primary School), co-workers in a workplace (Workplace) and patients and staff in a hospital (Hospital). Hyperedges represent a group of people that were in close proximity at some point in time. Each dataset contains attributes that describe either the classes, the departments, or the roles the nodes belong to.

We measure the ability of our model to explain group interactions by assessing its performance on a hyperedge prediction task. To this end, we infer the parameters using only a fraction of the hyperedges in the dataset. Then we utilize the held out hyperedges to measure the AUC metric, which represents the fraction of times the model predicts an observed interaction as more likely than a non-observed one (higher values mean better performance).

Models that do not utilize any attribute have been previously shown to perform well on such a task on these datasets [41, 17, 1] when a large fraction of the dataset was given as input. Here, we vary the amount of structural information available to the algorithms more pronouncedly to assess their robustness in realistic situations where the full data is unavailable and investigate how making use of attributes can compensate for this. To simulate this setting, we delete an increasing fraction of the existing hyperedges (keeping the hypergraph connected) and perform 5-fold cross-validation on the remaining dataset.

The results in fig. 5.2 show a significant and monotonic drop in performance

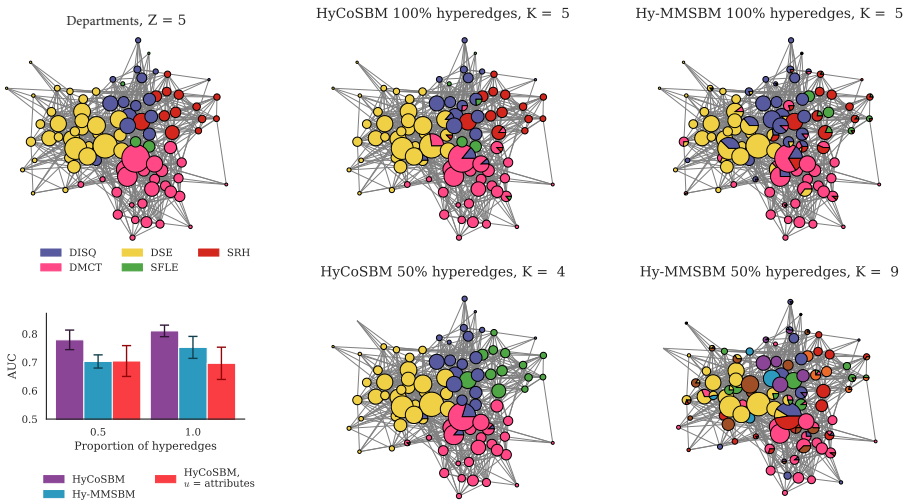


Figure 5.3: Communities detected in a Workplace dataset from partial observations of close-proximity interactions. We vary the fraction of hyperedges given in input to the algorithms (top: 100%, bottom: 50%) and compare the inferred communities against the attribute departement (top left). The AUC barplot (bottom-left) shows the performance of the models in hyperedge prediction. This plot shows that HyCoSBM is able to use the attributes effectively to keep performance high even at a low fraction of input observations.

for Hy-MMSBM as we decrease the fraction of hyperedges, consequently reducing the amount of structural information available to the algorithm. In contrast, HyCoSBM maintains an almost constant and high performance, all the way down to having access only to 20% of the hyperedges, owing to its usage of the additional attribute information. In addition, even in the favorable setting when all hyperedges are available, HyCoSBM yields higher AUC in Workplace (with $\gamma = 0.995$), indicating that incorporating attributes can be beneficial even when robust results are obtained using structural information alone. Focusing on other datasets where HyCoSBM attains AUC similar to that of other algorithms when all the interactions are utilized, we still observe a difference in the types of communities detected. As an example, in the High School dataset the community assignments u inferred via Hy-MMSBM have cosine similarity of 0.59 with the class attribute of the nodes, as opposed to the cosine similarity of 0.94 observed for HyCoSBM.

These different levels of correlation between inferred communities and attributes, together with observing similar AUC (indicating a similar ability to explain the structural information), could be explained by the presence of competing network divisions, as already observed in network datasets [196, 135, 31]. Our model allows selecting among divisions, finding ones that correlate with the attribute of interest.

We highlight that, although the communities inferred by HyCoSBM correlate

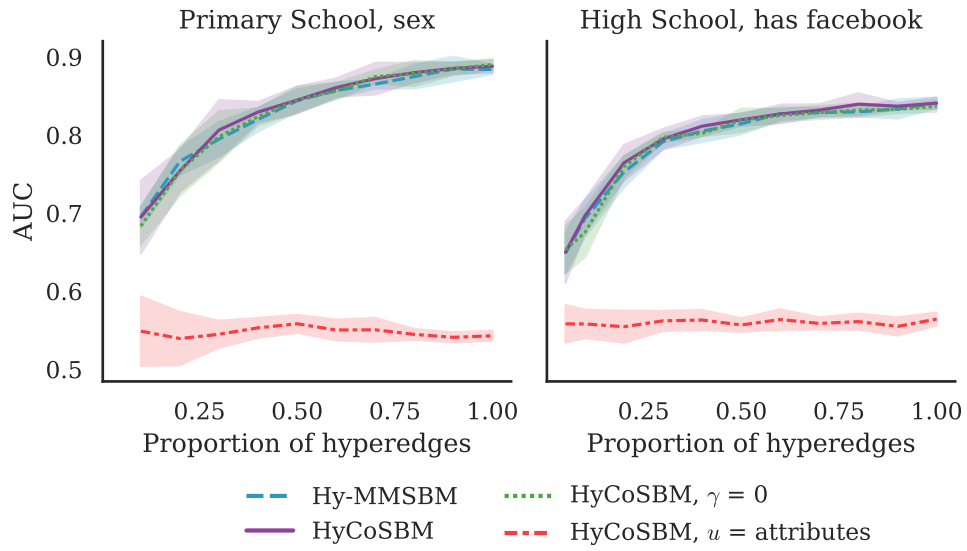


Figure 5.4: AUC on contacts dataset with partial hyperedges: uncorrelated attributes. Using *sex* and *has facebook* as the attributes, the performance of all models drops as the hyperedges are removed.

with the attributes, these two are not equivalent. In fact, we observe several cases where the number of detected communities is not equal to the number of attributes. For example, we observe cases where the model detects fewer communities than the number of attributes available. In fig. 5.3 the nodes with attribute SFLE (green) are included within the community formed mainly by DISQ nodes (purple) by our model when 50% of the edges are given in input. This partition achieves higher AUC than the model with community assignments fixed and equal to the attributes. In other cases, our model finds smaller communities within the bigger partitions determined by the attributes. We find such an example in the High School dataset in fig. 5.6, where HyCoSBM finds finer partitions ($K = 11$) than the one given by the $Z = 9$ classes, hierarchically splitting some classes into subgroups. The resulting partition attains a high AUC score. A high number of inferred communities is also observed in Hy-MMSBM, but, in this case, the AUC drops significantly, and the $K = 30$ communities inferred at 30% of the edges are much more mixed between the classes. In short, the communities inferred by our model do not simply replicate the attribute. Rather, this additional information is used to infer a community structure that better explains the interactions observed in the data.

5.2.2.2 Performance with uninformative attributes

In the previous sections, we have shown how attribute information can aid the recovery of effective communities and improve inference. In general, though, we cannot expect that any type of attribute added to a network dataset may help explaining the observed structure. This may be the case for instance when an attribute is uncorrelated or weakly correlated with the hyperedges, as in the synthetic experiments described above when ρ is close to 0.1.

In this section we study the performance of HyCoSBM in this adversarial regime and show that, when attributes are uninformative, these are readily discarded by our model to only perform inference based on structural information.

To this end, we feed the `sex` and `has facebook` attributes, respectively from the Primary School and High School datasets, into our model. As we show in fig. 5.4, the performance of HyCoSBM closely resembles that of the models that do not use any attribute in input, signaling that these attributes are not as informative as `class` to explain the observed group interactions. This is reinforced by a very low AUC for the model that fixes u as the attributes (red line).

We further illustrate this point in four datasets of US representatives. Here, nodes are representatives (in the House of Representatives or in the Senate) and hyperedges represent co-sponsorship of bills (Bills datasets) or co-participation in a committee during a Congress meeting (Committees datasets). The attribute indicates whether the representative is associated with the Republican or Democratic party ($Z = 2$). In table 5.1 we show that there is no advantage in using this binary attribute to explain the co-sponsorship nor the co-participation patterns, as the AUC is similar to that of models that do not use attribute information in input. As a confirmation, the value of γ obtained via cross-validation is equal to 0 in three out of four cases, and 0.1 in one case, showing that the algorithm tends to discard the attribute information and prefers to rely solely on structural data.

5.2.2.3 Improving prediction of Gene-Disease associations

Our final application is to a biological dataset containing Gene-Disease associations [197]. Here, nodes represent genes, and hyperedges represent a combination of genes specific to a disease. For each node, its Disease Pleiotropy Index (DPI) is available as an attribute, indicating the tendency of a gene to be associated with many types of diseases, with $Z = 25$ possible discrete values. The dataset is highly sparse, as many nodes are present only in one hyperedge. Previous results have shown that inferring missing associations improves sensibly when using all hyperedges in the datasets [1]

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 5.1: AUC scores on co-sponsorship and co-participation datasets of US representatives. We report the results of cross-validation in terms of selected K, γ, and obtained AUC. Here the node attribute used by HyCoSBM is the political party of the representative (Democrat or Republican, $Z = 2$).

(with AUC scores up to 0.84), compared to using only hyperedges up to size $D = 25$ [17]. In this paragraph, we investigate whether these results can be further improved when additional information is available in the form of the DPI attribute. We find that running HyCoSBM achieves an AUC score of 0.9, indicating that this attribute is informative. Furthermore, we observe that the communities detected by HyCoSBM are similar to those obtained from the attributes, see fig. 5.5a), but with a finer division into $K = 30$ communities, which is larger than the $Z = 25$ covariate categories.

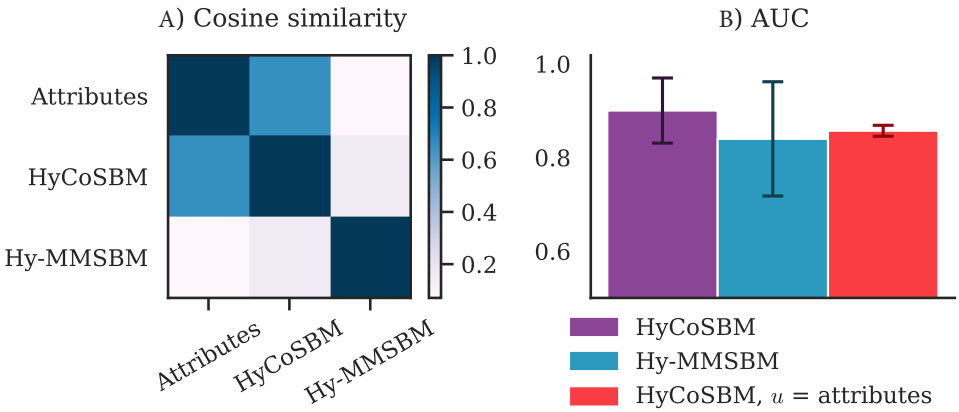


Figure 5.5: Cosine similarity and AUC in a Gene Disease dataset. a) Cosine similarity between the three types of communities: attribute, HyCoSBM and Hy-MMSBM. The membership u detected by HyCoSBM correlates with the DPI attribute and achieves higher AUC than both Hy-MMSBM and the model trained with u fixed as the attribute.

All the results of the previous analysis are summarized in table 5.2.

5.2.3 Discussion

We have analyzed how node attributes can be used to guide investigations of higher-order data. We focused on the problem of community detection,

Dataset	Attribute	N	$\lvert E\rvert$	Z	HyCoSBM			Hy-MMSBM		Source
					K	γ	AUC	K	AUC	
Gene Disease	DPI	9262	3128	25	30	0.500	0.9 ± 0.07	2	0.84 ± 0.122	[197]
High School	class	327	7818	9	11	0.995	0.899 ± 0.011	24	0.884 ± 0.006	[198]
	has filled 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	[198]
	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	[198]
Workplace	department	92	788	5	5	0.995	0.81 ± 0.02	5	0.752 ± 0.039	[198]
House Bills	political party	1494	54933	2	22	0.000	0.952 ± 0.003	25	0.952 ± 0.001	[169, 170]
House Committees	political party	1290	335	2	13	0.100	0.985 ± 0.015	24	0.972 ± 0.011	[179]
Senate Bills	political party	294	21721	2	23	0.000	0.929 ± 0.006	19	0.923 ± 0.003	[169, 170]
Senate Committes	political party	282	301	2	23	0.000	0.972 ± 0.01	21	0.963 ± 0.023	[179]

Table 5.2: AUC scores on real datasets. We report the AUC scores resulting from 5-fold cross-validation on various real datasets. We report the number of nodes N, number of hyperedges $|E|$, number of attributes Z and the values of K and γ as obtained from cross-validation.

introducing a mixed-membership probabilistic generative model for hypergraphs. Our model can explicitly incorporate both hyperedges and node attributes, and find more expressive community partitions by exploiting the combination of these information sources.

We have applied our model to a variety of social, political and biological hypergraphs, showing how prediction of missing interactions can be boosted by the addition of informative attributes, in particular in the regime of incomplete or noisy data. We have also illustrated various scenarios where attributes can be used to select between competing divisions, or cases where they are not informative and can be discarded.

There are a number of possible extensions of this work. One could include additional attribute types, such as attributes on hyperedges, continuous variables or vector variables. Similarly, one could consider alternative probabilistic expressions for the structural data, but this would require efforts to derive closed form updates and maintain a low computational complexity. On a related note, our model is based on the assumption that attributes and structure are independent conditionally on the latent variables. This approach is rather general, as the latent variables can potentially take on different semantics. It would be interesting to study other types of dependencies between structure and attributes. Finally, our model might be extended to consider dynamical hypergraphs, where communities and interactions can change in time, and assess what role attributes play in this case.

5.3 Methods

5.3.1 Inference of the latent variables

The likelihood of HyCoSBM factorizes over all hyperedges $e \in \Omega$, and single hyperedges are modeled with a Poisson distribution:

$$P_A(A_e|u,w) = Pois\left(A_e; \frac{\lambda_e}{k_e}\right) . \tag{5.6}$$

Similarly, the probability of attributes factorizes into Bernoulli probabilities:

$$P_X(X|u,\beta) = \prod_{i=1}^{N}\prod_{z=1}^{Z} \pi_{iz}^{x_{iz}}(1-\pi_{iz})^{(1-x_{iz})} \quad . \tag{5.7}$$

Under the Poisson distribution in eq. 5.6, it can be shown that the log-likelihood $L_A(u,w)$ of the full hypergraph evaluates to

$$L_A(u,w) = -C \sum_{i<j\in V} u_i^T w u_j + \sum_{e\in E} A_e \log \sum_{i<j\in e} u_i^T w u_j \,, \tag{5.8}$$

where $C = \sum_{d=2}^{D} \binom{N-2}{d-2} \frac{1}{\kappa_d}$ and D is the maximum hyperedge size observed [1]. Instead, eq. 5.7 yields the log-likelihood

$$\begin{aligned} L_X(u,\beta) = &\sum_{i=1}^{N}\sum_{z=1}^{Z} x_{iz} \log\left(\sum_{k=1}^{K} u_{ik}\,\beta_{kz}\right) \\ &+ \sum_{i=1}^{N}\sum_{z=1}^{Z} (1-x_{iz}) \log\left(\sum_{k=1}^{K} (1-u_{ik})\,\beta_{kz}\right) \quad . \end{aligned} \tag{5.9}$$

As we assumed conditional independence of the network part and the attributes part, the total log-likelihood becomes the sum of those two terms. In practice though, performance improves by introducing a balancing parameter $\gamma \in [0,1]$ that tunes the relative contribution of the two terms [37, 185, 195, 194], yielding a total log-likelihood as:

$$L(u,w,\beta) = (1-\gamma)\, L_A(u,w) + \gamma\, L_X(u,\beta) \quad . \tag{5.10}$$

The value of γ is not known a priori, and it can be learned from the data using standard techniques for hyperparameter learning. In our experiments, we utilize cross-validation. The γ parameter is necessary to better balance the contribution of the structural and covariate information, as the magnitude of the two different log-likelihood terms can be on different scales, with the risk of biasing the total likelihood maximization towards one of the two terms. This balancing is also useful when attribute data are somehow more (or less) reliable than structural data, for instance when we believe

that one is less (or more) subject to noise. Furthermore, γ is reminiscent of any hyperparameter of approaches that adjust inference based on prior distributions on the community assignments, as done in some attributed network models, e.g. [135, 186].

We note here that the value of γ has a clear interpretation only for the extreme cases of 0 or 1, which discards entirely the contribution of one of the two terms. In all the other intermediate cases, its value is not simply interpreted as a percentage contribution of the attributes over the network. This is because γ balances the magnitudes of two likelihood terms. In general, the network part is much larger than the attribute one, which draws γ to values closer to 1, e.g. 0.995, to compensate for the difference in scales. This does not necessarily mean that the network information is barely used, but rather that it has to be rescaled to allow the attribute information to be effectively considered.

As a final remark, our definition of X allows modeling several discrete attributes at the same time, and the dimension Z is the total number of values, including all the attribute types. Formally, $Z = \sum_{p=1,\dots,P} z_p$, where P is the number of attribute types (e.g. age and class would give $P = 2$), and z_p is the number of discrete values an attribute of type p can take. Alternatively, the presence of more than one attribute can be modeled by considering separate terms L_X, each with a different multiplier γ. While this formulation would allow for tuning the contribution of attributes more specifically, this comes at a price of higher model complexity (in case of using different expressions for the L_X) or higher computational complexity, as one needs to cross-validate more than one type of γ. We do not explore this here.

5.3.1.1 Variational lower bound

To maximize the total log-likelihood in eq. 5.10 we adopt a standard variational approach to lower bound the summation terms inside the logarithm. 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:

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

$$\sum_{e \in E} A_e \sum_{i<j \in e} \sum_{k,q=1}^{K} \rho_{ijkq}^{(e)} \log \left(\frac{u_{ik} u_{jq} w_{kq}}{\rho_{ijkq}^{(e)}} \right) ; \tag{5.11}$$

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

$$\sum_{i=1}^{N} \sum_{z=1}^{Z} x_{iz} \sum_{k=1}^{K} h_{izk} \log \left(\frac{u_{ik} \beta_{kz}}{h_{izk}} \right) ; \tag{5.12}$$

$$\sum_{i=1}^{N} \sum_{z=1}^{Z} (1 - x_{iz}) \log \left(\sum_{k=1}^{K} (1 - u_{ik}) \beta_{kz} \right) \geq$$

$$\sum_{i=1}^{N} \sum_{z=1}^{Z} (1 - x_{iz}) \sum_{k=1}^{K} h'_{izk} \log \left(\frac{(1 - u_{ik}) \beta_{kz}}{h'_{izk}} \right) ; \tag{5.13}$$

with equality reached when

$$\rho_{ijkq}^{(e)} = \frac{u_{ik} u_{jq} w_{kq}}{\lambda_e} ; \tag{5.14}$$

$$h_{izk} = \frac{\beta_{kz} u_{ik}}{\sum_{k'} \beta_{k'z} u_{ik'}} ; \tag{5.15}$$

$$h'_{izk} = \frac{\beta_{kz} (1 - u_{ik})}{\sum_{k'} \beta_{k'z} (1 - u_{ik'})} ; \tag{5.16}$$

respectively.

Plugging eq. 5.11 into eq. 5.8 yields a lower bound $\mathcal{L}_A$ of the structural log-likelihood

$$\mathcal{L}_A(u, w, \rho) = -C \sum_{i<j \in e} u_i^T w u_j$$

$$+ \sum_{e \in E} A_e \sum_{i<j \in e} \sum_{k,q=1}^{K} \rho_{ijkq}^{(e)} \log \left(\frac{u_{ik} u_{jq} w_{kq}}{\rho_{ijkq}^{(e)}} \right) . \tag{5.17}$$

Similarly, eqs. 5.12–5.13 yield a lower bound $\mathcal{L}_X$ of the log-likelihood of the attributes:

$$\mathcal{L}_X(u, \beta, h, h') = \sum_{i=1}^{N} \sum_{z=1}^{Z} x_{iz} \sum_{k=1}^{K} h_{izk} \log \left(\frac{u_{ik} \beta_{kz}}{h_{izk}} \right)$$

$$+ \sum_{i=1}^{N} \sum_{z=1}^{Z} (1 - x_{iz}) \sum_{k=1}^{K} h'_{izk} \log \left(\frac{(1 - u_{ik}) \beta_{kz}}{h'_{izk}} \right) , \tag{5.18}$$

so that

$$\mathcal{L} := (1-\gamma)\mathcal{L}_A + \gamma\mathcal{L}_X\,, \tag{5.19}$$

is a lower bound of the full log-likelihood.

5.3.1.2 Expectation-Maximization

We now aim to optimize the variational lower bound in eq. 5.19 with respect to the model parameters u, w and β. To account for the constraint on β and u, we introduce the Lagrange multipliers $\lambda^{(\beta)}$ and $\lambda^{(u)}$ obtaining the following objective:

$$\mathcal{L}_{constr} := \mathcal{L} - \sum_{z=1}^{Z}\lambda_z^{(\beta)}\left(\sum_{k=1}^{K}\beta_{kz}-1\right) - \sum_{i}^{N}\sum_{k}^{K}\lambda_{ik}^{(u)}u_{ik}\,. \tag{5.20}$$

We proceed as in the Expectation-Maximization algorithm [199], by alternating two optimization steps until convergence. In one step, we maximize eq. 5.20 with respect to the model parameters u, w, β and the Lagrange multipliers $\lambda^{(\beta)}, \lambda^{(u)}$. In the other, we utilize the closed-form updates in eqs. 5.14–5.16 for the variational parameters. The procedure is described in detail in algorithm 4.

Differentiating objective eq. 5.20 with respect to the w, β parameters and the multipliers $\lambda^{(\beta)}$ yields the following closed-form updates:

$$w_{kq} = \frac{\sum_{e\in E} A_e \sum_{i<j\in e}\rho_{ijkq}^{(e)}}{C\sum_{i<j\in V}u_{ik}u_{jq}} \quad , \tag{5.21}$$

$$\beta_{kz} = \frac{\sum_i (x_{iz}h_{izk} + (1-x_{iz})h'_{izk})}{\sum_{i,k'}(x_{iz}h_{izk'} + (1-x_{iz})h'_{izk'})} \quad . \tag{5.22}$$

Equation 5.21 is valid when $\gamma \neq 1$ and eq. 5.22 is valid when $\gamma \neq 0$.

To obtain the updates for u we distinguish two cases. In the case of $\gamma \neq 0$, differentiating eq. 5.20 with respect to u_{ik} yields the condition:

$$a_{ik}\,u_{ik}^2 - (a_{ik}+b_{ik}+c_{ik})\,u_{ik} + b_{ik} = 0 \quad , \tag{5.23}$$

where

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

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

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

The updated values for u_{ik} are found by numerically solving eq. 5.23. We take the smallest root of eq. 5.23, as this is guaranteed to be in $(0, 1)$, as we show in Supplementary Note B. This update automatically yields a value of u_{ik} in $[0, 1]$, therefore the constraints on u are inactive and we do not need to differentiate with respect to the Lagrange multipliers $\lambda_{ik}^{(u)}$.

In the case $\gamma = 0$, we differentiate eq. 5.20 with respect to both u_{ik} and the Lagrangian multipliers $\lambda_{ik}^{(u)}$ to obtain the update

$$u_{ik} = \frac{\sum_{e \in E: i \in e} A_e \sum_{j \neq i \in e} \sum_{q=1}^{K} \rho_{ijkq}^{(e)}}{C \sum_{j \in V, j \neq i} \sum_{q=1}^{K} u_{jq} w_{kq} + \lambda_{ik}^{(u)}}, \tag{5.24}$$

which is exactly the same as those of the Hy-MMSBM model [1], except that in our case we have $\lambda_{ik}^{(u)}$ which constrains $u_{ik} \in [0, 1]$. Thus, our model is as powerful as Hy-MMSBM when $\gamma = 0$, but, when the attributes correlate well with the communities, our model can utilize this information to boost performance. In practice, in the latter case, cross-validation would yield $\gamma > 0$.

The EM algorithms finds a local maximum for a given starting point, which is not guaranteed to be the global maximum. Therefore, the algorithm is run several times and the best parameters are chosen based on the run that gives the highest log-likelihood.
A pseudocode for the algorithmic implementation is given in algorithm 4.

Algorithm 4: HyCoSBM: EM algorithm

Input: Hypergraph A, covariates X, hyperparameters γ and K
Result: Inferred (u, w, β)

```
u, w, β ← init(u, w, β) :          Randomly initialize the parameters
while convergence not reached do
    ρ, h, h' ← update(ρ, h, h')                          eqs. 5.14–5.16
    u ← update(u)                                  eq. 5.23 or eq. 5.24
    if γ ≠ 1 then
        w ← update(w)                                          eq. 5.21
    end
    if γ ≠ 0 then
        β ← update(β)                                          eq. 5.22
    end
end
```

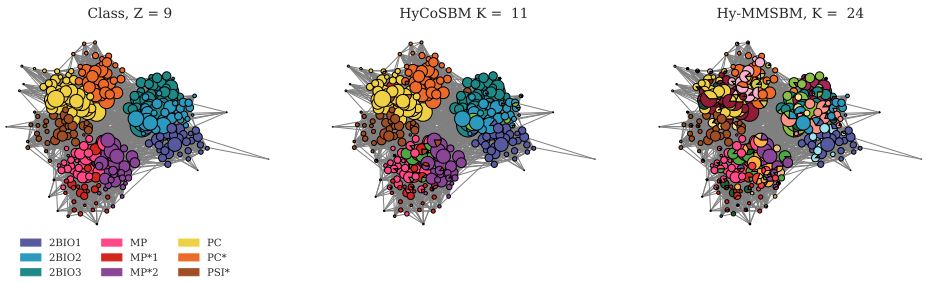


Figure 5.6: Communities detected in a High School dataset of close-proximity interactions. We give the whole dataset as input to the algorithms, and compare the inferred communities against the `class` attribute (top left). The plot shows that both HyCoSBM and Hy-MMSBM detect communities aligned with the attribute, but with a number of communities greater than the number of attribute values. AUC values are slightly higher for HyCoSBM , see table 5.2.

5.3.1.3 Hyperedge prediction and cross-validation

For all experiments with real datasets we used 5-fold cross-validation with the test AUC as performance metric to select the hyperparameters K and γ. We varied $K \in \{2, \ldots, 30\}$ and $\gamma \in [0.0, 1.0]$. The set of hyperedges was split into 80% and 20% for training and testing. The AUC is calculated by comparing the Poisson probabilities assigned to a given existing hyperedge against that of a randomly generated hyperedge of the same size. Since comparing all possible pairs of observed-unobserved edges is unfeasible, we estimate the AUC via sampling. For every observed edge in the dataset, we draw an edge of the same size uniformly at random, and compute the relative Poisson probabilities. The resulting Poisson probabilities are saved in a vector R_1 for the observed edges and R_0 for the randomly generated ones. We then compute the AUC as

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

where $\sum(R_1 > R_0)$ stands for the number of times the Poisson probability of the positive hyperedge was higher than the negative one, $\sum(R_1 == R_0)$ when they were equal, and the total number $|R_1|$ of comparisons made is equal to the number of hyperedges in the test set.

5.3.2 Extended data

We provide additional results in figs. 5.6 and 5.7.

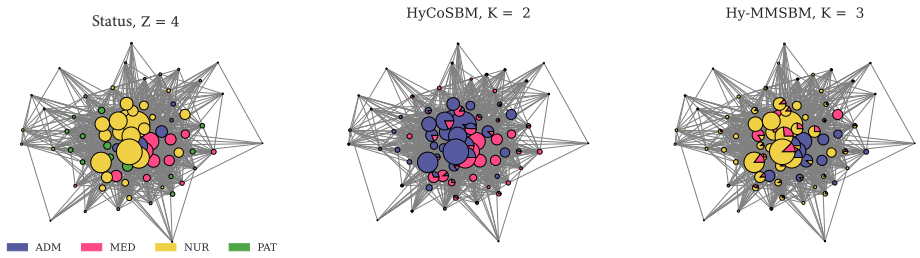


Figure 5.7: Communities detected in Hospital dataset using 60% **of hyperedges**. We give in input to the algorithms 60% of hyperedges and compare the inferred communities against the attribute `status` (NUR=paramedical staff; PAT=Patient; MED=Medical doctor; ADM=administrative staff) (top left). This plot shows that both HyCoSBM and Hy-MMSBM detect fewer communities than the division indicated by attributes, with HyCoSBM achieving a higher AUC that Hy-MMSBM, see Fig. 2 in the main manuscript.

Chapter 6

Message-Passing on Hypergraphs: Detectability, Phase Transitions and Higher-Order Information

Abstract

Hypergraphs are widely adopted tools to examine systems with higher-order interactions. Despite recent advancements in methods for community detection in these systems, we still lack a theoretical analysis of their detectability limits. Here, we derive closed-form bounds for community detection in hypergraphs. Using a Message-Passing formulation, we demonstrate that detectability depends on hypergraphs' structural properties, such as the distribution of hyperedge sizes or their assortativity. Our formulation enables a characterization of the entropy of a hypergraph in relation to that of its clique expansion, showing that community detection is enhanced when hyperedges highly overlap on pairs of nodes. We develop an efficient Message-Passing algorithm to learn communities and model parameters on large systems. Additionally, we devise an exact sampling routine to generate synthetic data from our probabilistic model. With these methods, we numerically investigate the boundaries of community detection in synthetic datasets, and extract communities from real systems. Our results extend the understanding of the limits of community detection in hypergraphs and introduce flexible mathematical tools to study systems with higher-order interactions.

6.1 Introduction

Modeling complex systems as graphs has broadened our understanding of the macroscopic features that emerge from the interaction of individual units. Among the various aspects of this problem, community detection stands out as a fundamental task, as it provides a coarse-grained description of a

network's structural organization. Notably, community structure is observed across different systems, such as food webs [200], spatial migration and gene flow of animal species [201], as well as in social networks [202], power grids [203], and others [29].

In the case of networks with only pairwise interactions, there are solid theoretical results on detectability limits, describing whether the task of community detection can or cannot succeed [55, 204, 205, 206, 207, 208]. However, many complex systems with interactions that extend beyond pairs are better modeled by hypergraphs [22], which generalize the simpler case of dyadic graphs. Phenomena that have been investigated on graphs are now readily explored on hypergraphs, with examples including diffusion processes, synchronization, phase transitions [16] and, more recently, community structure [115, 41, 39, 1, 40].

Extending the rigorous results of detectability transitions for networks to higher-order interactions is a relevant open question.

One of the main obstacles in modeling hypergraphs is their intrinsic complexity, which poses both theoretical and computational challenges and restricts the range of results available in the literature. The difficulty of defining communities in hypergraphs and of deriving theoretical thresholds for their recovery has limited investigations to the study of d-uniform hypergraphs, i.e., hypergraphs that only contain interactions among exactly d nodes [117, 209, 210, 211, 212, 213, 214, 215, 216].

A related line of literature focuses on the detection of planted sub-hypergraphs [217, 218] and testing for the presence of community structure in hypergraphs [219, 220]. Generally, extracting recovery results on non-uniform hypergraphs proved to be demanding, with scarce literature on the subject.

Recently, Chodrow *et al.* [42] conjectured a recoverability threshold for their spectral clustering algorithm on non-uniform hypergraphs. Closer to the scope of our work, Dumitriu and Wang [40] provide a probabilistic model and bounds for the theoretical recovery of communities under the same model. However, such detectability bounds are based on algorithms which are not feasible in practice, and no empirical demonstration of the predicted recovery is provided. Furthermore, all these methods lack a variety of desirable probabilistic features, such as the estimation of marginal probabilities of a node to belong to a community, a principled procedure to sample synthetic hypergraphs with prescribed community structure, and the possibility to investigate the energy landscape of a problem via free energy estimations.

In this work, we address these issues by deriving a precise detectability threshold for hypergraphs that depends on the node degree distribution, the assortativity of the hyperedges, and crucially, on higher-order properties such as the distribution of hyperedge sizes. Additionally, we show how these

properties can be formally described via notions of entropy and information, leading to a clear interpretation of the role of higher-order interaction in detectability.

Our approach is based on a probabilistic generative model and a related Bayesian inference procedure, which we utilize to study the limits of the community detection problem using a Message-Passing (MP) formulation [221, 11, 14], originating from the cavity method in statistical physics [222, 54]. We focus on an extension to hypergraphs of the stochastic block model (SBM) [33, 223], a generative model for networks with community structure. Several variants of the SBM [41], and of its mixed-membership version [39, 1], have been extended to hypergraphs. The model we utilize is an extension of the dyadic SBM to hypergraphs and allows generalizing the seminal detectability results of Decelle *et al.* [55, 204] to higher-order interactions.

In addition to our theoretical contributions, we derive an algorithmic implementation for inferring both communities and parameters of the models from the data. Our implementation scales well to both large hypergraphs and large hyperedges, owing to a dynamic-program formulation.

Finally, we show how, with additional combinatorial arguments, one can efficiently sample hypergraphs with arbitrary communities from our probabilistic model. This problem, often studied in conjunction with inference, deserves its own attention when dealing with hypergraphs, as recently discussed in related work [224, 2].

Through numerical experiments, we confirm our theoretical calculations by showing that our algorithm accurately recovers the true community structure in synthetic hypergraphs all the way down to the predicted detectability threshold. We also illustrate that our approach gives insights into the community organization of real hypegraphs by analyzing a dataset of group interactions between students in a school. To facilitate reproducibility, we release open source the code that implements our inference and sampling procedures at github.com/nickruggeri/hypergraph-message-passing.

6.2 The hypergraph stochastic block model

Consider a hypergraph $H = (V, E)$ where $V = \{1, ..., N\}$ is the set of nodes and E the set of hyperedges. A hyperedge e is a set of two or more nodes. We define $\Omega = \{e : 2 \leq |e| \leq D\}$, the set of all possible hyperedges up to some maximum dimension $D \leq N$, with $|e|$ being the size of a hyperedge, i.e., the number of nodes it contains. Notice that $E \subseteq \Omega$. We denote with $A_e = 1$ all $e \in E$ and with $A_e = 0$ hyperedges $e \in \Omega \setminus E$.

Our Hypergraph Stochastic Block Model (HySBM) is an extension of the classical SBM for graphs [33, 223]. It partitions nodes into K communities

by assigning a hard membership $t_i \in [K] \equiv \{1, \dots, K\}$ to each node $i \in V$, with $t = \{t_i\}_{i \in V}$ being the membership vector. It does so probabilistically, assuming that the likelihood to observe a hyperedge A_e is a Bernoulli distribution with a parameter that depends on the memberships $\{t_i\}_{i \in e}$ of its nodes.Formally, the probabilistic model is summarized as

$$t_i \sim \text{Cat}(n) \qquad \forall i \in V \tag{6.1}$$

$$A_e \,|\, t \sim \text{Be}\left(\frac{\pi_e}{\kappa_{|e|}}\right) \quad \forall e \in \Omega\,, \tag{6.2}$$

where $n = (n_1, \dots, n_K)$ is a vector of prior categorical probabilities for the hard assignments t_i. The Bernoulli probabilities are given by

$$\pi_e = \sum_{i<j \in e} p_{t_i t_j}\,, \tag{6.3}$$

with $0 \leq p_{ab} \leq 1$ being elements of a symmetric probability matrix (also referred to as affinity matrix) and $\kappa_{|e|}$ a normalizing constant that only depends on the hyperedge size $|e|$. This can take on any values, provided that it yields sparse hypergraphs where $\pi_e/\kappa_{|e|} = O(1/N)$ and valid probabilities $\pi_e/\kappa_{|e|}$. We develop our theory for a general form of $\kappa_{|e|}$ and elaborate more on its choice in Supp. Mat. In our experiments we utilize the value $\kappa_d = \binom{N-2}{d-2}\frac{d(d-1)}{2}$ [1, 2].

Our specific formulation of the likelihood is only one among many alternatives to model communities in hypergraphs. The likelihood we propose has three main properties. First, HySBM reduces to the standard SBM when only pairs are present (as $\kappa_2 = 1$). Since we aim to develop a model that generalizes the SBM to hypergraphs, this is an important condition to satisfy. Second, it enables to develop the MP equations presented in the following section, which in turn lead to a theoretical characterization of the detectability limits and a computationally efficient algorithmic implementation. Third, the likelihoods based on expressions similar to eq. 6.3 have been shown to well describe higher-order interactions that possibly contain nodes from different communities [2].

For convenience, we work with a rescaled affinity matrix $c = Np$, which is of order $c = O(1)$ (elementwise) in sparse hypergraphs. The log-likelihood

$\mathcal{L} \equiv \mathcal{L}(A, t \mid p, n)$ evaluates to

$$\begin{aligned}\mathcal{L} &= \sum_{e\in\Omega}\left[A_e \log\left(\frac{\pi_e}{\kappa_e}\right) + (1-A_e)\log\left(1-\frac{\pi_e}{\kappa_e}\right)\right] \\ &\quad + \sum_{i\in V}\log n_{t_i} \\ &= \sum_{e\in\Omega}\left[A_e \log\left(\sum_{i<j\in e} c_{t_i t_j}\right) + (1-A_e)\log\left(1-\frac{\sum_{i<j\in e} c_{t_i t_j}}{N\kappa_e}\right)\right] \\ &\quad + \sum_{i\in V}\log n_{t_i} \\ &\quad + \text{const.}\,, \end{aligned} \tag{6.4}$$

where const. denotes quantities that do not depend on the parameters of the model.

6.3 Inference and generative modeling

6.3.1 Induced factor graph representation

The probabilistic model in eqs. 6.1–6.2 has a negative log-likelihood that can be interpreted as the Hamiltonian of a Gibbs-Boltzmann distribution on the community assignments t:

$$p(t \mid A, p, n) = \frac{p(A, t \mid p, n)}{p(A \mid p, n)} = \frac{\exp \mathcal{L}(A, t \mid p, n)}{Z}\,, \tag{6.5}$$

where Z is the partition function of the system, that corresponds to the marginal likelihood of the data. The quantity $F = -\log Z$ is also called the free energy. The equivalence in Equation 6.5 allows interpreting the probabilistic model in terms of factor graphs [11]. Here, the function nodes are hyperedges $f \in \Omega$, and variable nodes are elements of V. The interactions between function and variable nodes can be read directly from the log-likelihood in eq. 6.4. In other words, the probabilistic model induces a factor graph $F = (\mathcal{V}, \mathcal{F}, \mathcal{E})$ with variable nodes $\mathcal{V} = V$, function nodes $\mathcal{F} = \Omega$ and edges $\mathcal{E} = \{(i, e) \in \mathcal{V} \times \mathcal{F} : i \in e\}$. In fig. 6.1 we show a graphical representation of the equivalence between hypergraphs and factor graphs. For any variable node i and function node f of the factor graph we define the neighbors, or boundaries, as $\partial i = \{f \in \mathcal{F} : (i, e) \in \mathcal{E}\}$, being all function nodes adjacent to i, and $\partial f = \{i \in \mathcal{V} : (i, e) \in \mathcal{E}\}$ being all variable nodes adjacent to f.

6.3.2 Message-Passing (MP)

Given the factor graph representation of HySBM, we can perform Bayesian inference of the community assignments via message-passing. Originally

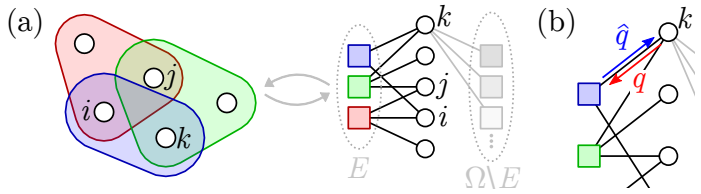


Figure 6.1: Representing hypergraphs as factor graphs. (a) We depict a hypergraph and its factor graph equivalent. In the factor graph $\mathcal{F}$, function nodes represent hyperedges. Notice that, while the node sets are the same in both representations, due to the presence of all possible hyperedges in the log-likelihood in Equation 6.4, the factor graph does not only contain the observed interactions E (black), but also the unobserved ones $\Omega \setminus E$ (gray). **(b)** In factor graphs, there are two types of messages: variable-to-function node q (red), and function-to-variable node $\hat{q}$ (blue).

obtained from the cavity method on spin glasses [222, 54], MP allows estimating marginal distributions on the variable nodes of a graphical model by iteratively updating messages, auxiliary variables that operate on the edges of the factor graph. The efficiency of MP comes from the fact that the structure of the factor graph favors locally distributed updates. Although exact theoretical results are only proven on trees, MP has been shown to obtain strong performance also on locally tree-like graphs [11] and it has been extended to dense graphs with short loops [56, 57].

Applying MP to our model, the inference procedure yields expressions for the marginal probabilities $q_i(a)$ of a node i to be assigned to any given community $a \in [K]$. Their values are obtained as solutions to closed-form fixed-point equations, which involve messages $q_{i\to e}(t_i)$ from variable to function nodes, and $\widehat{q}_{e\to i}(t_i)$, from function to variable nodes. The messages follow the sum-product updates

$$q_{i\to e}(t_i) \propto n_{t_i} \prod_{f\in\partial i\setminus e} \widehat{q}_{f\to i}(t_i) \tag{6.6}$$

$$\widehat{q}_{e\to i}(t_i) \propto \sum_{t_j : j\in\partial e\setminus i} \left(\frac{\pi_e}{\kappa_e}\right)^{A_e} \left(1-\frac{\pi_e}{\kappa_e}\right)^{1-A_e} \prod_{j\in\partial e\setminus i} q_{j\to e}(t_j)\,, \tag{6.7}$$

and yield marginal distributions as

$$q_i(t_i) \propto n_{t_i} \prod_{e\in\partial i} \widehat{q}_{e\to i}(t_i)\,. \tag{6.8}$$

Notice that, compared to those for graphs, the MP equations for hypergraphs in Equations 6.6–6.8 present additional challenges. First, in graphs the updates simplify. One can in fact collapse the two types of messages (and equations) into a unique one, since paths (i, f, j) in the factor graph reduce to pairwise interactions (i, j) between nodes. This simplification is not possible in hypergraphs, as one function node may connect more than two variable

nodes. Second, the dimensionality of the MP equations grows faster when accounting for higher-order interactions. Here, the number of function nodes is equal to $|\mathcal{F}| = |\Omega| = \sum_{d=2}^{D} \binom{N}{d}$, yielding $|\mathcal{F}| = O(2^N)$ at large $D = N$. In contrast, one gets $O(N^2)$ pairwise messages in the updates for graphs. To produce computationally feasible MP updates one can assume sparsity, as already done in the dyadic case. We outline such updates in the following theorem.

Theorem 6.1 *Assuming sparse hypergraphs where $c = O(1)$, the MP updates satisfy the following fixed-point equations to leading order in N. For all hyperedges $e \in E$ and nodes $i \in e$, the messages and marginals are given by:*

$$q_{i\to e}(t_i) \propto n_{t_i} \left(\prod_{\substack{f\in E \\ f\in \partial i \setminus e}} \widehat{q}_{f\to i}(t_i) \right) \exp(-h(t_i)) \tag{6.9}$$

$$\widehat{q}_{e\to i}(t_i) \propto \sum_{t_j : j\in \partial e\setminus i} \pi_e \prod_{j\in \partial e\setminus i} q_{j\to e}(t_j) \tag{6.10}$$

$$q_i(t_i) \propto n_{t_i} \left(\prod_{\substack{f\in E \\ f\in \partial i}} \widehat{q}_{f\to i}(t_i) \right) \exp(-h(t_i)) \tag{6.11}$$

$$h(t_i) = \frac{C'}{N} \sum_{j\in V} \sum_{t_j} c_{t_i t_j} q_j(t_j)\,, \tag{6.12}$$

where $C' = \sum_{d=2}^{D} \binom{N-2}{d-2} \frac{1}{\kappa_d}$.

A proof of theorem 6.1 is provided in Supp. Mat. The updates in Equations 6.9–6.12 are in principle computationally feasible, as products of function nodes $f \in E$ have replaced products over the entire space $f \in \Omega$. In sparse graphs, that we observe in many real datasets, E is much smaller than the original Ω, thus significantly decreasing the computation cost. An intuitive justification of theorem 6.1, which we formalize in its proof, is that the observed interactions $f \in E$ hold most of the weight in the updates of their neighbors, while the unobserved ones $f \in \Omega \setminus E$ send approximately constant messages and thus can be absorbed in the external field h introduced in Equation 6.12. This idea is inspired by the dyadic MP equations in Decelle *et al.* [55]. However, in contrast to MP on graphs, a vanilla implementation of the updates is still not scalable in hypergraphs, as the computational cost of Equation 6.10 is $O(K^{|e|-1})$. To tackle this issue, we develop a dynamic programming approach that reduces the complexity to $O(K^2|e|)$. Dynamic programming is exact, as it does not rely on further approximations on the MP updates, its detailed derivations are provided in Supp. Mat.

The fixed-point equations of theorem 6.1 naturally suggest an algorithmic implementation of the MP inference procedure. We present a pseudocode for it in Supp. Mat.

6.3.3 Expectation-Maximization to learn the model parameters

We have presented a MP routine for inferring the community assignments $\{t_i\}_{i\in V}$. Now, we derive closed-form updates for the model parameters c, n via an Expectation-Maximization (EM) routine [199]. Differentiating the log-likelihood in eq. 6.4 with respect to n, and imposing the constraint $\sum_{a=1}^{K} n_a = 1$, yields the update

$$n_a = \frac{N_a}{N}\,. \tag{6.13}$$

Notice that this update depends on the MP results, as $N_a = |\{i \in V : \arg\max_b q_i(b) = a\}|$ is the count of nodes assigned to community a according to the inferred marginals. To update the rescaled affinity c we adopt a variational approach, where we maximize a lower bound of the log-likelihood, or, equivalently, minimize a variational free energy. In Supp. Mat., we show detailed derivations for the following fixed-point updates

$$c_{ab}^{(t+1)} = c_{ab}^{(t)} \frac{2\sum_{e\in E} \#_{ab}^{e}/\pi_e}{N\,C'\,(N n_a n_b - \delta_{ab} n_a)}\,, \tag{6.14}$$

where $\#_{ab}^{e} = \sum_{i<j\in e} \delta_{t_i a}\delta_{t_j b}$ is the count of dyadic interactions between two communities a, b within a hyperedge e. In practice, when inferring t, n, c one proceeds by alternating MP inference of t, as presented in section 6.3.2, with the updates of c and n in eqs. 6.13–6.14 until convergence. A pseudocode for the EM procedure is presented in Supp. Mat.

6.3.4 Sampling from the generative model

One of the main advantages of using a probabilistic formulation is the ability to generate data with a desired community structure. Among other tasks, this can be used in particular to test detectability results like the ones we theoretically derive in the following section. However, in hypergraphs, writing a probabilistic model does not directly imply the ability to sample from it, as is typically the case for graphs [2, 224]. In fact, while the $O(N^2)$ configuration space of graphs allows performing sampling explicitly, in the case of hypergraphs the exploding configuration space Ω makes this task prohibitive, even for hypergraphs with moderate number of nodes and hyperedge sizes.

We propose a sampling algorithm that can efficiently scale and produce hypergraphs of dimensions in the tens or hundreds of thousands of nodes. We exploit the hard-membership nature of the assignments to obtain exact sampling via combinatorial arguments, as opposed to the approximate sampling in recent work for mixed-membership models [2]. The key observation to obtain an efficient algorithm is that the hyperedge probabilities do not

depend on the nodes they contain, but only on their community assignments, as implied by Equation 6.3.

With this in mind, we define the auxiliary quantity

$$\#_a^e = \sum_{i \in e} \delta_{t_i a} \,, \tag{6.15}$$

for a hyperedge e and community $a \in [K]$, which is the count of nodes in e that belong to community a. Crucially, the hyperedge probability depends only on these counts:

$$\pi_e = \sum_{a<b \in [K]} \#_a^e \, \#_b^e \, p_{ab} + \sum_{a \in [K]} \frac{\#_a^e (\#_a^e - 1)}{2} \, p_{aa} \,. \tag{6.16}$$

Therefore, all hyperedges with different nodes, but same counts $\#_1^e, \ldots, \#_K^e$, have equal probability.

Using Equation 6.16, we sample hypergraphs as in Algorithm 5 with the following steps:

1. Iterate over the combinations.
   For hyperedges of size $d = 2$, sample all the $N(N-1)/2$ edges directly. Otherwise, iterate the steps (ii), (iii), (iv) for the hyperedge sizes $d = 3, \ldots, D$ and vectors $\# = (\#_1, \ldots, \#_K)$ of community counts (where we omitted the superscript e to highlight that same counts yield identical Equation 6.16) satisfying $\sum_{a=1}^{K} \#_a = d$.

2. Compute the probability.
   For a given count vector #, the hyperedge probability $\pi_\#$ is given in eq. 6.16. Notice that there are $N_\# = \binom{N_1}{\#_1} \cdot \ldots \cdot \binom{N_K}{\#_K}$ hyperedges satisfying the count #, since we can choose $\#_a$ nodes from the N_a nodes in each community a.

3. Sample the number of hyperedges.
   Importantly, we do not sample the individual hyperedges, but the *number* of observed hyperedges. Since the individual hyperedges are independent Bernoulli variables with same probability, their sum X follows a binomial distribution:

   $$X \sim \text{Binom}\left(N_\#, \frac{\pi_\#}{\kappa_d}\right) \tag{6.17}$$

   with probability $\pi_\#$ fixed, determined by #, and number of realizations $N_\#$. Sampling directly from eq. 6.17 is numerically challenging for large $N_\#$ and κ_d, hence we adopt a series of numerical approximation summarized in Supp. Mat.

4. Sample the hyperedges.
   Given the count X of hyperedges sampled from Equation 6.17, we can sample the hyperedges. This operation is performed by independently sampling X times $\#_a$ nodes from each community a. Notice that this procedure might yield repeated hyperedges, which are not allowed. In sparse regimes, this event has low probability [52]. As a sensible approximation, we delete repeated hyperedges.

Owing to this sampling procedure, our results are not limited to theoretical derivations, but can be tested numerically on synthetic data, as we show in Supp. Mat.. In Supp. Mat. we give a detailed analysis of the complexity, which is asymptotically upper bounded by $O(N \log N)$. A pseudocode for this procedure is shown in Algorithm 5 and we provide an open source implementation of the sampling procedure at github.com/nickruggeri/hypergraph-message-passing.

Algorithm 5: Sampling hypergraphs

Input: D maximum size of hyperedges
N number of nodes
K number of communities
n prior of the community memberships
p affinity matrix

Sample node memberships using eq. 6.1
for $d = 2, \dots, D$ **do**
    (i)
    if $d = 2$ **then**
        Sample $N(N-1)/2$ (hyper)edges from eq. 6.2
    else
        for *each* $\# = (\#_1, \dots, \#_K)$ *such that* $\sum_{a=1}^{K} \#_a = d$ **do**
            (i)
            Compute $\pi_{\#}$ with Equation 6.16 (ii)
            Sample X from Equation 6.17 (iii)
        end
        for $a = 1, \dots, K$ **do**
            Sample X times $\#_a$ nodes (iv)
        end
    end
    Delete repeated hyperedges
end

6.4 Phase transition

6.4.1 Detectability bounds

Beside providing a valid and efficient inference algorithm, one of the main advantages of MP is the possibility of deriving closed-form expressions for the detectability of planted communities. The transition from detectable to undetectable regimes has been first shown to exist in MP-based inference models for graphs [55], and gave rise to an extensive body of literature on theoretical detectability limits and sharp phase transitions [205, 206]. Here, we extend these classical arguments to hypergraphs, and find relevant differences when higher-order interactions are considered.

In line with previous work, we restrict our study to the case where groups have constant expected degrees. In fact, in settings where such an assumption does not hold, it is possible to obtain good classification by simply clustering nodes based on their degrees [55]. Formally, we assume

$$\sum_{b=1}^{K} c_{ab} n_b = c\,, \tag{6.18}$$

for some fixed constant c. Notice that eq. 6.18 does not immediately imply a constant degree for the groups, as in hypergraphs the expected degree is defined differently than the left-hand-side of the equation above. Nevertheless, in Supp. Mat. we prove that imposing the condition in eq. 6.18 does indeed imply a constant average degree. More precisely,

Proposition 6.2 *Assuming eq.* 6.18*, the following holds:*

- *all the groups have the same expected degree;*
- *the fixed points for the messages read*

$$q_{i\to e}(t_i) = n_{t_i} \quad \forall e \in E, i \in e \tag{6.19}$$

$$\widehat{q}_{e\to i}(t_i) = \frac{1}{K} \quad \forall e \in E, i \in e\,. \tag{6.20}$$

We want to study the propagation of perturbations around the fixed points of Equations 6.19–6.20. We assume that the factor graph is locally tree-like, i.e., neighborhoods of nodes are approximately trees. We provide a visualization of this in fig. 6.2. Classically, it has been proven that for sparse graphs almost all nodes have local tree-like structures up to distances of order $O(\log N)$ [11]. We are not aware of similar statements for hypergraphs. While our empirical results prove that these assumptions are reasonable and approximately valid, we leave the formalization of such an argument for future work.

Referring to Figure 6.2(b), one can see that between every leaf and the root, there is a single connecting path. Thus, perturbations on the leaves propagate

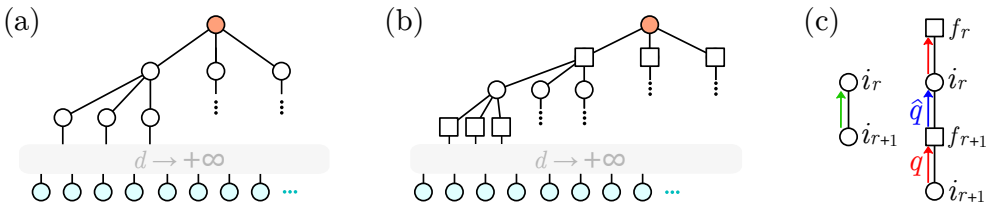


Figure 6.2: Local tree assumption. (a) The classical local tree assumption for graphs. Here, it is assumed that the neighborhoods of nodes are approximately trees. **(b)** The tree assumption for factor graphs. Here, a path from a leaf (light blue) to a root (orange) consists of steps alternating variable nodes and function nodes. These two representations coincide in the case of graphs. **(c)** The perturbations propagate up the tree via the messages. In graphs (a), they reach the root passing from nodes i_{r+1} to i_r (green). In hypergraph-induced factor graphs, perturbations spread from a node i_{r+1}, at depth $r+1$, to its neighboring function nodes f_{r+1} (red), and up to node i_r at depth r (blue) in an alternating fashion.

through a tree to the root, and transmit via the following transition matrix

$$\tilde{T}_r^{ab} = \frac{\partial q_{i_r \to f_r}(a)}{\partial q_{i_{r+1} \to f_{r+1}}(b)}\,, \tag{6.21}$$

where i_r, f_r are respectively the r-th variable node and function node in the path. In words, this is the dependency of a message on the message one level below in the path. In Supp. Mat. we show that, to leading terms in N, the transition matrix evaluates to

$$\tilde{T}_r^{ab} = \frac{2\,n_a}{|f_r|(|f_r|-1)}\left(\frac{c_{ab}}{c}-1\right)\,. \tag{6.22}$$

A related expression was previously obtained for the transition matrix on graphs is $T^{ab} = n_a\,(c_{ab}/c - 1)$ [55]. Hence, we can compactly write

$$\tilde{T}_i^{ab} = \left[2/(|f_r|(|f_r|-1))\right] T^{ab}\,.$$

This connection highlights an important difference between the two cases: hyperedges induce a higher-order prefactor with a "dispersion" effect. The larger the hyperedge, the lower is the magnitude of this transition. Instead, if the hyperedge is a pair, this prefactor reduces to one, and we recover the result on graphs. A perturbation $\epsilon_{t_d}^{k_d}$ of a leaf node k_d influences the perturbation $\epsilon_{t_0}^{k_0}$ on the root t_0 by

$$\epsilon_{t_0}^{k_0} = \sum_{\{t_r\}_{r=1,\dots,d}} \left(\prod_{r=0}^{d-1} \tilde{T}_i^{t_r t_{r+1}}\right) \epsilon_{t_d}^{k_d}\,. \tag{6.23}$$

We can also express this connection in matrix form as

$$\epsilon^{k_0} = \left(\prod_{r=0}^{d-1} \frac{2}{|f_r|(|f_r|-1)}\right) T^d \epsilon^{k_d}\,, \tag{6.24}$$

where T is the matrix with entries T^{ab} (in Equation 6.24 raised to the power of d), and ϵ^{k_d} the array of $\epsilon_{t_d}^{k_d}$ values. Now, similarly to Decelle *et al.* [55], we consider paths of length $d \to +\infty$. In such a case, the r-dependent prefactor in Equation 6.24 converges almost surely to

$$\mu = \exp\left(\mathbb{E}\left[d \log \frac{2}{|f|(|f|-1)}\right]\right), \tag{6.25}$$

where the expectation is taken with respect to randomly drawn hyperedges $f \in E$. If λ is the leading eigenvector of T, then

$$\epsilon^{k_0} \approx \mu\, \lambda^d \epsilon^{k_d}. \tag{6.26}$$

Aggregating over the leaves, and since the perturbations have an expected value of zero, we obtain variance:

$$\langle (\epsilon_{t_0}^{k_0})^2 \rangle \approx \left\langle \left(\sum_{k=1}^{[d_0(F-1)]^d} \mu\, \lambda^d \epsilon_t^k \right)^2 \right\rangle \tag{6.27}$$

$$\overset{\text{i.i.d.}}{=} (d_0(F-1))^d \mu^2\, \lambda^{2d} \langle (\epsilon_t^k)^2 \rangle, \tag{6.28}$$

where d_0 is the average node degree and F the average hyperedge size. The expression in Equation 6.28 yields the following stability criterion, the key result of our derivations:

$$d_0(F-1)\left(\exp \mathbb{E}\left[\log \frac{2}{|f|(|f|-1)}\right]\right)^2 \lambda^2 < 1. \tag{6.29}$$

This generalizes the seminal result $c\lambda^2 < 1$ of Decelle *et al.* [55] to hypergraphs. When Equation 6.29 holds, the influence of the leaves to the root decays when propagating up the tree in Figure 6.2(b). Conversely, if Equation 6.29 is not satisfied, it grows exponentially.

To obtain more interpretable bounds, we focus on a benchmark scenario where the affinity matrix contains all equal on- and off-diagonal elements, i.e., $c_{aa} = c_{\text{in}}$ for all $a \in [K]$ and $c_{ab} = c_{\text{out}}$ for all $a \neq b$. In this case, condition eq. 6.18 becomes $c_{\text{in}} + (K-1)c_{\text{out}} = Kc$, the leading eigenvalue of T is $\lambda = (c_{\text{in}} - c_{\text{out}})/Kc$, and the stability condition in eq. 6.29 reads

$$|c_{\text{in}} - c_{\text{out}}| > \frac{Kc}{\sqrt{d_0(F-1)}} \exp\left(-\mathbb{E}\left[\log \frac{2}{|f|(|f|-1)}\right]\right). \tag{6.30}$$

When hypergraphs only contain dyadic interactions, Equation 6.30 reduces to the bound $|c_{\text{in}} - c_{\text{out}}| > K\sqrt{c}$ previously derived for graphs [55], also known as Kesten-Stigum bound [225, 226].

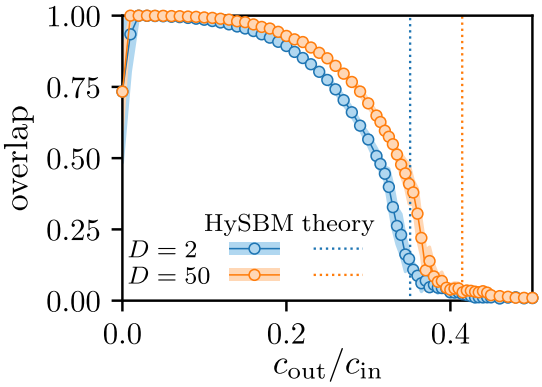


Figure 6.3: Phase transition. The overlap between ground truth and inferred communities varies for different $c_{\text{out}}/c_{\text{in}}$ ratios. The values attained are positive on the detectable region (left of the dotted theoretical bounds) and continuously drop to zero as the phase transition boundary approaches. We attribute the small drop at c_{out}/c_{in} to the the hypergraph being possibly disconnected, hence resulting in some communities being assigned the same label when the initialization in unfavourable. Values for hyperedges up to size $D = 50$ (orange) always yield higher overlap compared to $D = 2$ (light blue). Shaded areas are standard deviations over 5 random initializations of MP.

6.4.2 Phase transition in hypergraphs

We test the bound obtained in Equation 6.30 by running MP on synthetic hypergraphs generated via the sampling algorithm of Section 6.3.4. In our experiments, we fix $K = 4$ and sample hypergraphs with $N = 10^4$ nodes. We also fix $c = 10$ and change the ratio $c_{\text{out}}/c_{\text{in}}$. In this setup, for graphs, one expects a continuous phase transition between two regimes where the system is undetectable and detectable [55]. In the former, where the inequality yielded by the Kesten-Stigum bound does not hold, and the graph does not carry sufficient information about the community assignments, community detection is impossible. In the latter, communities can be efficiently recovered by MP. In Figure 6.3 we plot the overlap $= (\sum_i q_i^\star/N - \max_a n_a)/(1 - \max_a n_a)$ with $q_i^\star \equiv q_i(a_i^\star)$ and $a_i^\star = \arg\max_b q_i(b)$, against $c_{\text{out}}/c_{\text{in}}$. Our results are in agreement with the theoretical predictions: the overlap is low in the undetectable region, high in the detectable region, and we observe a continuous phase transition at the Kesten-Stigum bound for graphs, i.e., when $D = 2$.

We expect the presence of higher-order interactions to improve detectability, as it yields greater overlap for any $c_{\text{out}}/c_{\text{in}}$ and it shifts the theoretical transition to larger values. We empirically validate this prediction by evaluating Equation 6.30 for hyperedges up to size $D = 50$ and performing MP inference in Figure 6.3. Diverging convergence times for larger $c_{\text{out}}/c_{\text{in}}$, i.e., when the free energy landscape gets progressively rugged, further demonstrate this behavior, as shown in Supp. Mat.

6.4.3 The impact of higher-order interactions on detectability

As mentioned above, the transition matrix in eq. 6.22 reduces to the classic T^{ab} [55] when only dyadic interactions are present. In fact, the additional prefactor $2/(|f_r|(|f_r|-1))$ is equal to one for 2-dimensional hyperedges. However, when hyperedges of higher sizes are present, this prefactor is strictly smaller than one. This dampens the perturbations ϵ^{k_0} when they propagate up the tree in fig. 6.2(b). It is unclear whether this higher-order effect aids or hurts detectability, as it could prevent signal from being propagated, but also noise from accumulating at the root.

With this in mind, we investigate the impact of higher-oder interactions on detectability by disentangling the effect that K, c and, most importantly, D have on the detectability bound set by Equation 6.30. To this end, we rewrite Equation 6.30 as

$$\left|\rho_{\text{in}} - \frac{1}{Kc}\right| > \Phi(K, c, D)\,. \tag{6.31}$$

Here, we utilized $c_{\text{in}}/Kc = \rho_{\text{in}} \in [0,1]$, a degree-independent rescaling of c_{in}, where we normalize by its maximum possible value Kc, as per Equation 6.18. The term $\Phi(K,c,D)$ is the value of the theoretical bound at the r.h.s. of Equation 6.30, normalized by Kc as well. This way, we get the decomposition $\Phi(K,c,D) = \alpha(K)\beta(c)\gamma(D)$ as a product of three independent terms:

$$\alpha(K) = \frac{K-1}{K} \tag{6.32}$$

$$\beta(c) = \frac{1}{\sqrt{c}} \tag{6.33}$$

$$\gamma(D) = \frac{\exp\left(-\mathbb{E}\left[\log \frac{2}{|f|(|f|-1)}\right]\right)}{\sqrt{C(F-1)/2}}\,, \tag{6.34}$$

where $C = \sum_{d=2}^{D} \binom{N-2}{d-2} \frac{d}{\kappa_d}$

In our experiments we choose of $\kappa_d = \binom{N-2}{d-2}\frac{d(d-1)}{2}$, which conveniently returns $C = 2H_{D-1}$ (see Supp. Mat.), with H_{D-1} being the $(D-1)$-th harmonic number. However, our theory holds true for any κ_d yielding sparse hypergraphs.

The classic effect of $\alpha(K)$ and $\beta(c)$ is summarized in Figure 6.4(a), where the maximum hyperedges size is fixed to $D = 2$, hence $\gamma(D) = 1$. Here, we observe that the undetectability gap reduces when increasing c. Graphs with higher average degrees are more detectable even when there is a larger inter-community mixing. The effect of larger K is that of skewing the detectability phase transition. This is because edges contributing to c_{out} are spread over $K-1$ communities, while those accounted for c_{in} concentrate in a single one. Intuitively, increasing K allows to have more in-out edges, and detectability

is still possible because of the dominating c_{in} term. The limit value $\rho_{\text{in}} = 1/K$ constitutes the perfect mixing case $c_{\text{in}} = c_{\text{out}} = c$, where detectability is unfeasible for any K and finite degree c. One should notice that, while the bounds drawn in Figure 6.4 hold theoretically, for large K it may be exponentially hard to retrieve communities even in the detectable region [55, 227].

The higher-order effects on detectability are shown in Figure 6.4(b)-(c). The presence of hyperedges with $D > 2$ enters in Equation 6.34 as the product of two separate contributions, $\gamma(D) = \gamma_1(D)\gamma_2(D)$, where

$$\gamma_1(D) = \exp\left(-\mathbb{E}\left[\log \frac{2}{(|f|(|f|-1))}\right]\right) \tag{6.35}$$

$$\gamma_2(D) = \frac{1}{\sqrt{C(F-1)/2}}\,. \tag{6.36}$$

These two terms have contrasting effects that multiply to obtain the overall trend of $\gamma(D)$: $\gamma_1(D)$ is monotonically increasing while $\gamma_2(D)$ is monotonically decreasing. If we were to consider only the "dispersion" contribution γ_1, we would enlarge the detectability gap by increasing Φ. However, the γ_2 term factors in the increasing number of interactions observed with larger hyperedges. The result is the overall higher-order contribution to detectability $\gamma(D) = \gamma_1(D)\gamma_2(D)$, where the value of γ_2 dominates over γ_1, giving rise to the non-trivial, monotonically decreasing, profile of Figure 6.4(b).

The overall effect of higher-order terms is illustrated by plotting the relative difference $\Delta\Phi(K, c, D) = (\Phi(K, c, D) - \Phi(K, c, 2))/\Phi(K, c, 2)$ for a range of c and D values, with $K = 4$, as shown in Figure 6.4(c). We observe how higher-order interactions lead to better detectability for all c, especially in sparse regimes, where c is small and pairwise information is not sufficient for the recovery of the communities.

6.4.4 Entropy and higher-order information

Hypergraphs are often compared against their clique decomposition, i.e., the graph obtained by projecting all hyperedges onto their pairwise connections, as a baseline network structure [228, 229, 230].

The clique decomposition yields highly dense graphs. For this reason, most theoretical results on sparse graphs are not directly applicable, algorithmic implementations become heavier—many times unfeasible—and storage in memory is suboptimal. Previous work also showed that algorithms developed for hypergraphs tend to work better in many practical scenarios [39]. Intuitively, hypergraphs "are more informative" than graphs [231], as there exists only one clique decomposition induced by a given hypergraph, but possibly more hypergraphs corresponding to a given clique decomposition.

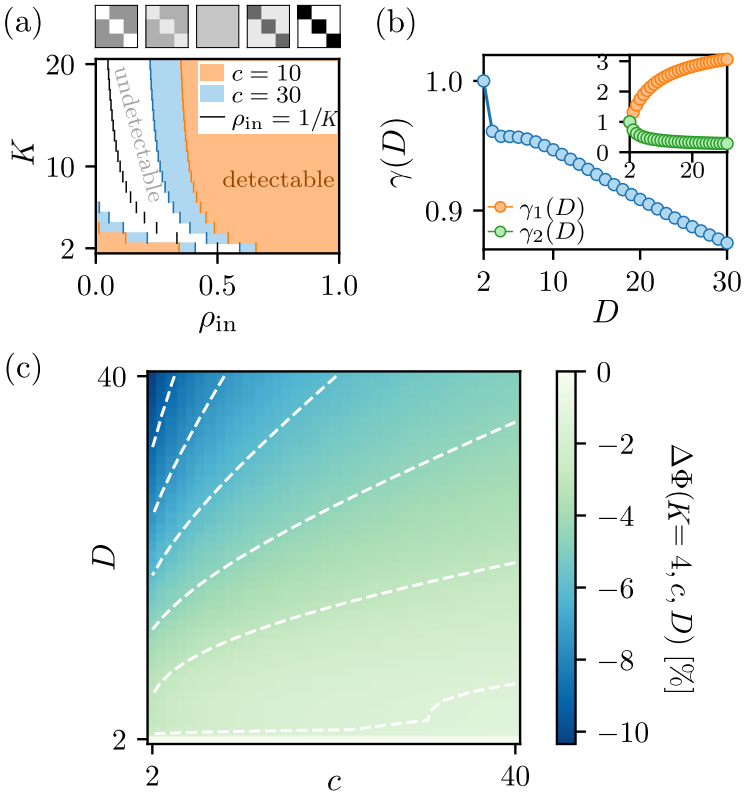


Figure 6.4: Theoretical phase transition. Due to the decomposition of our bound in Equations 6.32–6.34 it is possible to separately describe the effects of K, c and D on the predicted phase transition. **(a)** Detectability bounds for networks $(D = 2)$. Increasing c yields a broader range of detectable configurations (colored areas) for ρ_{in}. The number of communities skews detectability: while for $K = 2$ communities can be detected in extremely disassortative regimes (ρ_{in} close to zero), when more communities are present, only assortative networks are detectable. **(b)** Effect of the maximum hyperedge size D. The term $\gamma(D)$ in Equation 6.34 can be split into the product $\gamma_1(D)\gamma_2(D)$, as defined in Equations 6.35–6.36. The non-trivial decrease of $\gamma(D)$ results from the interplay of $\gamma_1(D)$ and $\gamma_2(D)$, having opposite monotonicity. **(c)** The percentage decrease $\Delta\Phi(K, c, D) = (\Phi(K, c, D) - \Phi(K, c, 2))/\Phi(K, c, 2)$ in detectability for different c, D values shows that higher-order interactions steadily improve detection, especially in sparse regimes.

Here we give a theoretical basis to this common intuition and find that, within our framework, we can quantify the extra information carried by higher-order interactions.

For a given hypergraph $H = (V, E)$, edge $(i, j) \in V^2$ and hyperedge $e \in E$, we define the probability distribution

$$p_H(\{i, j\}, e) = \begin{cases} \dfrac{1}{E} \dfrac{2}{|e|(|e|-1)} & \text{if } i, j \in e \\ 0 & \text{otherwise}. \end{cases} \tag{6.37}$$

This distribution represents the joint probability of drawing a hyperedge uniformly at random among the possible E in the hypergraph and a dyadic

interaction $\{i, j\}$ out of the possible $\binom{|e|}{2}$ within the hyperedge e. From eq. 6.37 we can derive the following marginal distributions:

$$p_E(e) = \frac{1}{E} \tag{6.38}$$

$$p_{\mathcal{C}}(\{i,j\}) = \frac{1}{E} \sum_{e \in E : i,j \in e} \frac{2}{|e|(|e|-1)}\,, \tag{6.39}$$

for all $e \in E$ and pairs of nodes $i \neq j$. The distribution p_E is a uniform random draw of hyperedges. The distribution $p_{\mathcal{C}}$ represents the probability of drawing a weighted interaction $\{i, j\}$ in the clique decomposition of H.

With Equations 6.37–6.39 at hand, it is possible to rewrite $\gamma_1(D)$ in Equation 6.35 as

$$\log \gamma_1(D) = \mathcal{H}(\{i,j\} \,|\, f)\,, \tag{6.40}$$

where $\mathcal{H}(\cdot \,|\, \cdot)$ is the conditional entropy. This entropy is minimized when $p_{\mathcal{C}}(\{i,j\})$ is very different than $p_H(\{i,j\}|f)$, i.e., when conditioning a pair $\{i, j\}$ to be in f brings additional information with respect to the interaction $\{i, j\}$ alone. This happens when $\{i, j\}$ appears in several hyperedges and it is difficult to reconstruct the hypergraph from its clique decomposition. As lower values of γ_1 imply easier recovery, Equation 6.40 suggests that recovery is favored in hypergraphs where hyperedges overlap substantially and that cannot be easily distinguished from their clique decomposition.

We obtain a similar result by rewriting Equation 6.40 as

$$\gamma_1(D) = \frac{\exp \mathcal{H}(p_H)}{\exp \mathcal{H}(p_E)} = \frac{\mathrm{PP}(p_H)}{\mathrm{PP}(p_E)}\,, \tag{6.41}$$

which is the ratio of two exponentiated entropies. In information theory, PP is referred to as perplexity [232], and it is an effective measure of the number of possible outcomes in a probability distribution [233]. Once we fix the number of hyperedges E (and therefore $\mathrm{PP}(p_E)$), the number of effective outcomes is given by the number of likely drawn $\{i, j\}$ pairs. This number is minimized when there is high overlap between hyperedges, thus confirming the interpretation of Equation 6.40.

Finally, we set a different focus by rewriting γ_1 as

$$\log \gamma_1(D) = \mathcal{H}(p_{\mathcal{C}}) - \mathrm{KL}\left(p_H \,||\, p_{\mathcal{C}} \otimes p_E\right)\,, \tag{6.42}$$

where KL is the Kullback-Leibler divergence and $\otimes$ the product probability distribution. Here we pose the question: given a fixed clique decomposition and number of hyperedges, what is the hypergraph attaining the highest

detectability? From the equation, such hypergraph is that with the highest $\mathrm{KL}\left(p_H \,||\, p_{\mathcal{C}} \otimes p_E\right) = I(\{i,j\}, f)$. In this case, the KL-divergence between a joint distribution and its marginals, also called mutual information I [234] of the two random variables, describes the information shared between pairwise interactions and single hyperedges. Hypergraphs with high KL-divergence, i.e, high information about a given $\{i,j\}$ in a single hyperedge f, will yield better detectability. In other words, it is preferable to choose hypergraphs that, while still producing the observed clique decomposition (thus achieving low entropy $\mathcal{H}(p_H)$), have largely overlapping hyperedges. The results discussed in this section provides a theoretically guidance for the construction of hypergraphs that explain an observed graph made of only pairwise interactions [108], a problem relevant in datasets where higher-oder interactions are not explicitly tracked.

6.5 Experiments on real data

Our model leads to a natural algorithmic implementation to learn communities in hypergraphs. In fact, alternating MP and EM rounds, our algorithm outputs marginal probabilities $q_i(t_i)$ for a node i to belong to a community t_i, as well as the community ratios n and the affinity matrix p. We illustrate an application of this procedure on a dataset of interactions between high school students (High School) [177]. Here, nodes are students and hyperedges represent whether a group of students was observed in close proximity, as recorded by wearable devices. The hypergraph contains $N = 327$ nodes and $E = 7818$ hyperedges. In Figure 6.5(a) we show the communities inferred on the dataset where only hyperedges up to size $D = 2, 3, 4$ are kept. We observe a clear progression in how the nodes are gradually allocated into different groups when higher-order interactions are progressively taken into account. This suggests that interactions beyond pairs carry information that would get lost if only edges were to be observed.

To get a qualitative interpretation, we compare the communities inferred with the nine classes attended by the students, an attribute available with the dataset. We illustrate the hypergraph of student interactions, coloring each node according to its class, in Figure 6.5(b). Previous studies have shown that in this dataset a number of interactions happen with stronger prevalence within students of the same class [177]. In Figure 6.5(c), we compare the communities inferred with different maximum hyperedge size D with the classes, and observe that there is a stronger alignment between them when larger hyperedges are utilized for inference. In Figure 6.5(d) we show, at $D = 2, 3, 4$, the Normalized Mutual Information (NMI) between inferred communities and class attributes,the AUC with respect to the full dataset, and the fraction ρ_D of hyperedges with size equal to D. In addition, our algorithm detects connection patterns that were previously observed between

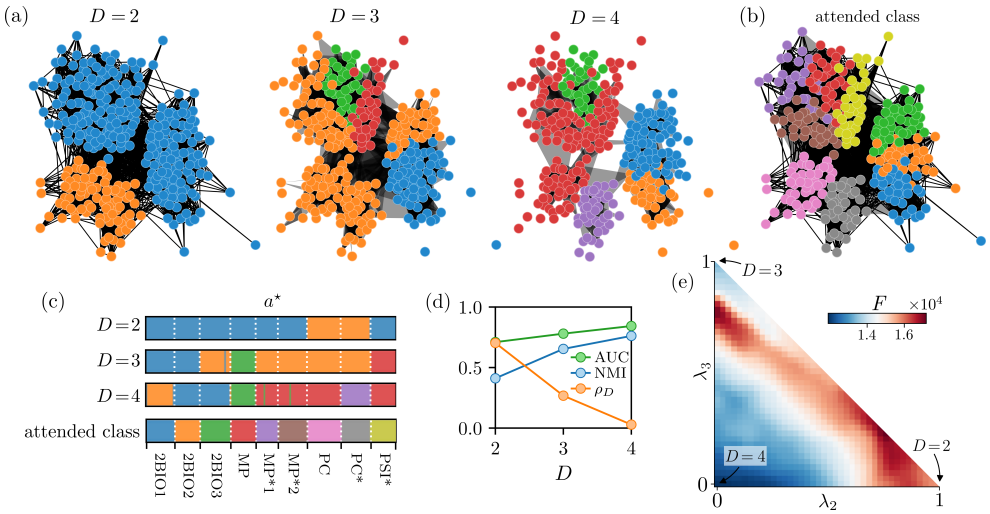


Figure 6.5: Experiments on the High School dataset. We infer the communities via MP and EM on the High School dataset. In all cases, we run inference with $K = 10$ communities. **(a)** Inferred communities on the High School dataset, only utilizing hyperedges up to a maximum size D. Taking into account higher-order information, up to $D = 4$, results in more granular partitions. **(b)** Graphical representation of the students' partition into classes. We draw only hyperedges of size D. **(c)** We compare the inferred partitions with the "attended class" covariate of the nodes, i.e., the classes students participate in. We comment further on this comparison in Supp. Mat.. **(d)** A quantitative measurement complementing that of panel (b): the Normalized Mutual Information (NMI) between inferred communities and attended classes, the AUC on the full dataset, as well as the ratio ρ_D of hyperedges of size equal to D. **(e)** Free energy landscape. We consider the parameters (p_2, n_2), (p_3, n_3) and (p_4, n_4) inferred from the dataset with, respectively, $D = 2, 3, 4$. With these, we build the simplex of convex combinations $p = \sum_{i\in\{2,3,4\}} \lambda_i p_i$, where $\sum_{i\in\{2,3,4\}} \lambda_i = 1$ and $0 \leq \lambda_i \leq 1$ (similarly for n). For every point in the simplex, we compute the free energy on the full dataset, i.e., with $D = 5$. More details on these computations are provided in Supp. Mat..

the different student classes as captured by the affinity matrix p, see Supp. Mat. for details.

A feature that sets MP apart from other inference methods is the possibility to approximately compute the evidence $Z = p(A \mid p, n)$ of the whole dataset, or, equivalently, the free energy $F = -\log Z$. In Supp. Mat. we discuss how to make the free energy computations feasible by exploiting classical cavity arguments, as well as a dynamic program similar to that employed for MP. We present the results of these estimates on the High School dataset in Figure 6.5(e). Here we take the values of n and p inferred by cutting the dataset at maximum hyperedge sizes $D = 2, 3, 4$. Then, we compute the free energy on the full dataset ($D = 5$) in the simplex of n, p parameters outlined by the three vertices. We notice that interactions of size $D = 5$ seem to be less informative and lead to suboptimal inference, see Supp. Mat. Similarly to what observed on graphs [55], the energy landscape appears rugged and complex. EM converges to solutions that are local attraction points, i.e.,

valleys of low-energy configurations. Moreover, the free energy of the p, n parameters inferred with only pairwise interactions (i.e., $D = 2$, lower-right) is higher than that inferred for $D = 3$ (upper-left), which is in turn higher that the one of $D = 4$ (bottom-left).

6.6 Conclusion

We developed a probabilistic generative model and a message-passing-based inference procedure that lead to several results advancing community detection on hypergraphs. In particular we obtained closed-form bounds for the detectability of community configurations, extending the seminal results of Decelle *et al.* [55] to higher-order interactions. Experimental validation of such bounds shows the emergence of a detectability phase transition when spanning from disassortative to assortative community structures. With these theoretical bounds at hand, we investigate the relationship between hypergraphs and graphs from an information-theoretical perspective. Characterizing the entropy and perplexity of pairs of nodes in hyperedges, we find that hypergraphs with many overlapping hyperedges are easier to detect. Beside these theoretical advancements, we develop two relevant algorithmic ones. First, we derive an efficient and scalable Message-Massing algorithm to learn communities and model parameters. Second, we propose an exact and efficient sampling routine that generates synthetic data with desired community structure according to our probabilistic model in order of seconds. Both of these implementations are released open source at github.com/nickruggeri/hypergraph-message-passing.

The mathematical tools we propose here to obtain our results are valid for standard hypergraphs. We can foresee that they could be generalized to dynamic hypergraphs where interactions change in time, using intuitions derived for dynamic graphs [207]. Similarly, it would be interesting to see how detectability bounds change when accounting for node attributes, as results in networks have shown that adding extra information can boost community detection [135, 37, 3]. Finally, from an empirical perspective, it would be interesting to see how our theoretical insights in terms of entropy of hypergraphs and clique expansion match measures that relate hypergraphs to simplicial complexes [235].

Chapter 7

Provable Concept Learning for Interpretable Predictions Using Variational Inference

Abstract

In safety-critical applications, practitioners are reluctant to trust neural networks when no interpretable explanations are available. Many attempts to provide such explanations revolve around pixel-based attributions or use previously known concepts. In this paper we aim to provide explanations by provably identifying *high-level, previously unknown ground-truth concepts*. To this end, we propose a probabilistic modeling framework to derive (C)oncept (L)earning and (P)rediction (CLAP) – a VAE-based classifier that uses visually interpretable concepts as predictors for a simple classifier. Assuming a generative model for the ground-truth concepts, we prove that CLAP is able to identify them while attaining optimal classification accuracy. Our experiments on synthetic datasets verify that CLAP identifies distinct ground-truth concepts on synthetic datasets and yields promising results on the medical Chest X-Ray dataset.

7.1 Introduction

Suppose a hospital aims to deploy a model that classifies diseases $\mathbf{Y}$ from medical images $\mathbf{X}$ and informs the doctor about relevant predictive features. There may be multiple diseases such as lung atelectasis and lung infiltration and multiple *interpretable* ground-truth *features* (or *concepts*) $\mathbf{Z}_c$, such as lung or heart shape, that are relevant for predicting each disease. Ideally, in addition to identifying and utilizing these interpretable features, the model should perform prediction in an interpretable manner itself. The domain expert can then check whether the model is reasonable and also potentially make new scientific discoveries – i.e. discover new factors relevant for prediction.

Thus, in this paper, we seek an interpretable predictive model that uses the ground-truth features for prediction. But what makes a predictive model interpretable from a practical perspective? Even though the definite answer depends on the application domain, practitioners often agree on the following desiderata: first of all, the model should be *simple* – e.g. additive in the predictive features with a small number of relevant features. Simplicity allows us to interpret the relevance of each variable [236], and ensure that the interpretation is robust to small changes to the input [237, 238]. Furthermore, the model ideally assigns *global and local* importance to the features used for prediction [239, 240]; in the context of medical imaging for example, the former corresponds to the population-level importance, the latter to the patient-level one.

While there have been many works on interpretable predictions, none of them provide a prediction model that identifies and uses these previously unknown ground-truth features (see relate works for more discussion). This paper tries to go bottom-up, starting from a generative model to derive a procedure based on variational inference that satisfies all the desiderata. Our proposed framework i) mathematically formalizes concept learning and ii) provably identifies the ground-truth concepts and provides an accurate and simple prediction model using these discovered concepts.

More concretely, we view the recovery of the ground-truth concepts as a latent variable estimation problem. We start by assuming an explicit graphical model for the joint distribution of $(\mathbf{X}, \mathbf{Z}, \mathbf{Y})$. Here, the latent variables $\mathbf{Z}$ include all ground-truth latent features, as well as others irrelevant for prediction. Together, the latent variables $\mathbf{Z}$ generate the raw observation $\mathbf{X}$. The task of concept learning can then be mathematically thought of as obtaining identifiability and performing inference on the latent factors. Using a VAE-based architecture, we enable both visualization (and thus facilitate human interpretation) of the learned concepts, as well as prediction based on these.

In summary, we make the following contributions:

1. We present a framework to model ground-truth latent features $\mathbf{Z}_c$ (Section 7.2), and derive C(oncept) (L)earning and (P)rediction (in short CLAP), an inherently interpretable prediction framework based on variational autoencoders (Section 7.3)
2. We prove that CLAP enables identification of the ground-truth concepts underlying the data and learns a simple optimal prediction model based on these. Importantly, our framework does not require knowing the number of latent features (Section 7.4)
3. We validate CLAP on various multi-task prediction scenarios on synthetic (MPI3D, Shapes3D and SmallNorbs) datasets that yield encour-

aging results on domain-specific application of the framework on real data (Section 7.5)

We believe that our theoretical framework is a useful step for formalizing interpretable predictions. In particular, in settings where it's reasonable to assume that the ground-truth features are themselves interpretable by a domain expert, CLAP provably provides an end-to-end interpretable prediction model. Even when the assumption does not hold, we can still guarantee that CLAP finds a simple and accurate prediction model using ground-truth features.

7.1.1 Related work

In this section, we compare existing interpretable prediction methods with CLAP in detail, with a concise summary provided in Table 7.1. Previous methods proposed in the context of explainable/interpretable AI can be broadly divided into two categories: (i) providing post-hoc explanations for black-box prediction models and (ii) designing interpretable models that explicitly incorporate transparency into the model design, where the explanation is learned during training.

	Post-hoc explanations			Inherently interpretable	
Desiderata	pixel attribution+ counterfactual	pre-defined concepts	StyleGANs	existing VAEs/ autoencoders	CLAP
Learning visually distinct features	×	×	✓	✓*	✓
Global importance of predictive features	×	✓	×	×	✓
Guarantees: concept learning+prediction	×	×	×	×	✓

Table 7.1: Comparison of CLAP with post-hoc explanation methods and other inherently interpretable techniques. The symbol ✓* highlights that for learning visually distinct features, existing predictive VAEs require strong knowledge of the latent variables or auxiliary variables (in addition to labels).

Post-hoc methods The majority of work on interpretability so far has focused on (i), providing post-hoc explanations for a given prediction model. These include pixel attribution methods [241, 242, 243], counterfactual explanations [244, 245], explanations based on pre-defined concepts [246, 247, 248], and recently developed StyleGANs [249, 250]. Post-hoc methods have a number of shortcomings given our desired objectives: First, it is unclear whether post-hoc explanations indeed reflect the black-box model's true "reasoning" [251, 236]. Even if an expert deems the output of the explanation model as unreasonable, one is unable to determine whether the explanation method or the original model is at fault. Furthermore, by construction, post-hoc methods cannot come with statistical inference guarantees and ensure that the learned concepts align with the ground-truth features. Finally, post-hoc methods are typically used to explain complex classifiers; as a result, they

are unable to provide meaningful global and local importance of features for prediction.

VAE-based methods for inherently interpretable prediction Our procedure CLAP is an inherently interpretable prediction model and similar in spirit to VAE-based prediction techniques. On a high level, existing procedures either are unable to identify the ground-truth latent features or require additional labels. Therefore, they are not applicable in the traditional supervised learning setting considered in this paper (where only $\mathbf{X}, \mathbf{Y}$ are available). Further, none of the existing methods provide simultaneous guarantees for learning the underlying concept and obtaining optimal predictions using these learned features. We provide more specific comparisons next.

Unsupervised VAEs [75] can easily be used for prediction tasks by training a classifier on the latent features. A massive literature proposes various structural adjustments to improve disentanglement [252, 253, 254, 255, 256]. However, [257] empirically and theoretically demonstrate that these methods generally do not successfully identify the ground-truth latent features. Recently proposed VAE methods address the issue of non-identifiability by assuming access to additional data and improve identifiability. However, they either require the label as direct input [76], or labels for auxiliary variables that contain information about the ground-truth latent factors [77, 258] or the ground-truth factors themselves [257]. None of these scenarios are applicable to the traditional supervised learning setting in our paper.

Other works With respect to model architecture, our method is similar to Self-Explaining Neural Networks (SENN) [238] which decomposes a complex prediction model into learning interpretable concepts (using an autoencoder) and a simple (linear) predictor. More broadly, methods based on contrastive learning or multi-view data (e.g. [259, 260, 261, 262, 263]) can identify underlying latent features, albeit with access to pairs of images that share similar sources. Furthermore, the focus of these methods is on representation learning rather than interpretable predictions.

7.2 Modeling interpretable and predictive concepts

We present a probabilistic graphical model that statistically relates the ground-truth latent features $\mathbf{Z}_c$ to the labels and observed variables; our proposed method later uses this model to learn the latent concepts as well as a simple classifier based on these features. We remark that, although the methodology in this paper is presented under a specific generative model, the framework is general and flexible to other modeling choices.

Let $\mathbf{X}$ be raw observations and $\mathbf{Y} \in \mathcal{Y}$ be the associated label vector taking a finite collection of values. In general, $\mathbf{X}$ is comprised of *style factors* $\mathbf{Z}_s$, that should not be relevant for prediction, and high-level *core factors* $\mathbf{Z}_c$ that are

the desired ground-truth concepts. For example, in the context of medical imaging, $\mathbf{Y}$ are various disease labels such as the presence of lung atelectasis and lung infiltration. Core factors $\mathbf{Z}_c$ that one can see in the X-ray image $\mathbf{X}$, such as heart and lung shapes, are typically direct consequences of a patient contracting the disease. Style factors $\mathbf{Z}_s$ such as physiological characteristics of the subject or specialities of the scanner are also factors that appear in the image but are not related to the disease.

A natural model for settings such as the one above is to assume an *anti-causal* model as in Figure 7.1a, where $\mathbf{Z}_c$ is a child of $\mathbf{Y}$, and combines with $\mathbf{Z}_s$ to produce the raw observation $\mathbf{X}$. We assume $\mathbf{Z}_c$ to be independent conditionally on $\mathbf{Y}$, as in the X-ray example, they may often vary independently (across patients) given a disease label. We instead allow arbitrary dependencies within $\mathbf{Z}_s$ and $\mathbf{Y}$.

Aggregating style and core factors in the vector $\mathbf{Z} = (\mathbf{Z}_c, \mathbf{Z}_s)$, we impose the following structural equation model on the graph in Figure 7.1a:

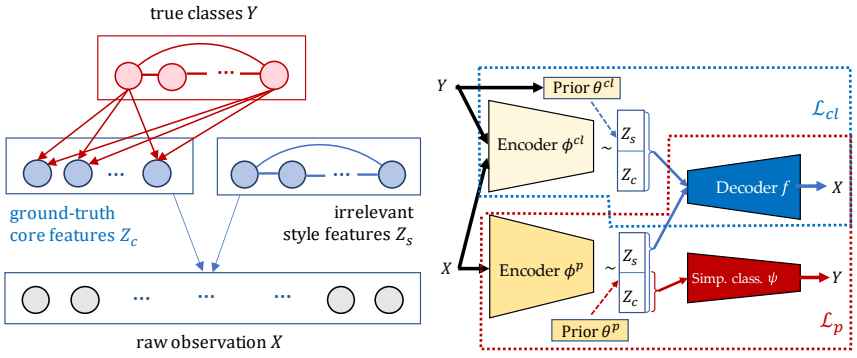
$$\begin{aligned} &\mathbf{X} = f^\star(\mathbf{Z}) + \epsilon \;\text{ where } \epsilon \perp\!\!\!\perp \mathbf{Z}, \mathbf{Y} \text{ and for all } y \in \mathcal{Y}: \\ &\mathbf{Z}|\mathbf{Y}{=}y \sim \mathcal{N}\left(\begin{pmatrix}\mu_y^\star\\ \mu^\star\end{pmatrix}, \begin{pmatrix}D_y^\star & 0\\ 0 & G^\star\end{pmatrix}\right); D_y^\star \text{ diagonal}, \end{aligned} \tag{7.1}$$

for some continuous one-to-one function $f^\star$, vectors $\mu_y^\star, \mu^\star$, and positive-definite matrices $D_y^\star, G^\star$. The model Equation 7.1 encodes the conditional independence relationships in Figure 7.1a: the covariance of the distribution $\mathbf{Z}_c|\mathbf{Y}$ is diagonal; the mean and covariance corresponding to $\mathbf{Z}_s$ are not a function of y and the noise ϵ is independent of $\mathbf{Y}$ so that $\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z}_c$ and $\mathbf{Z}_s \perp\!\!\!\perp \mathbf{Y}$.

7.3 CLAP: interpretable predictions using ground-truth concepts

Given data of $\mathbf{X}$ and $\mathbf{Y}$ arising from the graphical model in Figure 7.1a, our objective is to identify the ground-truth concepts and learn a simple classifier that uses these to accurately predict $\mathbf{Y}$. Additionally, to facilitate human interpretability, we aim to enable experts in the loop to visually interpret the learned concepts. For concreteness, we specialize our exposition to images, although our framework can in principle be used on other types of data.

Our proposed framework is based on variational autoencoders (VAEs) [75, 247]. VAEs offer a number of favorable properties for our objectives. First, they can be derived in a principled manner from the underlying data generating mechanism. Second, the encoder/decoder pair in VAEs provide an effective approach to visualize and thus interpret the learned latent features via latent traversals (see Section 7.3.4 for more details).



(a) Generative graphical model **(b)** VAE architecture of CLAP

Figure 7.1: The graphical model in (a) describes how the desired high-level core latent features $\mathbf{Z}_c$ are related to the remaining variables $\mathbf{Y}, \mathbf{X}, \mathbf{Z}_s$. The VAE architecture in (b) is derived by lower-bounding the evidence values $p(\mathbf{X}, \mathbf{Y})$ and $p(\mathbf{X}|\mathbf{Y})$ and incorporating the generative assumptions from (a) (see main text). We utilize two separate encoders, correspondent to the $\mathcal{L}_{cl}$ and $\mathcal{L}_p$ terms of objective Equation 7.3, and impose sharing of the decoder. The two encoders define two different sets of latents $\mathbf{Z} = (\mathbf{Z}_c, \mathbf{Z}_s)$, which are separately passed through f to get the relative reconstructions. The two resulting objectives $\mathcal{L}_p$ and $\mathcal{L}_{cl}$ are then summed in the full objective $\mathcal{L}_{CLAP}$. A simple classifier based on $\mathbf{Z}_c$ is trained as part of the model inside $\mathcal{L}_p$.

In that light, a natural first approach that might come to mind would be to train a VAE that uses the estimated latent features for prediction. In Section 7.3.1 we derive such a model, and show why, in its vanilla version, it can perform prediction but cannot identify the ground-truth core concepts.

In Section 7.3.2, we overcome these challenges by introducing a novel VAE architecture CLAP shown in Figure 7.1b. Our proposed method combines the predictive VAE structure from earlier with a second VAE which helps with identifying the underlying ground-truth concepts.

7.3.1 Vanilla predictive VAE and its shortcomings

A natural first attempt at learning a predictive VAE procedure is to maximize the following ELBO of the log-evidence of $(\mathbf{X}, \mathbf{Y})$:

$$\log p(\mathbf{X}, \mathbf{Y}) \geq \mathbb{E}_{q_{\phi^p}(\mathbf{Z}|\mathbf{X})} \log \frac{p_f(\mathbf{X}|\mathbf{Z}) p_\psi(\mathbf{Y}|\mathbf{Z}_c) p_{\theta^p}(\mathbf{Z})}{q_{\phi^p}(\mathbf{Z}|\mathbf{X})} =: \mathcal{L}_p(\phi^p, \theta^p, f, \psi; \mathbf{X}, \mathbf{Y}) \,. \tag{7.2}$$

The objective $\mathcal{L}_p$ corresponds to the VAE architecture in the red box in Figure 7.1b. Here, q is the approximate posterior with encoder parameters ϕ^p, ψ parameterizes a simple classifier, f is the decoder's parameters, and θ^p the prior distribution's parameters. Specifically, from the data generating mechanism Equation 7.1, the prior $p_{\theta^p}(\mathbf{Z})$ is a density of a Gaussian mixture distribution with $|\mathbf{Y}|$ (number of labels) components, where the covariance corresponding to the core features for each mixture component is diagonal.

The ELBO Equation 7.2 is derived in a classical fashion by using Jensen's inequality $\log p(\mathbf{X}, \mathbf{Y}) \geq \mathbb{E}_{q(\mathbf{Z}|\mathbf{X},\mathbf{Y})} \log \frac{p(\mathbf{X},\mathbf{Y}|\mathbf{Z})p(\mathbf{Z})}{q(\mathbf{Z}|\mathbf{X},\mathbf{Y})}$ and leveraging the assumed generative model Equation 7.1 to simplify the right-hand side.

The model learned by maximizing the objective $\mathcal{L}_p$ naturally yields a classifier $p_\psi(\mathbf{Y}|\mathbf{Z}_c)$ based on core features extracted from the encoder $q_{\phi^p}(\mathbf{Z}|\mathbf{X})$, which should approximate the ground-truth ones. Since the encoder does not rely on $\mathbf{Y}$ as an input, we can readily use it for end-to-end classification during test time. In fact, under a regularity condition, we show in Supp. Mat. that this architecture is optimal for prediction. However, it *does not* guarantee that the estimated core features $\hat{\mathbf{Z}}_c$ correspond to the ground-truth factors $\mathbf{Z}_c$. In fact, they can be arbitrary linear transformations of $\mathbf{Z}_c$ without sacrificing prediction performance [257] (see ablation studies in Section 7.4), thus not satisfying our desired properties. In addition, as the dimensionality of the core features $\mathbf{Z}_c$ is typically unknown, a conservative choice for the number of latent features (over-parameterized setting) may wrongly include style features or redundant core features in the prediction model (see ablation study in Section 7.4). In the next section, we propose our framework CLAP that mitigates the aforementioned issues: it learns a prediction model using the ground-truth core concepts (even in the over-parameterized setting), without sacrificing classification accuracy.

7.3.2 CLAP to overcome shortcomings

To overcome the aforementioned challenges, we augment the objective $\mathcal{L}_p$ with two additional terms to arrive at our proposed objective function for CLAP:

$$\mathcal{L}_{CLAP} := \mathcal{L}_p + \mathcal{L}_{cl} - \lambda_n \rho \,. \tag{7.3}$$

On a high level, the additional component $\mathcal{L}_{cl}$ ensures identifiability of the ground-truth concepts $\mathbf{Z}_c$ (*concept learning*) and the regularization term $\lambda_n \rho$ helps to identify a minimal number of ground-truth concepts in an over-parameterized latent space. In the following, we formalize each term.

Concept-learning component $\mathcal{L}_{cl}$ While the objective $\mathcal{L}_p$ is designed to maximize the full likelihood of image data $\mathbf{X}$ and target labels $\mathbf{Y}$, the term $\mathcal{L}_{cl}$ maximizes the likelihood of $\mathbf{X}$ conditioned on $\mathbf{Y}$. The fact that the labels act as additional input data in this likelihood objective, plays a central role in provably obtaining identifiability. Furthermore, the conditional independence of $\mathbf{Z}_c$ given $\mathbf{Y}$ can be more naturally captured when $\mathbf{Y}$ is considered as an input. Similarly to above, for any posterior q, we can lower-bound the conditional log-evidence as $\log p(\mathbf{X}|\mathbf{Y}) \geq \mathbb{E}_{q(\mathbf{Z}|\mathbf{X},\mathbf{Y})} \log \frac{p(\mathbf{X}|\mathbf{Z},\mathbf{Y})p(\mathbf{Z}|\mathbf{Y})}{q(\mathbf{Z}|\mathbf{X},\mathbf{Y})}$, and incorporate the generative assumptions in Equation 7.1 to obtain the final

ELBO objective:

$$\log p(\mathbf{X}|\mathbf{Y}) \geq \mathbb{E}_{q_{\phi^{cl}}(\mathbf{Z}|\mathbf{X},\mathbf{Y})} \log \frac{p_f(\mathbf{X}|\mathbf{Z}) p_{\theta^{cl}}(\mathbf{Z}|\mathbf{Y})}{q_{\phi^{cl}}(\mathbf{Z}|\mathbf{X},\mathbf{Y})} := \mathcal{L}_{cl}(\phi^{cl}, \theta^{cl}, f; \mathbf{X}, \mathbf{Y}) \,. \quad (7.4)$$

The component of CLAP corresponding to $\mathcal{L}_{cl}$ is highlighted in blue in Figure 7.1b. Here, ϕ^{cl} are the parameters of the encoder, and f those of the decoder. Appealing to the data generating mechanism Equation 7.1, we can further factorize the prior in the form $p_{\theta^{cl}}(\mathbf{Z}|\mathbf{Y}) = p(\mathbf{Z}_c|\mathbf{Y})p(\mathbf{Z}_s)$. Here, $p(\mathbf{Z}_c|\mathbf{Y})$ is a Gaussian density function with diagonal covariance and different parameters for different $\mathbf{Y}$ while we model the prior $p(\mathbf{Z}_s)$ as a standard Gaussian distribution without loss of generality. We aggregate all these parameters in θ^{cl}.

In general, maximizing the ELBO or even the true log-evidence would not allow for of identification the true concepts. However, a simple heterogeneity assumption can alleviate this issue, formally stated in Supp. Mat.

Theorem 7.1 (Concept learning, informal) *The functions $f, f^\star$ satisfy a regularity condition and the distribution of core features change 'enough' when conditioned on different realizations of $\mathbf{Y}$.*

We now utilize these assumptions to prove the following result.

Lemma 7.2 (Maximizing $\mathcal{L}_{cl}$ identifies the ground-truth concepts) *Suppose the data is generated according to the model in Equation 7.1 with no noise, i.e. $\epsilon \equiv 0$ and Theorem 7.1 holds. Suppose $\mathcal{L}_{cl}$ is maximized in the infinite data limit with the correct number of latent features included in the model. Then, the posterior samples $\hat{\mathbf{Z}}_c$ obtained from the encoder $q_{\hat{\phi}^{cl}}$ are equal to the ground-truth features $\mathbf{Z}_c$ up to permutation and scaling.*

We prove this lemma in Supp. Mat., and also extend to the noisy setting in Supp. Mat. Theoretical results for identifiability were previously established in [77]. We note that our guarantees differ substantially and refer to Supp. Mat. for more details. Despite the concept-learning capabilities, a model trained only on $\mathcal{L}_{cl}$ *cannot* be used for prediction since it requires the labels as input to the encoder $q_{\phi^{cl}}(\mathbf{Z}|\mathbf{X},\mathbf{Y})$.

Therefore, we combine the objectives $\mathcal{L}_p$ and $\mathcal{L}_{cl}$ by utilizing the same decoder f in Equation 7.2 and Equation 7.4, as represented in Figure 7.1b. This coupling via a shared decoder is crucial, as it forces the $\mathcal{L}_p$ architecture to also perform concept learning. To see why, first note that in joint training, the two encoders of $\mathcal{L}_{cl}$ and $\mathcal{L}_p$ learn approximately the same latent space. In fact, we show in Theorem 7.3 that the latent spaces align in the infinite data limit. [1]Since $\mathcal{L}_{cl}$ provably identifies the ground-truth features in the latent space, it

[1]Informally speaking, the reason for this is that the latent features in each architecture reconstruct the image via the same decoder. Since the common decoder defines a generative model , the posteriors (i.e. the different encoders) need to be similar as well.

then follows that the estimated core features obtained by the encoder of $\mathcal{L}_p$ closely align with $\mathbf{Z}_c$. Thus, after training the combined objective $\mathcal{L}_p + \mathcal{L}_{cl}$, the trained VAE architecture corresponding to $\mathcal{L}_p$ provides an interpretable prediction model: an input image is mapped to accurate ground-truth core features, which are then used on top of a simple classifier to predict the target label $\mathbf{Y}$. We refer the reader to Section 7.3.4 for more discussion on how the trained CLAP is used at test time.

Sparsity penalty ρ to account for overparameterized latent space We add a regularization term $\lambda_n \rho(f, \psi)$ to impose simultaneous group sparsity on the prediction weights and decoder weights – this ensures that if an estimated core feature feature is predictive, it has non-negligible effect in the reconstruction of the image and vice versa. In particular, let k_c, k_s be the conservative choice on the dimensionality of the core and style features in our VAE model, respectively. Further, let $k = k_c + k_s$ be the total number of latent variables. We consider the following parameterization for the decoder $f = f' \circ B,\ B \in \mathbb{R}^{k \times k}$ and classifier $\psi = \psi' \circ C,\ C \in \mathbb{R}^{k_c \times k_c}$, where $|\mathbf{Y}|$ is the number of labels to be predicted and f', ψ' are one-to-one and continuous. Then, the sparsity inducing penalty $\rho(f, \psi)$ in the combined objective function Equation 7.3

takes the form:

$$\rho(f, \psi) := \sum_{i=1}^{k_c} \mathbb{I}\left[\left\| \begin{pmatrix} B_{:,i}^T & C_{:,i}^T \end{pmatrix} \right\|_2 > 0 \right] + \sum_{i=k_c+1}^{k} \mathbb{I}\left[\left\| B_{:,i}^T \right\|_2 > 0 \right], \qquad (7.5)$$

where the indicator function $\mathbb{I}[\cdot]$ counts the number of latent features effectively utilizes by the model. Note that the nonzero columns of C correspond to core features in the model with predictive power, and the nonzero columns of B correspond to core and style features that are used for reconstruction with the decoder f. For practical considerations, we consider the following convex surrogate in our experiments: $\rho(f, \psi) = \sum_{i=1}^{k_c} \left\| \begin{pmatrix} B_{:,i}^T & C_{:,i}^T \end{pmatrix} \right\|_2$.

7.3.3 Theoretical guarantees for CLAP

In Section 7.3.2, we described how after the training of CLAP, the component corresponding to $\mathcal{L}_p$ can be used as an interpretable prediction model. We next provide guarantees that this prediction model is optimal in terms of accuracy and is based on high-level features that align with the ground-truth concepts. In the sequel, we denote k_c, k_s to be the number of core and style features chosen in the VAE architecture and $k_c^\star, k_s^\star$ to be the dimensions of the true features of the generative model in Figure 7.1a. Further, we use $q_{\hat{\phi}^p}$, $q_{\hat{\phi}^{cl}}$ to denote the encoders obtained by maximizing the objective in Equation 7.3 in the infinite data limit and let $\hat{\mathbf{Z}}$ be the posterior samples obtained from $q_{\hat{\phi}^p}$. Finally, we denote the trained classifier as $\hat{\psi} = \hat{\psi}' \circ \hat{C}$, and the core features

$\hat{\mathbf{Z}}_c$ are specified as the elements corresponding to nonzero columns of $\hat{C}$. Our theory requires Theorem 7.1 for concept learning as well as an assumption about a simple classifier being optimal:

Assumption 1 (optimal classifier) *The Bayes optimal classifier for predicting* $\mathbf{Y}$ *using* $\mathbf{Z}_c$ *belongs to the set of simple classifiers used in CLAP.*

We utilize Assumptions 1 and 2 to prove the following result, which informally states that CLAP learns an optimal prediction model using interpretable ground-truth features.

Theorem 7.3 (CLAP: Optimal and interpretable prediction model) *Consider the same setup as Lemma 7.2. Suppose* $k_c \geq k_c^\star$, $k_s \geq k_s^\star$, *and that Assumptions 1 and 2 hold. Then, the posterior samples* $\hat{\mathbf{Z}}$ *obtained from the encoder* $q_{\hat{\phi}^p}$ *are identical to the posterior samples obtained from the encoder* $q_{\hat{\phi}^{cl}}$. *Furthermore, the core features* $\hat{\mathbf{Z}}_c$ *are 1) optimally predictive:* $\mathbf{Y}|\hat{\mathbf{Z}}_c \overset{dist}{=} \mathbf{Y}|\mathbf{X}$, *and 2) aligned with the ground truth:* $\hat{\mathbf{Z}}_c$ *is equal to* $\mathbf{Z}_c$ *up to scaling and permutation.*

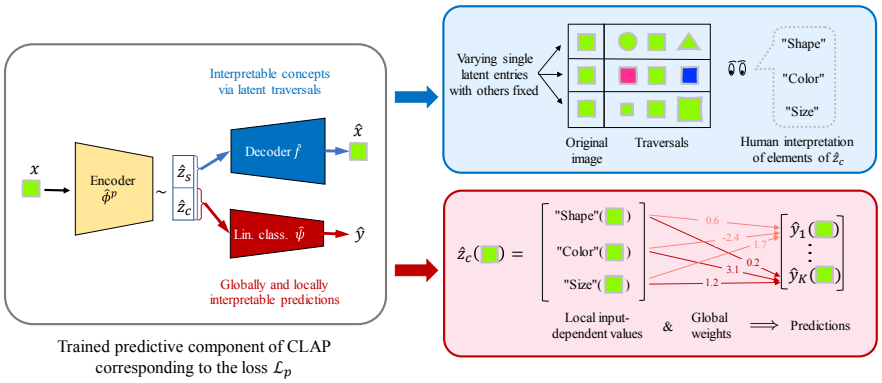
The proof of Theorem 7.3 is presented in Supp. Mat. Our guarantees in Theorem 7.3 ensure that the prediction model obtained by CLAP is optimal. Furthermore, the core features $\hat{\mathbf{Z}}_c$ align with the ground-truth concepts. Finally, the number of predictive factors equals to the number of ground-truth concepts; that is, our model obtains the minimal set of predictive features.

7.3.4 Visualizing and evaluating CLAP's output for interpretation

We now discuss how CLAP's trained model can be used to produce an end-to-end interpretable prediction model pipeline, which we represent in Section 7.3.4.

At inference time, the part of CLAP's model corresponding to $\mathcal{L}_p$ is utilized, since it does not require a label as an input (Section 7.3.4 left). As we describe in detail next, the learned concepts are visualized using latent traversals; to conclude the pipeline, a human expert visually inspects these traversals and assigns a meaning to the relative latent variables.

Interpretations via latent traversals Generally, the visual explanations provided by the model need to be evaluated by a human expert (see Section 7.1). As is customary for VAE models, we provide such visualizations via latent traversals. Specifically, let x be an input image. The core concepts associated to x are obtained via the posterior mean $\hat{\mu}(x) := \mathbb{E}_{q_{\hat{\phi}^p}(\hat{\mathbf{Z}}_c|x)}[\hat{\mathbf{Z}}_c]$. The semantics of $\hat{\mathbf{Z}}_c$ are then discovered by performing latent traversals. In these, we change one component of $\hat{\mu}(x)$ at a time, while keeping the others fixed, and observe the reconstructions obtained through the decoder $\hat{f}$. Owing to the concept-learning capabilities of CLAP, the traversals on the core latent features will



Trained predictive component of CLAP corresponding to the loss $\mathcal{L}_p$

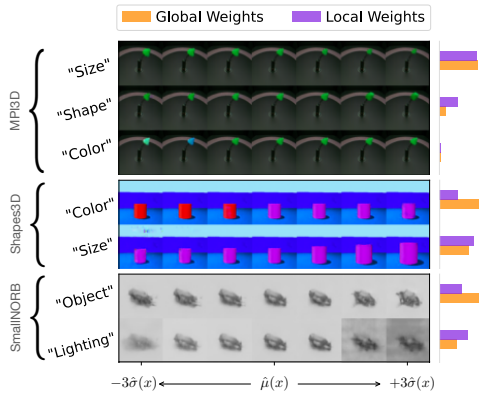
Figure 7.2: We present how the prediction model obtained by training CLAP can be used and interpreted at test time. Supplying a test images x to the component $\mathcal{L}_p$ of CLAP, we learn core features $\hat{\mathbf{Z}}_c$. These features are visualized using latent traversals and interpreted by a human, who assigns them to high-level concepts. Furthermore, the estimated linear classifier predicts a label and provides global (population wise) and local (instance wise) importance for the interpreted concepts.

produce distinct changes in the reconstructed images corresponding to the different discovered ground-truth concepts, which will allow the human expert to assign them with a semantic meaning. This procedure is represented in the top-right of Section 7.3.4. There, for example, upon visual inspection, the first latent is assigned the meaning of *"Shape"* from the expert, the second *"Color"*, and so on.

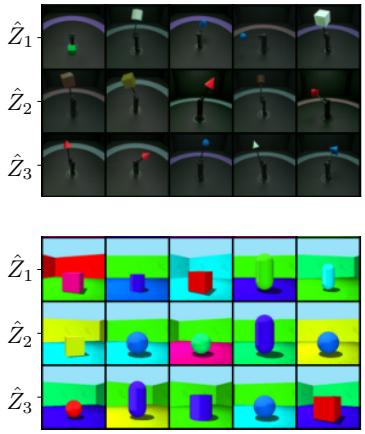
Interpretable predictions using learned concepts We note here that in our experiments, we found a linear classifier to be well-performing across all datasets. For this reason, the following description assumes ψ to simply be the linear weights of the corresponding linear classifier $p_\psi(\mathbf{Y}|\hat{\mathbf{Z}}_c)$. For each concept, we provide both a global and local relevance for prediction, as depicted in the bottom right of Section 7.3.4. The global relevance represents the importance of a concept for prediction at a population level (i.e. across images) and is thus directly encoded in the entries of $\hat{\psi}$. The local relevance is instead image-specific, and is observed in the summands of the linear combination $\langle \hat{\mu}(x), \hat{\psi} \rangle$. These two measures allow the practitioner to transparently assess the decision process of the model, as they assign a prediction weight to human interpretable features.

7.4 Experiments: using CLAP for interpretable predictions

We next present experiments on synthetic data to corroborate our theoretical results, and evaluate the ability of CLAP to learn an accurate prediction model using the ground-truth features. Since in most real-world datasets,



(a) CLAP traversals and interpretations



(b) SENN prototypes

Figure 7.3: a) CLAP traversals on (in order) the MPI3D, Shapes3D and SmallNORB datasets, and b) SENN prototypes on (in order) the MPI3D and Shapes3D datasets.

ground-truth factors are unknown but necessary to verify whether CLAP can work in practice, we resort to three standard "disentanglement" datasets MPI3D [264], Shapes3D [265] and SmallNORB [266]. These datasets consist of collections of objects generated synthetically according to some ground-truth factors of variation. The images are a priori unlabeled; thus, we select some of the ground-truth factors, which represent the concepts $\mathbf{Z}_c$ to be discovered, and generate artificial binary labels $\mathbf{Y}$. The ground-truth factors $\mathbf{Z}_c$ are object shape, size and color for MPI3D, object color and size for Shapes3D and object type and lighting for SmallNORB (see Supp. Mat.). For all the experiments and baselines in Section 7.4, details on training and architectures employed are deferred to Supp. Mat.[2]. In general, for all methods, we used neural network architectures comparable in complexity to those utilized in [76, 267].

[2]Our code is publicly available at https://github.com/nickruggeri/CLAP-interpretable-predictions

As explained in Section 7.3.4, we proceed with the evaluation of CLAP by first generating latent traversals. The goal is to determine whether the discovered concepts have a one-to-one correspondence with the ground-truth $\mathbf{Z}_c$ that we used to generate the data. In Figure 7.3a, every row corresponds to the traversal for one latent feature. As can be observed, the estimated core features indeed represent the ground-truth ones; this means that the model identifies the ground-truth concepts underlying the data generating mechanism. Importantly, we remark that the concept names assigned to the single rows (e.g. *"Size"*, *"Shape"*) are obtained by visual inspection; the model doesn't have direct access to them, but only to the images $\mathbf{X}$ and labels $\mathbf{Y}$.

Finally, the discovered $\mathbf{Z}_c$ are also fully predictive, as CLAP achieves classification accuracy above 0.99 on all the datasets. We include additional traversals in Supp. Mat.; there, we also show that, due to the sparsity regularization penalty $\rho(f, \psi)$, the model accurately assigns negligible global and local weights (i.e. no predictive value) to the remaining latent features included in the model. This is in contrast to the concepts shown in Figure 7.3a that have non-negligible global and local weights. In other words, in line with our theory, estimated core features that have prediction power align with the ground-truth concepts.

Comparison with baselines We compare the outputs of CLAP with those of SENN [238] and CCVAE [76], two prediction models in the existing literature that are closest to CLAP. To explain its predictions and visualize the learned concepts, SENN uses prototypes – a set of training images that "best represent" every latent variable. In Figure 7.3b, we depict the prototypes relative to some of these features. Similarly to CLAP, human inspection is needed to describe the concepts that such latents encode. However, the task here is substantially more difficult: for any of the latents, we can observe many different changes, e.g in the first row objects of different colors and shapes are observed, and from different camera angles. This indicates that not only SENN is not able to identify the ground-truth $\mathbf{Z}_c$, thus hindering interpretability, but also mixes them with non-predictive style features $\mathbf{Z}_s$. We also apply CCVAE on synthetic data and observe that its learned latent features do not align with the ground-truth ones; due to space constraints, we show these results in Supp. Mat.

Ablation studies In order to demonstrate the importance of each of our design choices, we also perform various ablation studies on the MPI3D dataset, presented in Supp. Mat. Firstly, we show that if the sparsity penalty $\lambda_n \rho(f, \psi)$ is removed from the learning objective, the resulting model utilizes separately some latent variables for visualization, and some others for prediction. On the other hand, with the use of $\lambda_n \rho(f, \psi)$, CLAP ensures correspondence between features utilized for prediction and visualization. Furthermore, we show latent traversals for a model trained only on $\mathcal{L}_p$. As explained in

Section 7.3.1, the learned features are fully predictive, but do not correspond to the ground-truth one. In fact, it can be observed that various ground-truth features change jointly within one single traversal. Further, we empirically confirm that the concept-learning capabilities of CLAP rely on the labels $\mathbf{Y}$ being informative enough, as highlighted by the assumptions in Section 7.3.2. Practically, this means that multiple labels help with more accurate recovery of the ground-truth $\mathbf{Z}_c$; we show that the concept learning capabilities of CLAP indeed decrease on a dataset where only one label is available.

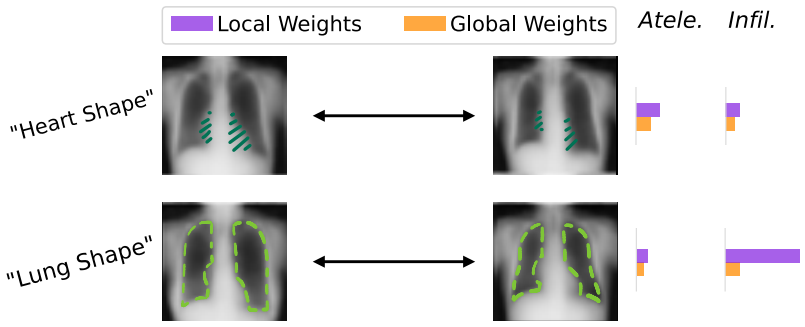
7.5 Future Outlook

So far, we have evaluated CLAP in synthetic scenarios where we know the ground-truth data generating mechanism and the core factors are easy to recognize for a layperson. For many scientific scenarios such as the example in the introduction, evaluating whether learned concepts correspond to the "ground-truth" can only be done by domain experts. Nevertheless, we provide the outputs of CLAP for some challenging real datasets to highlight some of its favorable properties compared to other competing methods.

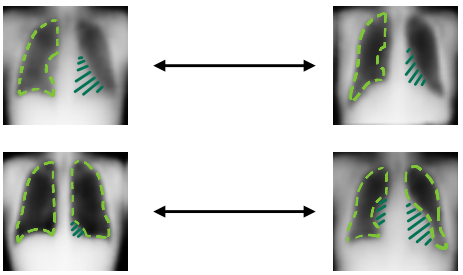
In this section, we present results on the Chest X-ray dataset, and defer additional experiments on the PlantVillage dataset [268] to Supp. Mat. The Chest X-ray dataset [269] consists of radiography images; each image has 14 associated binary disease labels. We emphasize that only the disease labels may be used to learn the underlying concepts and no additional supervision is available. As explained in Section 7.1.1, many inherently interpretable models cannot be applied successfully in this setting, since they generally assume further information on the ground-truth factors. Due to the negative results for SENN in Section 7.4, we only compare our method with CCVAE.

Both CLAP and CCVAE attain similar classification accuracies of 0.903 and 0.898, respectively. In Figure 7.4, we compare the traversals obtained by both methods. First we observe that CLAP manages to learn concepts that are localized in the X-Ray image, corresponding to separate properties, such as *"Heart shape"* and *"Lung shape"*. Instead, in both the traversals presented for CCVAE, characteristics that can be associated to both the heart and lung shapes vary together. Thus, while CCVAE finds similar concepts for prediction, they do not appear distinctly as separate components of $\mathbf{Z}_c$. For this reason, it is harder for a human expert to uniquely label the learned concepts and, consequently, interpret the model's output.

Another desirable characteristic of CLAP is that the global and local weights reflect the importance of the concepts in predicting different diseases. For example, compared to atelectasis, the concept "lung shape" has higher weight (both global and local) in determining the presence of lung infiltration. Since



(a) CLAP traversals and interpretations



(b) CCVAE traversals

Figure 7.4: Output of CLAP and traversals of CCVAE for the Chest X-ray dataset. In (a), we present the weights for both the atelectasis and lung infiltration disease predictions, as well as the human interpretations of the discovered concepts. For better visual comparison, we only show the images obtained at the extremes of the latent traversals. Moreover, we highlight the changes that occur during the traversals. We include magnified figures with full traversals in Supp. Mat., as well as a glossary on how to read the results.

lung infiltration is a condition related to dense substances in the lungs, the concept ”lung shape” learned by CLAP is natural and indicative.

Further, we remark that the discovered concepts manifest through very nuanced traversals. This is sensible, as it is to be expected that real life examples come with subtle and less pronounced features than synthetic and commonly used datasets. In conclusion, these experiments show the advancement and potential of CLAP compared to existing methods for providing real-life interpretable predictions.

There are a number of exciting future directions that can further improve CLAP for broader and more effective use in real-world scenarios. For example, the visualizations of the VAE are not optimally sharp compared to the status quo for GANs. Hence, it would be interesting to explore whether one can obtain provable concept learning when the VAE is replaced by a GAN structure. Further, in many scientific applications, the number of available images can be quite small. An interesting avenue for future research could be to develop solutions for the small data regime, e.g. via transfer learning.

Bibliography

[1] N. Ruggeri, M. Contisciani, F. Battiston, C. De Bacco, Community detection in large hypergraphs. *Science Advances* **9**, eadg9159 (2023).

[2] N. Ruggeri, F. Battiston, C. De Bacco, Framework to generate hypergraphs with community structure. *Physical Review E* **109**, 034309 (2024).

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

[4] N. Ruggeri, A. Lonardi, C. De Bacco, Message-passing on hypergraphs: Detectability, phase transitions and higher-order information. *arXiv preprint arXiv:2312.00708* (2023).

[5] A. Taeb, N. Ruggeri, C. Schnuck, F. Yang, Provable concept learning for interpretable predictions using variational inference. *arXiv preprint arXiv:2204.00492* **160** (2022).

[6] Q. F. Lotito, M. Contisciani, C. De Bacco, L. Di Gaetano, L. Gallo, A. Montresor, F. Musciotto, N. Ruggeri, F. Battiston, Hypergraphx: a library for higher-order network analysis. *Journal of Complex Networks* **11**, cnad019 (2023).

[7] N. Ruggeri, C. De Bacco, Sampling on networks: estimating spectral centrality measures and their impact in evaluating other relevant network measures. *Applied Network Science* **5**, 1–29 (2020).

[8] N. Ruggeri, C. De Bacco, *Complex Networks and Their Applications VIII: Volume 1 Proceedings of the Eighth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2019 8* (Springer, 2020), pp. 90–101.

[9] D. Sherrington, S. Kirkpatrick, Solvable model of a spin-glass. *Physical Review Letters* **35**, 1792 (1975).

[10] B. A. Cipra, An introduction to the ising model. *The American Mathematical Monthly* **94**, 937–959 (1987).

[11] M. Mézard, A. Montanari, *Information, Physics, and Computation* (Oxford University Press, 2009).

[12] E. Aurell, M. Ekeberg, Inverse ising inference using all the data. *Physical Review Letters* **108**, 090201 (2012).

[13] M. I. Jordan, *Learning in graphical models* (MIT press, 1999).

[14] K. P. Murphy, *Machine learning: a probabilistic perspective* (MIT press, 2012).

[15] K. Donhauser, N. Ruggeri, S. Stojanovic, F. Yang, *International Conference on Machine Learning* (PMLR, 2022), pp. 5397–5428.

[16] F. Battiston, E. Amico, A. Barrat, G. Bianconi, G. Ferraz de Arruda, B. Franceschiello, I. Iacopini, S. Kéfi, V. Latora, Y. Moreno, *et al.*, The physics of higher-order interactions in complex systems. *Nature Physics* **17**, 1093–1098 (2021).

[17] M. Contisciani, F. Battiston, C. De Bacco, Inference of hyperedges and overlapping communities in hypergraphs. *Nature Communications* **13**, 7229 (2022).

[18] A. R. Benson, D. F. Gleich, J. Leskovec, Higher-order organization of complex networks. *Science* **353**, 163–166 (2016).

[19] G. Petri, P. Expert, F. Turkheimer, R. Carhart-Harris, D. Nutt, P. J. Hellyer, F. Vaccarino, Homological scaffolds of brain functional networks. *Journal of The Royal Society Interface* **11**, 20140873 (2014).

[20] J. Grilli, G. Barabás, M. J. Michalska-Smith, S. Allesina, Higher-order interactions stabilize dynamics in competitive network models. *Nature* **548**, 210–213 (2017).

[21] S. Klamt, U.-U. Haus, F. Theis, Hypergraphs and cellular networks. *PLOS Computational Biology* **5**, e1000385 (2009).

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

[23] G. Gallo, G. Longo, S. Pallottino, S. Nguyen, Directed hypergraphs and applications. *Discrete Applied Mathematics* **42**, 177–201 (1993).

[24] A. Myers, C. Joslyn, B. Kay, E. Purvine, G. Roek, M. Shapiro, *International Workshop on Algorithms and Models for the Web-Graph* (Springer, 2023), pp. 127–146.

[25] K. Alaluusua, K. Avrachenkov, B. V. Kumar, L. Leskelä, *International Workshop on Algorithms and Models for the Web-Graph* (Springer, 2023), pp. 83–98.

[26] N. W. Landry, M. Lucas, I. Iacopini, G. Petri, A. Schwarze, A. Patania, L. Torres, Xgi: A python package for higher-order interaction networks. *Journal of Open Source Software* **8**, 5162 (2023).

[27] B. K. Fosdick, D. B. Larremore, J. Nishimura, J. Ugander, Configuring random graph models with fixed degree sequences. *Siam Review* **60**, 315–355 (2018).

[28] R. Amritkar, S. Jalan, C.-K. Hu, Synchronized clusters in coupled map networks. ii. stability analysis. *Physical Review E* **72**, 016212 (2005).

[29] S. Fortunato, Community detection in graphs. *Physics Reports* **486**, 75–174 (2010).

[30] W. W. Zachary, An information flow model for conflict and fission in small groups. *Journal of Anthropological Research* **33**, 452–473 (1977).

[31] L. Peel, D. B. Larremore, A. Clauset, The ground truth about metadata and community detection in networks. *Science Advances* **3**, e1602548 (2017).

[32] T. P. Peixoto, A. Kirkley, Implicit models, latent compression, intrinsic biases, and cheap lunches in community detection. *arXiv preprint arXiv:2210.09186* (2022).

[33] P. W. Holland, K. B. Laskey, S. Leinhardt, Stochastic blockmodels: First steps. *Social Networks* **5**, 109-137 (1983).

[34] C. Aicher, A. Z. Jacobs, A. Clauset, Learning latent block structure in weighted networks. *Journal of Complex Networks* **3**, 221–248 (2015).

[35] E. M. Airoldi, D. Blei, S. Fienberg, E. Xing, Mixed membership stochastic blockmodels. *Advances in Neural Information Processing Systems* **21** (2008).

[36] C. De Bacco, E. A. Power, D. B. Larremore, C. Moore, Community detection, link prediction, and layer interdependence in multilayer networks. *Physical Review E* **95**, 042317 (2017).

[37] M. Contisciani, E. A. Power, C. De Bacco, Community detection with node attributes in multilayer networks. *Scientific Reports* **10**, 1–16 (2020).

[38] D. Duan, Y. Li, Y. Jin, Z. Lu, *Proceedings of the 1st ACM International Workshop on Complex Networks Meet Information & Knowledge Management* (2009), pp. 11–18.

[39] M. Contisciani, F. Battiston, C. De Bacco, Inference of hyperedges and overlapping communities in hypergraphs. *Nature Communications* **13**, 7229 (2022).

[40] I. Dumitriu, H. Wang, Exact recovery for the non-uniform Hypergraph Stochastic Block Model (2023).

[41] P. S. Chodrow, N. Veldt, A. R. Benson, Generative hypergraph clustering: From blockmodels to modularity. *Science Advances* **7**, eabh1303 (2021).

[42] P. Chodrow, N. Eikmeier, J. Haddock, Nonbacktracking spectral clustering of nonuniform hypergraphs. *arXiv preprint arXiv:2204.13586* (2022).

[43] T. P. Peixoto, Bayesian stochastic blockmodeling. *Advances in Network Clustering and Blockmodeling* pp. 289–332 (2019).

[44] R. E. Kass, L. Wasserman, The selection of prior distributions by formal rules. *Journal of the American Statistical Association* **91**, 1343–1370 (1996).

[45] C. P. Robert, The metropolis-hastings algorithm (2016).

[46] R. H. Swendsen, J.-S. Wang, Replica monte carlo simulation of spin-glasses. *Physical Review Letters* **57**, 2607–2609 (1986).

[47] D. J. Earl, M. W. Deem, Parallel tempering: theory, applications, and new perspectives. *Physical Chemistry Chemical Physics : PCCP* **7 23**, 3910-6 (2005).

[48] G. Casella, C. P. Robert, M. T. Wells, Generalized accept-reject sampling schemes. *Lecture Notes-Monograph Series* pp. 342–347 (2004).

[49] S. T. Tokdar, R. E. Kass, Importance sampling: a review. *Wiley Interdisciplinary Reviews: Computational Statistics* **2**, 54–60 (2010).

[50] A. Azevedo-Filho, R. D. Shachter, *Uncertainty in Artificial Intelligence* (Elsevier, 1994), pp. 28–36.

[51] M. Welling, Y. W. Teh, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)* (Citeseer, 2011), pp. 681–688.

[52] P. S. Chodrow, Configuration models of random hypergraphs. *Journal of Complex Networks* **8**, cnaa018 (2020).

[53] M. Mézard, G. Parisi, M. A. Virasoro, *Spin glass theory and beyond: An Introduction to the Replica Method and Its Applications*, vol. 9 (World Scientific Publishing Company, 1987).

[54] M. Mézard, G. Parisi, The bethe lattice spin glass revisited. *The European Physical Journal B-Condensed Matter and Complex Systems* **20**, 217–233 (2001).

[55] A. Decelle, F. Krzakala, C. Moore, L. Zdeborová, Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E* **84**, 066106 (2011).

[56] G. T. Cantwell, M. E. J. Newman, Message passing on networks with loops. *Proceedings of the National Academy of Sciences* **116**, 23398-23403 (2019).

[57] A. Kirkley, G. T. Cantwell, M. Newman, Belief propagation for networks with loops. *Science Advances* **7**, eabf1211 (2021).

[58] J. S. Yedidia, W. T. Freeman, Y. Weiss, Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory* **51**, 2282–2312 (2005).

[59] A. Pelizzola, Cluster variation method in statistical physics and probabilistic graphical models. *Journal of Physics A: Mathematical and General* **38**, R309 (2005).

[60] Y. Li, R. E. Turner, *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, R. Garnett, eds. (Curran Associates, Inc., 2016), vol. 29.

[61] D. M. Blei, A. Kucukelbir, J. D. McAuliffe, Variational inference: A review for statisticians. *Journal of the American Statistical Association* **112**, 859–877 (2017).

[62] M. D. Hoffman, M. J. Johnson, *Workshop in Advances in Approximate Bayesian Inference, NIPS* (2016), vol. 1.

[63] S. Zhao, J. Song, S. Ermon, Towards deeper understanding of variational autoencoding models. *arXiv preprint arXiv:1702.08658* (2017).

[64] A. Alemi, B. Poole, I. Fischer, J. Dillon, R. A. Saurous, K. Murphy, *International Conference on Machine Learning* (PMLR, 2018), pp. 159–168.

[65] R. Ranganath, D. Tran, D. Blei, *Proceedings of The 33rd International Conference on Machine Learning*, M. F. Balcan, K. Q. Weinberger, eds. (PMLR, New York, New York, USA, 2016), vol. 48 of *Proceedings of Machine Learning Research*, pp. 324–333.

[66] R. Kingma, M. Welling, Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114* (2013).

[67] D. Rezende, S. Mohamed, *Proceedings of the 32nd International Conference on Machine Learning*, F. Bach, D. Blei, eds. (PMLR, Lille, France, 2015), vol. 37 of *Proceedings of Machine Learning Research*, pp. 1530–1538.

[68] C. M. Bishop, Pattern recognition and machine learning. *Springer* **2**, 5–43 (2006).

[69] R. Ranganath, S. Gerrish, D. Blei, *Artificial Intelligence and Statistics* (PMLR, 2014), pp. 814–822.

[70] R. Ranganath, S. Gerrish, D. Blei, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, S. Kaski, J. Corander, eds. (PMLR, Reykjavik, Iceland, 2014), vol. 33 of *Proceedings of Machine Learning Research*, pp. 814–822.

[71] Y. Burda, R. B. Grosse, R. Salakhutdinov, Importance weighted autoencoders. *CoRR* **abs/1509.00519** (2015).

[72] J. Domke, D. Sheldon, Importance weighting and variational inference. *arXiv preprint arXiv:1808.09034* **abs/1808.09034** (2018).

[73] D. P. Kingma, T. Salimans, M. Welling, Variational dropout and the local reparameterization trick. *Advances in Neural Information Processing Systems* **28** (2015).

[74] E. Jang, S. Gu, B. Poole, Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* (2016).

[75] D. Kingma, D. Rezende, S. Mohamed, M. Welling, *Neural Information Processing Systems* (2014).

[76] T. Joy, S. Schmon, P. Torr, N. Siddharth, T. Rainforth, *International Conference in Learning Representations* (2021).

[77] I. Khemakhem, R. Kingma, P. Monti, A. Hyvärinen, *International Conference on Artificial Intelligence and Statistics* (2020).

[78] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, D.-U. Hwang, Complex networks: Structure and dynamics. *Physics Reports* **424**, 175–308 (2006).

[79] L. Torres, A. S. Blevins, D. Bassett, T. Eliassi-Rad, The why, how, and when of representations for complex systems. *SIAM Review* **63**, 435-485 (2021).

[80] F. Battiston, G. Petri, *Higher-Order Systems* (Springer, 2022).

[81] A. Patania, G. Petri, F. Vaccarino, The shape of collaborations. *EPJ Data Science* **6**, 1–16 (2017).

[82] A. Zimmer, I. Katzir, E. Dekel, A. E. Mayo, U. Alon, Prediction of multidimensional drug dose responses based on measurements of drug pairs. *Proceedings of the National Academy of Sciences* **113**, 10442–10447 (2016).

[83] G. Cencetti, F. Battiston, B. Lepri, M. Karsai, Temporal properties of higher-order interactions in social networks. *Scientific Reports* **11**, 1–10 (2021).

[84] F. Musciotto, D. Papageorgiou, F. Battiston, D. R. Farine, Beyond the dyad: uncovering higher-order structure within cohesive animal groups. *bioRxiv* (2022).

[85] C. Giusti, R. Ghrist, D. S. Bassett, Two's company, three (or more) is a simplex. *Journal of Computational Neuroscience* **41**, 1–14 (2016).

[86] A. Santoro, F. Battiston, G. Petri, E. Amico, Higher-order organization of multivariate time series. *Nature Physics* pp. 1–9 (2023).

[87] T. Carletti, F. Battiston, G. Cencetti, D. Fanelli, Random walks on hypergraphs. *Physical Review E* **101**, 022308 (2020).

[88] C. Bick, P. Ashwin, A. Rodrigues, Chaos in generically coupled phase oscillator networks with nonpairwise interactions. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **26**, 094814 (2016).

[89] P. S. Skardal, A. Arenas, Higher order interactions in complex networks of phase oscillators promote abrupt synchronization switching. *Communications Physics* **3**, 1–6 (2020).

[90] A. P. Millán, J. J. Torres, G. Bianconi, Explosive higher-order kuramoto dynamics on simplicial complexes. *Physical Review Letters* **124**, 218301 (2020).

[91] M. Lucas, G. Cencetti, F. Battiston, Multiorder laplacian for synchronization in higher-order networks. *Physical Review Research* **2**, 033410 (2020).

[92] L. V. Gambuzza, F. Di Patti, L. Gallo, S. Lepri, M. Romance, R. Criado, M. Frasca, V. Latora, S. Boccaletti, Stability of synchronization in simplicial complexes. *Nature Communications* **12**, 1–13 (2021).

[93] Y. Zhang, M. Lucas, F. Battiston, Higher-order interactions shape collective dynamics differently in hypergraphs and simplicial complexes. *Nature Communications* **14**, 1605 (2023).

[94] I. Iacopini, G. Petri, A. Barrat, V. Latora, Simplicial models of social contagion. *Nature Communications* **10**, 1–9 (2019).

[95] S. Chowdhary, A. Kumar, G. Cencetti, I. Iacopini, F. Battiston, Simplicial contagion in temporal higher-order networks. *Journal of Physics: Complexity* **2**, 035019 (2021).

[96] L. Neuhäuser, A. Mellor, R. Lambiotte, Multibody interactions and nonlinear consensus dynamics on networked systems. *Physical Review E* **101**, 032310 (2020).

[97] U. Alvarez-Rodriguez, F. Battiston, G. F. de Arruda, Y. Moreno, M. Perc, V. Latora, Evolutionary dynamics of higher-order interactions in social networks. *Nature Human Behaviour* **5**, 586–595 (2021).

[98] A. Civilini, N. Anbarci, V. Latora, Evolutionary game model of group choice dilemmas on hypergraphs. *Physical Review Letters* **127**, 268301 (2021).

[99] A. Civilini, O. Sadekar, F. Battiston, J. Gómez-Gardeñes, V. Latora, Explosive cooperation in social dilemmas on higher-order networks. *arXiv preprint arXiv:2303.11475* (2023).

[100] C. Berge, *Graphs and hypergraphs* (North-Holland Pub. Co., 1973).

[101] A. R. Benson, Three hypergraph eigenvector centralities. *SIAM Journal on Mathematics of Data Science* **1**, 293–312 (2019).

[102] F. Tudisco, D. J. Higham, Node and edge nonlinear eigenvector centrality for hypergraphs. *Communications Physics* **4**, 1–10 (2021).

[103] A. R. Benson, R. Abebe, M. T. Schaub, A. Jadbabaie, J. Kleinberg, Simplicial closure and higher-order link prediction. *Proceedings of the National Academy of Sciences* **115**, E11221–E11230 (2018).

[104] Q. F. Lotito, F. Musciotto, A. Montresor, F. Battiston, Higher-order motif analysis in hypergraphs. *Communications Physics* **5**, 79 (2022).

[105] Q. F. Lotito, F. Musciotto, F. Battiston, A. Montresor, Exact and sampling methods for mining higher-order motifs in large hypergraphs. *arXiv preprint arXiv:2209.10241* (2022).

[106] F. Musciotto, F. Battiston, R. N. Mantegna, Detecting informative higher-order interactions in statistically validated hypergraphs. *Communications Physics* **4**, 1–9 (2021).

[107] F. Musciotto, F. Battiston, R. N. Mantegna, Identifying maximal sets of significantly interacting nodes in higher-order networks. *arXiv preprint arXiv:2209.12712* (2022).

[108] J.-G. Young, G. Petri, T. P. Peixoto, Hypergraph reconstruction from network data. *Communications Physics* **4**, 1–11 (2021).

[109] K. Balasubramanian, D. Gitelman, H. Liu, Nonparametric modeling of higher-order interactions via hypergraphons. *Journal of Machine Learning Research* **22**, 146–1 (2021).

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

[111] K. Turnbull, S. Lunagomez Coria, C. Nemeth, E. Airoldi, Latent space representations of hypergraphs. *arXiv preprint* (2019).

[112] T. L. J. Ng, T. B. Murphy, Model-based clustering for random hypergraphs. *Advances in Data Analysis and Classification* **16**, 691–723 (2022).

[113] T. Carletti, D. Fanelli, R. Lambiotte, Random walks and community detection in hypergraphs. *Journal of Physics: Complexity* **2**, 015011 (2021).

[114] A. Eriksson, D. Edler, A. Rojas, M. de Domenico, M. Rosvall, How choosing random-walk model and network representation matters for flow-based community detection in hypergraphs. *Communications Physics* **4**, 1–12 (2021).

[115] D. Zhou, J. Huang, B. Schölkopf, Learning with hypergraphs: Clustering, classification, and embedding. *Advances in Neural Information Processing Systems* **19** (2006).

[116] D. Ghoshdastidar, A. Dukkipati, *International Conference on Machine Learning* (PMLR, 2015), pp. 400–409.

[117] M. C. Angelini, F. Caltagirone, F. Krzakala, L. Zdeborová, *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)* (IEEE, 2015), pp. 66–73.

[118] X. Gong, D. J. Higham, K. Zygalakis, Generative hypergraph models and spectral embedding. *Scientific Reports* **13**, 1–13 (2023).

[119] D. Ghoshdastidar, A. Dukkipati, Consistency of spectral partitioning of uniform hypergraphs under planted partition model. *Advances in Neural Information Processing Systems* **27** (2014).

[120] C.-Y. Lin, I. E. Chien, I.-H. Wang, *2017 IEEE International Symposium on Information Theory (ISIT)* (IEEE, 2017), pp. 2178–2182.

[121] K. Ahn, K. Lee, C. Suh, Community recovery in hypergraphs. *IEEE Transactions on Information Theory* **65**, 6561–6579 (2019).

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

[123] N. Veldt, A. R. Benson, J. Kleinberg, Combinatorial characterizations and impossibilities for higher-order homophily. *Science Advances* **9**, eabq3200 (2023).

[124] S. P. Borgatti, M. G. Everett, Models of core/periphery structures. *Social Networks* **21**, 375–395 (2000).

[125] P. Csermely, A. London, L.-Y. Wu, B. Uzzi, Structure and dynamics of core/periphery networks. *Journal of Complex Networks* **1**, 93-123 (2013).

[126] V. Colizza, A. Flammini, M. A. Serrano, A. Vespignani, Detecting rich-club ordering in complex networks. *Nature Physics* **2**, 110–115 (2006).

[127] A. Ma, R. J. Mondragón, Rich-cores in networks. *PloS one* **10**, e0119678 (2015).

[128] I. Amburg, J. Kleinberg, A. R. Benson, Planted hitting set recovery in hypergraphs. *Journal of Physics: Complexity* **2**, 035004 (2021).

[129] F. Tudisco, D. J. Higham, Core-Periphery Detection in Hypergraphs. *SIAM Journal on Mathematics of Data Science* **5**, 1–21 (2023).

[130] B. Klimt, Y. Yang, *European Conference on Machine Learning* (Springer, 2004), pp. 217–226.

[131] H. Safdari, M. Contisciani, C. De Bacco, Generative model for reciprocity and community detection in networks. *Physical Review Research* **3**, 023209 (2021).

[132] M. Contisciani, H. Safdari, C. De Bacco, Community detection and reciprocity in networks by jointly modelling pairs of edges. *Journal of Complex Networks* **10**, cnac034 (2022).

[133] H. Safdari, M. Contisciani, C. De Bacco, Reciprocity, community detection, and link prediction in dynamic networks. *Journal of Physics: Complexity* **3**, 015010 (2022).

[134] N. Nakis, A. Çelikkanat, M. Mørup, *Complex Networks and Their Applications XI: Proceedings of The Eleventh International Conference on Complex Networks and Their Applications: COMPLEX NETWORKS 2022—Volume 1* (Springer, 2023), pp. 350–363.

[135] M. E. Newman, A. Clauset, Structure and inference in annotated networks. *Nature Communications* **7**, 1–11 (2016).

[136] C. Matias, V. Miele, Statistical clustering of temporal networks through a dynamic stochastic block model. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **79**, 1119–1141 (2017).

[137] D. J. Watts, S. H. Strogatz, Collective dynamics of 'small-world'networks. *Nature* **393**, 440–442 (1998).

[138] V. Latora, M. Marchiori, Efficient behavior of small-world networks. *Physical Review Letters* **87**, 198701 (2001).

[139] A.-L. Barabási, R. Albert, Emergence of scaling in random networks. *science* **286**, 509–512 (1999).

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

[141] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms. *Physical review E* **78**, 046110 (2008).

[142] S. Fortunato, D. Hric, Community detection in networks: A user guide. *Physics Reports* **659**, 1–44 (2016).

[143] A. Arenas, A. Diaz-Guilera, C. J. Pérez-Vicente, Synchronization reveals topological scales in complex networks. *Physical Review Letters* **96**, 114102 (2006).

[144] A. Nematzadeh, E. Ferrara, A. Flammini, Y.-Y. Ahn, Optimal network modularity for information diffusion. *Physical Review Letters* **113**, 088701 (2014).

[145] M. Coscia, L. Rossi, How minimizing conflicts could lead to polarization on social media: An agent-based model investigation. *PloS one* **17**, e0263184 (2022).

[146] C. Bordier, C. Nicolini, A. Bifone, Graph analysis and modularity of brain functional connectivity networks: searching for the optimal threshold. *Frontiers in Neuroscience* **11**, 441 (2017).

[147] D. Lusher, J. Koskinen, G. Robins, *Exponential random graph models for social networks: Theory, methods, and applications* (Cambridge University Press, 2013).

[148] E. A. Hobson, M. J. Silk, N. H. Fefferman, D. B. Larremore, P. Rombach, S. Shai, N. Pinter-Wollman, A guide to choosing and implementing reference models for social network analysis. *Biological Reviews* **96**, 2716–2734 (2021).

[149] M. T. Schaub, A. R. Benson, P. Horn, G. Lippner, A. Jadbabaie, Random walks on simplicial complexes and the normalized hodge 1-laplacian. *SIAM Review* **62**, 353–391 (2020).

[150] O. T. Courtney, G. Bianconi, Generalized network structures: The configuration model and the canonical ensemble of simplicial complexes. *Physical Review E* **93**, 062311 (2016).

[151] J.-G. Young, G. Petri, F. Vaccarino, A. Patania, Construction of and efficient sampling from the simplicial configuration model. *Physical Review E* **96**, 032312 (2017).

[152] M. Barthelemy, Class of models for random hypergraphs. *Phys. Rev. E* **106**, 064310 (2022).

[153] K. Kovalenko, I. Sendiña-Nadal, N. Khalil, A. Dainiak, D. Musatov, A. M. Raigorodskii, K. Alfaro-Bittner, B. Barzel, S. Boccaletti, Growing scale-free simplices. *Communications Physics* **4**, 1–9 (2021).

[154] A. P. Millán, R. Ghorbanchian, N. Defenu, F. Battiston, G. Bianconi, Local topological moves determine global diffusion properties of hyperbolic higher-order networks. *Physical Review E* **104**, 054302 (2021).

[155] P. Krapivsky, Random recursive hypergraphs. *Journal of Physics A: Mathematical and Theoretical* **56**, 195001 (2023).

[156] J. Lerner, M. Tranmer, J. Mowbray, M.-G. Hancean, Rem beyond dyads: relational hyperevent models for multi-actor interaction networks. *arXiv preprint arXiv:1912.07403* (2019).

[157] G. Robins, P. Pattison, Y. Kalish, D. Lusher, An introduction to exponential random graph (p*) models for social networks. *Social Networks* **29**, 173–191 (2007).

[158] J. Park, M. E. Newman, Statistical mechanics of networks. *Physical Review E* **70**, 066117 (2004).

[159] J. Mulder, P. D. Hoff, A latent variable model for relational events with multiple receivers. *arXiv preprint arXiv:2101.05135* (2021).

[160] A. Schein, J. Paisley, D. M. Blei, H. Wallach, *Proceedings of the 21th ACM SIGKDD International conference on knowledge discovery and data mining* (2015), pp. 1045–1054.

[161] T. G. Kolda, B. W. Bader, Tensor decompositions and applications. *SIAM review* **51**, 455–500 (2009).

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

[163] B. Karrer, M. E. Newman, Stochastic blockmodels and community structure in networks. *Physical review E* **83**, 016107 (2011).

[164] S. L. Hakimi, On realizability of a set of integers as degrees of the vertices of a linear graph. i. *Journal of the Society for Industrial and Applied Mathematics* **10**, 496–506 (1962).

[165] S. A. Choudum, A simple proof of the erdos-gallai theorem on graph sequences. *Bulletin of the Australian Mathematical Society* **33**, 67–70 (1986).

[166] W. K. Hastings, Monte carlo sampling methods using markov chains and their applications (1970).

[167] P. L. Erdosa, I. Miklósa, The second order degree sequence problem is np-complete. *arXiv preprint arXiv:1606.00730* (2016).

[168] U. Dutta, Sampling random graphs with specified degree sequences, Ph.D. thesis, University of Colorado at Boulder (2022).

[169] J. H. Fowler, Connecting the congress: A study of cosponsorship networks. *Political Analysis* **14**, 456–487 (2006).

[170] J. H. Fowler, Legislative cosponsorship networks in the US house and senate. *Social Networks* **28**, 454–465 (2006).

[171] E. Estrada, J. A. Rodríguez-Velázquez, Subgraph centrality and clustering in complex hyper-networks. *Physica A: Statistical Mechanics and its Applications* **364**, 581–594 (2006).

[172] Y. Zhang, M. Lucas, F. Battiston, Do higher-order interactions promote synchronization? *arXiv preprint arXiv:2203.03060* (2022).

[173] F. Baccini, F. Geraci, G. Bianconi, Weighted simplicial complexes and their representation power of higher-order network data and topology. *Physical Review E* **106**, 034319 (2022).

[174] P. Bonacich, Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology* **2**, 113–120 (1972).

[175] A. Bretto, Hypergraph theory. *An Introduction. Mathematical Engineering. Cham: Springer* (2013).

[176] E. Estrada, J. A. Rodriguez-Velazquez, Complex networks as hypergraphs. *arXiv preprint* (2005).

[177] R. Mastrandrea, J. Fournet, A. Barrat, Contact patterns in a high school: a comparison between data collected using wearable sensors, contact diaries and friendship surveys. *PloS one* **10**, e0136497 (2015).

[178] I. Amburg, N. Veldt, A. Benson, *Clustering in Graphs and Hypergraphs with Categorical Edge Labels* (Association for Computing Machinery, 2020), pp. 706–717.

[179] C. Stewart III, J. Woon, Congressional committee assignments, 103rd to 114th congresses, 1993–2017: House, *Tech. rep.*, MIT mimeo (2008).

[180] D. R. Hunter, S. M. Goodreau, M. S. Handcock, Goodness of fit of social network models. *Journal of the American Statistical Association* **103**, 248–258 (2008).

[181] H. J. Spaeth, L. Epstein, A. D. Martin, J. A. Segal, T. J. Ruger, S. C. Benesh, 2022 supreme court database, version 2020 release 1., http://Supremecourtdatabase.org (2020).

[182] X. Zhang, C. Moore, M. E. Newman, Random graph models for dynamic networks. *The European Physical Journal B* **90**, 1–14 (2017).

[183] M. M. Mayfield, D. B. Stouffer, Higher-order interactions capture unexplained complexity in diverse communities. *Nature Ecology & Evolution* **1**, 0062 (2017).

[184] J. Yang, J. McAuley, J. Leskovec, *2013 IEEE 13th international conference on data mining* (IEEE, 2013), pp. 1151–1156.

[185] O. Fajardo-Fontiveros, R. Guimerà, M. Sales-Pardo, Node metadata can produce predictability crossovers in network inference problems. *Physical Review X* **12**, 011010 (2022).

[186] C. Tallberg, A bayesian approach to modeling stochastic blockstructures with covariates. *Journal of Mathematical Sociology* **29**, 1–23 (2004).

[187] A. Vazquez, Finding hypergraph communities: a bayesian approach and variational solution. *Journal of Statistical Mechanics: Theory and Experiment* **2009**, P07006 (2009).

[188] B. Ball, B. Karrer, M. E. Newman, Efficient and principled method for detecting communities in networks. *Physical Review E* **84**, 036103 (2011).

[189] A. Goldenberg, A. X. Zheng, S. E. Fienberg, E. M. Airoldi, A survey of statistical network models. *Foundations and Trends in Machine Learning* **2**, 129-233 (2010).

[190] Y. Li, R. Yang, J. Shi, Efficient and effective attributed hypergraph clustering via k-nearest neighbor augmentation. *Proceedings of the ACM on Management of Data* **1**, 1–23 (2023).

[191] B. Fanseu Kamhoua, L. Zhang, K. Ma, J. Cheng, B. Li, B. Han, *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (2021), pp. 453–463.

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

[193] M. Contisciani, H. Safdari, C. De Bacco, Community detection and reciprocity in networks by jointly modelling pairs of edges. *Journal of Complex Networks* **10**, cnac034 (2022).

[194] T. Hofmann, *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and development in Information Retrieval* (1999), pp. 50–57.

[195] Y. Zhu, X. Yan, L. Getoor, C. Moore, *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2013), pp. 473–481.

[196] B. H. Good, Y.-A. De Montjoye, A. Clauset, Performance of modularity maximization in practical contexts. *Physical review E* **81**, 046106 (2010).

[197] J. Piñero, J. M. Ramírez-Anguita, J. Saüch-Pitarch, F. Ronzano, E. Centeno, F. Sanz, L. I. Furlong, The disgenet knowledge platform for disease genomics: 2019 update. *Nucleic Acids Research* **48**, D845–D855 (2020).

[198] M. Génois, A. Barrat, Can co-location be used as a proxy for face-to-face contacts? *EPJ Data Science* **7**, 1–18 (2018).

[199] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* **39**, 1–22 (1977).

[200] M. Girvan, M. E. Newman, Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* **99**, 7821–7826 (2002).

[201] R. J. Fletcher, A. Revell, B. E. Reichert, W. M. Kitchens, J. D. Dixon, J. D. Austin, Network modularity reveals critical scales for connectivity in ecology and evolution. *Nature Communications* **4**, 2572 (2013).

[202] M. E. J. Newman, The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences* **98**, 404-409 (2001).

[203] L. M. Shekhtman, S. Shai, S. Havlin, Resilience of networks formed of interdependent modular networks. *New Journal of Physics* **17**, 123007 (2015).

[204] A. Decelle, F. Krzakala, C. Moore, L. Zdeborová, Inference and Phase Transitions in the Detection of Modules in Sparse Networks. *Physical Review Letters* **107**, 065701 (2011).

[205] C. Moore, The Computer Science and Physics of Community Detection: Landscapes, Phase Transitions, and Hardness (2017).

[206] E. Abbe, Community Detection and Stochastic Block Models: Recent Developments. *Journal of Machine Learning Research* **18**, 1–86 (2018).

[207] A. Ghasemian, P. Zhang, A. Clauset, C. Moore, L. Peel, Detectability Thresholds and Optimal Algorithms for Community Structure in Dynamic Networks. *Physical Review X* **6**, 031005 (2016).

[208] D. Taylor, S. Shai, N. Stanley, P. J. Mucha, Enhanced detectability of community structure in multilayer networks through layer aggregation. *Physical Review Letters* **116**, 228301 (2016).

[209] I. Chien, C.-Y. Lin, I.-H. Wang, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, A. Storkey, F. Perez-Cruz, eds. (PMLR, 2018), vol. 84 of *Proceedings of Machine Learning Research*, pp. 871–879.

[210] J. Liang, C. Ke, J. Honorio, *2021 IEEE International Symposium on Information Theory (ISIT)* (2021), pp. 2578–2583.

[211] S. Pal, Y. Zhu, Community detection in the sparse hypergraph stochastic block model. *Random Structures & Algorithms* **59**, 407–463 (2021).

[212] Q. Zhang, V. Y. F. Tan, Exact recovery in the general hypergraph stochastic block model. *IEEE Transactions on Information Theory* **69**, 453-471 (2023).

[213] Y. Gu, Y. Polyanskiy, Weak recovery threshold for the hypergraph stochastic block model (2023).

[214] S. Cole, Y. Zhu, Exact recovery in the hypergraph stochastic block model: A spectral algorithm. *Linear Algebra and its Applications* **593**, 45-73 (2020).

[215] C.-Y. Lin, I. E. Chien, I.-H. Wang, *2017 IEEE International Symposium on Information Theory (ISIT)* (2017), pp. 2178–2182.

[216] D. Ghoshdastidar, A. Dukkipati, *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K. Weinberger, eds. (Curran Associates, Inc., 2014), vol. 27.

[217] M. Yuan, Z. Shang, Information limits for detecting a subhypergraph. *Stat* **10**, e407 (2021).

[218] L. Corinzia, P. Penna, W. Szpankowski, J. Buhmann, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, G. Camps-Valls, F. J. R. Ruiz, I. Valera, eds. (PMLR, 2022), vol. 151 of *Proceedings of Machine Learning Research*, pp. 11615–11640.

[219] J. Jin, Z. T. Ke, J. Liang, *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan, eds. (Curran Associates, Inc., 2021), vol. 34, pp. 7220–7231.

[220] M. Yuan, R. Liu, Y. Feng, Z. Shang, Testing community structure for hypergraphs. *The Annals of Statistics* **50**, 147 – 169 (2022).

[221] J. Pearl, *Proceedings of the Second AAAI Conference on Artificial Intelligence*, AAAI'82 (AAAI Press, 1982), p. 133–136.

[222] M. Mézard, G. Parisi, M. Virasoro, *Spin Glass Theory and Beyond* (World Scientific, 1986).

[223] S. Wasserman, K. Faust, *Social Network Analysis: Methods and Applications*, Structural Analysis in the Social Sciences (Cambridge University Press, 1994).

[224] B. Kamiński, P. Prałat, F. Théberge, Hypergraph Artificial Benchmark for Community Detection (h–ABCD). *Journal of Complex Networks* **11**, cnad028 (2023).

[225] H. Kesten, B. Stigum, Limit theorems for decomposable multidimensional Galton-Watson processes. *Journal of Mathematical Analysis and Applications* **17**, 309-338 (1967).

[226] H. Kesten, B. P. Stigum, Additional Limit Theorems for Indecomposable Multidimensional Galton-Watson Processes. *The Annals of Mathematical Statistics* **37**, 1463 – 1481 (1966).

[227] M. Mézard, A. Montanari, Reconstruction on Trees and Spin Glass Transition. *Journal of Statistical Physics* **124**, 1317-1350 (2006).

[228] E. Schneidman, M. J. Berry, R. Segev, W. Bialek, Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature* **440**, 1007-1012 (2006).

[229] C. Giusti, E. Pastalkova, C. Curto, V. Itskov, Clique topology reveals intrinsic geometric structure in neural correlations. *Proceedings of the National Academy of Sciences* **112**, 13455-13460 (2015).

[230] L. Merchan, I. Nemenman, On the Sufficiency of Pairwise Interactions in Maximum Entropy Models of Networks. *Journal of Statistical Physics* **162**, 1294-1308 (2016).

[231] E. Schneidman, S. Still, M. J. Berry, W. Bialek, Network information and connected correlations. *Physical Review Letters* **91**, 238701 (2003).

[232] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (2003).

[233] L. L. Campbell, Exponential entropy as a measure of extent of a distribution. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete* **5**, 217-225 (1966).

[234] T. Cover, J. Thomas, *Elements of Information Theory* (Wiley, 2006).

[235] N. W. Landry, J.-G. Young, N. Eikmeier, The simpliciality of higher-order networks (2023).

[236] C. Rudin, *arXiv preprint arXiv:1811.10154* (2018).

[237] D. Alvarez-Melis, T. Jaakkola, *arXiv preprint arXiv:1806.08049* (2018).

[238] D. Alvarez-Melis, T. Jaakkola, *Neural Information Processing Systems* (2018).

[239] M. Reyes, R. Meier, S. Pereira, C. Silva, F. Dahlweid, H. von Tengg-Kobligk, R. Summers, R. Wiest, On the interpretability of artificial intelligence in radiology: Challenges and opportunities. *Radiology: Artificial intelligence* **23**, e190043 (2020).

[240] G. Stiglic, P. Kocbek, N. Fijavko, M. Zitnik, K. Verbert, L. Cilar, Interpretability of machine learning-based prediction models in healthcare. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **10** (2020).

[241] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS ONE* **10**, 1–46 (2015).

[242] R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, *International Conference on Computer Vision* (2017).

[243] K. Simonyan, A. Vedaldi, A. Zisserman, *arXiv preprint arxiv:1312.6034* (2014).

[244] J. Antoran, U. Bhatt, T. Adel, A. Weller, J. M. Hernández-Lobato, *International Conference in Learning Representations* (2021).

[245] C.-H. Chang, E. Creager, A. Goldenberg, D. K. Duvenaud, *International Conference in Learning Representations* (2019).

[246] D. Kazhdan, B. Dimanov, M. Jamnik, P. Liò, A. Weller, *International Conference on Information and Knowledge Management* (2020).

[247] D. Rezende, S. Mohamed, D. Wierstra, *International Conference in Machine Learning* (2014).

[248] C. Yeh, B. Kim, S. Arik, C.-L. Li, T. Pfister, P. Ravikumar, *Neural Information Processing Systems* (2020).

[249] O. Lang, Y. Gandelsman, M. Yarom, Y. Wald, G. Elidan, A. Hassidim, W. T. Freeman, P. Isola, A. Globerson, M. Irani, I. Mosseri, *International Conference in Computer Vision* (2021).

[250] Z. Wu, D. Lischinski, E. Shechtman, *Conference on Computer Vision and Pattern Recognition* (2021).

[251] I. E. Kumar, S. Venkatasubramanian, C. E. Scheidegger, S. A. Friedler, *International Conference in Machine Learning* (2020).

[252] C. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, A. Lerchner, *arXiv preprint arXiv:1804.03599* (2018).

[253] T. Q. Chen, X. Li, R. Grosse, D. K. Duvenaud, *Neural Information Processing Systems* (2018).

[254] I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, A. Lerchner, *International Conference in Learning Representations* (2017).

[255] H. Kim, A. Mnih.

[256] A. Kumar, P. Sattigeri, A. Balakrishnan, *International Conference in Learning Representations* (2017).

[257] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, O. Bachem, *International Conference in Machine Learning* (2019).

[258] G. Mita, M. Filippone, P. Michiardi, *International Conference on Machine Learning* (2021).

[259] L. Gresele, P. Rubenstein, A. Mehrjou, F. Locatello, B. Schölkopf, *Uncertainty in Artificial Intelligence* (2019).

[260] A. Hyvärinen, H. Sasaki, R. Turner, *International Conference on Artificial Intelligence and Statistics* (2019).

[261] F. Locatello, B. Poole, G. Raetsch, B. Schölkopf, O. Bachem, M. Tschannen, *International Conference in Machine Learning* (2020).

[262] R. Shu, Y. Chen, A. Kumar, S. Ermon, B. Poole, *International Conference on Learning Representations* (2020).

[263] J. von Kügelgen, Y. Sharma, L. Gresele, W. Brendel, B. Scholkopf, M. Besserve, F. Locatello, *Neural Information Processing Systems* (2021).

[264] M. W. Gondal, M. Wüthrich, D. Miladinović, F. Locatello, M. Breidt, V. Volchkov, J. Akpo, O. Bachem, B. Schölkopf, S. Bauer, *Neural Information Processing Systems* (2019).

[265] C. Burgess, H. Kim, 3D Shapes Dataset, https://github.com/deepmind/3dshapes-dataset/ (2018).

[266] Y. LeCun, F. Huang, L. Bottou, *Computer Vision and Pattern Recognition* (2004).

[267] J. Qiao, Z. Li, B. Xu, R. Cai, K. Zhang, Disentanglement challenge: From regularization to reconstruction. *arXiv preprint arXiv:1912.00155* (2019).

[268] D. Hughes, M. Salathé, *et al.*, *arXiv preprint arXiv:1511.08060* (2015).

[269] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, R. Summers, *Computer Vision and Pattern Recognition* (2017).