Model-based clustering in simple hypergraphs through a stochastic blockmodel

Luca Brusa, Catherine Matias

Model-based clustering in simple hypergraphs through a stochastic blockmodel

Luca Brusa[1] & Catherine Matias[2]
1. University of Milano Bicocca
2. Sorbonne Université, Université de Paris Cité, Centre National de la Recherche Scientifique
luca.brusa@unimib.it

October 12, 2022

Abstract

We present a new hypergraph stochastic blockmodel and an associated inference procedure for model-based clustering of the nodes in simple hypergraphs. Simple hypergraphs, where a node may not appear several times in a same hyperedge, have been overlooked in the literature, though they appropriately model some high-order interactions (such as co-authorship). The model assumes latent groups for the nodes and conditional independence of the hyperedges given the latent groups. We establish the first proof of generic identifiability of the parameters in such a model. We develop a variational approximation Expectation-Maximization algorithm for parameter inference and node clustering, and derive an integrated classification likelihood criterion for model selection. We illustrate the performance of our algorithm on synthetic data and analyse a real dataset of co-authorship. Our method called `HyperSBM` is implemented in `C++` for efficiency and available as an `R` package at `https://github.com/LB1304/HyperSBM`.

Keywords: co-authorship network, higher-order interactions, hypergraph clustering, latent variable model, variational expectation-maximization

1 Introduction

Over the past two decades a broad variety of models has been developed for pairwise interactions, encoded in graphs. However, modern applications in various fields highlight the need to account for high-order interactions, to include the information deriving from groups of three or more nodes. Simple examples include triadic and larger groups interactions in social networks (whose importance has early been acknowledged in Simmel, 1950), scientific co-authorship (Estrada and Rodríguez-Velázquez, 2006), interactions between more than two species in ecological systems (Muyinda et al., 2020; Singh and Baruah, 2021) or high-order correlations between neurons in

brain networks (Chelaru et al., 2021). Hypergraphs provide the most general formalization of high-order interactions: similarly to a graph, a hypergraph is defined as a set of nodes and a set of hyperedges; each hyperedge is a subset of nodes taking part in an interaction. Here we distinguish these simple hypergraphs from *multisets-hypergraphs* where multiset hyperedges are allowed. A multiset is the generalisation of a set, where each element may appear with some multiplicity. Thus, multisets-hypergraphs occur when nodes may be repeated in a hyperedge. We refer to Battiston et al. (2020), Bick et al. (2021), and Torres et al. (2021) for recent reviews on high-order interactions.

Despite an increasing interest for these high-order interactions, the statistical literature on this topic remains scarce. Statistics such as centrality or clustering coefficient have been extended from graphs to hypergraphs (Estrada and Rodríguez-Velázquez, 2006). These statistics help understand the structure and extract information from the data but do not fill the need for random hypergraphs models. Some early analyses of hypergraphs rely on their embedding into the space of bipartite graphs (see, *e.g.*, Battiston et al., 2020). Indeed, hypergraphs with self-loops and multiple hyperedges (*i.e.* weighted hyperedges, with integer-valued weights) are equivalent to bipartite graphs. However, bipartite graphs models were not specifically designed for hypergraphs and may induce artefacts (we refer to Section A in the Supplementary Material for more details).

Generalising Erdös-Rényi's model of random graphs leads to uniformly random hypergraphs. The model consists in drawing uniformly at random from the set of all m-uniform hypergraphs (*i.e.* with hyperedges of fixed cardinality m) over a set of n nodes. However, similarly to Erdös-Rényi, the model is too simple and homogeneous to be used to statistically analyse datasets. The configuration model for random graphs draws uniformly at random from the set of all graphs over a set of n nodes with some prescribed degrees sequence. A first generalisation appears in Ghoshal et al. (2009) focusing on tripartite and 3-uniform hypergraphs, while Chodrow (2020) extends it to a more general hypergraphs setup. In these references, both the nodes degrees and the hyperedges sizes are kept fixed. The configuration model is useful to sample (hyper)-graphs with the same degree sequence (and same hyperedges sizes) as an observed one through shuffling algorithms, and thus is often used as a null model in a statistical perspective. However sampling exactly (and not approximately) from this model is challenging, in particular in the hypergraph case. We refer to Section 4 in Chodrow (2020) for a thorough discussion on this issue.

Another popular way of extracting information from heterogeneous data is clustering. In the graphs context, stochastic blockmodels (SBMs) were introduced in the early eighties (Frank and Harary, 1982; Holland et al., 1983) and have flourished in many directions. These models assume that nodes are clustered into groups and the connection probabilities between nodes are driven by their groups memberships. Variants handling weighted graphs and degree corrected versions have been developed among others. In the hypergraphs context, Ghoshdastidar and Dukkipati

(2014) introduce a planted partition model for m-uniform hypergraphs, which is a particular case of a SBM. They assume that nodes are clustered into equally-sized groups and two parameters determine intra-group and inter-groups connection probabilities, the former being larger than the latter. They develop a spectral partitioning method and establish its consistency. This result is extended to the non-uniform and weighted sparse (*i.e.* most weights are close to zero) setting in Ghoshdastidar and Dukkipati (2017). Introducing hypergraphons, Balasubramanian (2021) extends the hypergraph SBM ideas to a nonparametric setting. In a parallel vein, Turnbull et al. (2021) propose a latent space model for hypergraphs, by generalizing random geometric graphs to hypergraphs, though not designed to capture clustering. A proposal linked to SBM appears in Vazquez (2009), where nodes belong to latent groups and participate in a hyperedge with a probability that depends on their group and that hyperedge.

Modularity is also a widely used criterion for clustering entities in the context of interaction data. It is designed to obtain specific clusters, called communities, characterized by large intra-group and low inter-groups connection probabilities (exactly as in the above partition model from Ghoshdastidar and Dukkipati, 2014). In the hypergraph context, the definition of modularity is not unique. In particular, Kamiński et al. (2019) introduce a "strict" modularity criterion such that only hyperedges with all their nodes belonging to the same group contribute to an increase in the modularity. Their criterion measures a deviation of the number of these homogeneous hyperedges from a new null model: it constitutes a configuration-like model for hypergraphs where the average values of the degrees are kept fixed. Further in this direction, Chodrow et al. (2021) introduce a degree-corrected hypergraph SBM and propose two new modularity criteria. Similarly to Kamiński et al. (2019), one of these criteria relies on an "all-or-nothing" affinity function that only distinguishes whether a given hyperedge is contained entirely within a single cluster or not. In this setup, they establish a link between approximate maximum likelihood estimation and their modularity criterion. This echoes the work of Newman (2016) in the graph context. It is important to note that the developments in Kamiński et al. (2019) and Chodrow et al. (2021) are done in a multisets-hypergraphs context where hyperedges are multisets, *i.e.* nodes are allowed to appear with a certain multiplicity in each hyperedge. The multisets-hypergraphs setup simplifies some challenges raised by the computation of the modularity and to our knowledge, modularity approaches still lack instantiation in the simple hypergraph case. We further discuss this point in Section 2.2. Focusing on community detection, random walks approaches have also been used for hypergraph clustering (Swan and Zhan, 2021), as well as low-rank tensor decompositions (Ke et al., 2020). The misclassification rate for the community detection problem in hypergraphs and its limits have been analysed in various contexts (see for instance Ahn et al., 2018; Chien et al., 2019; Cole and Zhu, 2020). We mention that a recent approach has proposed to cluster hyperedges (Ng and Murphy, 2021) while our focus in this work is on nodes clustering.

The literature about high-order interactions often discusses simplicial complexes in parallel with hypergraphs (Battiston et al., 2020). However the peculiarity of these structures (namely

the fact that, stating it in the hypergraphs terminology, each subset of a hyperedge should also be a hyperedge) puts them out of the scope of this introduction.

In this paper, we focus on model-based clustering for simple hypergraphs and study stochastic hypergraphs blockmodels. We start by discussing the multisets-hypergraphs assumption, often presented as a harmless one in the literature, and highlight its consequences on datasets analysis (Section 2). These consequences motivate our focus on simple hypergraphs, where much less has been done, while computational challenges are higher. Then, we formulate a general stochastic simple hypergraphs blockmodel as well as different submodels and briefly highlight the main differences with previous proposals (Section 3.1). We provide the first result of generic identifiability of the parameters in a hypergraph stochastic blockmodel (Section 3.2). Parameter inference and node clustering are then performed through a variational Expectation-Maximization (`VEM`) algorithm (Section 3.3) and model selection on the number of groups relies on an integrated classification likelihood (ICL) criterion (Section 3.4). An illustration of the performance of our methods on synthetic datasets follows (Section 4) and a co-authorship dataset is analysed (Section 5). All the proofs of theoretical results are postponed to Section 6. An `R` package `HyperSBM` implementing the method (in `C++` for efficiency) and all the codes are available online (see Section 7). This manuscript comes with a Supplementary Material (SM) that contains additional information.

2 The need for simple hypergraphs models

In this section, we discuss modeling differences between multisets-hypergraphs where multiset hyperedges are allowed, versus simple hypergraphs where hyperedges are subsets of nodes. We recall that multisets-hypergraphs allow for repeated nodes in a same hyperedge, the latter being defined as a multiset of nodes. Multiset hyperedges generalize in some sense the notion of self-loops in graphs and thus are a natural extension to consider. However, they are not appropriate in all contexts. For instance, a co-authorship dataset cannot contain hyperedges with repeated nodes (but may contain a self-loop of a unique author). In the same way, a social interaction hypergraph does not contain multisets hyperedges; triadic interactions are restricted to 3 different individuals and self-loops are not allowed. In the meantime, they are natural in other contexts; consider, *e.g.*, chemical reaction hypergraphs where the multiplicity plays the role of the stoichiometric coefficients (Flamm et al., 2015). We first argue that multisets-hypergraph models are inappropriate for analysing simple hypergraphs.

2.1 A motivating example

For the sake of simplicity, we restrict our attention to 3-uniform hypergraphs on a set of n nodes and consider two different models. The first one, denoted as MH, acts on 3-uniform multisets-hypergraphs and draws a hyperedge between any 3 nodes, not necessarily distinct, with probability p_{MH}. The second one, denoted as SH, acts on 3-uniform simple hypergraphs and draws a hyperedge between any 3 distinct nodes with probability p_{SH}.

Now, we consider a toy example of observing a simple hypergraph $\mathcal{H}$ with $n = 3$ nodes and only one hyperedge $e = \{1, 2, 3\}$. This dataset could correspond to observing for instance one publication with 3 authors. When analysed under the MH model, the density of our observed hypergraph is estimated by

$$\hat{p}_{\mathrm{MH}} = 1/27$$

because there are $n^3 = 27$ possible size-3 multiset hyperedges under this model, and just one of these is observed. On the contrary, when analysed under the SH model, we infer a density of

$$\hat{p}_{\mathrm{SH}} = 1$$

because the only possible size-3 hyperedge is observed. As a consequence, the statistical conclusions drawn on this dataset will highly differ depending on whether we restrict attention to simple hypergraphs or work with more general multisets-hypergraphs. This choice of the ambient space has to be made according to the specificities of the dataset. This simple and elementary example shows that it is not possible to statistically analyse a simple hypergraph with a multisets-hypergraphs model without erroneous conclusions.

2.2 Computational challenge in the simple hypergraph case

The main technical difference between multisets-hypergraphs and simple hypergraphs analysis comes from the enumeration of m-tuples of nodes. In the multisets-hypergraphs setting, the summations over multisets of nodes $\{i_1, \ldots, i_m\} \in \{1, \ldots, n\}^m$ factorize into m independent sums. On the contrary, in the simple hypergraph setting, the summations involve sets of nodes $\{i_1, \ldots, i_m\}$ that are constrained to be distinct. As a consequence, such a factorization is impossible.

Let us consider a concrete example. We already emphasized the fact that modularity criteria for hypergraphs have been proposed only in the multisets-hypergraphs setting (Kamiński et al., 2019; Chodrow et al., 2021). Modularities are generally constructed as deviation measures of the number of hyperedges from their expected number under a null model. For instance in the graphs context, the Newman and Girvan modularity of a partition $(C_1, \ldots, C_Q)$ of the nodes into

Q clusters is computed as

$$\begin{aligned}\text{Modularity}(C_1,\dots,C_Q) &= \frac{1}{2|E|}\sum_{q=1}^{Q}\sum_{i,j\in C_q}\Big(A_{ij}-\frac{d_i d_j}{2|E|}\Big)\\ &= \frac{1}{2|E|}\sum_{q=1}^{Q}\sum_{i,j\in C_q}A_{ij}-\frac{1}{2|E|}\sum_{q=1}^{Q}\sum_{i,j\in C_q}\frac{d_i d_j}{2|E|},\end{aligned}$$

where $A=(A_{ij})_{i,j}$ is the graph adjacency matrix, d_i is the degree of node i, and $2|E|=\sum_i d_i$ is twice the number of edges. While the first part of these criteria enumerates only the occurring hyperedges, a quantity that is small in general as most hypergraph datasets are sparse, the second part needs to account for all tuples of nodes in the graph (or at least in the same group C_q). In the case of multisets-graphs this second term factorizes to

$$\frac{1}{2|E|}\sum_{q=1}^{Q}\sum_{i,j\in C_q}\frac{d_i d_j}{2|E|}=\frac{1}{2|E|}\sum_{q=1}^{Q}\frac{(\sum_{i\in C_q}d_i)(\sum_{j\in C_q}d_j)}{2|E|}=\sum_{q=1}^{Q}\frac{\text{Vol}(q)^2}{(2|E|)^2},$$

where the computation of the volume $\text{Vol}(q)=\sum_{i\in C_q}d_i$ has time complexity of $O(n)$. Similarly, for multisets-hypergraphs the modularity computed in Chodrow et al. (2021) uses two main terms: the first is a cut term that depends only on occurring hyperegdes while the second relies on volumes of latent configurations of the nodes (see Eq. (12) and (13) in Chodrow et al., 2021). On the contrary, in the simple hypergraph setting, enumerating all constrained tuples of nodes requires enumerating

$$\sum_{m=2}^{M}\binom{n}{m}$$

elements for a hypergraph with n nodes and maximum hyperedge size M. This quantity is huge and represents the main computational limit when analysing hypergraphs (our approach to this issue is detailed in Section C from SM).

3 A stochastic blockmodel for hypergraphs

3.1 Model formulation

Let $\mathcal{H}=(\mathcal{V},\mathcal{E})$ denote a binary hypergraph, where $\mathcal{V}=\{1,\dots,n\}$ is a set of n nodes and $\mathcal{E}$ is the set of hyperedges. A hyperedge of size $m\geq 2$ is a collection of m distinct nodes in $\mathcal{V}$; we do not allow for hyperedges being multisets nor self-loops. We indicate by $M=\max_{e\in\mathcal{E}}|e|$ the largest possible size of the hyperedges in $\mathcal{E}$ (so that $M\geq 2$, with $M=2$ for graphs). Let us denote by

$$\begin{aligned}\mathcal{V}^{(m)} &= \big\{\{i_1,\dots,i_m\}: i_1,\dots,i_m\in\mathcal{V} \text{ and } i_1\neq\ldots\neq i_m\big\},\\ \mathcal{E}^{(m)} &= \big\{\{i_1,\dots,i_m\}\in\mathcal{V}^{(m)}: \{i_1,\dots,i_m\}\in\mathcal{E}\big\},\end{aligned}$$

the sets of unordered node tuples and hyperedges of size m, respectively. Obviously $\mathcal{E} = \bigcup_{m=2}^{M} \mathcal{E}^{(m)} \subseteq \bigcup_{m=2}^{M} \mathcal{V}^{(m)}$. In particular, for each tuple $\{i_1, \dots, i_m\} \in \mathcal{V}^{(m)}$, we define the indicator variable

$$Y_{i_1,\dots,i_m} = \mathbf{1}_{\{i_1,\dots,i_m\}\in\mathcal{E}} = \begin{cases} 1 & \text{if } \{i_1, \dots, i_m\} \in \mathcal{E}, \\ 0 & \text{if } \{i_1, \dots, i_m\} \notin \mathcal{E}. \end{cases}$$

We let $\boldsymbol{Y} = (Y_{i_1,\dots,i_m})_{\{i_1,\dots,i_m\}\in\mathcal{V}^{(m)}}$ represent a random hypergraph.

Likewise the formulation of the SBM for graphs, we assume that the nodes belong to Q unobserved groups. We let $Z_1, \dots, Z_n$ denote n independent and identically distributed latent variables with prior distribution $\pi_q = \mathbb{P}(Z_i = q)$ for each $q = 1, \dots, Q$ and such that $\pi_q \geq 0$ and $\sum_{q=1}^{Q} \pi_q = 1$. With a slight abuse of notation, we sometimes write $Z_i = (Z_{i1}, \dots, Z_{iQ}) \in \{0, 1\}^Q$, with only one value Z_{iq} equal to 1. We also let $\boldsymbol{Z} = (Z_1, \dots, Z_n)$. Every m-tuple of nodes is associated with a latent configuration, simply defined as the set of latent groups these nodes belong to. We let

$$\mathcal{Q}^{(m)} = \big\{\{q_1, \dots, q_m\} : q_1, \dots, q_m \in \{1, \dots, Q\}\big\},$$

the set of all possible latent configurations of elements in $\mathcal{V}^{(m)}$. Note that groups values may be repeated. Now, conditional on the latent variables $\boldsymbol{Z}$, all indicator variables $Y_{i_1,\dots,i_m}$ are assumed independent and follow a Bernoulli distribution whose parameter depends on the latent configuration:

$$Y_{i_1,\dots,i_m} | \{Z_1 = q_1, \dots, Z_m = q_m\} \sim \mathcal{B}(B^{(m)}_{q_{i_1},\dots,q_{i_m}}) \qquad \text{for any } \{i_1, \dots, i_m\} \in \mathcal{V}^{(m)}.$$

Here $B^{(m)}_{q_1,\dots,q_m}$ denotes the probability that m unordered nodes with latent configuration $\{q_1, \dots, q_m\}$ are connected into a hyperedge. Note that each $B^{(m)}$ is a fully symmetric tensor of rank m, namely

$$B^{(m)}_{q_1,\dots,q_m} = B^{(m)}_{q_{\sigma(1)},\dots,q_{\sigma(m)}}, \ \forall q_1, \dots, q_m \text{ and } \forall \sigma \text{ permutation of } \{1, \dots, m\}. \tag{1}$$

We let $\theta = (\pi_q, B^{(m)}_{q_1,\dots,q_m})_{q,m,q_1\leq\dots\leq q_m}$ denote the parameter vector and $\mathbb{P}_\theta, \mathbb{E}_\theta$ the corresponding probability distribution and expectation, respectively.

Lemma 1. *The number of different parameters in each tensor* $B^{(m)} = (B^{(m)}_{q_1,\dots,q_m})_{q_1\leq\dots\leq q_m}$ *is* $\binom{Q+m-1}{m}$.

As a consequence, the total number of parameters of our hypergraph stochastic blockmodel (HSBM) is given by

$$(Q-1) + \sum_{m=2}^{M} \binom{Q+m-1}{m}.$$

As shown in Table 1, the number of parameters increases quite rapidly as the values of Q and M grow. To significantly reduce the model complexity, we introduce submodels by assuming equality of some conditional probabilities $B^{(m)}_{q_1,\dots,q_m}$. We mention that Chodrow et al. (2021) have

also defined submodels in the context of degree-corrected HSBM. In particular, we consider two "affiliation" submodels given by

$$B^{(m)}_{q_1,\dots,q_m} = \begin{cases} \alpha^{(m)} & \text{if } q_1 = \cdots = q_m, \\ \beta^{(m)} & \text{if there exist at least } q_i \neq q_j \text{ for } i \neq j \end{cases} \qquad \textbf{(Aff-m)}$$

and

$$B^{(m)}_{q_1,\dots,q_m} = \begin{cases} \alpha & \text{if } q_1 = \cdots = q_m \\ \beta & \text{if there exist at least } q_i \neq q_j \text{ for } i \neq j \end{cases} \quad \forall m = 2, \dots, M. \qquad \textbf{(Aff)}$$

The number of parameters is dropped to $(Q-1)+2(M-1)$ and to $(Q-1)+2$ under Assumptions (**Aff-m**) and (**Aff**), respectively. These submodels reflect the same ideas as in Kamiński et al. (2019) and Chodrow et al. (2021) when they consider that only hyperedges whose nodes all belong to the same group should increase the modularities.

	Q					
M	2	3	4	5	6	7
3	4	10	20	35	56	84
4	5	15	35	70	126	210
5	6	21	56	126	252	462
6	7	28	84	210	462	924
7	8	36	120	330	792	1716

Table 1: *Number of parameters of the full HSBM for given values of Q (number of latent groups) and M (largest possible hyperedge size).*

The choice of M. It is important to stress that when analysing a dataset, M is not necessarily the maximum observed value of the hyperedges sizes but rather a modelling choice. Indeed, take for example a co-authorship dataset with n authors and only 3 co-authors at most. If nothing prevents 4 persons to be co-authors, then the fact that there are no hyperedges of size 4 gives as much information as if all the possible size-4 hyperedges would be present. In the same way, the amount of information contained in a dataset where all but say 5 possible size-4 hyperedges are present is the same as the amount of information contained in the same dataset but with only 5 occurring size-4 hyperedges. In other words, occurring hyperedges and possible but non-occurring hyperedges carry as much information (0 and 1 values play a similar role). As a consequence, M should be chosen by the statistician, depending on the characteristics of the dataset at hand and on computational resources (see "Algorithm complexity" below for more on that point). One

should keep in mind that on any dataset, choosing $M > 2$ is already an improvement (in the sense of taking more information into account) with respect to a graph analysis of the data at hand.

Generalizations. Our model could allow for self-loops without any important changes (by authorizing $m = 1$). It could also be easily generalized to multiple hypergraphs (with or without self-loops) by putting a (zero-inflated or deflated) Poisson law on the conditional distribution of the hyperedges. More generally, the conditional Bernoulli distribution could be replaced by any parametric distribution to handle weighted hypergraphs. The case of multisets-hypergraphs could also be handled and would result in a fastest algorithm (though requiring a distinct implementation, which is not provided in our `R` package).

Differences with previous proposals. Clustering methods for hypergraphs SBMs have already been proposed in the literature and we highlight here what distinguishes our approach. The models considered in Ghoshdastidar and Dukkipati (2014, 2017) are restricted to our particular (**Aff-m**) model with equally-sized groups (and adding a sparsity parameter in the most recent reference). In the same way, the references Ke et al. (2020); Ahn et al. (2018); Chien et al. (2019) all focus on community detection and do not find clusters that are not communities. Then, Chodrow et al. (2021) introduced a very general degree-corrected SBM for multiple hypergraphs. However, their inference method solves the clustering problem in the space of multisets-hypergraphs while the one presented here solves it in the simple hypergraphs space. As already argued in Section 2.1, while both approaches are founded, they give rise to different statistical analyses. The choice of which should be used depends on the type of data at hand. Finally, the SBM for hypergraphs presented in Balasubramanian (2021) is very general; but his least-squares estimator of a hypergraphon model is untractable. Besides, Algorithm 1 in that reference is dedicated to community detection and does not recover general groups.

3.2 Parameter identifiability

In the next result, we first establish generic identifiability of the parameters of a HSBM restricted to simple m-uniform hypergraphs for any $m \geq 2$. Generic identifiability (in a parametric context) means that every parameter θ, except possibly for some lying in a subset whose dimension is strictly smaller than the dimension of the full parameter space, uniquely defines the distribution $\mathbb{P}_\theta$. In other words, when picking at random (*w.r.t.* Lebesgue measure) a parameter $\theta \in \Theta$, this parameter uniquely defines $\mathbb{P}_\theta$ almost surely (*w.r.t.* Lebesgue measure). Identifiability is established up to label switching on the node groups, as in any discrete latent variable model. The case $m = 2$ corresponds to Theorem 2 in Allman et al. (2011). Our proof follows the same ideas, building on a key result by Kruskal (1977), and relying in our case on a sufficient condition

for a sequence of nonnegative integers to be the degree sequence of a simple m-uniform hypergraph (Behrens et al., 2013).

Theorem 2. *For any $m \geq 2$ and any integer Q, the parameter $\theta^{(m)} = (\pi_q, B^{(m)}_{q_1,\dots,q_m})_{1\leq q\leq Q, 1\leq q_1\leq\dots\leq q_m\leq Q}$ of the HSBM restricted to m-uniform simple hypergraphs over n nodes, is generically identifiable, up to label switching on the node groups, for large enough n (depending only on m, Q).*

The case of fixed group proportions (*e.g.*, equal group proportions $\pi_q = 1/Q$) needs special attention. Indeed, our main result does not explicitly characterize the subspace of the parameter space on which identifiability may not be satisfied (we only know that its dimension is less than that of the full parameter space). When restricting to fixed group proportions, we are exactly on a lower dimensional space and may not obtain identifiability without specific care. In the same way, our result does not apply in the affiliation cases (**Aff-m**) and (**Aff**) that correspond to a restriction of the parameter space to a lower-dimensional subspace.

The result stated for m-uniform hypergraphs is enough to imply a similar one for non-uniform simple hypergraphs, as stated in the following corollary.

Corollary 3. *For any integer Q, the parameter $\theta = (\pi_q, B^{(m)}_{q_1,\dots,q_m})_{1\leq q\leq Q, 1\leq q_1\leq\dots\leq q_m\leq Q, 2\leq m\leq M}$ of the HSBM for simple hypergraphs over n nodes, is generically identifiable, up to label switching on the node groups, for large enough n (depending only on M, Q).*

Our proof of Corollary 3 specifically requires all the π_q's are distinct (a generic condition, thus not explicitly stated) and does not apply for instance in the restricted case of equal group proportions. In that case, it is not sufficient to identify the parameters for each value of m separately.

3.3 Parameter estimation via variational Expectation-Maximization

The likelihood of the model is given by

$$\begin{aligned}
\mathbb{P}_\theta(\boldsymbol{Y}) &= \sum_{q_1=1}^{Q}\cdots\sum_{q_n=1}^{Q}\mathbb{P}_\theta(\boldsymbol{Y}, \boldsymbol{Z}=(q_1,\dots,q_n)) \\
&= \sum_{q_1=1}^{Q}\cdots\sum_{q_n=1}^{Q}\left(\prod_{i=1}^{n}\mathbb{P}_\theta(Z_i=q_i)\right)\prod_{m=2}^{M}\prod_{\{i_1,\dots,i_m\}\in\mathcal{V}^{(m)}}\mathbb{P}_\theta(Y_{i_1,\dots,i_m}|Z_{i_1}=q_{i_1},\dots,Z_{i_m}=q_{i_m}) \\
&= \sum_{q_1=1}^{Q}\cdots\sum_{q_n=1}^{Q}\left(\prod_{i=1}^{n}\pi_{q_i}\right)\prod_{m=2}^{M}\prod_{\{i_1,\dots,i_m\}\in\mathcal{V}^{(m)}}(B^{(m)}_{q_{i_1},\dots,q_{i_m}})^{Y_{i_1,\dots,i_m}}(1-B^{(m)}_{q_{i_1},\dots,q_{i_m}})^{1-Y_{i_1,\dots,i_m}}.
\end{aligned} \tag{2}$$

The computation of the model likelihood is intractable in general. Indeed, Equation (2) involves a summation over all possible Q^n different latent configurations which cannot be done unless n and

Q are small. Latent variable models often rely on the Expectation-Maximization (`EM`) algorithm (Dempster et al., 1977) to solve this problem. Nonetheless, this approach can neither be applied in the context of SBMs. Indeed, the `E`-step computation of `EM` algorithm is typically based on the conditional posterior distribution of the latent variables $\mathbb{P}_\theta(\boldsymbol{Z}|\boldsymbol{Y})$, which is intractable itself in SBMs (see *e.g.* Matias and Robin, 2014). A possible remedy is to rely on variational approximations of `EM` algorithm (`VEM`, Jordan et al., 1999).

The complete data log-likelihood is

$$\begin{aligned}\ell_n^c(\theta) &= \log \mathbb{P}_\theta(\boldsymbol{Y}, \boldsymbol{Z}) = \log \mathbb{P}_\theta(\boldsymbol{Z}) + \log \mathbb{P}_\theta(\boldsymbol{Y}|\boldsymbol{Z}) \\ &= \sum_{q=1}^{Q}\sum_{i=1}^{n} Z_{iq} \log \pi_q \\ &\quad + \sum_{m=2}^{M}\sum_{\mathcal{V}^{(m)}}\sum_{\mathcal{Q}^{(m)}} Z_{i_1 q_1}\cdots Z_{i_m q_m}\left[Y_{i_1\ldots i_m}\log B^{(m)}_{q_1,\ldots,q_m} + (1-Y_{i_1\ldots i_m})\log(1-B^{(m)}_{q_1,\ldots,q_m})\right].\end{aligned} \tag{3}$$

The core idea at the basis of the variational method is to follow the same iterative two-steps structure as in `EM` algorithm and replace the intractable posterior distribution $\mathbb{P}_\theta(\boldsymbol{Z}|\boldsymbol{Y})$ by the best approximation (with respect to Kullback-Leibler divergence) in a class of simpler (often factorized) distributions. We thus introduce the class of factorized probability distributions $\mathbb{Q}_\tau$ over $\boldsymbol{Z} = (Z_1, \ldots, Z_n)$ given by

$$\mathbb{Q}_\tau(\boldsymbol{Z}) = \prod_{i=1}^{n} \mathbb{Q}_\tau(Z_i) = \prod_{i=1}^{n}\prod_{q=1}^{Q} \tau_{iq}^{Z_{iq}},$$

with the variational parameter $\tau_{iq} = \mathbb{Q}_\tau(Z_i = q) \in [0,1]$ and $\sum_{q=1}^{Q}\tau_{iq} = 1$, for any $i = 1, \ldots, n$ and $q = 1, \ldots, Q$. The expectation under distribution $\mathbb{Q}_\tau$ is $\mathbb{E}_{\mathbb{Q}_\tau}$ and $\mathcal{H}(\mathbb{Q}_\tau)$ denotes the entropy of $\mathbb{Q}_\tau$. Now we define the evidence lower bound (ELBO)

$$\begin{aligned}\mathcal{J}(\theta,\tau) &= \mathbb{E}_{\mathbb{Q}_\tau}[\log \mathbb{P}_\theta(\boldsymbol{Y}, \boldsymbol{Z})] + \mathcal{H}(\mathbb{Q}_\tau) \\ &= \mathbb{E}_{\mathbb{Q}_\tau}[\log \mathbb{P}_\theta(\boldsymbol{Y}, \boldsymbol{Z})] - \mathbb{E}_{\mathbb{Q}_\tau}[\log \mathbb{Q}_\tau(\boldsymbol{Z})] \\ &= \sum_{q=1}^{Q}\sum_{i=1}^{n} \tau_{iq}\log\frac{\pi_q}{\tau_{iq}} \\ &\quad + \sum_{m=2}^{M}\sum_{\mathcal{Q}^{(m)}}\sum_{\mathcal{V}^{(m)}} \tau_{i_1 q_1}\cdots\tau_{i_m q_m}\left[Y_{i_1,\ldots,i_m}\log B^{(m)}_{q_1,\ldots,q_m} + (1-Y_{i_1,\ldots,i_m})\log(1-B^{(m)}_{q_1,\ldots,q_m})\right].\end{aligned} \tag{4}$$

It is easy to see that $\mathcal{J}(\theta,\tau)$ satisfies

$$\mathcal{J}(\theta,\tau) = \log \mathbb{P}_\theta(\boldsymbol{Y}) - \mathrm{KL}(\mathbb{Q}_\tau(\boldsymbol{Z})||\mathbb{P}_\theta(\boldsymbol{Z}|\boldsymbol{Y})), \tag{5}$$

where $\mathrm{KL}(\cdot||\cdot)$ denotes the Kullback-Leibler divergence, and thus $\mathcal{J}$ is a lower bound of the model log-likelihood $\log \mathbb{P}_\theta(\boldsymbol{Y})$. Now, the `VEM` algorithm alternates the following two steps until a suitable convergence criterion is satisfied

- **VE-Step** maximizes $\mathcal{J}(\theta, \tau)$ with respect to τ

$$\widehat{\tau}^{(t)} = \arg\max_{\tau} \mathcal{J}(\theta^{(t-1)}, \tau); \quad \text{s.t. } \sum_{q=1}^{Q} \widehat{\tau}_{iq}^{(t)} = 1 \qquad \forall i = 1, \ldots, n \tag{6}$$

  this is equivalent to minimizing the Kullback-Leibler divergence term in (5), and thus finding the "best" approximation of the conditional distribution $\mathbb{P}_\theta(\boldsymbol{Z}|\boldsymbol{Y})$;

- **M-Step** maximizes $\mathcal{J}(\theta, \tau)$ with respect to θ

$$\widehat{\theta}^{(t)} = \arg\max_{\theta} \mathcal{J}(\theta, \tau^{(t-1)}), \quad \text{s.t. } \sum_{q=1}^{Q} \widehat{\pi}_q^{(t)} = 1 \tag{7}$$

  thus updating the value of the model parameters π_q and $B^{(m)}_{q_1,\ldots,q_m}$.

In the following we provide the solutions of the two maximization problems in Equations (6) and (7).

Proposition 4 (VE-Step). *Given the current model parameters $(\pi_q, B^{(m)}_{q_1,\ldots,q_m})_{q,m,q_1,\ldots,q_m}$ at any iteration of the VEM algorithm, the corresponding optimal values of the variational parameters $(\widehat{\tau}_{iq})_{i,q}$ defined in Equation (6) should satisfy the following fixed point equation*

$$\log \widehat{\tau}_{iq} = \log \pi_q + \sum_{m=1}^{M-1} \sum_{\{q_1,\ldots,q_m\} \in \mathcal{Q}^{(m)}} \sum_{\substack{\{i_1,\ldots,i_m\} \in \mathcal{V}^{(m)} \\ s.t. \{i, i_1, \ldots, i_m\} \in \mathcal{V}^{(m+1)}}} \widehat{\tau}_{i_1 q_1} \cdots \widehat{\tau}_{i_m q_m} \times \left[Y_{i i_1 \ldots i_m} \log(B^{(m+1)}_{q q_1 \ldots q_m}) + (1 - Y_{i i_1 \ldots i_m}) \log(1 - B^{(m+1)}_{q q_1 \ldots q_m}) \right] + c_i, \tag{8}$$

for any $1 \leq i \leq n$ and $1 \leq q \leq Q$ and where c_i are normalising constants such that $\sum_q \widehat{\tau}_{iq} = 1$.

Remark. From Proposition 4, the τ_i's are obtained using a fixed point algorithm. Although in all the situations we experienced, the algorithm converged in a reasonable number of iterations, we have no guarantee about existence nor uniqueness of a solution to (8).

Proposition 5 (M-Step). *Given the variational parameters $(\tau_{iq})_{i,q}$ at any iteration of the VEM algorithm, the corresponding optimal values of the model parameters $(\widehat{\pi}_q, \widehat{B}^{(m)}_{q_1 \ldots q_m})_{q,m,q_1,\ldots,q_m}$ defined in Equation (7) are given by*

$$\widehat{\pi}_q = \frac{1}{n} \sum_{i=1}^{n} \tau_{iq} \qquad \text{and} \qquad \widehat{B}^{(m)}_{q_1 \ldots q_m} = \frac{\sum_{\{i_1,\ldots,i_m\} \in \mathcal{V}^{(m)}} \tau_{i_1 q_1} \cdots \tau_{i_m q_m} Y_{i_1 \ldots i_m}}{\sum_{\{i_1,\ldots,i_m\} \in \mathcal{V}^{(m)}} \tau_{i_1 q_1} \cdots \tau_{i_m q_m}}.$$

We now express the solutions of the M-Step under the submodels given by (**Aff-m**) and (**Aff**). Note that the VE-Step is unchanged under these settings.

Proposition 6 (M-Step, affiliation setup). *In the particular affiliations submodels given by* (**Aff-m**) *and* (**Aff**) *respectively, given variational parameters $(\tau_{iq})_{i,q}$, at any iteration of the VEM algorithm, the corresponding optimal values of $(\widehat{\alpha}^{(m)}, \widehat{\beta}^{(m)})_m$ and $\widehat{\alpha}, \widehat{\beta}$ maximising $\mathcal{J}$ as in Equation (7) are now given by*

- *Under Assumption* (**Aff-m**),

$$\widehat{\alpha}^{(m)} = \frac{\sum_{q=1}^{Q} \sum_{\{i_1,\dots,i_m\}\in\mathcal{V}^{(m)}} \tau_{i_1 q} \dots \tau_{i_m q} Y_{i_1\dots i_m}}{\sum_{q=1}^{Q} \sum_{\{i_1,\dots,i_m\}\in\mathcal{V}^{(m)}} \tau_{i_1 q} \dots \tau_{i_m q}},$$

$$\widehat{\beta}^{(m)} = \frac{\sum_{\{q_1,\dots,q_m\}\in\mathcal{Q}^{(m)}} \sum_{\substack{\{i_1,\dots,i_m\}\in\mathcal{V}^{(m)} \\ |\{q_1,\dots,q_m\}|\geq 2}} \tau_{i_1 q_1} \dots \tau_{i_m q_m} Y_{i_1\dots i_m}}{\sum_{\substack{\{q_1,\dots,q_m\}\in\mathcal{Q}^{(m)} \\ |\{q_1,\dots,q_m\}|\geq 2}} \sum_{\{i_1,\dots,i_m\}\in\mathcal{V}^{(m)}} \tau_{i_1 q_1} \dots \tau_{i_m q_m}}.$$

- *Under Assumption* (**Aff**),

$$\widehat{\alpha} = \frac{\sum_{m=2}^{M} \sum_{q=1}^{Q} \sum_{\{i_1,\dots,i_m\}\in\mathcal{V}^{(m)}} \tau_{i_1 q} \dots \tau_{i_m q} Y_{i_1\dots i_m}}{\sum_{m=2}^{M} \sum_{q=1}^{Q} \sum_{\{i_1,\dots,i_m\}\in\mathcal{V}^{(m)}} \tau_{i_1 q} \dots \tau_{i_m q}},$$

$$\widehat{\beta} = \frac{\sum_{m=2}^{M} \sum_{\{q_1,\dots,q_m\}\in\mathcal{Q}^{(m)}} \sum_{\substack{\{i_1,\dots,i_m\}\in\mathcal{V}^{(m)} \\ |\{q_1,\dots,q_m\}|\geq 2}} \tau_{i_1 q_1} \dots \tau_{i_m q_m} Y_{i_1\dots i_m}}{\sum_{m=2}^{M} \sum_{\substack{\{q_1,\dots,q_m\}\in\mathcal{Q}^{(m)} \\ |\{q_1,\dots,q_m\}|\geq 2}} \sum_{\{i_1,\dots,i_m\}\in\mathcal{V}^{(m)}} \tau_{i_1 q_1} \dots \tau_{i_m q_m}}.$$

Algorithm initialization. We choose to start the algorithm with its `M`-step, hence providing an initial value for $\boldsymbol{\tau}$. This way, we may take advantage of smart initialization strategies based on a preliminary clustering of the nodes. Specifically, we rely on three different initialization strategies and keep the best result (that maximizes the criterion $\mathcal{J}$):

1. *random initialization*: This naive method simply draws each $(\tau_{iq})_{1\leq q\leq Q}$ uniformly in $(0,1)$ for every node i and normalise the vector τ_i;

2. *"soft" spectral clustering*: We rely on Algorithm 1 in Ghoshdastidar and Dukkipati (2017) combined with soft k-means. More precisely, we compute a hypergraph Laplacian and construct the column matrix X of its leading Q orthonormal eigenvectors. Then the rows of X are normalized to have unit norm (points 1 to 3 in Algorithm 1 from Ghoshdastidar and Dukkipati (2017)). We then perform a soft k-means algorithm on the rows of X and obtain τ_{iq} as the posterior probability for node i to belong to cluster q;

3. *graph-component absolute spectral clustering*: We restrict our attention to edges in the hypergraph ($m = 2$) and the corresponding adjacency matrix. We then perform the absolute spectral clustering (Rohe et al., 2011) on this adjacency matrix. This initialization does not use the whole information from the hypergraph (hyperedges of size $m \geq 3$ are not used). Nonetheless, absolute spectral clustering is believed to be superior to spectral clustering as it captures disassortative groups.

Note that in many cases, the random initialization gave as good results as the smart strategies.

Fixed point. The VE-Step is obtained by a fixed-point algorithm. In practice at iteration t of the VEM algorithm, starting from the previous values of the variational and the model parameters $\tau_{iq}^{(t-1)}, \theta^{(t-1)}$, respectively, we iterate over some index u the computation given by (8) and obtain a sequence of values $\tau_{iq}^{(t,u)}$. We stop these iterations whenever we reach a maximum number of fixed point iterations ($u > U_{\max}$) or the variational parameters converged ($\max_{iq} |\tau_{iq}^{(t,u-1)} - \tau_{iq}^{(t,u)}| \leq \varepsilon$).

Stopping criteria. The VEM algorithm iterations should stop whenever the ELBO $\mathcal{J}$ and the sequence of model parameter vectors $\theta^{(t)} = (\theta_s^{(t)})_s$ converged. However, experimenting with the above conditions, the algorithm sometimes stops when the VE-Step still requires a few iterations to reach a fixed point. In these cases, carrying on with the VEM iterations generally leads to higher values of the ELBO function, and hence to better estimates. Therefore we impose that the fixed point in the VE-Step is reached at its first iteration. This reduces the chance to converge to some local maxima of $\mathcal{J}$. Finally, when these conditions are not reached, we stop the algorithm if a maximum number of iterations has been reached. To summarize, we stop the algorithm whenever

$$\left\{ \frac{|\mathcal{J}(\theta^{(t-1)}) - \mathcal{J}(\theta^{(t)})|}{|\mathcal{J}(\theta^{(t)})|} \leq \varepsilon \quad \text{and} \quad \max_s |\theta_s^{(t-1)} - \theta_s^{(t)}| \leq \varepsilon \quad \text{and} \quad \max_{iq} |\tau_{iq}^{(t,0)} - \tau_{iq}^{(t,1)}| \leq \varepsilon \right\}$$
$$\text{or} \quad \{t > T_{\max}\}.$$

SM contains additional details about the algorithm's implementation (Section C) and the choice of some hyperparameters (Section D).

Algorithm complexity. The complexity of our algorithm is of the order $O(nQ^M \binom{n}{M})$, which is rather prohibitive for large datasets when M becomes large. We recall here that M must be chosen and is not necessarily the largest observed hyperedge size (see paragraph "The choice of M" above). Thus, for large datasets, we recommend limiting the analysis to $M = 3$ or 4.

3.4 Model selection

Ghoshdastidar and Dukkipati (2017) propose to select the number of groups by looking for the spectral gap. On the contrary here, we rely on a statistical approach to construct a model selection criterion.

We let $\hat{\theta}$ and $(\hat{\tau}_i)_i$ denote the estimated parameters obtained at the end of the VEM algorithm. We also let $\hat{Z}_i = \arg\max_q \hat{\tau}_{iq}$ denote the estimate of the group of node i. Then the integrated classification likelihood (ICL, Biernacki et al., 2000) is defined for the full model and the submodels

(**Aff-m**), (**Aff**) as

$$\mathrm{ICL}_{\text{full}}(q) = \log \mathbb{P}_{\hat{\theta}}(\boldsymbol{Y}, \hat{\boldsymbol{Z}}) - \frac{1}{2}(q-1)\log n - \frac{1}{2}\sum_{m=2}^{M}\binom{q+m-1}{m}\log\binom{n}{m},$$

$$\mathrm{ICL}_{\text{aff-m}}(q) = \log \mathbb{P}_{\hat{\theta}}(\boldsymbol{Y}, \hat{\boldsymbol{Z}}) - \frac{1}{2}(q-1)\log n - (M-1)\sum_{m=2}^{M}\log\binom{n}{m},$$

$$\mathrm{ICL}_{\text{aff}}(q) = \log \mathbb{P}_{\hat{\theta}}(\boldsymbol{Y}, \hat{\boldsymbol{Z}}) - \frac{1}{2}(q-1)\log n - \sum_{m=2}^{M}\log\binom{n}{m},$$

respectively. Then the number of groups is determined as $\hat{q} = \arg\max_q ICL(q)$.

4 Simulation results

4.1 Performance of parameter and groups estimation

We conduct a simulation study to assess the performance of `HyperSBM` package. In the following, we illustrate the simulation scheme and summarize the main results.

Hypergraphs are simulated from the HSBM, considering $Q = 2$ latent groups with prior probabilities equal to 0.6 and 0.4, respectively. The largest size M of hyperdeges is set to 3, and the number of nodes $n \in \{50, 100, 150, 200\}$. A simplified latent structure, according to the (**Aff**) submodel is assumed, and various scenarios, corresponding to different possible real-world situations, are analysed:

A. Communities: In this scenario, we focus on community detection and consider the case of high intra-group and low inter-groups connection probabilities. We thus set $\alpha = 0.7 > \beta = 0.3$;

B. Disassortative: In this scenario, we focus on disassortative behaviour and consider the case of low intra-group and high inter-groups connection probabilities. We thus set $\alpha = 0.3 < \beta = 0.7$;

C. Erdös-Rényi-like: In this scenario, we focus on the difficult case of very similar intra-group and inter-groups connection probabilities. We thus set $\alpha = 0.25$ very close to $\beta = 0.35$.

For each scenario and each value n of the number of nodes, 10 different datasets are simulated. We consider estimation under the full HSBM formulation with our `VEM` algorithm and rely on soft spectral clustering initialization only. The performance of `HyperSBM` is assessed in terms of both recovery of the correct clustering and estimation of the original parameters.

For the correct classification, the Adjusted Rand Index (ARI, Hubert and Arabie, 1985) is considered, measuring the similarity between the correct node clustering and the estimated one. This index is always smaller than or equal to 1 (two identical clusterings have an ARI exactly

equal to 1), and it can assume negative values when the agreement between the two clusterings is less than what is expected from a random result. Table 2 reports, for each setting, the average

n	Scenario A	Scenario B	Scenario C
50	1.00	1.00	0.50
100	1.00	1.00	0.90
150	1.00	1.00	1.00
200	1.00	1.00	1.00

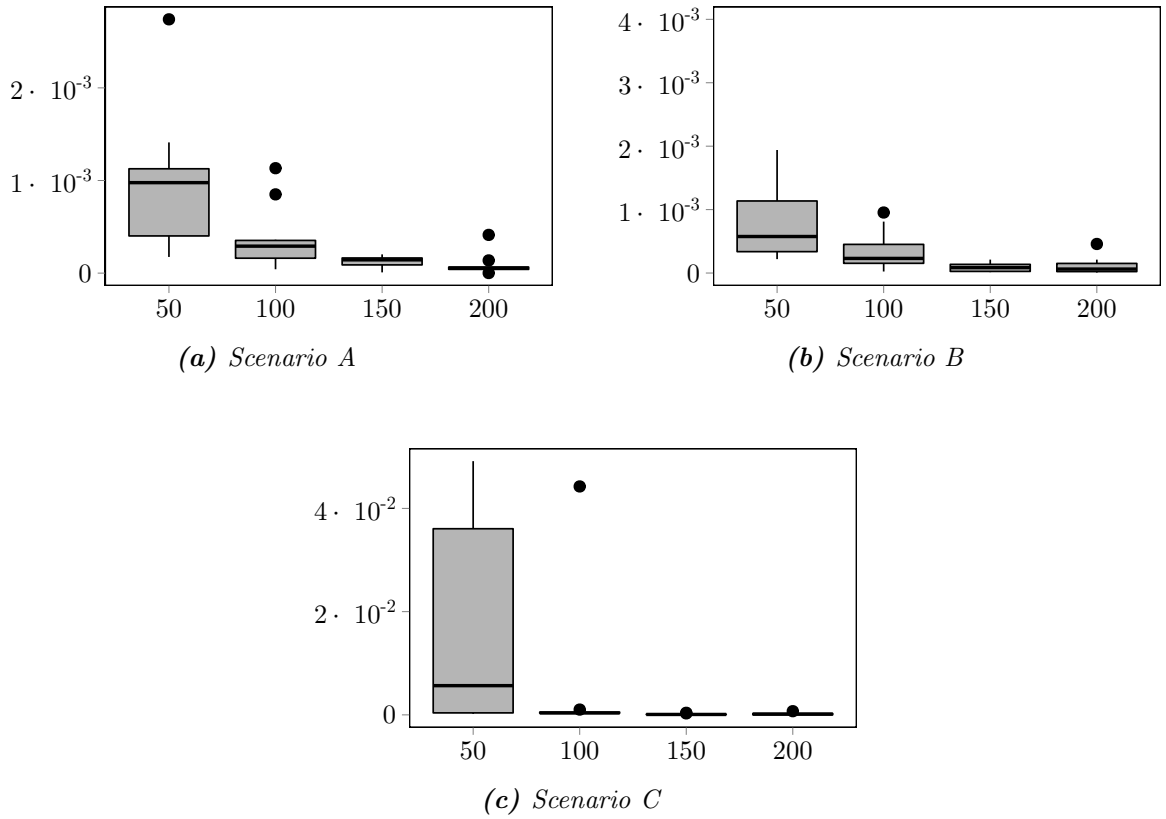
Table 2: *Adjusted Rand Index for different scenarios and number of nodes. Each value is obtained as the average over 10 simulated datasets.*

value of the ARI over the 10 simulated datasets. Considering scenarios A and B, the results are highly satisfactory, all values being equal to 1. The `VEM` algorithm perfectly recovers the correct clusters in all cases, hence showing an optimal performance in detecting communities as well as disassortative behaviours. Scenario C proves to be a more complex setting for clustering, especially when combined with a small number of nodes. Considering this setting, the proposed approach sometimes fails to recover the optimal clustering. This behavior is particularly evident in the case with $n = 50$ nodes, where the average ARI is rather low (0.5). In that scenario, the performances improve with the increase of the number of nodes. Note that such a behaviour could (partly) be explained by the results of Kim et al. (2018) about exact recovery. Indeed, in a similar setting, except for their assumption of equal group proportions and m-uniform hypergraphs, they establish that writing (with our notation) $\alpha = p \log n / \binom{n-1}{m-1}$ and $\beta = q \log n / \binom{n-1}{m-1}$ with p, q positive constants, then $I(p,q) = (\sqrt{p} - \sqrt{q})^2 / 2^{m-1}$ is the threshold for exact recovery of the latent groups; namely exact recovery is possible for $I(p,q) > 1$ and impossible for $I(p,q) < 1$. With our choice of parameters in scenario C and relying on $m = 3$, we have that $I(p,q) \simeq 0.63 < 1$ when $n = 50$ and $I(p,q) \simeq 2.21 > 1$ when $n = 100$. Again, our case is not exactly the same as we do not have equal group proportions and we observe hyperedges with both $m = 2$ and $m = 3$ but our results go in the same direction.

We also inspect the estimation of model parameters by computing the Mean Squared Error (MSE) between the true parameters and the estimated ones, for both the prior probabilities π_q, and the probabilities of hyperedge occurrence $B^{(m)}_{q_1,\dots,q_m}$. More specifically, we compute an aggregated MSE over all the components of θ, defined as

$$MSE = \frac{1}{10} \sum_{i=1}^{10} \Big\{ (\hat{\pi}_1^i - \pi_1)^2 + \sum_{m=2}^{M} \sum_{q_1,\dots,q_m} (\hat{B}^{(m),i}_{q_1,\dots,q_m} - B^{(m)}_{q_1,\dots,q_m})^2 \Big\},$$

where $(\hat{\pi}_1^i, \{\hat{B}^{(m),i}_{q_1,\dots,q_m}\}_{m,q_1,\dots,q_m})$ is the parameter estimated on the i-th dataset by the full model. The corresponding results are summarized through the boxplots in Figure 1. All values are rather



(a) *Scenario A*

(b) *Scenario B*

(c) *Scenario C*

Figure 1: *Mean Squared Error between true and estimated model parameters for different scenarios and number of nodes.*

small, showing that the model parameters are generally estimated with a high degree of accuracy. In particular, scenarios A and B provide the best results, with values of the MSE that are always lower than 0.5%. On the other hand, scenario C confirms to be the most difficult from the estimation perspective, showing the highest MSE for each value of n (up to 8%). This analysis also allows us to better outline the behavior of the `VEM` algorithm for different values of n; in particular, in each scenario, the parameters estimation becomes more accurate as the number of nodes increases. Estimates obtained assuming the submodel (**Aff**) formulation do not present any significant difference.

Section E from SM contains additional experiments showing that starting values do not have a strong influence on the result of the `VEM` algorithm.

4.2 Performance of model selection

In this section we assess the performance of ICL as a model selection criterion. To this aim we simulate 50 hypergraphs from the HSBM with $Q = 3$ latent states and assuming the submodel (**Aff**) for the latent structure. Two different values are tested for the number of nodes, $n = 100$ and $n = 200$, while the largest size M of hyperedges is set equal to 3 in both cases. The simulated data is then fitted with `HyperSBM` with a number of latent states ranging from 1 to 5.

	$n = 100$		$n = 200$	
Q	Percentage	ARI for 3 groups	Percentage	ARI for 3 groups
2	0%	-	2%	0.55
3	68%	1.00	90%	1.00
4	22%	0.57	6%	0.60
5	10%	0.58	2%	0.61

Table 3: *Frequency (as a percentage) of the selected number of groups and average Adjusted Rand Index of the classification obtained with $Q = 3$ depending on the selected number of groups. Model selection is carried out by means of the ICL criterion. Results are computed over 50 samples for each value of n.*

In Table 3 we show the frequency of the selected number of groups. Results are highly satisfactory: the correct model is selected in 68% of cases for $n = 100$ and in 90% of cases for $n = 200$. We also compute the value of ARI of the classification obtained with 3 clusters depending on the selected number of latent groups. This value is always equal to 1 when the correct model is recovered, thus confirming the optimal behavior of `HyperSBM` shown in Section 4.1. On the contrary, in cases where an incorrect number of groups is selected, values of ARI are quite low (around or smaller than 0.60). This behavior clarifies that in these cases, the estimation through

the `VEM` algorithm is responsible for the bad recovery more than the selection criterion. It is again confirmed that better results are obtained for higher values of n.

5 Analysis of a co-authorship dataset

5.1 Dataset description

We analyse a co-authorship dataset available at `http://vlado.fmf.uni-lj.si/pub/networks/data/2mode/Sandi/Sandi.htm`. The dataset was extracted from the bibliography of a book ("Product Graphs: Structure and recognition" by Imrich and Klavzăr) and is given as a bipartite author/article graph. Following Estrada and Rodríguez-Velázquez (2006), we constructed the hypergraph in which nodes are authors and hyperedges link the authors of a same paper. Details about pre-treatment of the dataset are given in Section F from SM, as well as further analyses. We chose $M = 4$ and worked with the induced main connected component of the hypergraph with 79 authors and 76 hyperedges (68.5% of which have size 2, while 29% have size 3 and 2.5% have size 4).

5.2 Analysis with `HyperSBM`

We performed an analysis of the constructed dataset with our `HyperSBM` package. The ICL criterion selected $Q = 2$ groups. We obtained a small group with only 8 authors (the remaining 71 authors being in the second group). Table 4 presents the distribution of the number of distinct co-authors per author. Among the 8 authors of the first group, 6 of them have the highest number of distinct co-authors (and the remaining 2 have 4 distinct co-authors each).

Nb co-authors	1	2	3	4	5	6	7	8	10	11	12
Count	23	27	13	6	2	2	1	1	2	1	1

Table 4: *Distribution of the number of distinct co-authors per author. The first group contains the 6 authors having the largest number of distinct co-authors (between 7 and 12) plus 2 authors with 4 co-authors each.*

Coming back to the bipartite graph of authors and (co-authored) papers, we looked at the degree distribution of the authors, given in Table 5. This corresponds to the distribution of the number of co-authored papers per author. We observed that 5 of the 8 authors from our first group are the ones that co-published the most, the three others having also high degree (one of degree 5 and two of degree 4). Thus, our first group is made of authors (among) the most collaborative ones, which are also (among) the most prolific ones.

Neither the first nor the second group inferred by `HyperSBM` are communities. Indeed we obtained the following estimated values from the size-2 hyperedges: $\hat{B}^{(2)}_{11} \simeq 4.2\%$ is of the same

Author degree	1	2	3	4	5	6	7	8	10	13
Count	44	14	6	6	4	1	1	1	1	1

Table 5: *Degree distribution of authors in the bipartite graph. Our first group contains the 5 most collaborating authors, one of the sixth, plus 2 authors with degree equal to 4.*

order as $\hat{B}^{(2)}_{12} \simeq 5.1\%$ while $\hat{B}^{(2)}_{22} \simeq 0.8\%$ is around five times smaller. This means that the first group contains authors that have written with authors from the two groups while the second group is made of authors who have less co-authored papers with people of their own group. Looking now at size-3 hyperedges, we get that $\hat{B}^{(3)}_{111} \simeq 2 \cdot 10^{-4}$; $\hat{B}^{(3)}_{112} \simeq 18 \cdot 10^{-4}$; $\hat{B}^{(3)}_{122} \simeq 7 \cdot 10^{-4}$ and $\hat{B}^{(3)}_{222} \simeq 0.6 \cdot 10^{-4}$. The most important estimated frequency is $\hat{B}^{(3)}_{112}$ that concerns 2 authors of the small first group co-authoring a paper with one author of the large second group. The second most important estimated frequency is $\hat{B}^{(3)}_{122}$ and is obtained for one author from small first group co-authoring a paper with two authors of the large second group. The remaining frequencies of size-3 hyperedges are negligible. This characterizes further the first groups as being composed by authors that do co-author with their own group as well as with authors from the second one.

Finally, looking now at size-4 hyperedges, the only non negligible estimated frequency is obtained for $\hat{B}^{(4)}_{1222} \simeq 4 \cdot 10^{-6}$. We note here that the quantities $B^{(3)}$'s and $B^{(4)}$'s are intrinsically on different scales, as are the quantities $B^{(2)}$'s and $B^{(3)}$'s. So again, authors from group one co-authored with the others authors. (Note that the first group is not large enough for a $B^{(4)}$ frequency with at least 2 authors in that group 1 to be non negligible).

5.3 Comparison with 2 other methods

We first compared our approach with the spectral clustering algorithm proposed in Ghoshdastidar and Dukkipati (2017). Let us recall that spectral clustering does not come with a statistical criterion to select the number of groups.

Looking at the partition obtained with $Q = 2$ groups, spectral clustering output groups with sizes 24 and 55, respectively. These groups are neither characterized by the number of co-authors nor their degrees in the bipartite graph (see details in SM). Indeed, in our case the best clusters are not communities and their sizes are very different, while we recall that spectral clustering tends to: i) extract communities ; ii) favor groups of similar size.

We then analysed the same dataset as a bipartite graph of authors/papers with the `R` package `SBM` through the function `estimateBipartiteSBM` (Chiquet et al., 2022). This method infers a latent blockmodel (that in fact corresponds to a SBM for bipartite graphs) and automatically selects a number of groups on both parts (authors and papers). Hereafter, we refer to this method as the `Bipartite-SBM` implementation. Let us underline here that while the bipartite

stochastic blockmodel can be written as a particular case of a HSBM, the converse is not true (See Section A.3 in the SM).

The `Bipartite-SBM` also selected 2 groups of authors (and one group of papers). There was one small group with 4 authors, entirely contained in our first small group; it corresponds to authors that have the highest degree in the bipartite graph and the highest number of co-authors. So, `Bipartite-SBM` output a very small group of the most prolific and the most collaborative authors in this dataset. Further details about the distinctions between these groups and the ones obtained by `HyperSBM` are given in SM.

As a conclusion, we see that while the outputs of `Bipartite-SBM` and `HyperSBM` may seem close on this specific dataset, they are nonetheless different. On the other hand, and still on this specific dataset, the spectral clustering approach outputs results that are completely different from those of `HyperSBM`.

6 Proofs

Proof of Lemma 1. We consider a fixed value of $m \geq 2$ and denote by $[\![a, b]\!]$ the set of integer values between a, b. Let us recall that $B^{(m)}$ is a fully symmetric tensor (1), so the number of free parameters in $B^{(m)}$ is equal to the number of ordered sequences $q_1 \leq \cdots \leq q_m$ of elements in $[\![1, Q]\!]$. We denote by $\mathcal{Q}^+$ this set. Then we define a function f which, to any such sequence $\underline{q} = (q_1, \ldots, q_m)$, associates the value $\underline{l} = f(\underline{q})$ defined by $f(\underline{q}) = (q_1, q_2+1, q_3+2, \ldots, q_m+m-1)$. We let $\mathcal{L}^+$ denote the set of sequences $\underline{l} = (l_1, \ldots, l_m)$ with coordinates in $[\![1, Q+m-1]\!]$ and such that $l_1 < l_2 < \cdots < l_m$. Thus, for any $\underline{q} \in \mathcal{Q}^+$ we get that $f(\underline{q}) \in \mathcal{L}^+$.

Conversely, for any $\underline{l} = (l_1, \ldots, l_m) \in \mathcal{L}^+$, we can associate the value $\underline{q} = g(\underline{l}) = (l_1, l_2-1, l_3-2, \ldots, l_m-m+1)$. It is easy to see that the image $\underline{q} = g(\underline{l})$ belongs to $\mathcal{Q}^+$.

As a consequence, the functions f and g are such that their composition is the identity function: $f \circ g = g \circ f = Id$. These are one-to-one functions mapping $\mathcal{Q}^+$ to $\mathcal{L}^+$ and conversely. This implies that the cardinalities of these two sets are equal. But an element in $\mathcal{L}^+$ is exactly a subset of size m of $[\![1, Q+m-1]\!]$ so that the cardinality of $\mathcal{L}^+$ is the number of subsets of size m of $[\![1, Q+m-1]\!]$. This concludes the proof of the lemma. □

Proof of Theorem 2. The proof mostly follows the same lines as the proof of Theorem 2 in Allman et al. (2011) for simple graphs SBM, generalizing it to simple m-uniform HSBM. The full details of this generalization are given in Section B from SM. In this section, we just establish the key element that strongly differs from the proof of Theorem 2 in Allman et al. (2011). It consists in exhibiting a set of degree sequences with some specific properties. As the characterization of which integer sequences may be realized as degree sequences strongly differs between graphs and m-uniform hypergraphs, our construction differs from the one in the proof of Theorem 2 in Allman et al. (2011).

Consider the following set of integer-valued sequences

$$\mathcal{D} = \Big\{ \mathbf{d} = (d_1, \dots, d_{n_0}) \mid \text{ for } 1 \leq i \leq n_0, d_i \in \{m, 2m, 3m, \dots, Qm\} \Big\}.$$

Lemma 7. *The set $\mathcal{D}$ of n_0-length integer sequences satisfies*

(i) for each $i \in \{1, \dots, n_0\}$, the set of i-th coordinates $\{d_i \mid \mathbf{d} \in \mathcal{D}\}$ has cardinality at most Q;

(ii) For large enough n_0 (depending on Q, m), any $\mathbf{d} \in \mathcal{D}$ is the degree sequence of a simple m-uniform hypergraph over n_0 nodes;

(iii) $|\mathcal{D}| \geq Q^{n_0}$.

Note that conditions $(i), (iii)$ imply that $\{d_i \mid \mathbf{d} \in \mathcal{D}\}$ should have cardinality exactly Q and that $|\mathcal{D}| = Q^{n_0}$.

Proof of Lemma 7. Points $(i), (iii)$ are a consequence of the definition of $\mathcal{D}$. For any integer sequence $\mathbf{d}$, a necessary condition for $\mathbf{d}$ to be a degree sequence of a simple m-uniform hypergraph over n_0 nodes is that m divides $\sum_i d_i$. Here, we rather need sufficient conditions in order to prove (ii). We rely on Corollary 2.2 in Behrens et al. (2013).

Corollary 2.2 in Behrens et al. (2013). *Let $\mathbf{d}$ be an integer-valued sequence with maximum term Δ and let p be an integer such that $\Delta \leq \binom{p-1}{m-1}$. If m divides $\sum_i d_i$ and $\sum_i d_i \geq (\Delta - 1)p + 1$ then $\mathbf{d}$ is the degree sequence of a simple m-uniform hypergraph.*

Fix some $\mathbf{d} \in \mathcal{D}$. Note that by construction, m divides $\sum_i d_i$. Let Δ be the maximum value of this sequence and note that $\Delta \leq Qm$. Thus we choose p an integer such that $Qm \leq \binom{p-1}{m-1}$. Moreover, $\sum_i d_i \geq mn_0$ and $(\Delta - 1)p + 1 \leq \Delta p \leq Qmp$. Then by choosing $n_0 \geq Qp$, we obtain the desired result. □

With Lemma 7 at hand, we are able to generalize the proof of Theorem 2 in Allman et al. (2011) for simple graphs SBM to the case of simple m-uniform HSBM. This concludes the proof of Theorem 2. □

Proof of Corollary 3. From the probability distribution $\mathbb{P}_\theta$ over simple hypergraphs $\mathcal{H}$ on a set of n nodes and hyperedges with largest size M, we automatically obtain all the probability distributions $\mathbb{P}_\theta$ restricted to simple m-uniform hypergraphs $\mathcal{H}_m$ on the same set of nodes. Applying the result of Theorem 2 for all values m is sufficient to obtain the desired result. Indeed, as M is finite, the union of the finite number of lower-dimensional subspaces where identifiability for fixed m may not be satisfied gives a lower-dimensional subspace, ensuring generic identifiability. Moreover, for each value of m, we recover the parameter $\theta^{(m)}$ up to a permutation on $\{1, \dots, Q\}$. Now, for any $m \neq m'$ it remains to be able to jointly order the parameters $\theta^{(m)}$ and $\theta^{(m')}$ up to a permutation on $\{1, \dots, Q\}$. If all the π_q's are different, which is a generic condition, this can be easily done because $\theta^{(m)}$ and $\theta^{(m')}$ share the same distinct π_q's. □

Proof of Proposition 4. We want to maximize $\mathcal{J}(\theta,\tau)$ with respect to τ_{iq} under the constraint $\sum_{q=1}^{Q}\tau_{iq}=1$ for all i. Using the method of Lagrange multipliers, this is equivalent to maximizing with respect to τ_{iq} the Lagrangian function

$$\begin{aligned}\Lambda(\theta,\tau,\lambda) &= \sum_{i=1}^{n}\lambda_i\left(\sum_{q=1}^{Q}\tau_{iq}-1\right)+\mathcal{J}(\theta,\tau)\\ &= \sum_{i=1}^{n}\lambda_i\left(\sum_{q=1}^{Q}\tau_{iq}-1\right)+\sum_{q=1}^{Q}\sum_{i=1}^{n}\tau_{iq}\log\frac{\pi_q}{\tau_{iq}}\\ &\quad+\sum_{i=1}^{n}\sum_{q=1}^{Q}\sum_{m=1}^{M-1}\sum_{\{q_1,\dots,q_m\}\in\mathcal{Q}^{(m)}}\sum_{\mathcal{V}^{(m)}\not\ni i}\tau_{iq}\tau_{i_1q_1}\cdots\tau_{i_mq_m}\Big[Y_{ii_1\dots i_m}\log B^{(m)}_{qq_1\dots q_m}\\ &\qquad+(1-Y_{ii_1\dots i_m})\log(1-B^{(m)}_{qq_1\dots q_m})\Big].\end{aligned}$$

Computing the partial derivative of $\Lambda(\boldsymbol{\theta},\boldsymbol{\tau},\boldsymbol{\lambda})$ with respect to τ_{iq}, we obtain the following expression

$$\begin{aligned}\frac{\partial\Lambda}{\partial\tau_{iq}} &= \lambda_i+\log\frac{\pi_q}{\tau_{iq}}-1\\ &\quad+\sum_{m=1}^{M-1}\sum_{\mathcal{Q}^{(m)}}\sum_{\mathcal{V}^{(m)}\not\ni i}\tau_{i_1q_1}\cdots\tau_{i_mq_m}\left[Y_{ii_1\dots i_m}\log B^{(m)}_{qq_1\dots q_m}+(1-Y_{ii_1\dots i_m})\log(1-B^{(m)}_{qq_1\dots q_m})\right]\\ &= \lambda_i+\log\pi_q-\log\tau_{iq}-1\\ &\quad+\log\prod_{m=1}^{M-1}\prod_{\mathcal{Q}^{(m)}}\prod_{\mathcal{V}^{(m)}\not\ni i}\left[(B^{(m)}_{qq_1\dots q_m})^{Y_{ii_1\dots i_m}}\cdot(1-B^{(m)}_{qq_1\dots q_m})^{1-Y_{ii_1\dots i_m}}\right]^{\tau_{i_1q_1}\cdots\tau_{i_mq_m}},\end{aligned}$$

which is equal to 0 if

$$\tau_{iq}=e^{\lambda_i-1}\,\pi_q\prod_{m=1}^{M-1}\prod_{\mathcal{Q}^{(m)}}\prod_{\mathcal{V}^{(m)}\not\ni i}\left[(B^{(m)}_{qq_1\dots q_m})^{Y_{ii_1\dots i_m}}\cdot(1-B^{(m)}_{qq_1\dots q_m})^{1-Y_{ii_1\dots i_m}}\right]^{\tau_{i_1q_1}\cdots\tau_{i_mq_m}}.$$

The term $e^{\lambda_i-1}=\frac{1}{\sum_{q=1}^{Q}\tau_{iq}}$ is the normalizing constant such that $\sum_{q=1}^{Q}\tau_{iq}=1$ for each i.
Finally, let us remark that the Lagrangian function Λ is concave with respect to each τ_{iq}, being the sum of a concave term ($\tau_{iq}\log(\pi_q/\tau_{iq})$) and linear terms. Then the critical point is a maximum. □

Proof of Proposition 5. For the prior probabilities π_q, we want to maximize $\mathcal{J}(\theta,\tau)$ with respect to π_q subject to the constraint $\sum_{q=1}^{Q}\pi_q=1$. Using again Lagrange multipliers, this is equivalent to maximizing

$$\Lambda(\theta,\tau,\lambda)=\lambda\left(\sum_{q=1}^{Q}\pi_q-1\right)+\mathcal{J}(\theta,\tau)$$

Noting that the second term of $\mathcal{J}(\theta,\tau)$ does not depend on π_q, the computation of the partial derivative of $\Lambda(\theta,\tau,\lambda)$ reduces to

$$\frac{\partial}{\partial\pi_q}\left[\lambda\Big(\sum_{q=1}^{Q}\pi_q-1\Big)+\sum_{q=1}^{Q}\sum_{i=1}^{n}\tau_{iq}\log\frac{\pi_q}{\tau_{iq}}\right]=\lambda+\sum_{i=1}^{n}\frac{\tau_{iq}}{\pi_q}.$$

This quantity is equal to 0 if

$$\pi_q=-\frac{1}{\lambda}\sum_{i=1}^{n}\tau_{iq},$$

where $\lambda=-n$ is the normalizing constant in order to satisfy $\sum_{q=1}^{Q}\pi_q=1$.
Note the Lagrangian function Λ is concave with respect to each π_q, being the sum of a concave term ($\log(\pi_q/\tau_{iq})$), of a linear term ($\lambda\sum_{q=1}^{Q}\pi_q$) and of a constant. The critical point is then a maximum.

Finally, the partial derivative *w.r.t.* $B^{(m)}_{q_1,\dots,q_m}$ is

$$\frac{\partial\mathcal{J}}{\partial B^{(m)}_{q_1,\dots,q_m}}=\sum_{\mathcal{V}^{(m)}}\tau_{i_1q_1}\cdots\tau_{i_mq_m}\left[Y_{i_1\dots i_m}\frac{1}{B^{(m)}_{q_1\dots q_m}}-(1-Y_{i_1\dots i_m})\frac{1}{1-B^{(m)}_{q_1\dots q_m}}\right].$$

Through some basic algebraic manipulations, this quantity results equal to 0 if

$$B^{(m)}_{q_1,\dots,q_m}=\frac{\sum_{\mathcal{V}^{(m)}}\tau_{i_1q_1}\cdots\tau_{i_mq_m}Y_{i_1\dots i_m}}{\sum_{\mathcal{V}^{(m)}}\tau_{i_1q_1}\cdots\tau_{i_mq_m}}.$$

Again, the Lagrangian function is the sum of a concave term ($\log B^{(m)}_{q_1,\dots,q_m}$) and of some constant terms, thus being a concave function. The critical point is then a maximum. □

Proof of Proposition 6. Let us define the following two subsets of $\mathcal{Q}^{(m)}$:

$$\mathcal{Q}^{(m,1)}=\big\{\{q_1,\dots,q_m\}\in\mathcal{Q}^{(m)}:q_1=\cdots=q_m\big\}\subset\mathcal{Q}^{(m)},$$
$$\mathcal{Q}^{(m,2)}=\big\{\{q_1,\dots,q_m\}\in\mathcal{Q}^{(m)}:|\{q_1,\dots,q_m\}|\geq 2\big\}\subset\mathcal{Q}^{(m)}.$$

It is straightforward to prove that $\mathcal{Q}^{(m,1)}\sqcup\mathcal{Q}^{(m,2)}=\mathcal{Q}^{(m)}$ (here $\sqcup$ denotes the disjoint union). Moreover, note that the summation $\sum_{\mathcal{Q}^{(m,1)}}$ is equivalent to $\sum_{q=1}^{Q}$. Then the following decomposition of $\mathcal{J}(\boldsymbol{\theta},\tau)$ naturally holds:

$$\begin{aligned}\mathcal{J}(\theta,\tau)=&\sum_{q=1}^{Q}\sum_{i=1}^{n}\tau_{iq}\log\frac{\pi_q}{\tau_{iq}}\\&+\sum_{m=2}^{M}\sum_{q=1}^{Q}\sum_{\mathcal{V}^{(m)}}\tau_{i_1q}\cdots\tau_{i_mq}\Big[Y_{i_1,\dots,i_m}\log\alpha^{(m)}+(1-Y_{i_1,\dots,i_m})\log(1-\alpha^{(m)})\Big]\\&+\sum_{m=2}^{M}\sum_{\mathcal{Q}^{(m,2)}}\sum_{\mathcal{V}^{(m)}}\tau_{i_1q_1}\cdots\tau_{i_mq_m}\Big[Y_{i_1,\dots,i_m}\log\beta^{(m)}+(1-Y_{i_1,\dots,i_m})\log(1-\beta^{(m)})\Big].\end{aligned}$$

The partial derivative *w.r.t.* $\alpha^{(m)}$ is

$$\frac{\partial \mathcal{J}}{\partial \alpha^{(m)}} = \sum_{q=1}^{Q} \sum_{\mathcal{V}^{(m)}} \tau_{i_1 q} \cdots \tau_{i_m q} \left[Y_{i_1 \dots i_m} \frac{1}{\alpha^{(m)}} - (1 - Y_{i_1 \dots i_m}) \frac{1}{1 - \alpha^{(m)}} \right],$$

hence it follows that:

$$\widehat{\alpha}^{(m)} = \frac{\sum_{q=1}^{Q} \sum_{\{i_1,\dots,i_m\} \in \mathcal{V}^{(m)}} \tau_{i_1 q} \cdots \tau_{i_m q} Y_{i_1 \dots i_m}}{\sum_{q=1}^{Q} \sum_{\{i_1,\dots,i_m\} \in \mathcal{V}^{(m)}} \tau_{i_1 q} \cdots \tau_{i_m q}}.$$

Analogously, the partial derivative *w.r.t.* $\beta^{(m)}$ is

$$\frac{\partial \mathcal{J}}{\partial \beta^{(m)}} = \sum_{\mathcal{Q}^{(m,2)}} \sum_{\mathcal{V}^{(m)}} \tau_{i_1 q_1} \cdots \tau_{i_m q_m} \left[Y_{i_1 \dots i_m} \frac{1}{\beta^{(m)}} - (1 - Y_{i_1 \dots i_m}) \frac{1}{1 - \beta^{(m)}} \right],$$

and

$$\widehat{\beta}^{(m)} = \frac{\sum_{\{q_1,\dots,q_m\} \in \mathcal{Q}^{(m,2)}} \sum_{\{i_1,\dots,i_m\} \in \mathcal{V}^{(m)}} \tau_{i_1 q_1} \cdots \tau_{i_m q_m} Y_{i_1 \dots i_m}}{\sum_{\{q_1,\dots,q_m\} \in \mathcal{Q}^{(m,2)}} \sum_{\{i_1,\dots,i_m\} \in \mathcal{V}^{(m)}} \tau_{i_1 q_1} \cdots \tau_{i_m q_m}}.$$

This concludes the proof for the formulas under assumption (**Aff-m**). The expressions for $\hat{\alpha}$ and $\hat{\beta}$ under assumption (**Aff**) are computed in the same way. □

7 Codes availability

The algorithm implementation in `C++` is available as an `R` package called `HyperSBM` at `https://github.com/LB1304/HyperSBM`. The Supplementary Material, the files to reproduce the synthetic experiments and the dataset analysis are available at `https://github.com/LB1304/Hypergraph-Stochastic-Blockmodel`.

Acknowledgements

Funding was provided by the French National Research Agency (ANR) grant ANR-18-CE02-0010-01 EcoNet.

References

Ahn, K., K. Lee, and C. Suh (2018). Hypergraph spectral clustering in the weighted stochastic block model. *IEEE Journal of Selected Topics in Signal Processing 12*(5), 959–974.

Allman, E., C. Matias, and J. Rhodes (2011). Parameters identifiability in a class of random graph mixture models. *Journal of Statistical Planning and Inference 141*, 1719–1736.

Balasubramanian, K. (2021). Nonparametric modeling of higher-order interactions via hypergraphons. *Journal of Machine Learning Research 22*(146), 1–35.

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

Behrens, S., C. Erbes, M. Ferrara, S. G. Hartke, B. Reiniger, H. Spinoza, and C. Tomlinson (2013). New results on degree sequences of uniform hypergraphs. *Electron. J. Comb. 20*(4), research paper p14, 18.

Bick, C., E. Gross, H. A. Harrington, and M. T. Schaub (2021). What are higher-order networks? Technical report, arXiv:2104.11329.

Biernacki, C., G. Celeux, and G. Govaert (2000). Assessing a mixture model for clustering with the Integrated Completed Likelihood. *IEEE Trans. Pattern Anal. Machine Intel. 22*(7), 719–725.

Chelaru, M. I., S. Eagleman, A. R. Andrei, R. Milton, N. Kharas, and V. Dragoi (2021). High-order correlations explain the collective behavior of cortical populations in executive, but not sensory areas. *Neuron 109*(24), 3954–3961.

Chien, I. E., C.-Y. Lin, and I.-H. Wang (2019). On the minimax misclassification ratio of hypergraph community detection. *IEEE Transactions on Information Theory 65*(12), 8095–8118.

Chiquet, J., S. Donnet, großBM team, and P. Barbillon (2022). `sbm`*: Stochastic Blockmodels*. R package version 0.4.4.

Chodrow, P. S. (2020). Configuration models of random hypergraphs. *Journal of Complex Networks 8*(3), cnaa018.

Chodrow, P. S., N. Veldt, and A. R. Benson (2021). Generative hypergraph clustering: From blockmodels to modularity. *Science Advances 7*(28), eabh1303.

Cole, S. and Y. Zhu (2020). Exact recovery in the hypergraph stochastic block model: A spectral algorithm. *Linear Algebra and its Applications 593*, 45–73.

Dempster, A., N. Laird, and D. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society, Series B 39*, 1–38.

Estrada, E. and J. A. Rodríguez-Velázquez (2006). Subgraph centrality and clustering in complex hyper-networks. *Physica A: Statistical Mechanics and its Applications 364*, 581–594.

Flamm, C., B. M. Stadler, and P. F. Stadler (2015). Chapter 13 - generalized topologies: Hypergraphs, chemical reactions, and biological evolution. In S. C. Basak, G. Restrepo, and J. L. Villaveces (Eds.), *Advances in Mathematical Chemistry and Applications*, pp. 300–328. Bentham Science Publishers.

Frank, O. and F. Harary (1982). Cluster inference by using transitivity indices in empirical graphs. *J. Amer. Statist. Assoc. 77*(380), 835–840.

Ghoshal, G., V. Zlatić, G. Caldarelli, and M. E. J. Newman (2009). Random hypergraphs and their applications. *Phys. Rev. E 79*, 066118.

Ghoshdastidar, D. and A. Dukkipati (2014). Consistency of spectral partitioning of uniform hypergraphs under planted partition model. In *Advances in Neural Information Processing Systems*, Volume 27.

Ghoshdastidar, D. and A. Dukkipati (2017). Consistency of spectral hypergraph partitioning under planted partition model. *The Annals of Statistics 45*(1), 289 – 315.

Holland, P., K. Laskey, and S. Leinhardt (1983). Stochastic blockmodels: some first steps. *Social networks 5*, 109–137.

Hubert, L. and P. Arabie (1985). Comparing partitions. *J. Classif. 2*, 193–218.

Jordan, M. I., Z. Ghahramani, T. S. Jaakkola, and L. K. Saul (1999). An introduction to variational methods for graphical models. *Machine Learning 37*(2), 183–233.

Kamiński, B., V. Poulin, P. Prałat, P. Szufel, and F. Théberge (2019). Clustering via hypergraph modularity. *PLoS ONE 14*(11), e0224307.

Ke, Z. T., F. Shi, and D. Xia (2020). Community detection for hypergraph networks via regularized tensor power iteration. Technical report, arXiv:1909.06503.

Kim, C., A. S. Bandeira, and M. X. Goemans (2018). Stochastic block model for hypergraphs: Statistical limits and a semidefinite programming approach. Technical report, arXiv:1807.02884.

Kruskal, J. (1977). Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra and Appl. 18*(2), 95–138.

Matias, C. and S. Robin (2014). Modeling heterogeneity in random graphs through latent space models: a selective review. *Esaim Proc. & Surveys 47*, 55–74.

Muyinda, N., B. De Baets, and S. Rao (2020). Non-king elimination, intransitive triad interactions, and species coexistence in ecological competition networks. *Theor Ecol 13*, 385–397.

Newman, M. E. J. (2016). Equivalence between modularity optimization and maximum likelihood methods for community detection. *Phys. Rev. E 94*, 052315.

Ng, T. and T. Murphy (2021). Model-based clustering for random hypergraphs. *Adv Data Anal Classif*.

Rohe, K., S. Chatterjee, and B. Yu (2011). Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics 39*(4), 1878 – 1915.

Simmel, G. (1950). *The sociology of Georg Simmel*. The free press, New York.

Singh, P. and G. Baruah (2021). Higher order interactions and species coexistence. *Theor Ecol 14*, 71–83.

Swan, M. and J. Zhan (2021). Clustering hypergraphs via the MapEquation. *IEEE Access 9*, 72377–72386.

Torres, L., A. S. Blevins, D. Bassett, and T. Eliassi-Rad (2021). The why, how, and when of representations for complex systems. *SIAM Review 63*(3), 435–485.

Turnbull, K., S. Lunagómez, C. Nemeth, and E. Airoldi (2021). Latent space modelling of hypergraph data. Technical report, arXiv:1909.00472.

Vazquez, A. (2009). Finding hypergraph communities: a Bayesian approach and variational solution. *Journal of Statistical Mechanics: Theory and Experiment 2009*(07), P07006.

Supplementary Material to: Model-based clustering in simple hypergraphs through a stochastic blockmodel

By Luca Brusa & Catherine Matias

All non-alphabetic references are concerned with the main text.

A Limits of the bipartite graphs representation of hypergraphs

A.1 Bipartite graphs and multiple hypergraphs with self-loops equivalence

Some early analyses of hypergraphs rely on the embedding of the former into the space of bipartite graphs (see for e.g. Battiston et al., 2020). Indeed, any hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ the set of hyperedges may be represented as a bipartite graph with two parts. The top part is simply the set $\mathcal{V}$ of hypergraph nodes, while the bottom part is the set $\mathcal{E}$ of hyperedges and there is a link between $v \in \mathcal{V}$ and $e \in \mathcal{E}$ whenever node v belongs to hyperedge e in the original hypergraph $\mathcal{H}$.

Now, it is possible to define a "converse" application from bipartite graphs to hypergraphs. Indeed, any bipartite graph can be projected into two distinct hypergraphs, by choosing one of the two parts as the nodes set and forming a hyperedge with any set of nodes that are neighbors (in the bipartite graph) of the same node (belonging to the second part). A major difference appears whether we consider simple hypergraphs or multiple hypergraphs with self-loops. In multiple hypergraphs (not to be confused with multisets-hypergraphs) hyperedges may appear several time so that these are weighted hypergraphs with integer valued weights. We also allow for self-loops, *i.e* hyperedges of cardinality 1. Then, this application from bipartite graphs to hypergraphs slightly differs depending on whether we allow the image of a bipartite graph to be a multiple hypergraphs with self-loops or a simple hypergraph. In the first case, all the information from the bipartite graph will be encoded in the multiple hypergraphs with self-loops; while in the second case, part of the information will be lost. This is illustrated on a toy example in Figure 2.

The embedding of the simple hypergraphs space into the bipartite graphs space is not the inverse of the natural projection of bipartite graphs into simple hypergraphs. Thus, models of bipartite graphs are inappropriate to handle simple hypergraphs, as the former generally put mass on any bipartite graph, notwithstanding the fact that not all of these may be realized as the image of a simple hypergraph. For the same reason, preferential attachment models of bipartite graphs (Guillaume and Latapy, 2004) may not be directly used for simple hypergraphs as they would produce unconstrained bipartite graphs that do not necessarily come from simple hypergraphs.

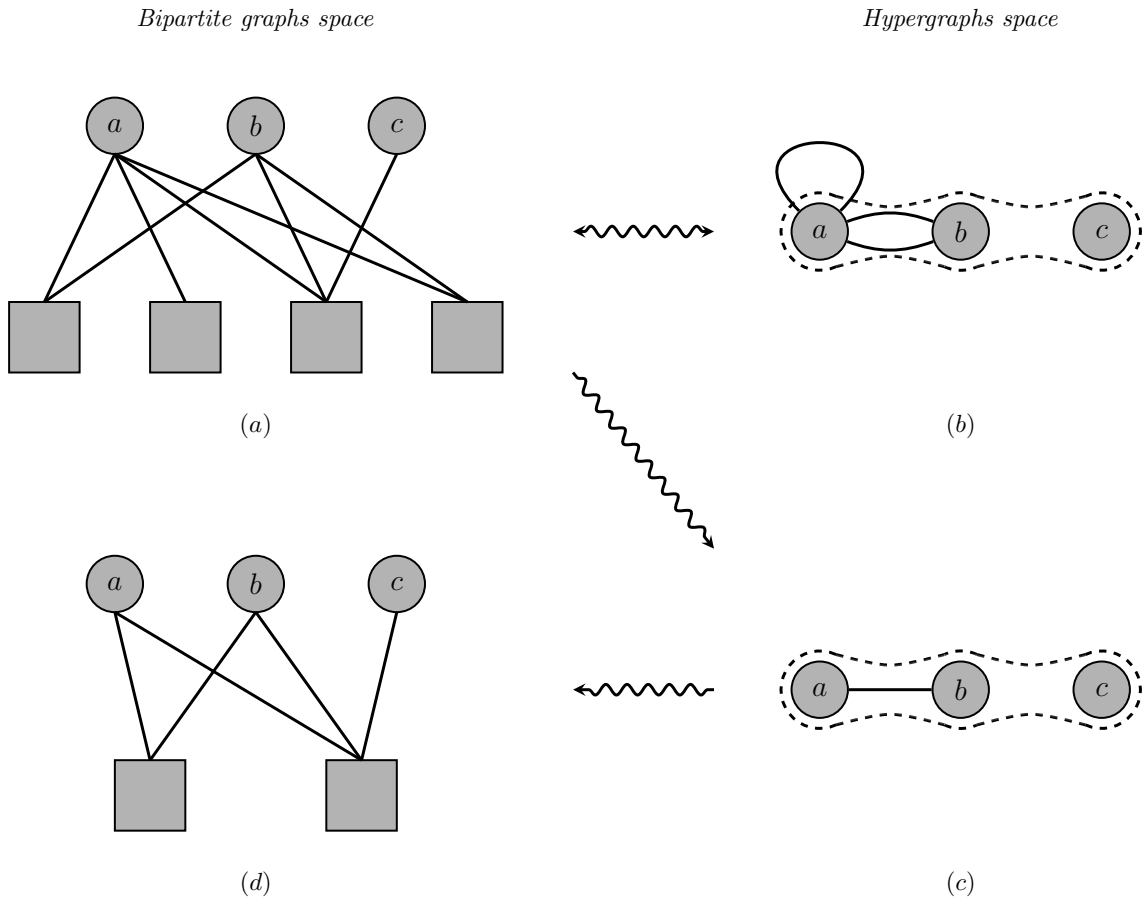


Figure 2: *(a) A bipartite graph G; (b) Projection of G into the multiple hypergraphs with self-loops space, choosing the top nodes as the new set of nodes. Hyperedges are $\{a\}, \{a, b\}, \{a, b\}, \{a, b, c\}$. The applications from (a) to (b) are invertible bijections, one being the inverse of the other; (c) Projection of G on the simple hypergraphs subspace. Hyperedges are $\{a, b\}, \{a, b, c\}$. (d) Embedding of the simple hypergraph in (c) in the bipartite graphs space. Note that (a) and (d) are not the same bipartite graph.*

A.2 Artefacts induced by bipartite graphs models

In order to view a bipartite graph as a hypergraph, one first needs to select the top and bottom parts. Swapping the role of the two parts will in general give another hypergraph. Most statistical models of bipartite graphs handle the two parts symmetrically and do not differentiate between a top and a bottom part. They are thus inadequate for modeling hypergraphs.

One may also note that most random bipartite graphs models are designed for fixed parts sizes, which induces, on top of a fixed number of nodes, a fixed number of hyperedges in the corresponding hypergraph model, an artifact which is not always desirable. For instance the uniformly random hypergraphs model allows for any possible density on the hyperedges.

A last example of inadequacy is given by configuration models on bipartite graphs that induce configuration models on hypergraphs. In these models, the degree distributions in each part are kept fixed. When projected in the hypergraphs space, that means that the degrees of the nodes and the sizes of the hyperedges are kept fixed. Then, relying on shuffling algorithms to explore the space of this configuration model, one will loose the labels on the bottom part (the hyperedges part) as these are automatically induced by the new edges of the bipartite graph and the labelling of the top part (the nodes part). As a consequence, if a specific node tends to take part in large size hyperedges, this information is lost in the configuration model issued from bipartite graphs.

To our knowledge, there is no configuration model on hypergraphs that only keeps the nodes degrees sequence fixed. We mention that Section 4 from Chodrow (2020) provides a discussion about the limitations of the embedding approach in terms of the types of hypergraph null models from which we can conveniently sample. In particular, Chodrow (2020) establishes that there is no obvious route for vertex-label sampling in hypergraphs through bipartite random graphs.

A.3 HyperSBM is not a bipartite SBM

In this section, we briefly outline that (i) while the bipartite stochastic blockmodel can be seen as a particular case of HSBM, (ii) the converse is not true in general.

To see point (i), let us consider a bipartite SBM on a graph G with nodes divided in 2 parts, say $\mathcal{V} = \{1, \dots, n\}$ and $\mathcal{U} = \{1, \dots, e\}$. The model has Q groups (resp. R groups) on the subset of nodes $\mathcal{V}$ (resp. $\mathcal{U}$), with group proportions π (resp. η). We let $Z_1, \dots, Z_n$ (resp. $W_1, \dots, W_e$) denote the latent groups of nodes $\mathcal{V}$ (resp. $\mathcal{U}$).

The model is also given by a connectivity matrix M of size $Q \times R$ whose entries M_{qr} are the conditional probabilities that a node in $\mathcal{V}$ from group q connects a node in $\mathcal{U}$ from group r. In other words $M_{qr} = \mathbb{P}(X_{iu} = 1 | Z_i = q, W_u = r)$ where $X = (X_{iu})$ is the $n \times e$ incidence matrix of G.

Now consider the hypergraph H constructed on the set of nodes $\mathcal{V}$ and whose hyperedges are obtained by looking at the set of nodes in $\mathcal{V}$ connected to a same node in $\mathcal{U}$. (A similar construction could be made with swapping the roles of $\mathcal{V}$ and $\mathcal{U}$). Then, the probability distribution of H

under the induced bipartite SBM is exactly a HSBM with Q groups, with group proportions π and parameters

$$\begin{aligned} B^{(m)}_{q_1,\dots,q_m} &= \mathbb{P}(Y_{i_1,\dots,i_m} = 1 | Z_{i_1} = q_1, \dots, Z_{i_m} = q_m) \\ &= \mathbb{P}(X_{i_1,u} = 1, \dots, X_{i_m,u} = 1 | Z_{i_1} = q_1, \dots, Z_{i_m} = q_m) \\ &= \sum_{r=1}^{R} \mathbb{P}(X_{i_1,u} = 1, \dots, X_{i_m,u} = 1, W_u = r | Z_{i_1} = q_1, \dots, Z_{i_m} = q_m) \\ &= \sum_{r=1}^{R} \eta_r \prod_{q=q_1}^{q_m} M_{qr}, \end{aligned}$$

where u is the node that connects $\{i_1, \dots, i_m\}$ into a hyperedge. So we see that the bibartite SBM induces a HSBM with constrained connection probabilities.

Let us now explain why (ii) the converse is not true in general. We start from a HSBM with Q groups and parameters $(\pi, (B^{(m)}_{q_1,\dots,q_m})_{q_1,\dots,q_m})_{2 \le m \le M}$ on a hypergraph H with set of nodes $\mathcal{V}$. Considering $\mathcal{U} = \{1, \dots, e\}$ where e is the number of hyperedges in H, we construct a bipartite graph G with nodes $\mathcal{V} \times \mathcal{U}$ and links between any $i \in \mathcal{V}$ and any $u \in \mathcal{U}$ whenever node i belongs to hyperedge u in the hypergraph H. Now, if there is a bipartite SBM on G with same distribution as HyperSBM, then necessarily it has Q groups on $\mathcal{V}$, with group proportions given by π. We let R denote the number of groups on such a model on $\mathcal{U}$, together with η the corresponding group proportions, and M the $Q \times R$ matrix of connection probabilities. Then we observe that η and M should satisfy the relations

$$\forall 2 \le m \le M, \forall q_1, \dots, q_m \in \{1, \dots, Q\}^m, \quad B^{(m)}_{q_1,\dots,q_m} = \sum_{r=1}^{R} \eta_r \prod_{q=q_1}^{q_m} M_{qr}. \tag{9}$$

Here, we first remark that the bipartite SBM fit on the co-authorship dataset (from Section 5) selected $R = 1$, thus inducing hyperedges connectivity parameters with a product form

$$B^{(m)}_{q_1,\dots,q_m} = \prod_{q=q_1}^{q_m} M_{q1}.$$

Our fitted HSBM on this same dataset did not result in hyperedges connectivity parameters with a product form, which establishes that the models are clearly different.

Now, more generally, we could ask whether for given parameters $(B^{(m)}_{q_1,\dots,q_m})_{2 \le m \le M}$, there exist some values of R, η and M such that (9) is satisfied. The answer is: not always. To see this, consider for instance $Q = 2$ and remark the relation between the two quantities

$$B^{(2)}_{11} = \sum_{r=1}^{R} \eta_r M^2_{1r},$$

$$B^{(3)}_{111} = \sum_{r=1}^{R} \eta_r M^3_{1r},$$

so that $B^{(2)}_{11}$ and $B^{(3)}_{111}$ cannot be chosen independently.

B The complete proof of Theorem 2

For the sake of completeness, we provide here the complete proof of Theorem 2. This proof mostly reproduces the proof of Theorem 2 in Allman et al. (2011).

The strategy relying on Kruskal's result. The proof strongly relies on an algebraic result from Kruskal (1977) that appeared to be a powerful tool to establish identifiability results in various models whose common feature is the presence of discrete latent groups and at least three conditionally independent random variables. We first rephrase Kruskal's result in a statistical context. Consider a latent random variable V with state space $\{1, \ldots, r\}$ and distribution given by the column vector $\mathbf{v} = (v_1, \ldots, v_r)$. Assume that there are three observable random variables U_j for $j = 1, 2, 3$, each with finite state space $\{1, \ldots, \kappa_j\}$. The U_js are moreover assumed to be independent conditional on V. Let M_j, $j = 1, 2, 3$ be the stochastic matrix of size $r \times \kappa_j$ whose ith row is $\mathbf{m}_i^j = \mathbb{P}(U_j = \cdot \mid V = i)$. Then consider the 3-dimensional array (or tensor) with dimensions $\kappa_1 \times \kappa_2 \times \kappa_3$ denoted $[\mathbf{v}; M_1, M_2, M_3]$ and whose (s, t, u) entry (for any $1 \leq s \leq \kappa_1, 1 \leq t \leq \kappa_2, 1 \leq u \leq \kappa_3$) is defined by

$$\begin{aligned}
[\mathbf{v}; M_1, M_2, M_3]_{s,t,u} &= \sum_{i=1}^{r} v_i m_i^1(s)\, m_i^2(t)\, m_i^3(u) \\
&= \sum_{i=1}^{r} \mathbb{P}(V = i)\mathbb{P}(U_1 = s|V = i)\mathbb{P}(U_2 = t|V = i)\mathbb{P}(U_3 = u|V = i) \\
&= \mathbb{P}(U_1 = s, U_2 = t, U_3 = u).
\end{aligned}$$

Note that $[\mathbf{v}; M_1, M_2, M_3]$ is left unchanged by simultaneously permuting the rows of all the M_j and the entries of $\mathbf{v}$, as this corresponds to permuting the labels of the latent classes. Knowledge of the distribution of (U_1, U_2, U_3) is equivalent to knowledge of the tensor $[\mathbf{v}; M_1, M_2, M_3]$.
Now, the *Kruskal rank* of a matrix M, denoted $\operatorname{rank}_K M$, is the largest number I such that *every* set of I rows of M are independent. Note that for any matrix M, its Kruskal rank is necessarily less than its rank, namely $\operatorname{rank}_K M \leq \operatorname{rank} M$, and equality of rank and Kruskal rank does not hold in general. However, in the particular case when a matrix M of size $p \times q$ has rank p, it also has Kruskal rank p. Now, let $I_j = \operatorname{rank}_K M_j$. Kruskal (1977) established the following result. If

$$I_1 + I_2 + I_3 \geq 2r + 2, \tag{10}$$

then the tensor $[\mathbf{v}; M_1, M_2, M_3]$ uniquely determines $\mathbf{v}$ and the M_j, up to simultaneous permutation of the rows. In other words, the set of parameters $\{(\mathbf{v}, \mathbb{P}(U_j = \cdot \mid V))\}$ is uniquely identified, up to label switching on the latent groups, from the distribution of the random variables (U_1, U_2, U_3).

Now, to obtain generic identifiability, it is sufficient to exhibit a single parameter value for which (10) is satisfied. Indeed, the set of parameter values for which $\operatorname{rank}_K M_j$ is fixed can be expressed through a Boolean combination of polynomial inequalities ($\neq$, or rather non-equalities) involving matrix minors in those parameters. In the same way, the converse condition of (10), namely inequality $I_1 + I_2 + I_3 \leq 2r + 1$ is the finite Boolean combination of polynomial non-equalities on the model parameters. This means that this set of parameters is an algebraic variety. But an algebraic variety can only be either the whole parameter space (in which case exhibiting a single value where (10) is satisfied would not be possible) or a proper subvariety, thus a subspace of dimension strictly lower than that of the whole parameter space.

The strategy of the proof for showing identifiability of certain discrete latent class models developed in Allman et al. (2011) and other papers by the same authors is to embed these models in the context of Kruskal's result just described. Applying Kruskal's result to the embedded model, the authors derive partial identifiability results on the embedded model, and then, using details of the embedding, relate these to the original model.

Embedding the HSBM into Kruskal's setup. For some number of nodes n (to be specified later), we let $V = (Z_1, Z_2, \ldots, Z_n)$ be the latent random variable, with state space $\{1, \ldots, Q\}^n$ and denote by $\mathbf{v}$ the corresponding vector of its probability distribution. The entries of $\mathbf{v}$ are of the form $\pi_1^{n_1} \cdots \pi_Q^{n_Q}$ for some integers $n_q \geq 0$ and such that $\sum_q n_q = n$. We fix $m \geq 2$ and consider simple m-uniform hypergraphs on the set of nodes $\mathcal{V} = \{1, \ldots, n\}$. Recall that $\mathcal{V}^{(m)}$ is the set of all distinct m-tuples of nodes in $\mathcal{V}$ and $\{Y_{i_1,\ldots,i_m}; \{i_1, \ldots, i_m\} \in \mathcal{V}^{(m)}\}$ the set of all indicator variables corresponding to possible (simple) hyperedges of a m-uniform hypergraph over $\mathcal{V}$. Now, we will construct below subsets $H_1, H_2, H_3 \subset \mathcal{V}^{(m)}$ of distinct m-tuples of nodes such that $H_i \cap H_j = \emptyset$ for any $i \neq j$. Then, we choose the 3 observed variables U_j ($1 \leq j \leq 3$) as the vectors of indicator variables $U_j = (Y_{i_1,\ldots,i_m})_{\{i_1,\ldots,i_m\} \in H_j}$. This induces that $\kappa_j = 2^{|H_j|}$ (where $|H_j|$ is the cardinality of H_j). As the subsets H_1, H_2, H_3 do not share any m-tuple of nodes, the random variables U_j are conditionally independent given V. We are in the statistical context of Kruskal's result.

The goal is now to construct the 3 subsets H_j of m-tuples such that their pairwise intersections are empty and such that condition (10) is satisfied (for at least one parameter value of the embedded model and thus generically for this embedded model). This construction of the H_j's proceeds in two steps: the base case and an extension step.

Starting with a small set $\mathcal{V}_0 = \{1, \ldots, n_0\}$ of nodes, we define a matrix A of dimension $Q^{n_0} \times 2^{\binom{n_0}{m}}$. Its rows are indexed by latent configurations $v \in \{1, \ldots, Q\}^{n_0}$ of the nodes in $\mathcal{V}_0$, its columns by the set of all possible states of the vector of indicator variables $(Y_{i_1,\ldots,i_m})_{\{i_1,\ldots,i_m\} \in \mathcal{V}_0}$, and the entries of A give the probability of observing the specified states of the vector of indicator variables, conditioned on the latent configurations v. Thus each column index corresponds to a different simple m-uniform hypergraph on $\mathcal{V}_0$. The base case consists in exhibiting a value of

n_0 such that this matrix A generically has full row rank. Then, in an extension step, relying on $n = n_0^2$ nodes, we construct the subsets H_1, H_2, H_3 with the desired properties (namely their pairwise intersections are empty and (10) is generically satisfied).

From Kruskal's theorem, we obtain that the vector $\mathbf{v}$ and the matrices M_1, M_2, M_3 are generically uniquely determined, up to simultaneous permutation of the rows from the distribution of a simple m-uniform HSBM.

With these embedded parameters $\mathbf{v}, M_1, M_2, M_3$ in hand, it is still necessary to recover the initial parameters of the simple m-uniform HSBM: the group proportions π_q and the connectivity matrix $B^{(m)} = (B^{(m)}_{q_1,\dots,q_m})_{1\le q_1\le\dots\le q_m\le Q}$. This will be done in the conclusion.

Base case. In the following, we drop the exponent (m) in the notation for the connection probabilities B and simply let $B_{q_1,\dots,q_m} = \mathbb{P}(Y_{i_1,\dots,i_m} = 1 \mid Z_{i_1} = q_1, \dots, Z_{i_m} = q_m) = 1 - \bar{B}_{q_1,\dots,q_m}$. The initial step consists in finding a value of n_0 such that the matrix A of size $Q^{n_0} \times 2^{\binom{n_0}{m}}$ containing the probabilities of any simple m-uniform hypergraph over these n_0 nodes, conditional on the hidden node states, generically has full row rank.

The condition of having full row rank can be expressed as the non-vanishing of at least one $Q^{n_0} \times Q^{n_0}$ minor of A. Composing the map sending the parameters $\{B_{q_1,\dots,q_m}\} \to A$ with this collection of minors gives polynomials in the parameters of the model. To see that these polynomials are not identically zero, and thus are non-zero for generic parameters, it is enough to exhibit a single choice of the $\{B_{q_1,\dots,q_m}\}$ for which the corresponding matrix A has full row rank. We choose to consider parameters $\{B_{q_1,\dots,q_m}\}$ of the form

$$B_{q_1,\dots,q_m} = \frac{s_{q_1}s_{q_2}\dots s_{q_m}}{s_{q_1}s_{q_2}\dots s_{q_m} + t_{q_1}t_{q_2}\dots t_{q_m}}, \text{ so } \bar{B}_{q_1,\dots,q_m} = \frac{t_{q_1}t_{q_2}\dots t_{q_m}}{s_{q_1}s_{q_2}\dots s_{q_m} + t_{q_1}t_{q_2}\dots t_{q_m}},$$

with $s_q, t_l > 0$ to be chosen later. However, since the property of having full row rank is unchanged under non-zero rescaling of the rows of the matrix A, and all entries of A are monomials with total degree $\binom{n_0}{m}$ in $B_{q_1,\dots,q_m}, \bar{B}_{q_1,\dots,q_m}\}$, we may simplify the entries of A by removing denominators, and consider the matrix (also called A) with entries in terms of $B_{q_1,\dots,q_m} = s_{q_1}s_{q_2}\dots s_{q_m}$ and $\bar{B}_{q_1,\dots,q_m} = t_{q_1}t_{q_2}\dots t_{q_m}$.

The rows of A are indexed by the composite node states $v \in \{1,\dots,Q\}^{n_0}$, while its columns are indexed by the m-uniform hypergraphs $\mathcal{H} = (y_{i_1,\dots,i_m})_{\{i_1,\dots,i_m\}\in\mathcal{V}_0} \in \{0,1\}^{\binom{n_0}{m}}$. For any composite hidden state $v \in \{1,\dots,Q\}^{n_0}$ and any node $i \in \{1,\dots,n_0\}$, let $v(i) \in \{1,\dots,Q\}$ denote the state of node i in the composite state v. With our particular choice of the parameters $B_{q_1,\dots,q_m}$, the $(v,\mathcal{H})$-entry of A is given by

$$\prod_{\{i_1\dots i_m\}\in\mathcal{V}_0^{(m)}} B^{y_{i_1,\dots,i_m}}_{v(i_1),\dots,v(i_m)} \bar{B}^{1-y_{i_1,\dots,i_m}}_{v(i_1),\dots,v(i_m)} = \prod_{i=1}^{n_0} s^{d_i}_{v(i)} t^{n_0-1-d_i}_{v(i)},$$

where

$$d_i = \sum_{\substack{\{i_1 \dots i_m\} \in \mathcal{V}_0^{(m)} \\ i \in \{i_1 \dots i_m\}}} y_{i_1,\dots,i_m}$$

is the degree of node i in the hypergraph $\mathcal{H} = (y_{i_1,\dots,i_m})_{\{i_1,\dots,i_m\} \in \mathcal{V}_0}$. With this choice of parameters $\{B_{q_1,\dots,q_m}\}$, the entries in a column of A are entirely determined by the degree sequence $\mathbf{d} = (d_i)_{1 \le i \le n_0}$ of the hypergraph under consideration. Two different hypergraphs may result in the same degree sequence, thus the same values in the two columns of A. For any degree sequence $\mathbf{d} = (d_i)_{1 \le i \le n_0}$ arising from a simple m-uniform hypergraph on n_0 nodes, let $A_{\mathbf{d}}$ denote a corresponding column of A. In order to prove that the matrix A has full row rank, it is enough to exhibit Q^{n_0} independent columns of A. To this aim, we introduce polynomial functions whose independence is equivalent to that of corresponding columns.

For each node $i \in \{1, \dots, n_0\}$ and each latent group $q \in \{1, \dots, Q\}$, introduce an indeterminate $X_{i,q}$ and a Q^{n_0}-size row vector $\mathbf{X} = (\prod_{1 \le i \le n_0} X_{i,v(i)})_{v \in \{1,\dots,Q\}^{n_0}}$. For each degree sequence $\mathbf{d}$, we have

$$\mathbf{X} A_{\mathbf{d}} = \sum_{v \in \{1,\dots,Q\}^{n_0}} \prod_{1 \le i \le n_0} s_{v(i)}^{d_i} t_{v(i)}^{n_0 - 1 - d_i} X_{i,v(i)} = \prod_{1 \le i \le n_0} \left(s_1^{d_i} t_1^{n_0 - 1 - d_i} X_{i,1} + \dots + s_Q^{d_i} t_Q^{n_0 - 1 - d_i} X_{i,Q} \right).$$

Now, independence of a set of columns $\{A_{\mathbf{d}}\}$ is equivalent to the independence of the corresponding set of polynomial functions $\{\mathbf{X} A_{\mathbf{d}}\}$ in the indeterminates $\{X_{i,q}\}$. For a set $\mathcal{D}$ of degree sequences, to prove that the polynomials $\{\mathbf{X} A_{\mathbf{d}}\}_{\mathbf{d} \in \mathcal{D}}$ are independent, we assume that there exist scalars $a_{\mathbf{d}}$ such that

$$\sum_{\mathbf{d} \in \mathcal{D}} a_{\mathbf{d}} \mathbf{X} A_{\mathbf{d}} \equiv 0, \tag{11}$$

and show that necessarily all $a_{\mathbf{d}} = 0$. This will be given by the following lemma from Allman et al. (2011). This lemma is originally formulated for a set $\mathcal{D}$ of degree sequences. However it is not specific to degree sequences; it applies for any sets $\mathcal{D}$ of sequences of integers indexed by $\{1, \dots, n_0\}$ and thus we phrase it in this way. We refer to Allman et al. (2011) for its proof.

Lemma 8. *(Lemma 18 in Allman et al. (2011).) Assume $n_0 \ge Q$. Let $\mathcal{D}$ be a set of n_0-length integer sequences such that for each $i \in \{1, \dots, n_0\}$, the set of i-th coordinates $\{d_i \mid \mathbf{d} \in \mathcal{D}\}$ has cardinality at most Q. Then for generic values of s_q, t_l, for each i and each $d^\star \in \{d_i \mid \mathbf{d} \in \mathcal{D}\}$ there exist values of the indeterminates $\{X_{i,q}\}_{1 \le q \le Q}$ that annihilate all the polynomials $\mathbf{X} A_{\mathbf{d}}$ for $\mathbf{d} \in \mathcal{D}$ except those for which $d_i = d^\star$.*

The next step is to construct a set $\mathcal{D}$ of n_0-length integer sequences that satisfies

- for each $i \in \{1, \dots, n_0\}$, the set of i-th coordinates $\{d_i \mid \mathbf{d} \in \mathcal{D}\}$ has cardinality at most Q (condition in Lemma 8);
- any $\mathbf{d} \in \mathcal{D}$ may be the degree sequence of a simple m-uniform hypergraph;

- $|\mathcal{D}| \geq Q^{n_0}$.

With such a set at hand, by choosing one column of A associated to each degree sequence in $\mathcal{D}$, we obtain a collection of $|\mathcal{D}| \geq Q^{n_0}$ different columns of A. These columns are independent since for each sequence $\mathbf{d}^\star \in \mathcal{D}$, by Lemma 8 we can choose values of the indeterminates $\{X_{i,q}\}_{1\leq i\leq n_0, 1\leq q\leq Q}$ such that all polynomials $\mathbf{X}A_{\mathbf{d}}$ vanish, except $\mathbf{X}A_{\mathbf{d}^\star}$, leading to $a_{\mathbf{d}^\star} = 0$ in equation (11). Thus, exhibiting such a set $\mathcal{D}$ is the last step to prove that A has generically full row rank.

Now, this is where our proof strongly differs from the one of Theorem 2 in Allman et al. (2011). Indeed, the characterizations of degree sequences for graphs and simple m-uniform hypergraphs are completely different. Relying on a result by Behrens et al. (2013), we have exhibited such a set in Lemma 7.

This concludes the proof of the base case.

The extension step. The extension step builds on the base case, in order to construct a larger set of $n = n_0^2$ nodes and subsets $H_1, H_2, H_3 \subset \mathcal{V}^{(m)}$ of distinct m-tuples of nodes in $\mathcal{V} = \{1, \ldots, n\}$ with the desired properties. This step was first stated as Lemma 16 in Allman et al. (2009) in the context of simple graphs SBM and we extend it below to our case.

Let us recall that we want to construct $H_1, H_2, H_3 \subset \mathcal{V}^{(m)}$ that are pairwise disjoint. Then, with notation from above, we choose the 3 observed variables U_j $(1 \leq j \leq 3)$ as the vectors of indicator variables $U_j = (Y_{i_1,\ldots,i_m})_{\{i_1,\ldots,i_m\}\in H_j}$. As the subsets H_1, H_2, H_3 do not share any m-tuple of nodes, the random variables U_j are conditionally independent given $V = (Z_1, \ldots, Z_n)$. We let M_j denote the $Q^n \times 2^{|H_j|}$ matrix of conditional probabilities of U_j given Z.

Lemma 9. *Suppose that for some number of nodes n_0, the matrix A of size $Q^{n_0} \times 2^{\binom{n_0}{m}}$ defined above has generically full row rank. Then with $n = n_0^2$ there exist pairwise disjoint subsets $H_1, H_2, H_3 \subset \mathcal{V}^{(m)}$ of m-tuples of nodes in $\mathcal{V} = \{1, \ldots, n\}$ such that for each j the $Q^n \times 2^{|H_j|}$ matrix M_j has generically full row rank (Q^n).*

Proof of Lemma 9. Let us describe the construction of H_j. We will partition the n_0^2 nodes into n_0 groups of size n_0 in three different ways, each way leading to one H_j. Then each H_j will be the union of the n_0 sets of all m-tuples made of some n_0 nodes. Thus each H_j has cardinality $n_0\binom{n_0}{m}$.

Labeling the nodes by $(u, v) \in \{1, \cdots, n_0\} \times \{1, \cdots, n_0\}$, we picture the nodes as lattice points in a square grid. We take as the partition leading to H_1 the rows of the grid, as the partition leading to H_2 the columns of the grid, and as the partition leading to H_3 the diagonals. In other words, H_1 is the union over n_0 rows of all m-tuples of nodes within each row. The same with columns and diagonals. Explicitly, we define two functions u, v that associate to any $i \in \{1, \ldots, n_0\}$ its coordinates $(u(i), v(i))$ on the $n_0 \times n_0$ grid. Then, the H_j are m-tuple of nodes

defined as

$$
\begin{aligned}
H_1 &= \cup_{u=1}^{n_0} H_1(u) = \cup_{u=1}^{n_0} \{\{i_1, \ldots, i_m\} \in \mathcal{V}^{(m)} \mid \forall k, u(i_k) = u, v(i_k) \in \{1, \cdots, n_0\}\},\\
H_2 &= \cup_{v=1}^{n_0} H_2(v) = \cup_{v=1}^{n_0} \{\{i_1, \ldots, i_m\} \in \mathcal{V}^{(m)} \mid \forall k, v(i_k) = v, u(i_k) \in \{1, \cdots, n_0\}\},\\
H_3 &= \cup_{s=1}^{n_0} H_3(s)\\
&= \cup_{s=1}^{n_0} \{\{i_1, \ldots, i_m\} \in \mathcal{V}^{(m)} \mid \forall k, u(i_k) = s, v(i_k) = s + t \bmod n_0 \text{ for some } t \in \{1, \cdots, n_0\}\}.
\end{aligned}
$$

The H_j are pairwise disjoints as required.

The matrix M_j of conditional probabilities of U_j given Z has Q^n rows indexed by composite states of all $n = n_0^2$ nodes, and $2^{n_0\binom{n_0}{m}}$ columns indexed by m-tuples in H_j.

Observe that with an appropriate ordering of the rows and columns (which is dependent on j), M_j has a block structure given by

$$M_j = A \otimes A \otimes \cdots \otimes A \ (n_0 \text{ factors}). \tag{12}$$

(Note that since A is $Q^{n_0} \times 2^{\binom{n_0}{m}}$, the tensor product on the right is $(Q^{n_0})^{n_0} \times \left(2^{\binom{n_0}{m}}\right)^{n_0}$ which is $Q^{n_0^2} \times 2^{n_0\binom{n_0}{m}}$, the size of M_j.) That M_j is this tensor product is most easily seen by noting the partitioning of the n_0^2 nodes into n_0 disjoint sets (rows, columns and diagonals of the grid) gives rise to n_0 copies of the matrix A, one for each set of all simple m-uniform hypergraphs over n_0 nodes. The row indices of M_j are obtained by choosing an assignment of states to the nodes in $H_j(u)$ for each u independently, and the column indices by the union of independently-chosen simple m-uniform hypergraphs subgraphs on $H_j(u)$ for each u. This independence in both rows and columns leads to the tensor decomposition of M_j.

Now since A has generically full row rank (Q^{n_0}), equation (12) implies that M_j does as well (*i.e* has row rank $Q^{n_0^2} = Q^n$). □

Next, with $\mathbf{v}, M_1, M_2, M_3$ defined by the embedding given in the previous paragraphs, we apply Kruskal's Theorem to the table $[\mathbf{v}; M_1, M_2, M_3]$. By construction of the M_j, condition (10) is generically satisfied since $3Q^n \geq 2Q^n + 2$. Thus the vector $\mathbf{v}$ and the matrices M_1, M_2, M_3 are generically uniquely determined, up to simultaneous permutation of the rows from the distribution of a simple m-uniform HSBM.

It now remains to recover the original parameters of the simple m-uniform HSBM: the group proportions π_q and the connectivity matrix $(B^{(m)}_{q_1,\ldots,q_m})_{1 \leq q_1 \leq q_m \leq Q}$.

Conclusion for the original model. The entries of $\mathbf{v}$ are of the form $\pi_1^{n_1} \cdots \pi_Q^{n_Q}$ with $\sum n_q = n$, while the entries of the M_j contain information on the $B^{(m)}_{q_1,\ldots,q_m}$. Although the ordering of the rows of the M_j is arbitrary, crucially we do know how the rows of M_j are paired with the entries of $\mathbf{v}$.

By focusing on one of the matrices, say M_1, and adding appropriate columns of it, we can obtain marginal conditional probabilities of single hyperedge variables, namely a column vector

with values $(\mathbb{P}_\theta(Y_{i_1,\dots,i_m}=1|(Z_1,\dots,Z_n)=v))_v$ for any m-tuple $\{i_1,\dots,i_m\}$. Indeed, this vector is obtained by summing all the columns of M_1 corresponding to simple m-uniform hypergraphs with $Y_{i_1,\dots,i_m}=1$. Thus, we recover the set of values $\{B^{(m)}_{q_1,\dots,q_m}\}_{1\le q_1\le\dots\le q_m\le Q}$, but without order. Namely, we still do not know the $B^{(m)}_{q_1,\dots,q_m}$ up to a permutation on $\{1,\dots,Q\}$ only, but rather up to a permutation on $\{1,\dots,Q\}^n$.

In the following, we assume without loss of generality, as it is a generic condition, that all $\{B^{(m)}_{q_1,\dots,q_m}\}_{1\le q_1\le\dots\le q_m\le Q}$ are distinct.

We look at the first $(m+1)$ nodes $\mathcal{V}_1=\{1,\dots,m,m+1\}$ and consider the $m+1$ different m-tuples $\{i_1,\dots,i_m\}\in\mathcal{V}_1^{(m)}$ that can be made from these nodes ($i_k\in\mathcal{V}_1$). Again, for each of these m-tuples, adding appropriate columns of M_1, we can jointly obtain the vectors of conditional marginal probabilities $(\mathbb{P}_\theta(Y_{\{i_1,\dots,i_m\}}=1|(Z_1,\dots,Z_n)=v))_v$. Jointly means that all those vectors share the same ordering over the index $v\in\{1,\dots,Q\}^n$. In other words, we recover the sets of values

$$\forall v\in\{1,\dots,Q\}^n,\quad R_v=\{B^{(m)}_{v_{i_1},\dots,v_{i_m}};\{i_1,\dots,i_m\}\in\mathcal{V}_1^{(m)}\}.$$

Now, we assumed the B's are all distinct so the cardinalities of the sets R_v will help us discriminate the different parameters (up to a permutation on $\{1,\dots,Q\}$ only). Indeed, there are exactly Q sets R_v with cardinality exactly one. These corresponds to the cases were $v=(q,q,\dots,q)$ for some $1\le q\le Q$. From this, we can distinguish the parameters of the form $\{B^{(m)}_{q,\dots,q};1\le q\le Q\}$ from the complete set of parameters. Note that the corresponding entries of v are given by π_q^m. So we also recover the paired values $\{(\pi_q,B^{(m)}_{q,\dots,q});1\le q\le Q\}$. Then, we continue with the sets R_v with cardinality two: these are of the form $\{B^{(m)}_{q,\dots,q};B^{(m)}_{q,\dots,q,l}\}$ for some $1\le q\ne l\le Q$. As we already identified the parameters $\{B^{(m)}_{q,\dots,q};1\le q\le Q\}$ and all B's are distinct, this enables us to identify the set of parameters $\{B^{(m)}_{q,\dots,q,l};1\le q\ne l\le Q\}$. By induction, we recover the set of parameters $\{B^{(m)}_{q,\dots,q,l,l'};1\le q,l,l'\le Q$ and q,l,l' distinct$\}$ *et caetera*, ending with the set of parameters $\{B^{(m)}_{q_1,\dots,q_m};1\le q_1<q_2<\dots<q_m\le Q\}$. This means that we finally have obtained the parameters $\{\pi_q,B^{(m)}_{q_1,\dots,q_m}\}_{1\le q\le Q;1\le q_1\le\dots\le q_m\le Q}$ up to a permutation over $\{1,\dots,Q\}$.

Finally, note that all generic aspects of this argument, in the base case and the requirement that the parameters $B^{(m)}_{q_1,\dots,q_m}$ be distinct, concern only the $B^{(m)}_{q_1,\dots,q_m}$. Thus if the group proportions π_q are fixed to any specific values, the theorem remains valid.

Remark. The requirement on large enough n is more precisely given as $n\ge Q^2p^2$ where p is the smallest integer such that $\binom{p-1}{m-1}\ge Qm$. A rough approximation gives that p is of the order $(Qm)^{1/(m-1)}$ which gives that n should be larger than $Q^2(Qm)^{2/(m-1)}$.

C Computational details on the algorithm's implementation

In order to provide an efficient implementation, the whole estimation algorithm is implemented in **C**++ language using the `Armadillo` library for linear algebra. Moreover the implementation

is made available in R by means of the R packages Rcpp (Eddelbuettel and François, 2011; Eddelbuettel, 2013) and RcppArmadillo (Eddelbuettel and Sanderson, 2014). In the following we consider some of the most relevant computational details.

Dealing with heavy computational cost. Dealing with very large data structures, the main drawback of the proposed algorithm is the intensive computational effort, in terms of both execution time needed to converge and required memory space. The most outstanding example regards the computation of the products $\tau_{i_1 q_1} \cdots \tau_{i_m q_m}$, required both in the VE-Step (see Proposition 4, for τ_{iq}) and in the M-Step (see Proposition 5, for $B^{(m)}_{q_1,\ldots,q_m}$). The huge computational cost of this calculation derives from the large number of potential unordered node tuples even for rather small values of n and m; indeed $|\mathcal{V}^{(m)}| = \binom{n}{m}$. A first possibility is to compute all the products $\tau_{i_1 q_1} \cdots \tau_{i_m q_m}$ in a recursive manner at the beginning of each VEM iteration and to store them in a matrix. Although this is actually very beneficial for the computational time, the resulting matrix is huge, having number of rows and columns equal to $\binom{n}{m}$ and $\binom{Q+m-1}{m}$ respectively. The result is a structure that is intractable except for very small values of n, Q, and (especially) m. Taking into account that every element requires 8 bytes, we report some examples in Table 6, in order to better clarify the magnitude of the quantity to store. Stated the impossibility to store a matrix of such size, the computation of the required products $\tau_{i_1 q_1} \cdots \tau_{i_m q_m}$ is implemented directly inside the VE- and M-Steps through nested loops; this process involves an important increase in the computing times, but on the other hand requires a minimal amount of memory. To handle the slowness of the computation, both the VE-Step and the M-Step are efficiently implemented in parallel through the RcppParallel package (Allaire et al., 2022).

n	m	Q	Memory size
100	3	2	$\approx 5.2\ MB$
100	3	4	$\approx 25.9\ MB$
100	6	2	$\approx 66.8\ GB$
100	6	4	$\approx 801.1\ GB$

n	m	Q	Memory size
500	3	2	$\approx 0.6\ GB$
500	3	4	$\approx 3.3\ GB$
500	6	2	$\approx 1179.2\ TB$
500	6	4	$\approx 14150.8\ TB$

Table 6: *Memory size of the matrix containing the products $\tau_{i_1 q_1} \cdots \tau_{i_m q_m}$ for given values of n (number of nodes), Q (number of latent groups) and m (hyperedge size).*

Floating point underflow. Another crucial aspect is the possible occurrence of numerical instability deriving from the multiplication of many small values in the computation of $\widehat{\tau}_{iq}$. A simple remedy is provided by the calculation of $\log \widehat{\tau}_{iq}$ instead of $\widehat{\tau}_{iq}$. So, denoting $b_{iq} = \log(\hat{\tau}_{iq} -$

c_i), we compute $\hat{\tau}_{iq}$ relying on

$$\widehat{\tau}_{iq} = \frac{\exp(b_{iq} - b_{\text{MAX},i})}{\sum_{p=1}^{Q} \exp(b_{iq} - b_{\text{MAX},i})},$$

where $b_{\text{MAX},i} = \max_{q=1\ldots Q} b_{iq}$ prevents the denominator to grow excessively large, thus avoiding new potential numerical issues related to the floating point underflow.

D Hyperparameters settings

All the experiments were made with the following hyperparameters. Concerning the soft spectral clustering initialization, the k-means algorithm (on the rows of the column leading eigenvectors matrix) is run with 100 initializations. The tolerance threshold ϵ used to stop the fixed point and the `VEM` algorithm is set to 10^{-6}. The maximum numbers of iterations for the fixed point and the `VEM` algorithm were set to $U_{\max} = 50$ and $T_{\max} = 50$, respectively.

E Influence of the initial value on `VEM`

As a final note, we also assess the influence of starting values on the behavior of the `VEM` algorithm. To this aim, we preliminary analyze the performance of spectral clustering algorithm, relying again on the ARI. Results are reported in Table 7 and clearly show the opposite behavior of this clustering method in detecting communities (scenario A) and disassortative behaviours (scenarios B and C): taking into account community detection, spectral clustering algorithm perfectly recovers the true clusters, apart from a few cases in which n is very small. On the contrary, considering disassortative behaviours, spectral clustering algorithm completely fails in determining the correct clusters, all values of the ARI being extremely close to 0. Hence, "soft" spectral clustering proves to be a very smart initialization strategy for scenario A, while for scenarios B and C, it behaves analogously to a random starting value. The optimal performance of `VEM` algorithm throughout all scenarios, therefore, highlights a very weak influence of the starting value on the behavior of the algorithm: a random initialization usually ensures a proper convergence and a correct clustering; instead, the real advantage deriving from the adoption of a smart initial value is a reduction in computing time (data not shown).

F Analyses on the co-authorship dataset

The original dataset has 274 papers and 314 authors, with 1 paper having 6 authors and 1 paper having 5 authors. We decided to consider $M = 4$ and discard these 2 papers with more than 5 authors. Then, we looked at the largest connected component of the resulting graph. It resulted in 76 papers and 79 authors.

n	Scenario A	Scenario B	Scenario C
50	0.69	$-0.43 \cdot 10^{-2}$	$-3.57 \cdot 10^{-3}$
100	0.99	$0.43 \cdot 10^{-2}$	$-7.01 \cdot 10^{-3}$
150	1.00	$-0.12 \cdot 10^{-2}$	$4.60 \cdot 10^{-5}$
200	1.00	$-0.42 \cdot 10^{-2}$	$-4.51 \cdot 10^{-3}$

Table 7: *Adjusted Rand Index for different scenarios and number of nodes with respect to the soft spectral clustering initialization. Each value is obtained as the average over 10 simulated datasets.*

We ran `HyperSBM` with Q ranging from 2 to 5, with 2 different initialisations: 1 random and 1 relying on the soft spectral clustering. The random initialisation always gave the best result. The results were robust to different tries. ICL selected $Q = 2$ groups, as shown in Figure 3.

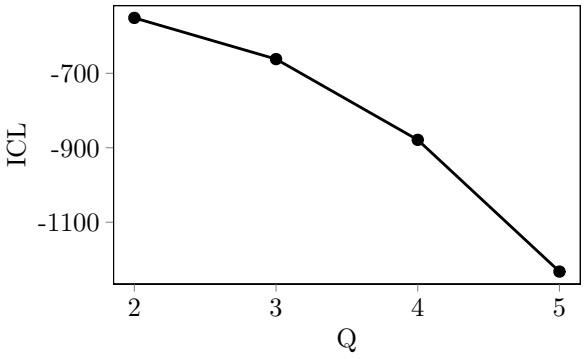


Figure 3: *Integrated Classification Likelihood index resulting from fitting the HSBM to the co-authorship dataset with number of latent groups ranging from 2 to 5.*

We obtained a first small group with only 8 authors (the remaining 71 authors being in the second large group). Inspecting more closely the variational parameters τ_{iq} for all the nodes, we found that a total of 4 nodes could be considered as ambiguously classified, while all other nodes had posterior probabilities to belong to one of the group larger than 0.8. More precisely, in the first small group, 2 nodes had posterior probabilities to belong to that group equal to 0.54 and 0.63, respectively; while in the second large group, 2 nodes had posterior probabilities to belong to that group equal to 0.56 and 0.72, respectively.

We discussed in the main text the number of co-authors and degrees in the bipartite graph (i.e. number of co-published papers) of the first small group of authors. We noticed that the 2 authors in this group that had smallest number of co-authors (namely 4) and smallest number of degrees (also 4) are the ones that are ambiguously clustered in this group. While the 2 other authors ambiguously clustered in the second large group have a number of co-authors of 6 and 4,

respectively; and both a degree of 4. This reinforces the conclusion that on this dataset, HyperSBM has grouped apart the authors which are both the most collaborative and the most prolific ones.

Then we ran the spectral clustering algorithm on our dataset. We looked at the spectral gap, that indicated 15 groups but the gap is not clear. Then we looked at the clustering obtained with $Q = 2$ groups. Spectral clustering output groups with sizes 24 and 55, respectively. We recall that spectral clustering tends to output comparable sizes groups. The small group contains the only author with 12 co-authors and the remaining authors have a number of co-authors ranging from 1 to 4. The second large group has a distribution of the number of co-authors ranging from 1 to 11. The small group contains authors with small degree in the bipartite graph, i.e having few co-published papers (all but one author have degrees less 4 and a last author has degree 7), while the second large group contains the 3 authors with largest degree, the rest of the authors having degrees ranging from 1 to 6. Thus, these groups are neither characterized by the number of co-authors nor by their degrees in the bipartite graph.

Finally, we analyzed the same dataset as a bipartite graph under a Bipartite SBM. We relied on the `R` package `SBM` through the function `estimateBipartiteSBM` (Chiquet et al., 2022).

The `Bipartite-SBM` also selected 2 groups of authors (and one group of papers). There was one small group with 4 authors, which are exactly the ones that have the highest degree in the bipartite graph and also correspond to the 4 authors having the highest number of co-authors.

Here, 2 nodes could be considered as ambiguously classified: one node from the first small (resp. second large) group had posterior probability to belong to that group of 0.73 only (resp. 0.67 only). These 2 nodes where not ambiguously classified by HyperSBM and both appeared in our first small group.

It is interesting to compare the situation of three particular authors here. Author with index 48 has 7 co-authors (the 6th highest) and 6 co-authored papers (the 5th highest). It is outside the small first group with `Bipartite-SBM` method (posterior probability $1 - 0.67 = 0.33$ to belong to that group); while `HyperSBM` clusters it unambiguously in the first small group. Similarly, author with index 27 has 12 coauthors (1st highest) and only 7 co-authored papers (the 4th highest). This node was ambiguously classified by `Bipartite-SBM` method in the first small group (posterior probability 0.73 only); while `HyperSBM` clusters it unambiguously in the first small group. Now, conversely, author with index 35 has 8 co-authors (the 6th highest) and 5 co-authored papers (also the 5th highest). This author is unambiguously clustered from the two methods; but while `HyperSBM` puts it in the first small graph, `Bipartite-SBM` excludes it from that group. The examination of these 3 particular tangent cases seem to show that on this dataset, `Bipartite-SBM` was more sensible to authors's degrees in the bipartite graph while `HyperSBM` paid more attention to the sizes of the hyperedges (i.e. number of co-authors) an author was

involved in

We also looked at estimated connection probabilities in the bipartite SBM. The authors from the first small group of `Bipartite-SBM` have many papers (estimated connection probability with the unique group pf papers in the bipartite graph is 11.5% whereas only 2.5% for the other large group). Finally, we computed the parameters values $B^{(m)}_{q_1,\dots,q_m}$ obtained with the groups estimated by `Bipartite-SBM`. We obtained with $m = 2$ that $\hat{B}^{(2)}_{11} \simeq 16,6\%$ (to be compared with 4.2% in `HyperSBM`); while $\hat{B}^{(2)}_{12} \simeq 7\%$ and $\hat{B}^{(2)}_{22} \simeq 1\%$ (more similar to the results of `HyperSBM`, which are 5.1% and 0.8%, respectively). In this case, the first group of authors behaves differently with respect to intra-group connections compared to outer-group connections.

References for Supplementary Material

Allaire, J., R. Francois, K. Ushey, G. Vandenbrouck, G. M., and Intel (2022). *`RcppParallel`: Parallel Programming Tools for `Rcpp`*. R package version 5.1.5.

Allman, E., C. Matias, and J. Rhodes (2009). Identifiability of parameters in latent structure models with many observed variables. *Ann. Statist. 37*(6A), 3099–3132.

Allman, E., C. Matias, and J. Rhodes (2011). Parameters identifiability in a class of random graph mixture models. *Journal of Statistical Planning and Inference 141*, 1719–1736.

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

Behrens, S., C. Erbes, M. Ferrara, S. G. Hartke, B. Reiniger, H. Spinoza, and C. Tomlinson (2013). New results on degree sequences of uniform hypergraphs. *Electron. J. Comb. 20*(4), research paper p14, 18.

Chiquet, J., S. Donnet, großBM team, and P. Barbillon (2022). *`sbm`: Stochastic Blockmodels*. R package version 0.4.4.

Chodrow, P. S. (2020). Configuration models of random hypergraphs. *Journal of Complex Networks 8*(3), cnaa018.

Eddelbuettel, D. (2013). *Seamless `R` and C++ Integration with `Rcpp`*. New York: Springer.

Eddelbuettel, D. and R. François (2011). `Rcpp`: Seamless `R` and C++ integration. *Journal of Statistical Software 40*(8), 1–18.

Eddelbuettel, D. and C. Sanderson (2014, March). `RcppArmadillo`: Accelerating `R` with high-performance C++ linear algebra. *Computational Statistics and Data Analysis 71*, 1054–1063.

Guillaume, J.-L. and M. Latapy (2004). Bipartite structure of all complex networks. *Information Processing Letters 90*(5), 215–221.

Kruskal, J. (1977). Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra and Appl. 18*(2), 95–138.