Latent Space Modelling of Hypergraph Data

Kathryn Turnbull[1], Simón Lunagómez[2], Christopher Nemeth[2], and Edoardo Airoldi[3]

[1]STOR-i Centre for Doctoral Training, Lancaster University
[2]Department of Mathematics and Statistics, Lancaster University
[3]Fox School of Business, Temple University

Abstract

The increasing prevalence of relational data describing interactions among a target population has motivated a wide literature on statistical network analysis. In many applications, interactions may involve more than two members of the population and this data is more appropriately represented by a hypergraph. In this paper, we present a model for hypergraph data which extends the latent space distance model of Hoff et al. (2002) and, by drawing a connection to constructs from computational topology, we develop a model whose likelihood is inexpensive to compute. We obtain posterior samples via an MCMC scheme and we rely on Bookstein coordinates to remove the identifiability issues associated with the latent representation. We demonstrate that the latent space construction imposes desirable properties on the hypergraphs generated in our framework and provides a convenient visualisation of the data. Furthermore, through simulation, we investigate the flexibility of our model and consider estimating predictive distributions. Finally, we explore the application of our model to two real world datasets.

Keywords: Hypergraphs, Latent Space Networks, Simplicial complex, Bayesian Inference, Statistical Network Analysis.

1 Introduction

In this paper we present a model for relational data which describe interactions involving several members of a target population. Our focus is on modelling hypergraphs comprised of N nodes and M hyperedges, where a hyperedge corresponds to a set of nodes, and we assume throughout that hyperedges are modelled randomly given a fixed collection of labelled nodes. A common approach to modelling relationships involving more than two nodes is to project the hypergraph onto a graph in which the connections are assumed to occur between pairs of nodes only. Representing the data as a graph, so that each hyperedge is replaced by a clique, allows the data to be analysed according to an extensive graph modelling literature (see Kolaczyk (2009), Barabási and Pósfai (2016) and Salter-Townshend et al. (2012)) which includes the stochastic blockmodel (Holland et al. (1983)), exponential random graphs (Holland and Leinhardt (1981)), random graph models (Erdös and Rényi (1959), Barabási and Albert (1999)), and latent space network models (Hoff et al. (2002)). However, it is clear that representing a hypergraph as a graph results in a loss of information (see Figures 1a, 1b and 1c) and, although several models for hypergraphs have been introduced (see Stasi et al. (2014), Ng and Murphy (2018), and Liu et al. (2013)), this literature
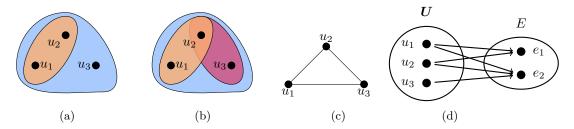
Figure 1: Figures 1a and 1b depict two possible hypergraphs, where a node belongs to a hyperedge if it lies within the associated shaded region. Figure 1c is the graph obtained by replacing hyperedges in Figure 1a, or equivalently in Figure 1b, by cliques. The hypergraphs in 1a and 1b cannot be recovered from 1c. Figure 1d presents the hypergraph relationships in Figure 1a as a bipartite graph, where an edge from a population node to a hyperedge node indicates membership of a hyperedge.

is currently less mature. Here we introduce a model for hypergraph data by considering the extension of the latent space approach for graphs as introduced in Hoff et al. (2002). In this framework the connections are modelled as a function of latent coordinates associated with the nodes, and this construction has many desirable properties which we wish to exploit when developing our model. In particular, the latent representation provides an intuitive visualisation of the graph, allows control in the joint distribution of subgraph counts, and can encourage transitive relationships.

Hypergraph data arise in a range of disciplines (see Kunegis (2013) and Leskovec and Krevl (2014)) including systems biology, neuroscience and marketing, and, depending on the context, the interactions may have different interpretations. For example, an interaction may indicate online communications, professional cooperation between individuals, or dependence between random variables. As a motivating example, consider coauthorship between academics where a connection indicates which authors contributed to an article. We let the nodes represent the population of academics and, since multiple academics may contribute to an article, it is natural to represent a publication by a hyperedge. Figure 1a shows a possible hypergraph relationship between three authors, where a shaded region indicates which authors contributed to an article. Whilst a range of questions can be posed about a hypergraph, we focus on the following.

(Q1) *"Conditional on the observed relationships, how do we expect a new set of new nodes to interact with the hypergraph?"* In the context of coauthorship, this is equivalent to asking who additional authors will collaborate with given the papers that have already been written.

(Q2) *"Which authors have a greater importance in the hypergraph?"* Important authors will not only have high degree but will also tend to have neighbours with high degree. In contrast to this, less important authors will have a smaller impact on the structure of the hypergraph.

(Q3) *"How do we formulate a latent space model for hypergraphs?"*

(Q3) arises naturally from (Q1) and (Q2) and, although there exists a rich latent space network modelling literature (see Krivitsky et al. (2009), Handcock et al. (2007), Friel et al. (2016) and Kim et al. (2017)), the extension to the hypergraph setting is largely unexplored. In the distance model of Hoff et al. (2002) nodes are more likely to be connected if their latent coordinates lie close together in a Euclidean sense. Since the Euclidean distance satisfies the

triangle inequality, transitive relationships in which "friends of friends are likely to be friends" are likely to occur. Properties of this model are well understood (see Rastelli et al. (2015)) and we wish to take advantage of analogous properties in the hypergraph setting. In general, it is unclear how to impose properties on a hypergraph when a bipartite representation is used (see Figure 1d) and so we rely on the representation shown in Figure 1a when developing our model.

The existing literature on hypergraph analysis covers many generalisations of well-studied network models. For example, Ghoshdastidar and Dukkipati (2014) extend the stochastic block-model (SBM) of Holland et al. (1983) to the k-uniform SBM where all hyperedges are of order k, Liu et al. (2013) consider the preferential attachment model of Barabási and Albert (1999) in the hypergraph setting, and Stasi et al. (2014) build upon the β-model of Holland and Lein-hardt (1981) to introduce an analogous model for hypergraphs. Several authors have considered community detection for hypergraphs, where Zhou et al. (2006) develop spectral clustering for hypergraphs and Ke et al. (2019) introduce a tensor decomposition based approach to community detection. Related work is presented in Aksoy et al. (2016), who focus on modelling commu-nity structure in bipartite graphs, and Ng and Murphy (2018), who develop a model to capture clustering in the hyperedges by extending methodology from latent class analysis. Although the latent space approach of Hoff et al. (2002) has yet to be extended to the hypergraph representa-tion in Figure 1a, Friel et al. (2016) have developed a latent space model for temporally evolving bipartite graphs, where the authors examine company directors and boards they associate with. An alternative class of network models in which the edges are assumed exchangeable have re-cently been introduced in the literature, and several authors have introduced models that are able to express interactions of arbitrary order in this framework. For example, see Campbell et al. (2018), Crane and Dempsey (2018) and Dempsey et al. (2019). Finally, in the probability literature, properties of simplicial hypergraphs are considered in Kahle (2016). Simplicial hy-pergraphs are a particular class of hypergraphs in which the presence of a hyperedge indicates the presence of all subsets of the hyperedge. This class of hypergraphs can be seen as a special case of a more general construction termed a simplicial complex, which appear more broadly in the statistics literature. For example, Pronzato et al. (2019) develop methodology for design of experiments and Lunagómez et al. (2017) introduce priors for graphical models using simplicial complexes. We note here that, although our work is related to Lunagómez et al. (2017), we focus on modelling non-simplicial hypergraph data.

The main contributions of this article are as follows. First, using the representation shown in Figure 1a, we develop a latent space model for non-simplicial hypergraph data by extending the distance model of Hoff et al. (2002). Second, by relying on tools from computational geometry (see Edelsbrunner and Harer (2010)), we develop a parsimonious model that is able to express complex data structures, while reducing the computational costs associated with the construction of Hoff et al. (2002). Third, to remove non-identifiability present in our model, we define the latent representation on the space of Bookstein coordinates which have so far not been explored in this context. Fourth, we perform inference on hypergraph data not within the reach of competing methods. Fifth, we investigate the theoretical properties of our model and present a framework for examining the degree distribution. This proves challenging for our model, and our discussion provides an outline which can be explored for other modelling choices.

The rest of this paper is organised as follows. In Section 2 we provide background to our hypergraph model presented in Section 3. Then, in Section 3.5 we discuss identifiability of our hypergraph model and, in Section 5, we describe our procedure for obtaining posterior samples. The simulation studies and real data examples are presented in Section 6 and 7, respectively. Finally, we conclude with a discussion.

2   Background

In this section we will review the network modelling literature relevant to the model for hypergraph data we introduce in Section 3. First, in Section 2.1, we discuss the framework introduced in Hoff et al. (2002), where connection probabilities between pairs of nodes are modelled as a function of a low-dimensional latent space. Then, in Section 2.2, we discuss random geometric graphs (RGGs), where the presence of an edge between a pair of nodes is determined by the intersection of convex sets. Finally, in Section 2.3 we demonstrate how RGGs can be extended to model a restricted class of hypergraphs.

2.1   Latent Space Network Modelling

Latent space models were introduced for network data in Hoff et al. (2002). The key assumption of this framework is that the nodes of a network can be represented in a low-dimensional latent space, and that the probability of an edge forming between each pair of nodes can be modelled as a function of their corresponding latent coordinates. We will first describe a general latent space modelling framework for a network with N nodes. Let $\boldsymbol{Y} = \{y_{ij}\}_{i,j=1,2,\ldots,N}$ denote the observed $(N \times N)$ adjacency matrix, where y_{ij} represents the connection between nodes i and j. For a binary network, we have that $y_{ij} = 1$ if i and j share an edge and $y_{ij} = 0$ otherwise. Additionally, we let $u_i \in \mathbb{R}^d$ represent the d-dimensional latent coordinate of the ith node, for $i = 1, 2, \ldots, N$. The presence of an edge is then given by the following model.

$$
\begin{aligned}
Y_{ij} &\sim \text{Bernoulli}(p_{ij}), \\
p_{ij} = P(y_{ij} = 1 | u_i, u_j, \theta) &= \frac{1}{1 + \exp\{-f(u_i, u_j, \theta)\}},
\end{aligned}
\tag{1}
$$

where θ represents additional model parameters and p_{ij} denotes the probability of an edge forming between nodes i and j. The connection probability depends on a function f that is monotonically decreasing in a measure of similarity between u_i and u_j. As an example, the distance model introduced in Hoff et al. (2002) is obtained by choosing

$$
f(u_i, u_j, \theta) = \alpha - \|u_i - u_j\|,
\tag{2}
$$

where $\|\cdot\|$ is the Euclidean distance, and $\theta = (\alpha)$ represents the base-rate tendency for edges to form. The function f may also be adapted to incorporate covariate information so that nodes which share certain characteristics are more likely to be connected.

The likelihood, conditional on $\boldsymbol{U}$ and θ, is given by

$$
\mathcal{L}(\boldsymbol{U}, \theta; \boldsymbol{Y}) \propto \prod_{i<j} P(y_{ij} = 1 | u_i, u_j, \theta)^{y_{ij}} \left[1 - P(y_{ij} = 1 | u_i, u_j, \theta) \right]^{1-y_{ij}},
\tag{3}
$$

where $\boldsymbol{U}$ is the $(N \times d)$ matrix of latent coordinates such that the ith row of $\boldsymbol{U}$ corresponds to u_i. Properties and modifications of this model are discussed in Rastelli et al. (2015).

2.2   Random Geometric Graphs

In this section we will discuss random geometric graphs (RGGs) which instead model the occurrence of edges as a deterministic function of the latent coordinates. As in the previous section, we will assume that the ith node can be represented by $u_i = (u_{i1}, u_{i2}, \ldots, u_{id}) \in \mathbb{R}^d$. The presence of an edge $\{i, j\}$ is modelled through the intersection of convex sets that are parameterised by u_i and u_j. There are many choices of convex sets, and to generate a RGG we choose the closed ball

in $\mathbb{R}^d$ with centre u_i and radius r. This set is represented $B_r(u_i) = \{u \in \mathbb{R}^d | \ \|u - u_i\| \le r\} = \left\{u \in \mathbb{R}^d \middle| \ \sqrt{\sum_{j=1}^d (u_j - u_{ij})^2} \le r\right\}$, and a graph is constructed by connecting each pair of nodes $\{i, j\}$ for which $B_r(u_i) \cap B_r(u_j) \ne \emptyset$. Generating a graph in this way is equivalent to connecting nodes i and j when $\|u_i - u_j\| \le 2r$. An example of this construction is given in the left and middle panel of Figure 2.

We now express the likelihood for this model as a function of the latent coordinates, keeping the notation from Section 2.1. The likelihood conditional on $\boldsymbol{U}$ and r is given by

$$\mathcal{L}(\boldsymbol{U}, r; \boldsymbol{Y}) \propto \prod_{i<j} \mathbb{1}(\|u_i - u_j\| \le 2r)^{y_{ij}} \left[1 - \mathbb{1}(\|u_i - u_j\| \le 2r)\right]^{1-y_{ij}}. \tag{4}$$

By comparing (3) and (4), we see that a RGG can be viewed as a latent space network model where the probability of a connection is given by $P(y_{ij} = 1|u_i, u_j, \theta) = \mathbb{1}(\|u_i - u_j\| \le 2r)$ where $\theta = r$. Therefore, conditional on a set of latent coordinates, a RGG is deterministic. Note that (4) is equal to 1 if there is a perfect correspondence between the observed connections $\boldsymbol{Y}$ and the connections induced by the latent coordinates $\boldsymbol{U}$ and the radius r. If there are any connections which do not correspond to each other, then (4) is equal to 0. For an in-depth discussion of RGGs see Penrose (2003).

2.3   Random Geometric Hypergraphs

The graph generating procedure described in Section 2.2 assumed that edges occur between pairs of nodes. We can generalise this framework to model hypergraphs by considering the full intersection pattern of convex sets, and we refer to these hypergraphs as random geometric hypergraphs (RGHs). In order to do this, we introduce the concept of a nerve (Section 3.2 of Edelsbrunner and Harer (2010)) which represents the set of indices for which their corresponding regions have a non-empty intersection, and it is given in Definition 2.1.

Definition 2.1. *(Nerve) Let $\mathcal{A} = \{A_i\}_{i=1}^N$ represent a collection of non-empty sets. The nerve of $\mathcal{A}$ is given by*

$$Nrv(\mathcal{A}) = \left\{\sigma \subseteq \{1, 2, \dots, N\} \middle| \underset{j \in \sigma}{\cap} A_j \ne \emptyset\right\}. \tag{5}$$

Note that the sets $\{1\}, \{2\}, \dots, \{N\}$ are included in $Nrv(\mathcal{A})$ and that $|\sigma| \le N$ for $\sigma \in Nrv(\mathcal{A})$, where $|\sigma|$ is the order, or dimension, of the set. For the rest of this discussion we will focus on the case where A_j is convex.

It is clear that the nerve defines a hypergraph where $\sigma \in Nrv(\mathcal{A})$ denotes a hyperedge. Consider the sets $\sigma_1 \in Nrv(\mathcal{A})$ and $\sigma_2 \subset \sigma_1$. It follows immediately that $\sigma_2 \in Nrv(\mathcal{A})$, and all hypergraphs generated by a nerve must have this property. Hypergraphs of this type are termed simplicial, and Kahle (2016) overview properties of simplicial random hypergraphs along with more general constructions.

In Section 2.2, we considered the choice $A_i = B_r(u_i)$ for generating a RGG. The nerve for this choice of $\mathcal{A}$ is well studied and it is referred to as the Čech complex (see Section 3.2 of Edelsbrunner and Harer (2010)), as given in Definition 2.2.

Definition 2.2. *(Čech Complex) For a set of coordinates $\boldsymbol{U} = \{u_i\}_{i=1}^N$ and a radius r, the Čech complex $\mathcal{C}_r$ is given by*

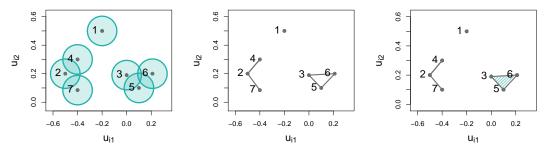$$\mathcal{C}_r = Nrv\left(\{B_r(u_i)\}_{i=1}^N\right). \tag{6}$$

Figure 2: Example of a Čech complex. Left: $B_r(u_i)$ for $\{u_i = (u_{i1}, u_{i2})\}_{i=1}^7$ in $\mathbb{R}^2$. Middle: the graph obtained by taking pairwise intersections. Right: the hypergraph obtained by taking intersections of arbitrary order. The shaded region between nodes 3, 5 and 6 indicates a hyperedge of order 3.

For this framework to be computationally appealing, it is important that the sets A_i are easy to parameterise and their intersections are efficient to compute. Other well studied examples of complexes are the Delaunay triangulation (Delaunay (1934)) and the Alpha complex (Edelsbrunner et al. (1983)).

We now introduce a subset of the Čech complex known as the k-skeleton, which is given in Definition 2.3. This will be revisited in Section 3.2.

Definition 2.3. *(k-skeleton of the Čech complex) Let $\mathcal{C}_r$ denote the Čech complex, as given in Definition 2.2. The k-skeleton of $\mathcal{C}_r$ is given by*

$$\mathcal{C}_r^{(k)} = \{\sigma \in \mathcal{C}_r | |\sigma| \leq k\}. \tag{7}$$

$\mathcal{C}_r^{(k)}$ represents the collection of sets in $\mathcal{C}_r$ which are of order that is less than or equal to k. Note that the k-skeleton can also be defined more generally for any nerve.

Example 2.1. *Figure 2 depicts an example of a Čech complex, where $\mathcal{C}_r = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{2,4\}, \{3,5\}, \{3,6\}, \{5,6\}, \{3,5,6\}\}$. The k-skeletons are given by $\mathcal{C}_r^{(1)} = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}\}$, $\mathcal{C}_r^{(2)} = \mathcal{C}_r^{(1)} \cup \{\{2,4\}, \{3,5\}, \{3,6\}, \{5,6\}\}$ and $\mathcal{C}_r^{(3)} = \mathcal{C}_r^{(2)} \cup \{3,5,6\}$.*

3 Latent space hypergraphs

In this section we introduce a model for hypergraph data which builds upon the models discussed in Section 2. Our model will balance the computational aspects of latent space network modelling (Section 2.1) with the approach of RGGs (Section 2.2) and RGHs (Section 2.3). The aims of our modelling framework are given in Section 3.1, notation and set-up are given in Section 3.2, and the generative model and likelihood are given in Section 3.3. Finally, in Section 3.4 we consider of the model in Section 3.3.

3.1 Motivation

Recall the motivating example of coauthorship, where hyperedges indicate the set of authors that have contributed to a given paper. In this example, we may observe relationships such as those depicted in Figures 1a and 1b. Therefore, the model discussed in Section 2.3 is likely
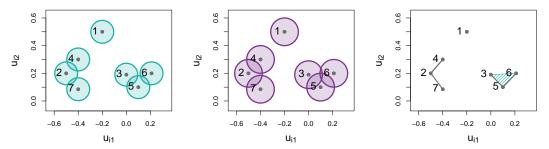
Figure 3: Example of a Čech complex. Left: $B_{r_2}(u)$ for each of 6 points in $\mathbb{R}^2$. Middle: $B_{r_3}(u)$ for each of 6 points in $\mathbb{R}^2$. Right: $\cup_{k=2}^3 \mathcal{D}_{r_k}^{(k)}$.

too restrictive and this motivates us to develop a model that is appropriate for non-simplicial hypergraphs.

In this section we will build upon the graph and hypergraph models introduced in Section 2, where our model for a hypergraph on N nodes has a maximal hyperedge order K ($2 \leq K \leq N$), with hyperedge $e \subseteq \{1, 2, \ldots, N\}$ present if $y_e = 1$, otherwise $y_e = 0$. Where necessary, we will denote a hyperedge of order k by e_k so that $|e_k| = k$. Our model for hypergraph data, which represents the nodes of the network in a low-dimensional space, is able to express a broad class of hypergraphs. Furthermore, our proposed model has (i) a convenient and computationally-efficient likelihood with (ii) a simple to describe support for non-simplicial hypergraph data.

3.2   Combining k-skeletons

To extend the model in Section 2.3 to express non-simplicial hypergraphs, we introduce additional radii. As an example, consider two radii which we denote by r_2 and r_3. For the same set of latent coordinates $\boldsymbol{U}$, each of these radii will give rise to Čech complex which we denote by $\mathcal{C}_{r_2}$ and $\mathcal{C}_{r_3}$, respectively. By varying r_2 and r_3 we are able to control the edges that are present in each of $\mathcal{C}_{r_2}$ and $\mathcal{C}_{r_3}$. Now suppose for each of these complexes we only consider hyperedges of a specific order. We can, for example, construct a hypergraph by taking the union of the order 2 hyperedges in $\mathcal{C}_{r_2}$ and the order 3 hyperedges in $\mathcal{C}_{r_3}$. This construction removes the simplicial constraint on the hypergraphs, and an example of this is given in Figure 3. We refer to hypergraphs constructed in this way as non-simplicial random geometric hypergraphs (nsRGH), and Definition 3.1 details this construction for a hypergraph on N nodes with maximal hyperedge order K.

Definition 3.1. *(non-simplicial RGH) Consider $\boldsymbol{r} = (r_2, r_3, \ldots, r_K)$ which satisfy $r_k > r_{k-1}$ for $k = 3, 4, \ldots, K$. We define the non-simplicial nsRGH on N nodes as the hypergraph with hyperedges given by $\cup_{k=2}^K \mathcal{D}_{r_k}^{(k)}$, where $\mathcal{D}_{r_k}^{(k)} = \mathcal{C}_{r_k}^{(k)} \setminus \mathcal{C}_{r_k}^{(k-1)}$ denotes the hyperedges of exactly order k in the Čech complex with radius r_k, and $\mathcal{C}_{r_k}^{(k)}$ is as in Definition 2.3.*

Example 3.1. *Consider the non-simplicial hypergraph shown in the right panel of Figure 3. We have $\mathcal{D}_{r_2}^{(2)} = \{\{2, 4\}, \{3, 5\}, \{5, 6\}\}$ and $\mathcal{D}_{r_3}^{(3)} = \{3, 5, 6\}$.*

In Definition 3.1, constraints are imposed on the radii $\boldsymbol{r} = (r_2, r_3, \ldots, r_K)$ to ensure that the generated hypergraphs are non-simplicial. For example, if $r_3 < r_2$ and the hyperedge $\{i, j, k\}$ is present in the hypergraph, then it follows that the hyperedges $\{i, j\}, \{i, k\}$ and $\{j, k\}$ must also be in the hypergraph. We further note that r_k is not explicitly modelled as a function of

---

Algorithm 1 Sample a hypergraph $g_{N,K}^*$ given $N, K, \boldsymbol{r}, \boldsymbol{\varphi}, \mu$ and Σ.

Sample $\boldsymbol{U} = \{u_i\}_{i=1}^N$ such that $u_i \overset{iid}{\sim} \mathcal{N}(\mu, \Sigma)$, for $i = 1, 2, \ldots, N$.

For $k = 2, 3, \ldots, K$,

    a) Given $\boldsymbol{U}$ and r_k, check which $e_k = \{i_1, i_2, \ldots, i_k\} \in \mathcal{E}_{N,k}$ satisfy $y_{e_k}^{(g)} = 1$.

        To determine if $y_{e_k}^{(g)} = 1$, check that $\cap_{l=1}^k B_{r_k}(u_{i_l}) \neq \emptyset$.

    b) For all $e_k \in \mathcal{E}_{N,k}$, sample $S_k \sim \text{Bernoulli}(\varphi_k)$.

        Let $y_{e_k}^{(g^*)} = \left(y_{e_k}^{(g)} + s_k \right) \mod 2$

---

k, however the constraint $r_k > r_{k-1}$ implies that hyperedges of larger order will have a larger associated radii.

3.3 Generative Model and Likelihood

The procedure of generating non-simplicial hypergraphs given in Definition 3.1 is deterministic conditional on $\boldsymbol{U}$. Therefore, similarly to the RGG in Section 2.2, the likelihood of an observed hypergraphs will be one only when there is a perfect correspondence between the observed and generated hyperedges. Furthermore, it is not straightforward to characterise the space of hypergraphs that can be expressed via the nsRGH procedure. We address these issues in this section by considering a modification of Definition 3.1.

Let $\mathcal{G}_{N,K}$ denote the space of hypergraphs on N nodes with maximum hyperedge order K. We write $\mathcal{G}_{N,K} = (\mathcal{V}_N, \mathcal{E}_{N,K})$, where $\mathcal{V}_N = \{1, 2, \ldots, N\}$ denotes the node labels and $\mathcal{E}_{N,K}$ denotes the set of possible hyperedges up to order K on N nodes. Let $\mathcal{E}_{N,k}$ represent the possible hyperedges of exactly order k on N nodes so that $\mathcal{E}_{N,K} = \cup_{k=2}^K \mathcal{E}_{N,k}$. Let $\varphi_k \in [0, 1]$ denote the probability of modifying the state of a hyperedge of order k, for $k = 2, 3, \ldots, K$. Then, for $e_k \in \mathcal{E}_{N,k}$, we sample a variable $S_k \sim \text{Bernoulli}(\varphi_k)$ and let

$$y_{e_k}^{(g^*)} = \begin{cases} 0, & \text{if } s_k = 1 \text{ and } y_{e_k}^{(g)} = 1 \\ 1, & \text{if } s_k = 1 \text{ and } y_{e_k}^{(g)} = 0 \\ y_{e_k}, & \text{if } s_k = 0 \end{cases} \tag{8}$$

where $g_{N,K}(\boldsymbol{U}, \boldsymbol{r})$ denotes the nsRGH induced from $\boldsymbol{U}$ and $\boldsymbol{r}$ as in Definition 3.1, and $y_{e_k^{(g)}}$ denotes the state of the hyperedge e_k in $g_{N,K}(\boldsymbol{U}, \boldsymbol{r})$. We let $g_{N,K}^*(\boldsymbol{U}, \boldsymbol{r})$ denote the hypergraph sampled from our model. From (8), we see that φ_k controls the amount of modification of the hyperedges of order k.

The final aspect of the generative model is to assign a probability distribution on the latent coordinates $\boldsymbol{U}$, and we assume that $u_i \overset{iid}{\sim} \mathcal{N}(\mu, \Sigma)$, for $i = 1, 2, \ldots, N$. This reflects the intuition that nodes placed more centrally in the latent representation are likely to share the greatest number of connections. A hypergraph can now be generated by the procedure given in Algorithm 1.

We can express the likelihood of an observed hypergraph $h_{N,K} \in \mathcal{G}_{N,K}$, conditional on $\boldsymbol{U}, \boldsymbol{r}$ and $\boldsymbol{\varphi}$, by considering the discrepancy between the hyperedge configurations in $h_{N,K}$ and $g_{N,K}(\boldsymbol{U}, \boldsymbol{r})$. For $k = 2, 3, \ldots, K$, let

$$d_k(g_{N,K}(\boldsymbol{U}, \boldsymbol{r}), h_{N,K}) = \sum_{e_k \in \mathcal{E}_{N,k}} |y_{e_k}^{(g)} - y_{e_k}^{(h)}| \tag{9}$$

denote the distance between the order k hyperedges in $g_{N,K}(\boldsymbol{U}, \boldsymbol{r})$ and $h_{N,K}$, where $y_{e_k}^{(g)}$ and $y_{e_k}^{(h)}$ represent an order k hyperedge in $g_{N,K}(\boldsymbol{U}, \boldsymbol{r})$ and $h_{N,K}$, respectively. Note that no modifications have been made to the hypergraph $g_{N,K}(\boldsymbol{U}, \boldsymbol{r})$. The measure of distance (9) corresponds to the number of hyperedges which differ between $h_{N,K}$ and $g_{N,K}(\boldsymbol{U}, \boldsymbol{r})$. This is equivalent to the Hamming distance and is related related to the l_1 norm and the exclusive or (XOR) operator. Evaluating this distance does not require the $\sum_{k=2}^{K} \binom{N}{k}$ computations suggested by (9), and we can instead evaluate the discrepancy by only considering hyperedges that are present in $g_{N,K}(\boldsymbol{U}, \boldsymbol{r})$ and $h_{N,K}$. In practice this is likely to be far less than the number of possible hyperedges and details of this calculation are discussed in Appendix E.2.

Given this notion of hypergraph distance the likelihood of observing $h_{N,K}$, conditional on $\boldsymbol{U}, \boldsymbol{r}$ and $\boldsymbol{\varphi}$, can be written as

$$\mathcal{L}(\boldsymbol{U}, \boldsymbol{r}, \boldsymbol{\varphi}; h_{N,K}) \propto \prod_{k=2}^{K} \varphi_k^{d_k(g_{N,K}(\boldsymbol{U}, \boldsymbol{r}), h_{N,K})} (1 - \varphi_k)^{\binom{N}{k} - d_k(g_{N,K}(\boldsymbol{U}, \boldsymbol{r}), h_{N,K})}. \tag{10}$$

We obtain (10) by considering which hyperedges in $g_{N,K}(\boldsymbol{U}, \boldsymbol{r})$ must have their state modified to match the hyperedges in $h_{N,K}$, and which hyperedges are the same as in $h_{N,K}$. For order k hyperedges which differ, the probability of switching the hyperedge state is given by φ_k. Since our likelihood is of the same form as Lunagómez et al. (2019, proof of Proposition 3.1), it follows that hypergraphs with a greater number of hyperedge modification are less likely for $0 < \varphi_k < 1/2$ and so (10) behaves in an intuitive way.

The model specification is complete with the following priors for $k = 2, 3, \ldots, K$.

$$\mu \sim \mathcal{N}(m_\mu, \Sigma_\mu) \quad \Sigma_\mu \sim \mathcal{W}^{-1}(\Phi, \nu), \quad r_k \sim \exp(\lambda_k), \quad \text{and} \quad \varphi_k \sim \text{Beta}(a_k, b_k), \tag{11}$$

where the priors in (11) are chosen for computational convenience.

3.4 Can we improved model flexibility?

In Algorithm 1 the number of order k edges is controlled by varying the parameter r_k. However, the constraint $r_{k+1} > r_k$, for $k = 2, 4, \ldots, K - 1$, implies r_k will impact the higher order hyperedges. This may limit the types of hypergraphs that can be expressed.

We improve the model flexibility by introducing an additional modification parameter for each hyperedge order. In Algorithm 1 the noise φ_k is applied independently across all hyperedges of order k. Alternatively, we can modify each hyperedge depending on its state in $g_{N,K}(\boldsymbol{U}, \boldsymbol{r})$. For $k = 2, 3, \ldots, K$, let $\boldsymbol{\psi}^{(0)} = (\psi_2^{(0)}, \psi_3^{(0)}, \ldots, \psi_K^{(0)}) \in [0, 1]$ denote the probability of modifying the state of a hyperedge in $g_{N,K}(\boldsymbol{U}, \boldsymbol{r})$ from absent to present, and let $\boldsymbol{\psi}^{(1)} = (\psi_2^{(1)}, \psi_3^{(1)}, \ldots, \psi_K^{(1)}) \in [0, 1]$ denote the probability of modifying the state of a hyperedge in $g_{N,K}(\boldsymbol{U}, \boldsymbol{r})$ from present to absent. Suppose our observed hypergraph suggests that there are many hyperedges of order 2 and few hyperedges of order 3. By increasing the modification noise $\psi_3^{(1)}$ we can control additional hyperedges that may appear in the hypergraph due to the constraint $r_3 > r_2$. This generative model is summarised in Algorithm 2, and as commented in Section 3.3, Algorithm 2 can be implemented without the suggested $\sum_{k=2}^{K} \binom{N}{k}$ computations. Further details on the procedure for checking which hyperedges are present in $g_{N,K}(\boldsymbol{U}, \boldsymbol{r})$ are discussed in Appendix E.1 with details of the hyperedge modifications given in Appendix B.

As in Section 3.3, the likelihood of observing $h_{N,K}$ is based on a distance metric,

$$d_k^{(ab)}(g_{N,K}(\boldsymbol{U}, \boldsymbol{r}), h_{N,K}) = \#\{e_k \in \mathcal{E}_{N,k} | y_{e_k}^{(g)} = a \cap y_{e_k}^{(h)} = b\}, \tag{12}$$

9

Algorithm 2 Sample a hypergraph $g_{N,K}^*$ given $N, K, \boldsymbol{r}, \boldsymbol{\psi}^{(0)}, \boldsymbol{\psi}^{(1)}, \mu$ and Σ.

Sample $\boldsymbol{U} = \{u_i\}_{i=1}^N$ such that $u_i \stackrel{iid}{\sim} \mathcal{N}(\mu, \Sigma)$, for $i = 1, 2, \ldots, N$.
For $k = 2, 3, \ldots, K$,
    a) Given $\boldsymbol{U}$ and r_k, check which $e_k = \{i_1, i_2, \ldots, i_k\} \in \mathcal{E}_{N,k}$ satisfy $y_{e_k} = 1$.
       To determine if $y_{e_k} = 1$, that $\cap_{l=1}^k B_{r_k}(u_{i_l}) \neq \emptyset$.
    b) For all $e_k \in \mathcal{E}_{N,k}$
       If $y_{e_k}^{(g)} = 1$, set $y_{e_k}^{(g^*)} = 0$ with probability $\psi_k^{(1)}$.
       If $y_{e_k}^{(g)} = 0$, set $y_{e_k}^{(g^*)} = 1$ with probability $\psi_k^{(0)}$.

---

which records the number of hyperedges that have state $a \in \{0, 1\}$ in $g_{N,K}(\boldsymbol{U}, \boldsymbol{r})$ and state $b \in \{0, 1\}$ in $h_{N,K}$. For example, $d_k^{(01)}(g_{N,K}(\boldsymbol{U}, \boldsymbol{r}), h_{N,K})$ represents the number of hyperedges absent in $g_{N,K}(\boldsymbol{U}, \boldsymbol{r})$ and present in $h_{N,K}$. Efficient evaluation of (12) is discussed in Appendix (E.2) and the likelihood conditional on $\boldsymbol{U}, \boldsymbol{r}, \boldsymbol{\psi}^{(1)}$ and $\boldsymbol{\psi}^{(0)}$ is given by

$$\mathcal{L}\left(\boldsymbol{U}, \boldsymbol{r}, \boldsymbol{\psi}^{(1)}, \boldsymbol{\psi}^{(0)}; h_{N,K}\right) \propto \prod_{k=2}^K \left[\left(\psi_k^{(1)}\right)^{d_k^{(10)}(g_{N,K}(\boldsymbol{U},\boldsymbol{r}),h_{N,K})} \left(1 - \psi_k^{(1)}\right)^{d_k^{(11)}(g_{N,K}(\boldsymbol{U},\boldsymbol{r}),h_{N,K})} \right.$$
$$\left. \times \left(\psi_k^{(0)}\right)^{d_k^{(01)}(g_{N,K}(\boldsymbol{U},\boldsymbol{r}),h_{N,K})} \left(1 - \psi_k^{(0)}\right)^{d_k^{(00)}(g_{N,K}(\boldsymbol{U},\boldsymbol{r}),h_{N,K})} \right]. \tag{13}$$

We obtain (13) in a similar way to (10), where we distinguish between hyperedges that are present and absent in the induced hypergraph. Note that (13) is equivalent to (10) when $\psi_k^{(1)} = \psi_k^{(0)}$, for $k = 2, 3, \ldots, K$.

The model specification is complete with the following prior distributions, for $k = 2, 3, \ldots, K$,

$$\mu \sim \mathcal{N}(m_\mu, \Sigma_\mu), \qquad \Sigma \sim \mathcal{W}^{-1}(\Phi, \nu), \qquad r_k \sim \exp(\lambda_k),$$
$$\psi_k^{(0)} \sim \text{Beta}\left(a_k^{(0)}, b_k^{(0)}\right), \quad \text{and} \quad \psi_k^{(1)} \sim \text{Beta}\left(a_k^{(1)}, b_k^{(1)}\right). \tag{14}$$

where priors in (14) are chosen for computational convenience.

3.5 Identifiability

Let $g_{N,K}(\boldsymbol{U}, \boldsymbol{r})$ denote the nsRGH obtained from $\boldsymbol{U}$ and $\boldsymbol{r}$, and let $g_{N,K}^*$ be the hypergraph obtained by modifying the hyperedges in $g_{N,K}(\boldsymbol{U}, \boldsymbol{r})$ according to $\boldsymbol{\varphi}$ (see Algorithm 1). By conditioning on $g_{N,K}(\boldsymbol{U}, \boldsymbol{r})$, we can decompose the conditional distribution for $g_{N,K}^*$ in the following way.

$$p(g_{N,K}^*, g_{N,K}(\boldsymbol{U}, \boldsymbol{r}) | \mu, \Sigma, \boldsymbol{\varphi}, \boldsymbol{r}) = p(g_{N,K}^* | g_{N,K}(\boldsymbol{U}, \boldsymbol{r}), \boldsymbol{\varphi}) p(g_{N,K}(\boldsymbol{U}, \boldsymbol{r}) | \mu, \Sigma, \boldsymbol{r}). \tag{15}$$

An equivalent decomposition for the model outlined in Algorithm 2 can also be expressed.

The probability of occurrence of a hyperedge in $g_{N,K}(\boldsymbol{U}, \boldsymbol{r})$ is a function of the distances between the latent coordinates $\boldsymbol{U}$. Therefore the conditional distribution $p(g_{N,K}(\boldsymbol{U}, \boldsymbol{r}) | \mu, \Sigma, \boldsymbol{r})$ is invariant to distance-preserving transformations of $\boldsymbol{U}$ and will exhibit multimodality. Additionally, we observe that scaling $\boldsymbol{U}$ and $\boldsymbol{r}$ by the same factor results in a source of model non-identifiability.

To remove these sources of non-identifiability, we define $\boldsymbol{U}$ on the Bookstein space of coordinates (see Bookstein (1986) and Section 2.3.3 of Dryden and Mardia (1998)). Bookstein

coordinates define a translation, rotation and re-scaling of the points $\boldsymbol{U}$ with respect to a set of anchor points and, since these anchor points remain fixed throughout model fitting, the radii $\boldsymbol{r}$ are also appropriately re-scaled. For details of the Bookstein transformation see Appendix A.

In the latent space network modelling literature, it is typical to use Procrustes analysis (Section 5 of Dryden and Mardia (1998)) as a post-processing step to remove the effect of distance preserving transformations of $\boldsymbol{U}$. Due to the non-identifiability associated with scaling $\boldsymbol{U}$ and $\boldsymbol{r}$, we note that this approach is not sufficient for removing all sources of non-identifiability in our model.

From (15), we see that hyperedges can either arise from $g_{N,K}(\boldsymbol{U}, \boldsymbol{r})$ or the hyperedge modification. To maintain the properties imposed on the hypergraph from the construction of $g_{N,K}(\boldsymbol{U}, \boldsymbol{r})$, we wish to keep the parameters $\boldsymbol{\varphi}$ relatively small. However, when generating sparse hypergraphs from our model, it will become increasingly difficult to distinguish between these competing sources of hyperedges. Therefore we will observe model non-identifiability in the sparse regime.

4 Theoretical Results

In this section we study the behaviour of the node degree in the hypergraph model detailed in Algorithm 1. We consider the probability of a hyperedge occurring in a nsRGH and we present our results for the degree distribution in Section 4.2. We note here that the nodes in our hypergraph model are exchangeable, and therefore we can focus on the node degree properties of any node i. We comment here that it is straightforward to extend the results in this section to the model detailed in Algorithm 2. To begin, we make explicit the following properties of our hypergraph model.

(P1) A hypergraph generated from our model is a modification of a nsRGH.

(P2) Conditional on $\boldsymbol{U}$ and $\boldsymbol{r}$, hyperedges of each order occur independently in our model.

Additionally, to obtain our results, we make the following assumptions.

(A1) The number of nodes N and maximum hyperedge order K are fixed.

(A2) The covariance of the latent coordinates Σ is diagonal, where $\Sigma_{ll} = \sigma_{ll}^2$ for $l = 1, 2, \ldots, d$ and 0 otherwise.

We let $g(\boldsymbol{U}, \boldsymbol{r}) = g \in \mathcal{G}_{N,K}$ denote the nsRGH generated from the coordinates $\boldsymbol{U}$ and radii $\boldsymbol{r}$. A hypergraph is generated from our model by modifying g with noise $\boldsymbol{\varphi}$, and we denote this hypergraph $g^* \in \mathcal{G}_{N,K}$. Additionally, we denote the degree of order k hyperedges and the overall degree of the ith node in g by $\mathrm{Deg}_{(i,k)}^g = \sum_{\{e_k \in \mathcal{E}_{N,k} | i \in e_k\}} y_{e_k}^{(g)}$ and $\mathrm{Deg}_{(i)}^g = \sum_{k=2}^{K} \mathrm{Deg}_{(i,k)}$, respectively. Note that the decomposition of $h_{N,K}$ in (15) follows from Property (P1) and Assumption (A2) is not restrictive since for any normally distributed set of points in $\mathbb{R}^d$ we can apply a distance-preserving transformation which maps the covariance matrix onto a diagonal matrix.

4.1 Properties of a nsRGH

We are interested in the probability of an order k hyperedge occurring in a nsRGH generated from $\boldsymbol{U}$ and $\boldsymbol{r}$, and we denote this by $p_{e_k} = P(y_{e_k}^{(g)} = 1 | \mu, \Sigma, r_k)$. We consider two settings for the order of hyperedges, $k = 2$ and $k \geq 3$ and present these separately.
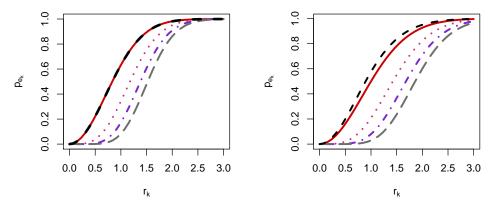
Figure 4: Estimate of probability of a hyperedge occurring for $k = 2$ (red, solid), $k = 3$ (pink, dot), $k = 4$ (purple, dash-dot), $k = 5$ (grey, dash) as a function of r_k. Points were simulated from a Normal distribution with $\mu = (0, 0)$, and $\Sigma = I_2$ in (left panel) and $\Sigma = \left(\begin{smallmatrix} 2 & 0 \\ 0 & 1 \end{smallmatrix} \right)$ in (right panel), and theoretical probability for $k = 2$ (black, dash).

4.1.1 Connection Probabilities for $k = 2$

An edge $e_2 = \{i, j\}$ is present in $g(\boldsymbol{U}, \boldsymbol{r})$ if $B_{r_2}(u_i) \cap B_{r_2}(u_j) \neq \emptyset$. Hence, the occurrence probability p_{e_2} is determined by the Euclidean distance between u_i and u_j. This probability is given in Proposition 4.1 and follows by considering the distribution of a squared Normal random variable.

Proposition 4.1. *Let $U_i \sim \mathcal{N}(\mu, \Sigma)$, for $i = 1, 2, \ldots, N$, and $\Sigma = diag(\sigma_1^2, \sigma_2^2, \ldots, \sigma_d^2)$. The probability of an edge $e_2 = \{i, j\}$ occurring in $g(\boldsymbol{U}, \boldsymbol{r})$ is given by*

$$p_{e_2} = P(\|U_i - U_j\| \leq 2r_2 | \Sigma) = \int_0^{(2r_2)^2} \sum_{l=1}^d f\left(z; \frac{1}{2}, 4\sigma_l^2\right) dz, \tag{16}$$
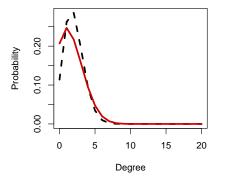
where $f(z; a, b) = \dfrac{b^a}{\Gamma(a)} z^{a-1} e^{-bz}$ is the pdf of a $\Gamma(a, b)$ random variable.

Proof. See Appendix F.1. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \square$

In Figure 4 we plot an empirical estimate of p_{e_2} against the theoretical result from Proposition 4.1. The results show a strong agreement between the theory and simulated results. Next, we will consider the connection probability p_{e_k} for $k \geq 3$.

4.1.2 Connection Probabilities for $k \geq 3$

The edge $e_k = \{i_1, i_2, \ldots, i_k\}$ is present in $g(\boldsymbol{U}, \boldsymbol{r})$ if $\cap_{l=1}^k B_{r_k}(u_{i_l}) \neq \emptyset$. This condition is equivalent to the coordinates $\{u_{i_l}\}_{l=1}^k$ lying within a ball of radius r_k (Edelsbrunner and Harer, 2010, see section 3.2). See Appendix E.1 for a visual representation. The connection probability p_{e_k} is then determined by the probability of exactly k points falling within a ball of radius r_k. As we assume that our latent coordinates are normally distributed, the probability of a point
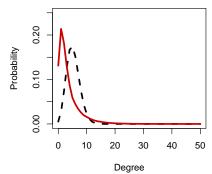
12

Figure 5: Comparison of theoretical (black, dashed) and simulated (red, solid) degree distribution for a hypergraph with $N = 20$ and $K = 3$. The figures show the degree distribution for hyperedges of order $k = 2$ (left panel) and $k = 3$ (right panel), with a Monte Carlo estimate for p_{e_3}.

falling within a ball of radius r_k and centre c is given by

$$P(u \in B_{r_k}(c)|\mu, \Sigma) = \int_{B_{r_k}(c)} p(u|\mu, \Sigma) \, \mathrm{d}u. \tag{17}$$

Solving this integral is intractable for p_{e_k} when $k \geq 3$ (see Gilliland (1962) for details) and so we rely on Monte Carlo approximations.

Figure 4 shows the empirical estimates of p_{e_k} for increasing r_k, where the case $k = 2$ is provided for reference. Points were sampled from $\mathcal{N}(0, \Sigma)$ and the left and right panel show the estimated connection probabilities for $\Sigma = I_2$ and $\Sigma = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$, respectively. The figure shows that a larger radius is required to obtain the same probability of connection as k grows. Additionally, by comparing the left and right panels of Figure 4, we see that, as the elements of Σ increase, the radii r_k must also increase to obtain the same probability of a connection.

4.2   Degree Distribution for the ith Node

We derive the degree distribution for the ith node and present this in Theorem 4.1 for hyperedges of order $k = 2$ and $k = 3$. We also obtain a result given in Lemma 4.1 for the expected degree of the ith node. In Theorem 4.1 we present an exact expression for the degree distribution of order $k = 2$ hyperedges, and rely on an approximation for $k = 3$ hyperedges. We comment on the quality of this approximation below.

Lemma 4.1. *Let p_{e_k} denote the probability of the hyperedge e_k occurring in the nsRGH g, and let g^* be the hypergraph obtained by modifying g with noise $\boldsymbol{\varphi}$. The expected degree of the ith node is given by*

$$\mathbb{E}\left[Deg_{(i)}^{g^*}|\boldsymbol{\varphi}, \Sigma, \boldsymbol{r}\right] = \sum_{k=2}^{K} \binom{N-1}{k-1} \left[(1 - \varphi_k)p_{e_k} + \varphi_k(1 - p_{e_k})\right]. \tag{18}$$

Proof. See Appendix F.2. □

13

Theorem 4.1. *Let g represent the nsRGH generated from* $\boldsymbol{U}$ *and* $\boldsymbol{r}$, *and let* g^* *be the hypergraph obtained by modifying g with noise* $\boldsymbol{\varphi}$. *It follows that*

1. *For k=2:*

$$Deg^{g^*}_{(i,2)}|\boldsymbol{\varphi}, \Sigma, \boldsymbol{r} \sim Binomial\left(N - 1, (1 - \varphi_2)p_{e_2} + \varphi_2(1 - p_{e_2})\right), \qquad (19)$$

   where $p_{e_2} = P(y^{(g)}_{e_2} = 1|\Sigma, r_2)$ *is the probability of* e_2 *being present in g.*

2. *For k=3:*

$$Deg^{g^*}_{(i,3)}|\boldsymbol{\varphi}, \Sigma, \boldsymbol{r} \,\dot{\sim}\, Poisson\left(\sum_{\{e_3 \in \mathcal{E}_{N,3}|i \in e_3\}} \binom{N-1}{2}\left[(1 - \varphi_3)p_{e_3} + \varphi_3(1 - p_{e_3})\right]\right), \quad (20)$$

   where $p_{e_3} = P(y^{(g)}_{e_3} = 1|\Sigma, r_3)$ *is the probability of* e_3 *being present in g and* $X \,\dot{\sim}\, f(x)$ *indicates that X is approximately distributed according to* $f(x)$.

Proof. See Appendix F.3. □

Lemma 4.1 and Theorem 4.1 indicate how the model parameters affect the degree distribution. The probability of connection $p_k = (1 - \varphi_k)p_{e_k} + \varphi_k(1 - p_{e_k})$ follows from Property (P1), and we note that equivalent degree distributions can be obtained from different choices for φ_k and p_{e_k}. For example, small p_{e_k} and large φ_k will behave similarly to large p_{e_k} and small φ_k. However, the characteristics of the resulting hypergraphs will differ significantly in each of these cases since large p_{e_k} and small φ_k will impose strong geometric constraints on the hyperedges. The Poisson approximation in Theorem 4.1 is only appropriate when p_3 is small, and we comment that improvements may be made to this approximation using methodology presented in Teerapabolarn (2014). In Figure 5 we compare the theoretical degree distribution with the simulated degree distribution for a hypergraph with $N = 20$ and $K = 3$ and, in Section 7.2, we compare the theoretical and observed degree distributions for a real data example.

5 Posterior Sampling

The posterior distributions for the models specified in Sections 3.3 and 3.4 do not have a closed-form expression. In this section we provide a high-level description of our Markov chain Monte Carlo (MCMC) algorithm that we use to sample from the posterior distribution of the model detailed in Algorithm 2. This MCMC scheme can easily be modified for the model detailed in Algorithm 1 and we refer the reader to the relevant sections of the Appendix for further details.

5.1 MCMC scheme

We use a Metropolis-Hastings-within-Gibbs MCMC scheme (see Section 6.4.2 of Gamerman and Lopes (2006)) to sample from the model presented in Algorithm 2 with priors given in (14). We update the latent coordinates $\boldsymbol{U}$ and radii $\boldsymbol{r}$ with a Metropolis-Hastings (MH) step, and the remaining parameters are updated via Gibbs samplers.

When updating the latent coordinates we use a random-walk MH. As discussed in Section 3.5, we define $\boldsymbol{U}$ on the Bookstein space of coordinates and fix a set of anchor points throughout the MCMC scheme. For $u_i \in \mathbb{R}^d$, let the anchor points be denoted by u_1 and u_2. For $i = 3, 4, \ldots, N$,

---

Algorithm 3 MCMC scheme to obtain posterior samples of $\boldsymbol{U}, \boldsymbol{r}, \boldsymbol{\psi}^{(0)}, \boldsymbol{\psi}^{(1)}, \Sigma$ and μ.

Specify $N, K, L, i_{max} \in \mathbb{N}$.
Initialise
    Determine initial values for $\boldsymbol{U}, \boldsymbol{r}, \boldsymbol{\psi}^{(0)}, \boldsymbol{\psi}^{(1)}, \Sigma$ and μ using Algorithm 6 (see Appendix D).
For i **in** $1, 2, \ldots, i_{max}$
    1) Sample $\mu^{(i)}$ from $p(\mu|\boldsymbol{U}, \Sigma, m_\mu, \Sigma_\mu)$ (see Appendix C.1).
    2) Sample $\Sigma^{(i)}$ from $p(\Sigma|\boldsymbol{U}, \mu, \Phi, \nu)$ (see Appendix C.2).
    3) Partition $\{u_3, u_4, \ldots, u_N\}$ into L sets U_l.
        For $l = 1, 2, \ldots, L$
            For $i \in U_l$, propose $u_i^* = u_i + \epsilon_u$, where $\epsilon_u \sim \mathcal{N}(0, \sigma_u I_d)$
            Accept proposal with probability (21).
    4) **For** $k = 2, 3, \ldots, K$, propose $\boldsymbol{r}^*$, where $r_k^* = r_k + \epsilon_r$ and $\epsilon_r \sim \mathcal{N}(0, \sigma_r)$
        Accept proposal $\boldsymbol{r}^*$ with probability (22).
    5) **For** $k = 2, 3, \ldots, K$
        Sample $\psi_k^{(0)}$ from $p\left(\psi_k^{(0)}|\boldsymbol{U}, \boldsymbol{r}, h_{N,K}, a_k^{(0)}, b_k^{(0)}\right)$ (see Appendix C.3).
        Sample $\psi_k^{(1)}$ from $p\left(\psi_k^{(1)}|\boldsymbol{U}, \boldsymbol{r}, h_{N,K}, a_k^{(1)}, b_k^{(1)}\right)$ (see Appendix C.4).

---

we propose $u_i^* = u_i + \epsilon_u$ where $\epsilon_u \sim \mathcal{N}(0, \sigma_u I_d)$, and for $i = 1, 2$ we let $u_i^* = u_i$. We then accept $\boldsymbol{U}^* = \{u_i^*\}_{i=1}^N$ as a sample from $p(\boldsymbol{U}|\mu, \Sigma, \boldsymbol{r}, \boldsymbol{\psi}^{(0)}, \boldsymbol{\psi}^{(1)}, h_{N,K})$ with probability

$$\min\left\{1, \frac{\mathcal{L}(\boldsymbol{U}^*, \boldsymbol{r}, \boldsymbol{\psi}^{(1)}, \boldsymbol{\psi}^{(0)}; h_{N,K})p(\boldsymbol{U}^*|\mu, \Sigma)}{\mathcal{L}(\boldsymbol{U}, \boldsymbol{r}, \boldsymbol{\psi}^{(1)}, \boldsymbol{\psi}^{(0)}; h_{N,K})p(\boldsymbol{U}|\mu, \Sigma)}\right\}, \tag{21}$$

where $p(\boldsymbol{U}|\mu, \Sigma) = \prod_{i=1}^N p(u_i|\mu, \Sigma)$. Since the proposal mechanism is symmetric in terms of $\boldsymbol{U}$ and $\boldsymbol{U}^*$, the term associated with this does not appear in (21).

The acceptance ratio (21) is for the full ($N \times d$) latent representation $\boldsymbol{U}$ and, as N grows, jointly proposing all latent coordinates will become increasingly inefficient. Alternatively, we can partition the latent coordinates into disjoint sets $\{U_l\}_{l=1}^L$ such that $\cup_{l=1}^L U_l = \{1, 2, \ldots, N\}$, and perform the MH update for each of these L sets separately. This approach will be used in the examples in Sections 6 and 7, and details of this are given in Algorithm 3.

To update $\boldsymbol{r}$, we let $\boldsymbol{r}^* = (r_2^*, r_3^*, \ldots, r_K^*)$ where $r_k^* = r_k + \epsilon_r$ and $\epsilon_r \sim \mathcal{N}(0, \sigma_r)$. Then we accept $\boldsymbol{r}^*$ as a sample from $p(r|\boldsymbol{U}, \boldsymbol{\psi}^{(0)}, \boldsymbol{\psi}^{(1)}, \mu, \Sigma)$ with probability

$$\min\left\{1, \frac{\mathcal{L}(\boldsymbol{U}, \boldsymbol{r}^*, \boldsymbol{\psi}^{(1)}, \boldsymbol{\psi}^{(0)}; h_{N,K})p(\boldsymbol{r}^*|\boldsymbol{\lambda})}{\mathcal{L}(\boldsymbol{U}, \boldsymbol{r}, \boldsymbol{\psi}^{(1)}, \boldsymbol{\psi}^{(0)}; h_{N,K})p(\boldsymbol{r}|\boldsymbol{\lambda})}\right\}, \tag{22}$$

where $p(\boldsymbol{r}|\boldsymbol{\lambda}) = \prod_{k=2}^K p(r_k|\lambda_k)$.

All other parameters can be sampled directly from their full conditionals, and the details of the MCMC scheme for i_{max} iterations are given in Algorithm 3. Initialisation for the MCMC is non-trivial, and a discussion of this can be found in Appendix D.

To implement the MCMC scheme, there are a number of computational considerations we must address. Firstly, to evaluate the likelihood we need to determine the hyperedges in the hypergraph generated from $\boldsymbol{U}$ and $\boldsymbol{r}$, $g(\boldsymbol{U}, \boldsymbol{r})$. We rely on methodology from the computational topology literature to do this, and details of the approach used are given in Appendix E.1. Secondly, to calculate the likelihood we note that we can avoid the summation over the entire set of hyperedges as suggested by (12) and, in Appendix E.2, we discuss the details of this.

Model	Case
Stasi et al. (2014)	1) All nodes equally likely to form connections
	2) Some nodes more likely to form connections
Ng and Murphy (2018)	1) Hyperedges in a single cluster
	2) Distinct topic clusters only
	3) Distinct size clusters only
	4) Fuzzy topic clusters
LSH	1) Strongly correlated Σ
	2) No correlation in Σ
	3) Dense in e_2, sparse in e_3, e_4
	4) Sparse e_2, e_4, dense in e_3
	5) Increase latent dimension from $d = 2$ to $d = 3$

Table 1: Summary of cases for each hypergraph model considered in the model depth comparison study. The case numbers correspond to the labels in Figure 7.

6 Simulations

In this section we describe two simulation studies. We begin in Section 6.1 with an investigation of the flexibility of our modelling approach in comparison with two other hypergraph models from the literature. Then, in Section 6.2, we examine the predictive degree distribution conditional on an observed hypergraph. An additional simulation study is presented in Appendix J, we consider the robustness of our model under misspecification.

6.1 Model depth comparisons

In this study we explore the range of hypergraphs that can be expressed in our modelling framework. We compare this with the uniform hypergraph model of Stasi et al. (2014) and the extended latent class analysis hypergraph model of Ng and Murphy (2018) (details of these models is given in Appendix G). For each model, we specify several cases which are designed to highlight particular aspects of the model. We then simulate hypergraphs for each case and record summary statistics to characterise the simulated hypergraphs.

We will focus on the hypergraph model described in Algorithm 2 in addition to the models of Stasi et al. (2014) and Ng and Murphy (2018). First, it is important to note that each of the models considered in this study have been designed for a different purpose. The uniform hypergraph model of Stasi et al. (2014) allows control over degree distribution via node specific parameters, and the extended latent class analysis hypergraph model of Ng and Murphy (2018) is designed to express hyperedges which exhibit a clustering structure according topic and size. Whereas our model utilises a latent geometry to model the connections, therefore we expect transitive connections. For example, if $\{i, j\}$ and $\{i, k\}$ are close in latent space, then $\{j, k\}$ is likely to be present in the hypergraph. A higher order transitivity will also occur in which, for example, the presence of hyperedges $\{i, j\}, \{i, k\}$ and $\{j, k\}$ suggests that the hyperedge $\{i, j, k\}$ is likely to be present. To showcase each of these models, we consider the cases described in Table (1) and the details of these are provided in Appendix H.

To characterise the simulated hypergraphs we consider several metrics. We record counts for the subgraphs depicted in Figure 6a. The degree distribution and spread of hyperedge orders are recorded using the percentiles of the node degrees and edge sizes, respectively, and we record the density of the hyperedges of order $k = 2$ and $k = 3$. Finally, to determine the clustering in the hypergraph, we project the hypergraph onto a pairwise graph such that the edge $\{i, j\}$ exists if
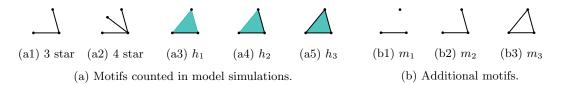
(a1) 3 star (a2) 4 star (a3) h_1 (a4) h_2 (a5) h_3 (b1) m_1 (b2) m_2 (b3) m_3

(a) Motifs counted in model simulations.   (b) Additional motifs.

Figure 6: Depiction of motifs considered in Sections 6 and 7.

i and j are present in the same hyperedge. Then, given this pairwise graph, we determine the community structure using the leading eigenvector with the function `cluster_leading_eigen()` in the *igraph* package in R (Csardi and Nepusz (2006)). We report the modularity of this clustering and the number of clusters, which measures the strength of the clustering and lies within $[-1, 1]$, where a high value indicates that the network can be divided clearly into clusters.

Figure 7 summarises the results of our study. For the model of Stasi et al. (2014) (Figure 7a) we see that, by comparing cases 1 and 2, this model can express hypergraphs with very different degree distributions. Additionally, since case 2 generates denser hypergraphs, we observe a larger number of motif counts. Note that the maximum hyperedge order was set to 4, and so no hyperedges for $k \geq 5$ are generated. For the model of Ng and Murphy (2018) (Figure 7b), we see that the topic clustering is consistently captured for all simulated hypergraphs. Generally, the degree distributions are similar, however by comparing cases 1 and 4 we note that this model can express different levels of connectivity. We also observe reasonably little variation in the motif counts and, for most cases, find that triangles are more prevalent than the hypergraph motifs. Finally, for our latent space hypergraph model (Figure 7c), we observe a greater control over the motif counts. For example, consider cases 3 and 4 where the order $k = 2$ and $k = 3$ hyperedges are denser. The counts for triangles and h_1 subgraphs clearly reflect the number of hyperedges of each order, and we also observe some control over the degree distribution and density. When we increase the latent dimension to $d = 3$, and fix all other parameters, we obtain a sparser hypergraph and to see this compare cases 1 and 5. Finally, since our graph is not designed to capture clustering, we do not observe consistent estimates for the number of clusters. We may alter aspects of our model to incorporate community structure or to vary the degree distribution.

In this study, we have explored the advantages of each modelling approach. It is clear that, for the construction we have chosen, our model presents a flexible framework that is particularly appropriate for hypergraph data which exhibit many motifs. We note that our approach can be modified to express hypergraphs with different characteristics by changing the distribution of the latent coordinates or the choice of complex.

6.2   Prior Predictive vs Posterior Predictive

In this section we examine the predictive degree distribution conditional on an observed hypergraph. To explore the predictive distribution, we rely on the latent representation to simulate new nodes and their associated connections given estimated model parameters. Since the models of Stasi et al. (2014) and Ng and Murphy (2018) contain node specific parameters, we comment that it is not immediately obvious how to implement an analogue of this in either of their frameworks. We begin by describing the study and set up, and then present our findings.

Using the latent space representation, we are able to examine how newly simulated nodes connect to an observed hypergraph. Suppose that we have fitted the hypergraph model detailed in Algorithm 2 to a hypergraph h_{obs} and we obtain the parameter estimates $\hat{\mu}, \hat{\Sigma}, \hat{r}, \hat{\varphi}^{(0)}$ and $\hat{\varphi}^{(1)}$. Conditional on these estimates, we may simulate new nodes and determine the hyperedges

(a) Uniform hypergraph model of Stasi et al. (2014).

(b) Extended latent class analysis hypergraph model of Ng and Murphy (2018).

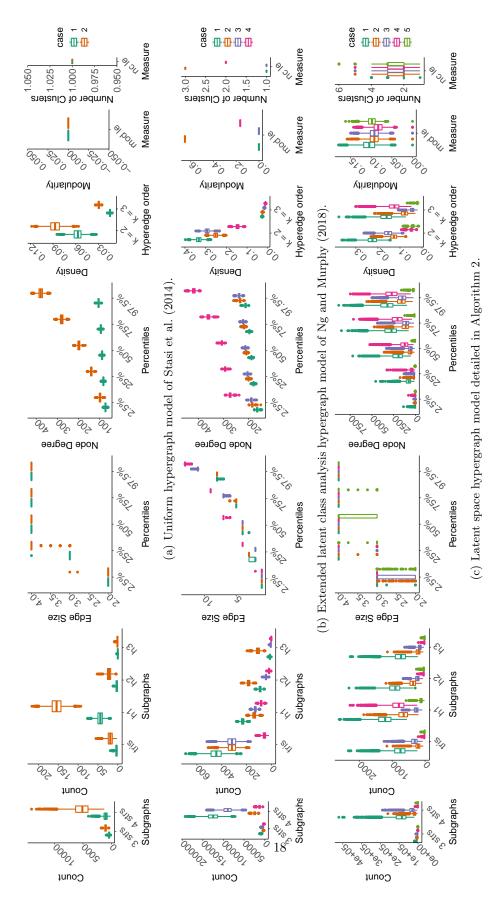(c) Latent space hypergraph model detailed in Algorithm 2.

Figure 7: Summary of hypergraphs simulated from each of the models considered in Section 6.1. The cases considered are summarised in Table 2 in Appendix H.
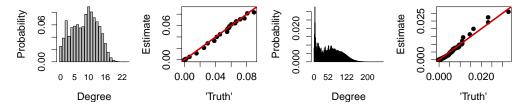
Figure 8: Comparison of prior and posterior predictive degree distributions for $N^* = 10$ simulated nodes. The left two panels show the prior predictive degree distribution and qq-plot of the prior and posterior predictive degree distributions for order 2 hyperedges, and the right two panels show the same, but for order 3 hyperedges.

induced from these additional nodes. Through repeated simulation we can then empirically estimate the predictive degree distribution of the newly simulated nodes and h_{obs}.
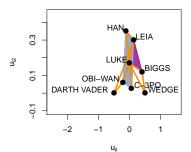
To implement this procedure, we being by simulating a hypergraph h_{sim} according to Algorithm 2 with $N = 50, d = 2, K = 3, \boldsymbol{r} = (0.32, 0.4), \psi_k^{(0)} = \psi_k^{(1)} = 0.001, \mu = (0.16, 1.24)$ and $\Sigma = \left(\begin{smallmatrix} 0.58 & 0 \\ 0 & 0.58 \end{smallmatrix}\right)$. After 10,000 post burn-in MCMC iterations, we obtain the model parameter estimates $\hat{\boldsymbol{U}}, \hat{\boldsymbol{r}} = (0.13, 0.16), \hat{\boldsymbol{\psi}}^{(0)} = (0.0058, 0.0014), \hat{\boldsymbol{\psi}}^{(1)} = (0.0057, 0.0035), \hat{\mu} = (-0.13, 0.44)$ and $\hat{\Sigma} = \left(\begin{smallmatrix} 0.14 & -0.0039 \\ -0.0039 & 0.078 \end{smallmatrix}\right)$.

To estimate the posterior predictive degree distribution we apply the following procedure n_{rep} times.

1. Set $u_i^* = \hat{u}_i$ for $i = 1, 2, \ldots, N$.

2. Simulate coordinates $u_i^* \sim \mathcal{N}(\hat{\mu}, \hat{\Sigma})$, for $i = N + 1, N + 2, \ldots, N + N^*$.

3. Determine the hypergraph h_{sim}^* obtained by taking the union of h_{obs} and the additional hyperedges induced from $\boldsymbol{U}^* = \{u_i^*\}_{i=N+1}^{N+N^*}, \hat{\boldsymbol{r}}, \hat{\boldsymbol{\psi}}^{(0)}$ and $\hat{\boldsymbol{\psi}}^{(1)}$.

4. Calculate the degree distribution of h_{sim}^*.

Recall that, for $u_i \in \mathbb{R}^2$, two coordinates are fixed as anchor points throughout posterior sampling. These anchor points determine the scaling of Σ and $\boldsymbol{r}$ and it is therefore not appropriate to make a direct comparison between true and estimated model parameters. However, considering the prior and posterior predictive degree distributions presents a fair comparison between models.

The full estimated prior and posterior predictive degree distributions are presented in Figure 8, where the histograms show the estimated prior predictive and the qq-plots show the prior and posterior predictive degree distributions. We see that there is a strong correspondence between the two distributions, particularly for the order $k = 2$ hyperedges. However, there is a slight difference in the upper tail for hyperedges of order $k = 3$. This may be due to the complexity of the space or the greater number of constraints placed on the latent coordinates of highest degree. Further plots which examine the distributions for hyperedges occurring between newly simulated nodes only, and the hyperedges occurring between the nodes of h_{sim} and the newly simulated nodes, are presented in Appendix I. For both cases, we see a close correspondence between the prior and posterior predictive degree distributions.
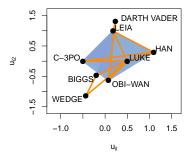
Figure 9: Posterior mean of latent coordinates for the Star Wars dataset with upper limits on $\boldsymbol{\varphi}$ set to 0.75 (left panel) and 1.5 (right panel) $\times$ the observed order k hyperedge density. Connections in orange, blue and purple correspond to hyperedges of order $k = 2, 3$ and 4, respectively.

7 Real data examples

In this section we analyse two real world hypergraph datasets using our model. First, we examine a dataset constructed from actor co-occurrence in 'Star Wars: A New Hope' and compare our analysis with that of Ng and Murphy (2018). Then, we consider a coauthorship dataset and asses predictive inference by comparing to the observed hypergraph. An additional data example in given in Appendix K.

7.1 Star Wars: A New Hope

In this section we consider a dataset constructed from the script of 'Star Wars: A New Hope' which describes co-occurrence between the eight main characters. We represent this as a hypergraph where the nodes represent characters and hyperedges indicate which characters appeared in a scene together. In this dataset we have $N = 8$ and $K = 4$, and we remove repeated hyperedges and hyperedges of order one to ensure the data is amenable to analysis under our model. This dataset was considered in Ng and Murphy (2018), and we compare our methodology to this approach.

Recall that, in our model, observed hyperedges can be explained by the latent geometry or the hyperedge modification. To ensure most hyperedges are explained by the latent representation, we fix an upper limit for the parameters $\boldsymbol{\varphi}$. In doing so, we encourage interpretable latent coordinates and improve the quality of predictive inference. To begin, we fit the model detailed in Algorithm 1 and set the upper limit for φ_k to be 0.75 or 1.5$\times$ the density of order k hyperedges, for $k = 3, 4$.

The posterior mean of the latent coordinates after 37500 post burn-in MCMC iterations is given in Figure 9 (left). In this figure, orange lines indicate a pairwise connection, and blue and purple regions correspond to order 3 and 4 hyperedges, respectively. We observe a group of well-connected nodes which contains the character *Luke* in its centre. In Ng and Murphy (2018), this character was highlighted to be likely to occur in the two largest topic clusters, and we comment that the latent representation reflects this characters importance. A similar observation is made in the network visualisation literature, where nodes with a greater number of connections are placed more centrally. The main group of nodes in Figure 9 (left) is largely determined by the order 3 and 4 hyperedges between *Leia*, *C-3PO*, *Luke*, *Obi-Wan* and *Han*. The characters *Wedge*
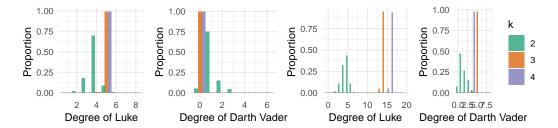
20

Figure 10: Predicted degree distributions conditional on the fitted model. Given $\hat{\boldsymbol{U}}$, $\hat{\boldsymbol{r}}$ and $\hat{\boldsymbol{\varphi}}$ we simulate the connections in the hypergraph to estimate the degree distribution, and the upper limit for $\boldsymbol{\varphi}$ is 0.75 the hyperedge density in left two panels and 1.5 the hyperedge density in right two panels, respectively.

and *Darth Vader* are less connected, and appear on the periphery of the latent representation. Figure 9 (right) shows the posterior mean of the latent coordinates when the upper limit for φ_k is $1.5 \times$ the density of order k hyperedges, for $k = 3, 4$. Again, the importance of the character *Luke* is reflected in the latent coordinates, however, the increased noise parameter means the latent coordinates are less constrained.

As mentioned in Section 6, our modelling framework allows us to explore predictive distributions. We consider the degree distributions for the characters *Luke* and *Darth Vader* by repeatedly simulating their connections given $\hat{\boldsymbol{U}}$, $\hat{\boldsymbol{r}}$ and $\hat{\boldsymbol{\varphi}}$ with both choices of upper limits on $\boldsymbol{\varphi}$. The results of this for upper limits of 0.75 and 1.5 times the hyperedge densities are shown in Figure 10 (left/right). In both cases we see a clear difference in the levels of connectivity for each character. Since *Luke* is more centrally located with respect to the latent coordinates, we observe that he is expected to be more connected than *Darth Vader*, who is located on the periphery. This tells us that nodes which are more centrally located are expected to be more connected in the hypergraph.

We now comment on the differences between our approach and the approach of Ng and Murphy (2018). Firstly, Ng and Murphy (2018) incorporate multiple occurrences of a hyperedge and hyperedges containing a single node into their analysis and our methodology does not facilitate this. Secondly, our model provides a visualisation of the hypergraph which reflects many observations of Ng and Murphy (2018), albeit it without the visualisation. Finally, our framework allows us to investigate the predictive distributions, which has allowed us to comment on the expected variability of the degree of certain nodes.

7.2 Coauthorship for Statisticians

We now return to our motivating example of coauthorship. We consider the dataset provided by Ji and Jin (2016) which describes coauthorship between $N = 3606$ statisticians where $K = 10$. Analysing the full dataset within our framework is computationally prohibitive, and so we consider a subset with $N = 48$ and $K = 3$. We obtain the subset by restricting the hyperedges to an order less than 4 and we select a seed node and include hyperedges involving this node with probability $p = 0.9$. We repeat this process with the nodes added to the hypergraph and we refer to each newly added set of nodes as a *wave*. To maintain a reasonably sized hypergraph subsample, we decrease the probability of inclusion by 0.15 for each new wave. Our interest is in assessing the quality of predictive inference by comparing the predictives to the next sampling wave.

We fit the model detailed in Algorithm 1 and the posterior mean of the latent coordinates
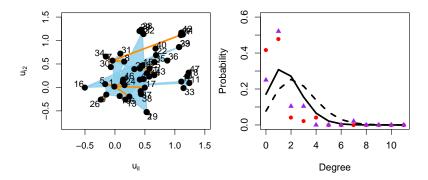
Figure 11: (Left) posterior mean of latent coordinates after 20000 iterations. Hyperedges of order 2 and 3 are shown in orange and blue, respectively. (Right), the observed and theoretical degree distribution, lines: solid is $k = 2$ theory, dashed is $k = 3$ theory, points: circles and triangles correspond to $k = 2$ and $k = 3$ observations, respectively.

is shown in Figure 11. As in the previous examples, we restrict the hyperedge modification to be at most 20% of the observed hyperedge density. By comparing the hyperedges in the graph $g_{N,K}(\hat{\boldsymbol{U}}, \hat{\boldsymbol{r}})$ to those present in the observed hypergraph, we find that 96% of order 2 hyperedges and 99% of order 3 hyperedges are explained by the latent representation. The nodes $1, 7$ and 25 have the highest degree and we see these nodes are placed close to the estimated mean $\hat{\mu}$ of the distribution of the latent coordinates. We also compare the theoretical degree distributions discussed in Section 4 with the observed degree distribution in Figure 11, where the theoretical degree distribution is calculated using the fitted model parameters and p_{e_3} is estimated via simulation. There is a reasonable agreement between the observed and theoretical degree distributions, though we see a slight discrepancy for the lower order degrees. We expect that this is due to the subsampling mechanism from which the data were obtained, since this will encourage a large number of nodes with low degree.

We assess the fitted model by comparing the subgraph counts for the next wave of sampling with the predictive distributions. We estimate the predictive distributions for the subgraphs shown in Figures 6a3, 6a4, 6a5, 6b1, 6b2 and 6b3, and compare this with the counts observed in the next wave of subsampling. The predictive distributions for the additional $N^* = 19$ nodes are shown in Figure 12, where the subgraph counts observed in the next wave of sampling are shown in red. Due to the subsampling mechanism, we expect the next 19 nodes to mostly be placed on the periphery of the point cloud. To mimic this, we sample $N + N^*$ nodes from $\mathcal{N}(\hat{\mu}, \hat{\Sigma})$ and take the N^* coordinates which lie furthest from $\hat{\mu}$ in terms of Euclidean distance. In Figure 12, we see a reasonable correspondence between the observed and predicted, however we note that many of the observed subgraph counts lie in the lower tail of the predictive.

8 Discussion

In this paper we have introduced a latent space model for hypergraph data in which the nodes are represented by coordinates in $\mathbb{R}^d$. We extend the framework in Hoff et al. (2002) through a modification of a nerve construction which allows us to express non-simplicial hypergraphs. This application of a nerve draws a connection between stochastic geometry and latent space network models, and allows us to develop a parsimonious hypergraph model. The latent representation
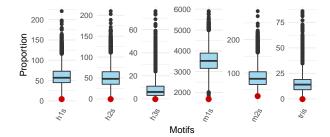
Figure 12: $N^* = 19$ predictive subgraph counts. Left to right: motifs depicted in Figure 6a3, 6a4, 6a5, 6b1, 6b2 and 6b3. The red dots correspond to the observed motif counts for the newly sampled nodes.

imposes properties on the hypergraphs generated from our model, including a type of 'higher-order transitivity'. This property, in which the presence of an order k hyperedge is more likely given the presence of subsets of the hyperedge, is highlighted in the model depth simulations in Section 6.1. It is important to note that, depending on the modelling choices, particular hypergraph relationships may be challenging to represent using our nerve construction. For example, the maximum number of possible leaves in a star will be limited by the dimension of the latent space. However this may be mediated by either choosing a different convex set to generate the nerve, increasing the probability of hyperedge modification or adopting a different specification for the latent positions.

The modification of the indicators for the hyperedges has two main motivations. Firstly, without this modification, the conditional distribution $\mathcal{L}(\boldsymbol{U}, \boldsymbol{r}; h_{N,K})$ would be equal to one only when there is a perfect correspondence between the observed and estimated hyperedges. Secondly, the modification extends the support of the model to the space of all hypergraphs. We note here that techniques such as tempering can be used to aid model fitting, but the challenge of characterising the support of the model would remain. From (15) we observed that a hypergraph generated from our model is a modification of a nsRGH and, as the probability of modification grows, the generated hypergraphs will behave more like the hypergraph analogue to an Erdős-Rényi random graph. In the context of networks, Le et al. (2018) investigate the recovery of an underlying true network given a set of noisy observations. Whilst this differs from our setting, we can view our model in a similar way in which the nsRGH represents the truth. This helps motivate our observations on the magnitude of the probability of hyperedge modification.

Our latent space hypergraph model provides a framework for modelling hypergraph data that could be extended to account for sparsity, community structure, or non-binary hypergraphs where hyperedges may occur multiple times. Spencer and Rohilla Shalizi (2017) have shown that modelling the latent coordinates according to a Poisson process allows the expression of graphs with various levels of sparsity, and a similar approach may be explored in the hypergraph setting. To model community structures we may consider modelling the latent coordinates as a mixture of Gaussians, which is typical in the latent space network literature (for example, see Handcock et al. (2007)). Although the models we have presented can in theory express hypergraphs with arbitrary K, this proves computationally challenging in practice. This is a key limitation of our approach, and alternatives to the Čech complex may be explored as an approach to improve scalability. For example, a popular alternative in computational topology is the Vietoris-Rips complex (Vietoris (1927)) in which an order k hyperedge exists if each of the $\binom{k}{2}$ balls of radius r intersect. Then, evaluating the order k hyperedges only requires consideration of the pairwise intersections and so is more scalable in terms of K. This may be adapted to the

23

non-simplicial setting by letting r_k denote the radii which controls order k interactions, similarly to the construction in Definition 3.1. Finally, we have not considered explicitly modelling the radii r_k as a function of k. Whether this offers modelling advantages remains an open question.

Acknowledgements

References

Aksoy, S., Kolda, T. G., and Pinar, A. (2016). Measuring and Modeling Bipartite Graphs with Community Structure. *arXiv e-prints*, page arXiv:1607.08673.

Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439):509–512.

Barabási, A.-L. and Pósfai, M. (2016). *Network science*. Cambridge University Press, Cambridge.

Berg, M. d., Cheong, O., Kreveld, M. v., and Overmars, M. (2008). *Computational Geometry: Algorithms and Applications*. Springer-Verlag TELOS, Santa Clara, CA, USA, 3rd ed. edition.

Bookstein, F. L. (1986). Size and shape spaces for landmark data in two dimensions. *Statist. Sci.*, 1(2):181–222.

Campbell, T., Cai, D., and Broderick, T. (2018). Exchangeable trait allocations. *Electron. J. Statist.*, 12(2):2290–2322.

Cox, T. F. and Cox, M. A. A. (2000). *Multidimensional Scaling*. Chapman & Hall, 2 edition.

Crane, H. and Dempsey, W. (2018). Edge exchangeable models for interaction networks. *Journal of the American Statistical Association*, 113(523):1311–1326.

Csardi, G. and Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal*, Complex Systems:1695.

Delaunay, B. (1934). Sur la sphére vide. *Bulletin de l'Académie des Sciences de l'URSS, Classe des Sciences Mathématiques et Naturelles*, pages 793–800.

Dempsey, W., Oselio, B., and Hero, A. (2019). Hierarchical network models for structured exchangeable interaction processes. *arXiv e-prints*, page arXiv:1901.09982.

Dryden, I. L. and Mardia, K. V. (1998). *Statistical Shape Analysis*. Wiley, Chichester.

Duchesne, P. and Micheaux, P. L. D. (2010). Computing the distribution of quadratic forms: Further comparisons between the liutangzhang approximation and exact methods. *Computational Statistics and Data Analysis*, 54(4):858 – 862.

Edelsbrunner, H. and Harer, J. (2010). *Computational Topology - an Introduction.* American Mathematical Society.

Edelsbrunner, H., Kirkpatrick, D., and Seidel, R. (1983). On the shape of a set of points in the plane. *IEEE Trans. Inf. Theor.*, 29(4):551–559.

Erdös, P. and Rényi, A. (1959). On random graphs, I. *Publicationes Mathematicae (Debrecen)*, 6:290–297.

Friel, N., Rastelli, R., Wyse, J., and Raftery, A. E. (2016). Interlocking directorates in irish companies using a latent space model for bipartite networks. *Proceedings of the National Academy of Sciences*, 113(24):6629–6634.

Gamerman, D. and Lopes, H. F. (2006). *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference.* Chapman and Hall/CRC, 2 edition.

Ghoshdastidar, D. and Dukkipati, A. (2014). Consistency of spectral partitioning of uniform hypergraphs under planted partition model. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 397–405. Curran Associates, Inc.

Gilliland, D. C. (1962). Integral of the bivariate normal distribution over an offset circle. *Journal of the American Statistical Association*, 57(300):758–768.

Handcock, M. S., Raftery, A. E., and Tantrum, J. M. (2007). Model-based clustering for social networks. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 170(2):301–354.

Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098.

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social Networks*, 5(2):109 – 137.

Holland, P. W. and Leinhardt, S. (1981). An exponential family of probability distributions for directed graphs. *Journal of the American Statistical Association*, 76(373):33–50.

Ji, P. and Jin, J. (2016). Coauthorship and citation networks for statisticians. *Ann. Appl. Stat.*, 10(4):1779–1812.

Kahle, M. (2016). Random simplicial complexes. *arXiv e-prints*, page arXiv:1607.07069.

Ke, Z. T., Shi, F., and Xia, D. (2019). Community Detection for Hypergraph Networks via Regularized Tensor Power Iteration. *arXiv e-prints*, page arXiv:1909.06503.

Kim, B., Lee, K., Xue, L., and Niu, X. (2017). A Review of Dynamic Network Models with Latent Variables. *arXiv e-prints*, page arXiv:1711.10421.

Kolaczyk, E. D. (2009). *Statistical Analysis of Network Data: Methods and Models.* Springer Publishing Company, Incorporated, 1st edition.

KONECT (2017). Corporate leadership network dataset. `http://konect.uni-koblenz.de/networks/brunson_corporate-leadership`.

Krivitsky, P. N., Handcock, M. S., Raftery, A. E., and Hoff, P. D. (2009). Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models. *Social Networks*, 31(3):204 – 213.

Kunegis, J. (2013). Konect: The koblenz network collection. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13 Companion, pages 1343–1350, New York, NY, USA. ACM.

Le, C. M., Levin, K., and Levina, E. (2018). Estimating a network from multiple noisy realizations. *Electron. J. Statist.*, 12(2):4697–4740.

Leskovec, J. and Krevl, A. (2014). SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data.

Liu, D., Blenn, N., and Mieghem, P. V. (2013). Characterising and modelling social networks with overlapping communities. *Int. J. Web Based Communities*, 9(3):371–391.

Lunagómez, S., Mukherjee, S., Wolpert, R. L., and Airoldi, E. M. (2017). Geometric representations of random hypergraphs. *Journal of the American Statistical Association*, 112(517):363–383.

Lunagómez, S., Olhede, S. C., and Wolfe, P. J. (2019). Modeling Network Populations via Graph Distances. *arXiv e-prints*, page arXiv:1904.07367.

Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., and Hornik, K. (2019). *cluster: Cluster Analysis Basics and Extensions*. R package version 2.0.9 — For new features, see the 'Changelog' file (in the package source).

Marin, J.-M., Pudlo, P., Robert, C. P., and Ryder, R. J. (2012). Approximate bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180.

Ng, T. L. J. and Murphy, T. B. (2018). Model-based clustering for random hypergraphs. *ArXiv e-prints*.

Penrose, M. (2003). *Random Geometric Graphs*. Oxford Studies in Probability.

Petersen, K. B. and Pedersen, M. S. (2012). The matrix cookbook. Version 20121115.

Pronzato, L., Wynn, H. P., and Zhigljavsky, A. (2019). Bregman divergences based on optimal design criteria and simplicial measures of dispersion. *Statistical Papers*, 60(2):195–214.

Rastelli, R., Friel, N., and Raftery, A. E. (2015). Properties of Latent Variable Network Models. *arXiv e-prints*, page arXiv:1506.07806.

Salter-Townshend, M., White, A., Gollini, I., and Murphy, T. B. (2012). Review of statistical network analysis: models, algorithms, and software. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(4):243–264.

Sarkar, P. and Moore, A. W. (2006). Dynamic social network analysis using latent space models. In Weiss, Y., Schölkopf, B., and Platt, J. C., editors, *Advances in Neural Information Processing Systems 18*, pages 1145–1152. MIT Press.

Spencer, N. A. and Rohilla Shalizi, C. (2017). Projective, Sparse, and Learnable Latent Position Network Models. *arXiv e-prints*, page arXiv:1709.09702.

26

Stasi, D., Sadeghi, K., Rinaldo, A., Petrović, S., and Fienberg, S. E. (2014). β models for random hypergraphs with a given degree sequence. *ArXiv e-prints*.

Teerapabolarn, K. (2014). An improvement of poisson approximation for sums of dependent bernoulli random variables. *Communications in Statistics - Theory and Methods*, 43(8):1758–1777.

The GUDHI Project (2015). *GUDHI User and Reference Manual*. GUDHI Editorial Board.

Vietoris, L. (1927). Über den hheren zusammenhang kompakter räume und eine klasse von zusammenhangstreuen abbildungen. *Mathematische Annalen*, pages 454–472.

Zhou, D., Huang, J., and Schölkopf, B. (2006). Learning with hypergraphs: Clustering, classification, and embedding. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06, pages 1601–1608, Cambridge, MA, USA. MIT Press.

A   Bookstein coordinates

In Bookstein coordinates a set of points are chosen as the anchor points. These points are fixed in the space and all other points are translated, rotated and scaled accordingly. In Appendix A.1 we describe the Bookstein coordinates in $\mathbb{R}^2$, and in Appendix A.2 we describe the Bookstein coordinates in $\mathbb{R}^3$.

A.1   Bookstein coordinates in $\mathbb{R}^2$

In $\mathbb{R}^2$, we set the anchor points $u_1^B = (u_{11}^B, u_{12}^B)$ and $u_2^B = (u_{21}^B, u_{22}^B)$ to be $(-1/2, 0)$ and $(1/2, 0)$, respectively. Let $\boldsymbol{U}^B$ denote the Bookstein coordinates and $\boldsymbol{U}$ denote the untransformed coordinates. Then $\boldsymbol{U}^B$ is given by

$$\boldsymbol{U}^B = cR(\boldsymbol{U} - \boldsymbol{b})$$
$$= \frac{1}{\sqrt{(u_{21}^B - u_{11}^B)^2 + (u_{22}^B - u_{12}^B)^2}} \begin{bmatrix} \cos(a) & \sin(a) \\ -\sin(a) & \cos(a) \end{bmatrix} \left(\boldsymbol{U} - \frac{1}{2} \begin{bmatrix} u_{11}^B + u_{21}^B \\ u_{12}^B + u_{22}^B \end{bmatrix} \right), \quad (23)$$

where $a = \arctan\{(u_{22}^B - u_{12}^B)/(u_{21}^B - u_{11}^B)\}$. The Bookstein transformation can hence be seen as a translation, rotation and re-scaling of the coordinates $\boldsymbol{U}$. Figure 13 shows an example of the Bookstein transformation in $\mathbb{R}^2$.

Furthermore, if $\boldsymbol{U} \sim \mathcal{N}(\mu, \Sigma)$, then we know that $\boldsymbol{U}^B \sim \mathcal{N}(\mu^B, \Sigma^B)$ where

$$\mu^B = cR(\mu - b), \quad (24)$$
$$\Sigma^B = c^2 R \Sigma R^T. \quad (25)$$

A.2   Bookstein coordinates in $\mathbb{R}^3$

Section 4.3.3 of Dryden and Mardia (1998) gives the Bookstein transformation for $\boldsymbol{U}$ where $u_i \in \mathbb{R}^3$. Following from (23) we first set $u_1^B = (-1/2, 0, 0), u_2^B = (1/2, 0, 0)$ and $u_3^B = (u_{31}^B, u_{32}^B, 0)$. Then for $i = 4, 5, \ldots, N$ and $l = 1, 2, 3$ we calculate

$$w_{il} = u_{il} - \frac{1}{2}(u_{1l}^B + u_{2l}^B). \quad (26)$$
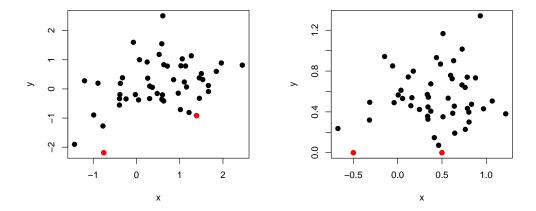
Figure 13: Bookstein transformation in $\mathbb{R}^2$. Left: original coordinates. Right: transformed Bookstein coordinates. The points highlighted in red are mapped to $(-1/2, 0)$ and $(1/2, 0)$.

The Bookstein coordinate u_i^B for $i = 4, 5, \ldots, N$ is then given by

$$u_i^B = R_1 R_2 R_3(w_{i1}, w_{i2}, w_{i3})/D_{12} \tag{27}$$

where

$$R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix}, \qquad R_2 = \begin{bmatrix} \cos(\omega) & 0 & \sin(\omega) \\ 0 & 1 & 0 \\ -\sin(\omega) & 0 & \cos(\omega) \end{bmatrix} \tag{28}$$

$$R_3 = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad D_{12} = 2(w_{21}^2 + w_{22}^2 + w_{23}^2)^{1/2}. \tag{29}$$

Furthermore, we have

$$\theta = \arctan(w_{22}/w_{21}) \tag{30}$$

$$\omega = \arctan(w_{23}/(w_{21}^2 + w_{22}^2)^{1/2}) \tag{31}$$

$$\phi = \arctan\left(\frac{(w_{21}^2 + w_{22}^2)w_{33} - (w_{21}w_{31} + w_{22}w_{32})w_{23}}{(w_{21}^2 + w_{22}^2 + w_{23}^2)^{1/2}(w_{21}w_{32} - w_{31}w_{22})} \right). \tag{32}$$

We see that the transformation in $\mathbb{R}^3$ is more involved than in $\mathbb{R}^2$ since we need to consider the effect of rotations over three different axes. Note that R_1, R_2 and R_3 correspond to a rotation around the x, y and z axes, respectively.

B   Modifying the Hyperedge Indicators

In the generative model detailed in Algorithm 1, the indicators for all order k hyperedges are modified with probability φ_k. Since the probability φ_k is small, we expect a small number of the $\binom{N}{k}$ possible order k hyperedges to be modified and so naively simulating a Bernoulli(φ_k)

random variable for each hyperedge is computationally wasteful. Here we discuss an alternative approach for this step which avoids considering all possible hyperedges, and we comment that this can easily be adapted for the model detailed in Algorithm 2.

Instead of sampling $\binom{N}{k}$ Bernoulli random variables, we instead begin by sampling the number of order k hyperedges whose indicator we modify, n_k, from a Binomial $\left(\binom{N}{k}, \varphi_k\right)$. Then, we randomly sample n_k hyperedges from $\mathcal{E}_{N,k}$. When sampling a hyperedge, we want to sample an index from $\{i_1 < i_2 < \cdots < i_k | i_1, i_2, \ldots, i_k \in \{1, 2, \ldots, N\}\}$ and we do this by sampling i_1 to i_k in incrementally. This avoids sampling from the $\binom{N}{k}$ possible combinations directly and so is more efficient.

We will now discuss this in more detail for hyperedges of order $k = 2$. To sample indices i_1 and i_2 such that $i_1 < i_2$ we

1. Sample i_1 with probability $p(i_1) = \dfrac{N - i_1}{\sum_{i=1}^{N-1}(N - i)}, \quad$ for $i_1 = 1, \ldots, (N - 1)$.

2. Sample $i_2 | i_1$ with probability $p(i_2 | i_1) = \dfrac{1}{N - i_1}, \quad$ for $i_2 = (i_1 + 1), \ldots, N$.

A similar procedure can be used for arbitrary k.

Note that this procedure ignores the dependence between samples since, once a hyperedge is sampled, the remaining hyperedges are sampled from a subset of hyperedges of size $\binom{N}{k} - 1$. However, we expect this effect to be negligible since the majority of hyperedges are not modified.

C   Conditional Posterior Distributions

C.1   Conditional posterior for μ

The conditional posterior for μ is given by

$$p(\mu | \boldsymbol{U}, \Sigma, m_\mu, \Sigma_\mu) \propto p(\boldsymbol{U} | \mu, \Sigma) p(\mu | m_\mu, \Sigma_\mu) \tag{33}$$

where $p(\mu | m_\mu, \Sigma_\mu) = \mathcal{N}(m_\mu, \Sigma_\mu)$ and $p(\boldsymbol{U} | \mu, \Sigma) = \prod_{i=1}^{N} \mathcal{N}(u_i | \mu, \Sigma)$.

We have

$$p(\mu | \boldsymbol{U}, \Sigma, m_\mu, \Sigma_\mu) \propto \exp\left\{ -\frac{1}{2} \sum_{i=1}^{N} (u_i - \mu)^T \Sigma^{-1} (u_i - \mu) - \frac{1}{2}(\mu - m_\mu)^T \Sigma_\mu^{-1} (\mu - m_\mu) \right\} \tag{34}$$

By recursively applying the result in Section of 8.1.7 Petersen and Pedersen (2012) we obtain

$$p(\mu | \boldsymbol{U}, \Sigma, m_\mu, \Sigma_\mu) = \mathcal{N}\left((N\Sigma^{-1} + \Sigma_\mu^{-1})^{-1} \left(\Sigma^{-1} \sum_{i=1}^{N} u_i + \Sigma_\mu^{-1} m_\mu \right), (N\Sigma^{-1} + \Sigma_\mu^{-1})^{-1} \right). \tag{35}$$

C.2   Conditional posterior for Σ

The conditional posterior for Σ is given by

$$p(\Sigma | \boldsymbol{U}, \mu, \Phi, \nu) \propto p(\boldsymbol{U} | \mu, \Sigma) p(\Sigma | \Phi, \nu) \tag{36}$$

where $p(\Sigma | \Phi, \nu) = \mathcal{W}^{-1}(\Phi, \nu)$ and $p(\boldsymbol{U} | \mu, \Sigma) = \prod_{i=1}^{N} \mathcal{N}(u_i | \mu, \Sigma)$.

We have

$$p(\Sigma|\boldsymbol{U},\mu,\Phi,\nu) \propto \prod_{i=1}^{N} \frac{1}{|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(u_i-\mu)^t\Sigma^{-1}(u_i-\mu)\right\} |\Sigma|^{-(\nu+d+1)/2} \exp\left\{-\frac{1}{2}tr\left(\Phi\Sigma^{-1}\right)\right\}$$
(37)

$$\propto |\Sigma|^{-(\nu+d+N+1)/2} \exp\left\{-\frac{1}{2}\left[tr\left(\Sigma^{-1}\sum_{i=1}^{N}(u_i-\mu)(u_i-\mu)^T\right)+tr(\Phi\Sigma^{-1})\right]\right\}$$
(38)

$$\propto |\Sigma|^{-(\nu+d+N+1)/2} \exp\left\{-\frac{1}{2}tr\left(\left[\sum_{i=1}^{N}(u_i-\mu)(u_i-\mu)^T+\Phi\right]\Sigma^{-1}\right)\right\}.$$
(39)

Line (39) follows from the symmetry of Σ and $\sum_{i=1}^{N}(v_i-\mu)(v_i-\mu)^T$, and properties of the trace operator.

Hence, we obtain

$$p(\Sigma|\boldsymbol{U},\mu,\Phi,\nu) = \mathcal{W}^{-1}\left(\Phi+\sum_{i=1}^{N}(u_i-\mu)(u_i-\mu)^T, \nu+N\right).$$
(40)

C.3 Conditional posterior for $\psi_k^{(0)}$

The conditional posterior for $\psi_k^{(0)}$ is given by

$$p\left(\psi_k^{(0)}|\boldsymbol{U},\boldsymbol{r},h_{N,K},a_k^{(0)},b_k^{(0)}\right) \propto \mathcal{L}\left(\boldsymbol{U},\boldsymbol{r},\boldsymbol{\psi}^{(1)},\boldsymbol{\psi}^{(0)};h_{N,K}\right) p\left(\psi_k^{(0)}|a_k^{(0)},b_k^{(0)}\right)$$
(41)

where $\mathcal{L}\left(\boldsymbol{U},\boldsymbol{r},\boldsymbol{\psi}^{(1)},\boldsymbol{\psi}^{(0)};h_{N,K}\right)$ is as in (13) and $p\left(\psi_k^{(0)}|a_k^{(0)},b_k^{(0)}\right) = \text{Beta}\left(a_k^{(0)},b_k^{(0)}\right)$.

We have

$$p(\psi_k^{(0)}|\boldsymbol{U},\boldsymbol{r},h_{N,K},a_k^{(0)},b_k^{(0)})$$
$$\propto \left(\psi_k^{(0)}\right)^{d_k^{(01)}(g_{N,K}(\boldsymbol{U},\boldsymbol{r}),h_{N,K})} \left(1-\psi_k^{(0)}\right)^{d_k^{(00)}(g_{N,K}(\boldsymbol{U},\boldsymbol{r}),h_{N,K})} \left(\psi_k^{(0)}\right)^{a_k^{(0)}-1} \left(1-\psi_k^{(0)}\right)^{b_k^{(0)}-1}$$
(42)

$$\propto \left(\psi_k^{(0)}\right)^{d_k^{(01)}(g_{N,K}(\boldsymbol{U},\boldsymbol{r}),h_{N,K})+a_k^{(0)}-1} \left(1-\psi_k^{(0)}\right)^{d_k^{(00)}(g_{N,K}(\boldsymbol{U},\boldsymbol{r}),h_{N,K})+b_k^{(0)}-1}.$$
(43)

Hence, we obtain

$$p(\psi_k^{(0)}|\boldsymbol{U},\boldsymbol{r},h_{N,K},a_k^{(0)},b_k^{(0)})$$
$$= \text{Beta}\left(d_k^{(01)}(g_{N,K}(\boldsymbol{U},\boldsymbol{r}),h_{N,K})+a_k^{(0)}, d_k^{(00)}(g_{N,K}(\boldsymbol{U},\boldsymbol{r}),h_{N,K})+b_k^{(0)}\right).$$
(44)

C.4 Conditional posterior for $\psi_k^{(1)}$

The conditional posterior for $\psi_k^{(1)}$ is given by

$$p\left(\psi_k^{(1)}|\boldsymbol{U},\boldsymbol{r},h_{N,K},a_k^{(1)},b_k^{(1)}\right) \propto \mathcal{L}\left(\boldsymbol{U},\boldsymbol{r},\boldsymbol{\psi}^{(1)},\boldsymbol{\psi}^{(0)};h_{N,K}\right) p\left(\psi_k^{(1)}|a_k^{(1)},b_k^{(1)}\right)$$
(45)

where $\mathcal{L}\left(\boldsymbol{U}, \boldsymbol{r}, \boldsymbol{\psi}^{(1)}, \boldsymbol{\psi}^{(0)}; h_{N,K}\right)$ is as in (13) and $p\left(\psi_k^{(1)}|a_k^{(1)}, b_k^{(1)}\right) = \text{Beta}\left(a_k^{(1)}, b_k^{(1)}\right)$.

We have

$$p(\psi_k^{(1)}|\boldsymbol{U}, \boldsymbol{r}, h_{N,K}, a_k^{(1)}, b_k^{(1)})$$

$$\propto \left(\psi_k^{(1)}\right)^{d_k^{(10)}(g_{N,K}(\boldsymbol{U},\boldsymbol{r}),h_{N,K})} \left(1 - \psi_k^{(1)}\right)^{d_k^{(11)}(g_{N,K}(\boldsymbol{U},\boldsymbol{r}),h_{N,K})} \left(\psi_k^{(1)}\right)^{a_k^{(1)}-1} \left(1 - \psi_k^{(1)}\right)^{b_k^{(1)}-1}$$

$$(46)$$

$$\propto \left(\psi_k^{(1)}\right)^{d_k^{(10)}(g_{N,K}(\boldsymbol{U},\boldsymbol{r}),h_{N,K})+a_k^{(1)}-1} \left(1 - \psi_k^{(1)}\right)^{d_k^{(11)}(g_{N,K}(\boldsymbol{U},\boldsymbol{r}),h_{N,K})+b_k^{(1)}-1}. \qquad (47)$$

Hence, we obtain

$$p(\psi_k^{(1)}|\boldsymbol{U}, \boldsymbol{r}, h_{N,K}, a_k^{(1)}, b_k^{(1)})$$

$$= \text{Beta}\left(d_k^{(10)}(g_{N,K}(\boldsymbol{U},\boldsymbol{r}), h_{N,K}) + a_k^{(1)}, d_k^{(11)}(g_{N,K}(\boldsymbol{U},\boldsymbol{r}), h_{N,K}) + b_k^{(1)}\right). \qquad (48)$$

D   MCMC initialisation

For the MCMC scheme in Algorithm 3, a random initialisation is likely to perform poorly. Here we discuss our approach for initialising the MCMC scheme, and we begin with the the latent coordinates $\boldsymbol{U}$.

In Sarkar and Moore (2006), the authors present a procedure for inferring the latent coordinates in the scenario where the network is temporally evolving. Their scheme begins with an initialisation which uses generalised multidimensional scaling (GMDS). Traditional MDS (see Cox and Cox (2000)) finds a set of coordinates in $\mathbb{R}^d$ whose pairwise distances correspond to a distance matrix specified as an input. In GMDS, the distance is extended beyond the Euclidean distance and, in our context, we use the shortest path between nodes i and j as the distance measure. To calculate the shortest paths we introduce a weighted adjacency matrix which incorporates the intuition than nodes which appear in a hyperedge are likely closer than nodes which are only connected by a pairwise edge. Once we have an initial value of $\boldsymbol{U}$ we then transform these coordinates onto the Bookstein space of coordinates. Our initialisation procedure for $\boldsymbol{U}$ is given in Algorithm 4.

Algorithm 4 Initialise $\boldsymbol{U}$.

Input: Observed hypergraph $h_{N,K}$
1) Let $A \in \mathbb{R}^{N \times N}$ denote a weighted adjacency matrix.
  For $i, j \in \{1, 2, \ldots, N\}$, if $\{i, j\}$ are connected by a hyperedge
   - let $A_{(i,j)} = 1$ if $\{i, j\}$ are only connected by a hyperedge of order $k = 2$,
   - let $A_{(i,j)} = \lambda$ if $\{i, j\}$ are connected by a hyperedge of order $k > 2$.
2) Find the distance matrix $D \in \mathbb{R}^{N \times N}$, where $D_{(i,j)}$ is the shortest path between
  nodes $\{i, j\}$ in the weighted graph determined by A. For $i = j$, let $D_{(i,j)} = 0$.
3) Apply MDS to D to obtain coordinates $\boldsymbol{U}_0 \in \mathbb{R}^{N \times d}$.
4) Specify the index of the anchor points, and transform $\boldsymbol{U}_0$ onto Bookstein
  coordinates (see Appendix A).

The radii $\boldsymbol{r}$ depend on the scale of $\boldsymbol{U}$, and so they are initialised in terms of $\boldsymbol{U}_0$. Given the initial latent coordinates, $\boldsymbol{r}_0$ is chosen to be the minimum radius which induces all edges that are present in $h_{N,K}$. The noise parameters $\boldsymbol{\psi}^{(0)}$ and $\boldsymbol{\psi}^{(1)}$ are initialised by sampling from their prior, where the prior values suggest that the perturbations are small.

Algorithm 5 Initialise μ and Σ.

Input: Hypergraph $h_{N,K}$, $\boldsymbol{r}_0$, $\boldsymbol{\psi}_0^{(0)}$, $\boldsymbol{\psi}_0^{(1)}$, $\mu \sim \mathcal{N}(m_\mu, \Sigma_\mu)$, $\Sigma \sim \mathcal{W}^{-1}(\Phi, \nu)$, N_{smp} and ϵ.

1) Calculate $T(h_{N,K})$, where $T(\cdot)$ is a vector of hypergraph summary statistics.
   Let $n = 0$.
2) While $n < N_{smp}$
   -Sample $\mu^* \sim \mathcal{N}(m_\mu, \Sigma_\mu)$ and $\Sigma^* \sim \mathcal{W}^{-1}(\Phi, \nu)$.
   -Sample $u_i^* \sim \mathcal{N}(\mu^*, \Sigma^*)$ for $i = 1, 2, \ldots, N$.
   Let $\boldsymbol{U}^*$ be the $N \times d$ matrix whose i^{th} row is u_i^*.
   -Given initial $\boldsymbol{r}_0$, determine the hypergraph $g_{N,K}(\boldsymbol{U}^*, \boldsymbol{r}_0)$.
   -Let $g_{N,K}^*$ by the hypergraph obtained by modifying $g_{N,K}(\boldsymbol{U}^*, \boldsymbol{r}_0)$ with noise
   $\boldsymbol{\psi}_0^{(0)}$ and $\boldsymbol{\psi}_0^{(1)}$
   -Calculate $T\left(g_{N,K}^*\right)$.
   -If $|T(h_{N,K}) - T(g_{N,K}^*)| < \epsilon$
       Accept samples μ^* and Σ^*.
3) Let μ_0 and Σ_0 be the average of N_{smp} samples.

To initialise the parameters μ and Σ we use an ABC scheme (see Marin et al. (2012) for an overview). In this scheme we first sample μ and Σ from their priors. Conditional on these samples, we then sample a hypergraph. By comparing summary statistics of the sampled hypergraph with the observed hypergraph, we determine whether or not our sampled hypergraph is similar enough to the observed hypergraph. If so, we accept the sampled μ and Σ. We choose the number of hyperedges of order $k = 2$, the number of hyperedges of order $k = 3$ and the number of triangles as our summary statistics. This initialisation scheme is detailed in Algorithm 5 and the full initialisation scheme is given in Algorithm 6.

Algorithm 6 Procedure for initialising MCMC scheme in Algorithm 3.

Input Observed hypergraph $h_{N,K}$
1) Determine $\boldsymbol{U}_0$ by applying Algorithm 4.
2) Let initial radii $\boldsymbol{r}_0$ be the smallest radii which induce all hyperedges observed in
   $h_{N,K}$, conditional on $\boldsymbol{U}_0$.
3) Sample $\boldsymbol{\psi}_0^{(0)}$ and $\boldsymbol{\psi}_0^{(1)}$ from their prior distributions.
4) Sample μ_0 and Σ_0 by applying Algorithm 5.

E Practicalities

To implement the MCMC scheme given in Algorithm 3 there are a number of practical considerations we must address. In this section we comment on these where, in E.1 we discuss an approach for determining the presence of a hyperedge of arbitrary order and in E.2 we discuss efficient evaluation of the likelihood.

E.1 Smallest Enclosing Ball

Here we discuss an approach for determining the presence of a hyperedge conditional on $\boldsymbol{U}$ and $\boldsymbol{r}$. Recall that a hyperedge $e_k = \{i_1, i_2, \ldots, i_k\}$ is present if $B_{r_k}(u_{i_1}) \cap B_{r_k}(u_{i_2}) \cap \cdots \cap B_{r_k}(u_{i_k}) \neq \emptyset$.
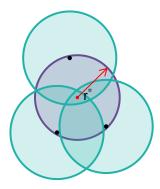
Figure 14: The blue shaded regions correspond to $B_r(u_i)$, for $i = 1, 2, 3$, and the purple shaded region is the smallest enclosing ball of the points. The statements $r^* < r$ and $B_r(u_1) \cap B_r(u_2) \cap B_r(u_3) \neq \emptyset$ are equivalent.

Hence, in order to determine whether $y_{e_k} = 1$, we must find whether or not the sets corresponding to the nodes in the hyperedge have a non-empty intersection.

Alternatively, note that it is equivalent to determine whether the coordinates $\{u_{i_1}, u_{i_2}, \ldots, u_{i_k}\}$ are contained within a ball of radius r_k (see section 3.2 of Edelsbrunner and Harer (2010)). Figure 14 shows this for an example with $k = 3$. This means that we can rephrase the above into the following procedure.

1. Determine the smallest enclosing ball B for the coordinates $\{u_{i_1}, u_{i_2}, \ldots, u_{i_k}\}$.

2. If the radius of B is less than r_k, the hyperedge $e_k = \{i_1, i_2, \ldots, i_k\}$ is present in the hypergraph.

To compute the smallest enclosing ball we can rely on the the miniball algorithm (see Section 3.2 of Edelsbrunner and Harer (2010)), which may be also be referred to as the minidisk algorithm (see Section 4.7 of Berg et al. (2008)). Before providing the algorithmic details, we will first discuss the intuition behind this algorithm. In the discussion below, we will follow the explanation of Edelsbrunner and Harer (2010).

For a set of points, it is clear that a given point is either contained within B or it lies on the boundary of B. The set of points on the boundary entirely determine B and, when the number of points is much larger than the dimension, the chance of a point lying on the boundary is small. Miniball exploits these facts to partition the set of points into those which are contained within B and those which lie on the boundary. The algorithm begins by sampling a point u from the full set of points $u_{i_1 \ldots i_k} = \{u_{i_1}, u_{i_2}, \ldots, u_{i_k}\}$. If u lies within the smallest enclosing ball of $u_{i_1 \ldots i_k} \setminus u$, then we know it lies within B and so it does not influence B. Alternatively, if u lies on the boundary then we must include it in the set which determines B. Miniball iterates over all points in this way to determine the set of points on the boundary. Then, once we have the minimum closing ball B with radius r^*, we check whether $r^* < r_k$ to determine the presence of a hyperedge.

To determine the set of order k hyperedges present in the graph, we rely on the simplicial property of the Čech complex (see Section 2.3). By observing that all subsets of an order k hyperedge must also be connected, we reduce the search space from all possible hyperedges to those whose subsets are present.

We now present the algorithmic details of the miniball algorithm (see Section 3.2 of Edelsbrunner and Harer (2010)). For the edge $e_k = \{i_1, i_2, \ldots, i_k\}$, we let $\sigma_1 \subseteq e_k$ and $\sigma_2 \subseteq e_k$ denote

subsets which partition e_k so that $\sigma_1 \cap \sigma_2 = e_k$. After a pass of the algorithm, the set σ_2 will contain the index of nodes which lie on the boundary of the smallest enclosing ball B. Hence, σ_2 represents the nodes which determine B. We initialise the miniball algorithm with $\sigma_1 = e_k$ and σ_2, and iteratively identify which nodes from σ_1 belong in σ_2. The procedure is given in Algorithm 7.

Algorithm 7 Miniball

---

  1) Set $\sigma_1 = e_k$ and $\sigma_2 = \emptyset$
  2) **if** $\sigma_1 = \emptyset$, compute the miniball B of σ_2
     else choose $u \in \sigma_1$
       -Calculate the miniball B which contains the points $\sigma_1 \setminus u$ in its interior and the
        points σ_2 on its boundary
      -if $u \notin B$, then set B to be the miniball B which contains the points $\sigma_1 \setminus u$
        in its interior and the points $\sigma_2 \cup u$ on its boundary

---

An alternative description of this algorithm can be found in Section 4.7 of Berg et al. (2008), and for efficient implementation of the Čech complex we rely on the GUDHI C++ library (The GUDHI Project (2015)).

E.2   Evaluating $\mathcal{L}(\boldsymbol{U}, \boldsymbol{r}, \boldsymbol{\psi}^{(1)}, \boldsymbol{\psi}^{(0)}; h_{N,K})$

The likelihood given in (13) requires the enumeration of hyperedge discrepancies between the observed hypergraph $h_{N,K}$ and the induced hypergraph $g_{N,K}(\boldsymbol{U}, \boldsymbol{r})$. In this section we note that this does not require a summation over all possible hyperedges, and so can be computed far more efficiently than (12) suggests. We first discuss evaluation of (13), and then observe that (10) can be evaluated in a similar way.

To evaluate the likelihood we have the hyperedges present in $h_{N,K}$ and the hyperedges present in $g_{N,K}(\boldsymbol{U}, \boldsymbol{r})$. In practice, as the data examples from Section 7 suggest, the number of hyperedges in each of these hypergraphs is much smaller than the number of possible hyperedges $\sum_{k=2}^{K} \binom{N}{k}$. Let $n_k^{(h)}$ and $n_k^{(g)}$ denote the number of order k hyperedges in $h_{N,K}$ and $g_{N,K}(\boldsymbol{U}, \boldsymbol{r})$, respectively. To evaluate the likelihood, we begin by enumerating the number of order k hyperedges which are present in both hypergraphs to obtain $d_k^{(11)}(g_{N,K}(\boldsymbol{U}, \boldsymbol{r}), h_{N,K})$. This can easily be computed by evaluating the number of intersection between the hyperedges in $h_{N,K}$ and $g_{N,K}(\boldsymbol{U}, \boldsymbol{r})$. Then, for $k = 2, 3, \ldots, K$, it follows that

$$d_k^{(10)}(g_{N,K}(\boldsymbol{U}, \boldsymbol{r}), h_{N,K}) = n_k^{(g)} - d_k^{(11)}(g_{N,K}(\boldsymbol{U}, \boldsymbol{r}), h_{N,K}), \tag{49}$$

$$d_k^{(01)}(g_{N,K}(\boldsymbol{U}, \boldsymbol{r}), h_{N,K}) = n_k^{(h)} - d_k^{(11)}(g_{N,K}(\boldsymbol{U}, \boldsymbol{r}), h_{N,K}), \tag{50}$$

$$d_k^{(00)}(g_{N,K}(\boldsymbol{U}, \boldsymbol{r}), h_{N,K}) = \binom{N}{k} - \Big[d_k^{(11)}(g_{N,K}(\boldsymbol{U}, \boldsymbol{r}), h_{N,K})$$
$$+ d_k^{(10)}(g_{N,K}(\boldsymbol{U}, \boldsymbol{r}), h_{N,K}) + d_k^{(01)}(g_{N,K}(\boldsymbol{U}, \boldsymbol{r}), h_{N,K}) \Big]. \tag{51}$$

Hence, we are able to avoid summation over all possible hyperedges. We can easily calculate the distance specified in (9) from the above, by observing that it is given by the sum of (49) and (50).

F Proofs for Section 4

F.1 Proof of Proposition 4.1

We have $U_i \sim \mathcal{N}(\mu, \Sigma)$ for $i = 1, 2, \ldots, N$ and $\Sigma = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_d^2)$. Our goal is to obtain an expression for p_{e_2}, and we begin by noting

$$p_{e_2} = P\left(\|U_i - U_j\| \le 2r_2 | \mu, \Sigma\right) = P\left((U_i - U_j)^T (U_i - U_j) \le 4(r_2)^2 | \mu, \Sigma\right). \tag{52}$$

Hence, we consider the distribution of $X_{ij}^T X_{ij}$ where $X_{ij} = U_i - U_j$.

From properties of the Normal distribution, we have that $X_{ij} = U_i - U_j \sim \mathcal{N}(0, 2\Sigma)$ and, from Section 1 of Duchesne and Micheaux (2010), we find

$$X_{ij}^T X_{ij} | \Sigma = \sum_{l=1}^{d} \lambda_l \chi_1^2 \tag{53}$$

where λ_l is the lth eigenvalue of 2Σ. Since Σ is diagonal, the eigenvalues of 2Σ are given by $\lambda_l = 2\sigma_l^2$. Furthermore, since a χ_1^2 distribution is equivalent to a $\Gamma(1/2, 2)$ distribution, we have

$$Z_{ij} | \Sigma = X_{ij}^T X_{ij} | \Sigma \sim \sum_{l=1}^{d} \Gamma\left(\frac{1}{2}, 2(2\sigma_l^2)\right). \tag{54}$$

Hence, we have the result.

F.2 Proof of Lemma 4.1

Recall that $y_{e_k}^{(g)} = 1$ and $y_{e_k}^{(g^*)} = 1$ indicate that the hyperedge e_k is present in g and in g^*, respectively. To begin we consider the probability of e_k being present in g^*. We may observe $y_{e_k}^{(g^*)} = 1$ from either $y_{e_k}^{(g)} = 1$ or $y_{e_k}^{(g)} = 0$. In the first case, we want to keep the state of e_k unaltered and, in the second case, we want to modify the state of the edge. Hence we have

$$P(y_{e_k} = 1^{(g^*)} | \Sigma) = (1 - \varphi_k) P(y_{e_k}^{(g)} = 1 | \Sigma) + \varphi_k (1 - P(y_{e_k}^{(g)} = 1 | \Sigma)). \tag{55}$$

The degree of the ith node with respect to order k hyperedges is obtained by summing over all possible hyperedges e_k that are incident to i. We have

$$\mathrm{Deg}_{(i,k)}^g = \sum_{\{e_k \in \mathcal{E}_{N,k} | i \in e_k\}} y_{e_k}^{(g)} \tag{56}$$

and, in total, there are $\binom{N-1}{k-1}$ possible order k hyperedges that are incident to i. Therefore, it follows that

$$\mathbb{E}\left[\mathrm{Deg}_{(i,k)}^g | \Sigma\right] = \binom{N-1}{k-1} P(y_{e_k} = 1^{(g)} | \Sigma). \tag{57}$$

Note that (57) is valid for dependent probabilities and we rely on this for hyperedges of order $k \ge 3$. As an example, consider the hyperedges $\{i, j, k\}$ and $\{i, j, l\}$ when $k = 3$. It is clear that both of these hyperedges depend on both i and j, and so there is dependence between the hyperedges and the summation in the calculation of $\mathrm{Deg}_{(i,k)}^g$.

Now we consider $\mathbb{E}\left[\text{Deg}^{g^*}_{(i,k)}\right]$. By the law of total expectation, we have

$$\mathbb{E}\left[\text{Deg}^{g^*}_{(i,k)}|\varphi_k, \Sigma\right] = \mathbb{E}\left[\mathbb{E}\left[\text{Deg}^{g^*}_{(i,k)}|\text{Deg}^{g}_{(i,k)}\right]|\varphi_k, \Sigma\right] \tag{58}$$

From (55), it follows that

$$\mathbb{E}\left[\text{Deg}^{g^*}_{(i,k)}|\varphi_k, \Sigma\right] = \mathbb{E}\left[(1-\varphi_k)\text{Deg}^{g}_{(i,k)} + \varphi_k\left(\binom{N-1}{k-1} - \text{Deg}^{g}_{(i,k)}\right)|\varphi_k, \Sigma\right] \tag{59}$$

$$= (1-\varphi_k)\mathbb{E}\left[\text{Deg}^{g}_{(i,k)}|\varphi_k, \Sigma\right] + \varphi_k\left(\binom{N-1}{k-1} - \mathbb{E}\left[\text{Deg}^{g}_{(i,k)}|\varphi_k, \Sigma\right]\right). \tag{60}$$

Using (56), we obtain

$$\mathbb{E}\left[\text{Deg}^{g^*}_{(i,k)}|\varphi_k, \mu, \Sigma\right] = \binom{N-1}{k-1}\left[(1-\varphi_k)P(y_{e_k} = 1^{(g)}|\mu, \Sigma) + \varphi_k(1 - P(y^{(g)}_{e_k} = 1|\mu, \Sigma))\right]. \tag{61}$$

The final result then follows directly Assumption 2.

F.3   Proof of Theorem 4.1

To begin we will consider the degree distribution of hyperedges of order $k = 2$. From (55) we have that the probability of an edge e_2 being present in g^* is given by

$$P\left(y^{(g^*)}_{e_2} = 1|\Sigma, r_2, \varphi_2\right) = (1-\varphi_2)p_{e_2} + \varphi_2(1 - p_{e_2}), \tag{62}$$

where p_{e_2} is the probability that e_2 is present in g.

The degree distribution of the ith node is a sum of independent Bernoulli trials since the edges $\{i, j\}$ and $\{i, k\}$ occur independently given conditioning on i. There are $N - 1$ possible order 2 edges which contain i. Hence it follows that

$$\text{Deg}^{g^*}_{(i,2)}|r_2, \varphi_2, \Sigma \sim \text{Binomial}\left(N - 1, (1-\varphi_2)p_{e_2} + \varphi_2(1 - p_{e_2})\right). \tag{63}$$

The degree of the ith node for order k hyperedges in g^* is given by

$$\text{Deg}^{g^*}_{(i,k)} = \sum_{\{e_k \in \mathcal{E}_{N,k}|i \in e_k\}} y^{g^*}_{e_k}. \tag{64}$$

Now we find the degree distribution for hyperedges of order $k = 3$. Consider the hyperedges $\{i, j, k\}$ and $\{i, j, l\}$ and note that these hyperedges both depend on i and j. It is clear that $P(y^{(g)}_{ijk} = 1)$ and $P(y^{(g)}_{ijl} = 1)$ are not independent and so the argument used for the degree distribution involving hyperedges of order $k = 2$ is no longer appropriate. Now we need to consider the sum of $\binom{N-1}{2}$ dependent Bernoulli trials. When p_{ijk} small, we can approximate this by a Poisson distribution with rate parameter $\lambda = \sum_{\{e_k \in \mathcal{E}_{N,3}|i \in e_k\}} y^{(g^*)}_{e_k}$. However, we note that this result can be extended using results presented in Teerapabolarn (2014)).

G Alternative hypergraph models

G.1 Stasi et al. (2014) model

In this model each node in the hypergraph is assigned a parameter which controls its tendency to form edges, and we denote this parameter by β_i, for $i = 1, 2, \ldots, N$. Let $y_{e_k} = 1$ denote the presence of the hyperedge $e_k = \{i_1, i_2, \ldots, i_k\} \subseteq \{1, 2, \ldots, N\}$ for $k \geq 2$. The probability of the hyperedge e_k occurring is then given by

$$p(y_{i_1 i_2 \ldots i_k} = 1) = \frac{\exp\{\beta_{i_1} + \beta_{i_2} + \ldots \beta_{i_k}\}}{1 + \exp\{\beta_{i_1} + \beta_{i_2} + \ldots \beta_{i_k}\}}. \tag{65}$$

Since the hyperedges are assumed to occur independently conditional on $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_N)$, the likelihood is obtained by taking the product of Bernoulli likelihoods over all possible hyperedges $\mathcal{E}_{N,K}$. This likelihood can be shown to belong to the exponential family. Stasi et al. (2014) introduce several variants of this model, however we only rely on the above for the study in Section 6.1.

G.2 Ng and Murphy (2018) model

The model introduced by Ng and Murphy (2018) assumes that hyperedges can be clustered according to their topic and size. In this context, the topic clustering implies that the hyperedges can be partitioned into latent classes and the probability of a node belonging to a hyperedge depends on its latent class. As an example, consider a coauthorship network where papers are represented as hyperedges. We may classify papers according to their academic discipline and impose that certain authors are more likely to contribute to papers within different disciplines. The size clustering is with respect to the hyperedge order, and this allows the model to capture variation in the size of hyperedges. To specify this model, we assume T topic clusters and S size clusters. It is assumed that the ith node belongs to a hyperedge with size label s and topic label t with probability $\alpha_s \phi_{it}$, so that $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_S)$ controls the size clusters and $\boldsymbol{\phi} = \{\phi_{it}\}_{i=1,2,\ldots,N, t=1,2,\ldots,T}$ controls the topic clusters. Additionally, we let $\pi = (\pi_1, \pi_2, \ldots, \pi_T)$ and $\tau = (\tau_1, \tau_2, \ldots, \tau_S)$ denote the prior topic and size assignment probabilities, respectively. To write down the likelihood, we let $x_{ij} = 1$ indicate that the ith node belongs to the jth hyperedge, $z_{jt}^{(1)} = 1$ indicate that the jth hyperedge has topic label t, and $z_{js}^{(2)} = 1$ indicate that the jth hyperedge has size label s. The likelihood is then given by

$$\mathcal{L}(x, z^{(1)}, z^{(2)}; \theta) = \prod_{j=1}^{M} \prod_{t=1}^{T} \prod_{s=1}^{S} \left[\pi_t \tau_s \prod_{i=1}^{N} (\alpha_s \phi_{it})^{x_{ij}} (1 - \alpha_s \phi_{it})^{1-x_{ij}} \right]^{z_{jt}^{(1)} z_{js}^{(2)}}. \tag{66}$$

Finally, to ensure the model is identifiable, we set $\alpha_S = 1$. Ng and Murphy (2018) also introduce a version of this model which only assumes a topic clustering, but we do not use this for our study in Section 6.1.

H Simulation set-up for Section 6.1

The model parameters for the cases described in Table (1) in Section 6.1 are provided in Table 2.
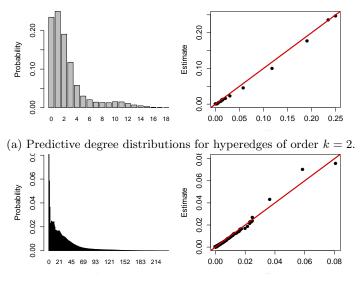
Model	Case	Parameters
Stasi et al. (2014)	1) All nodes equally likely to form connections	$\beta_i = -1.4$ for $i = 1, 2, \ldots, N$
	2) Some nodes more likely to form connections	$\boldsymbol{\beta} = (-0.5, -0.53, \ldots, -1.97, -2)$
Ng and Murphy (2018)	1) Hyperedges in a single cluster	$G = K = 1$, $a = 1$, $\phi_{i1} = 0.075$, $\pi = b1, \tau = 1$
	2) Distinct topic clusters only	$G = 3, K = 1$, $a = 1, \phi_{i1} = 0.25$ for $i \in \mathcal{A}$, $\phi_{i2} = 0.25$ for $i \in \mathcal{B}$, $\phi_{i3} = 0.25$ for $i \in \mathcal{C}$, $\pi = (1/3, 1/3, 1/3), \tau = 1$
	3) Distinct size clusters only	$G = 1, K = 3$, $a = (0.2, 0.5, 1)$, $\phi_{i1} = 0.15, \pi = 1, \tau = (1/3, 1/3, 1/3)$
	4) Fuzzy topic clusters	$G = 2, K = 3$, $a = (0.4, 1), \phi_{i1} = 0.3$ for $i \in \mathcal{A}$, $\phi_{i2} = 0.3$ for $i \in \mathcal{B}$, $\phi_{i1} = \phi_{i2} = 0.2$ for $i \in \mathcal{C}$, $\pi = (1/2, 1/2), \tau = (1/3, 1/3, 1/3)$
LSH	1) Strongly correlated Σ	$\boldsymbol{r} = (0.18, 0.3, 0.35), \mu = (0, 0)$, $\Sigma = 0.25 \left(\begin{smallmatrix} 1 & 0.9 \\ 0.9 & 1 \end{smallmatrix}\right)$, $\boldsymbol{\psi}_0 = (0.01, 0.01, 0.01)$, $\boldsymbol{\psi}_1 = (0.01, 0.01, 0.01)$
	2) No correlation in Σ	$\boldsymbol{r} = (0.18, 0.3, 0.35), \mu = (0, 0)$, $\Sigma = 0.25 \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$, $\boldsymbol{\psi}_0 = (0.01, 0.01, 0.01)$, $\boldsymbol{\psi}_1 = (0.01, 0.01, 0.01)$
	3) Dense in e_2, sparse in e_3, e_4	$\boldsymbol{r} = (0.2, 0.3, 0.35), \mu = (0, 0)$, $\Sigma = 0.25 \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$, $\boldsymbol{\psi}_0 = (0.01, 0.01, 0.01)$, $\boldsymbol{\psi}_1 = (0.01, 0.5, 0.01)$
	4) Sparse e_2, e_4, dense in e_3	$\boldsymbol{r} = (0.1, 0.35, 0.4), \mu = (0, 0)$, $\Sigma = 0.25 \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$, $\boldsymbol{\psi}_0 = (0.01, 0.01, 0.01)$, $\boldsymbol{\psi}_1 = (0.01, 0.01, 0.01)$
	5) Increase latent dimension from $d = 2$ to $d = 3$	$\boldsymbol{r} = (0.18, 0.3, 0.35), \mu = (0, 0)$, $\Sigma = 0.25 \left(\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right)$, $\boldsymbol{\psi}_0 = (0.01, 0.01, 0.01)$, $\boldsymbol{\psi}_1 = (0.01, 0.01, 0.01)$

Table 2: Cases for each hypergraph model considered in the model depth comparison study. The case numbers correspond to the labels in Figures 7a, 7b and 7c. For all cases set $N = 50$ and, where appropriate, $K = 4$.

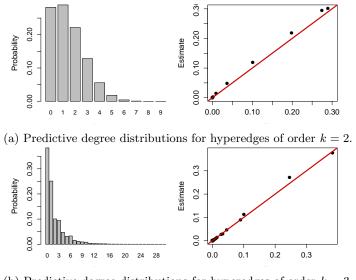# I   Prior and Posterior Predictive Degree Distributions

Here we present additional plots for the simulation study detailed in Section 6.2. Figure 15 compares the predictives for the connections between the observed nodes and the newly simulated nodes, and Figure 16 compares the predictives for connections occurring among the newly simulated nodes only. In both figures, we see a close correspondence between the prior and predictive

distributions.

(a) Predictive degree distributions for hyperedges of order $k = 2$.

(b) Predictive degree distributions for hyperedges of order $k = 3$.

Figure 15: Comparison of prior and posterior predictive degree distributions for hyperedges occurring between the $N^* = 10$ newly simulated nodes and the nodes of the observed hypergraph. Left: the prior predictive degree distribution. Right: qq-plot of the prior and posterior predictive degree distributions.

(a) Predictive degree distributions for hyperedges of order $k = 2$.

(b) Predictive degree distributions for hyperedges of order $k = 3$.

Figure 16: Comparison of prior and posterior predictive degree distributions for hyperedges occurring between the $N^* = 10$ newly simulated nodes only. Left: the prior predictive degree distribution. Right: panel shows a qq-plot of the prior and posterior predictive degree distributions.

Index	$\boldsymbol{U}$ misspecification	$\boldsymbol{r}$ misspecification
1	None	None
2	$u_i \sim \sum_{c=1}^{C} \lambda_c \mathcal{N}(\mu_c, \Sigma_c)$, $C = 2$ and distinct clusters	None
3	$u_i \sim \sum_{c=1}^{C} \lambda_c \mathcal{N}(\mu_c, \Sigma_c)$, $C = 2$ and fuzzy clusters	None
4	u_i sampled from a homogeneous Poisson point process	None
5	None	Simplicial: $r_k = r_{k-1}$
6	$u_i \sim \sum_{c=1}^{C} \lambda_c \mathcal{N}(\mu_c, \Sigma_c)$, $C = 2$ and distinct clusters	Simplicial: $r_k = r_{k-1}$
7	$u_i \sim \sum_{c=1}^{C} \lambda_c \mathcal{N}(\mu_c, \Sigma_c)$, $C = 2$ and fuzzy clusters	Simplicial: $r_k = r_{k-1}$
8	u_i sampled from a homogeneous Poisson point process	Simplicial: $r_k = r_{k-1}$

Table 3: Types of misspecification.

J   Model Misspecification

In this section we consider the robustness of our modelling approach under different cases of misspecification. This allows us to assess the suitability of our model for a range of hypergraph data which do not satisfy our modelling assumptions. We begin by detailing the types of misspecification we consider and then we describe the set up for the simulation study. Finally, we present our results.

In the specification of our model, we assume behaviour on the latent coordinates, the radii and the noise parameters which control the hyperedge modifications. To design the study, we consider alternative mechanisms which may have generated the observed hypergraphs. The types of misspecification we consider for each of these parameters are detailed in Table 3. We comment here that alternative cases of misspecification would be interesting to explore, such as node specific radii or non-homogeneous errors, however there are practical limitations which need to be addressed.

For each of the 8 cases described in Table 3, we simulate a hypergraph with $N = 50$ and $K = 3$ and fit our model. Then, using a similar procedure to the one used in Section 6.2 with $N^* = 10$, we explore the predictive distributions. We compare the predictives obtained by simulating from the true model and the estimated model. In particular, we record

- The degree distribution.

- The number of occurrences of the motifs depicted in Figures 6b3, 6a3, 6a4 and 6a5.

- The density of order 3 hyperedges.

- The number of clusters suggested by the latent representation, as obtained by the gap statistic of a k-means clustering using the cluster package in R (see Maechler et al. (2019)).

The results of our study are presented in Figure 17, where the ith row corresponds to the ith case detailed in Table 3. We generally observe a close correspondence between the prior and predictive distributions, however we see an overall poorer performance when the simulated hypergraphs are simplicial. It is interesting to note that, although there is generally a good fit in terms of the degree distribution, we see many of the motifs are over or under predicted by the posterior predictives. This reflects the contraints implied by the latent representation. Finally, there is little agreement in the number of clusters for all cases and this is likely due to variability in the gap statistic approach. From this study it is clear that our model imposes certain properties on the predictive distributions and it is important to verify whether or not these are appropriate for a specific data example.
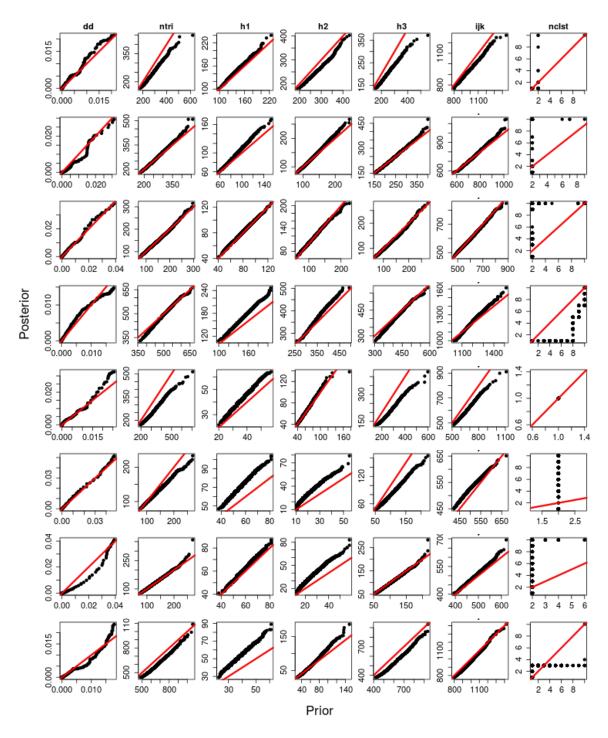
Figure 17: Summary of misspecification simulation study. Left to right: average degree distribution, number of triangles (Figure 6b3, number of Figure 6a3, number of Figure 6a4, number of Figure 6a5, density of order 3 hyperedges and number of clusters in latent representation. Each row corresponds to the misspecification cases summarised in Table 3. The y axes and x axes show the quantiles of the posterior and prior predictives, respectively, and the red lines correspond to the line $y = x$.
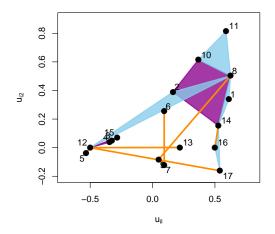
41

K Corporate Leadership Data Example
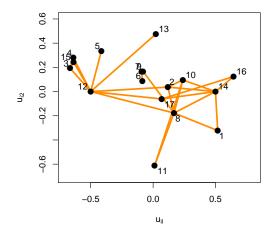
In this section we consider a dataset describing person-company leadership in which individuals are represented as nodes and hyperedges connect individuals who have held a position of leadership in the same company. This dataset is available from KONECT (2017), and to analyse this dataset we remove hyperedges of order larger than 4. We comment that our model can in theory express hyperedges of arbitrary order, however larger values of K become increasingly computationally challenging in our framework. Finally, we remove nodes with degree 0 to obtain a hypergraph with $N = 17$ nodes.

As noted in Section 1, many hypergraph datasets are analysed in terms of the graph obtained by connecting nodes i and j if they are contained within the same hyperedge. This results in a loss of information and in this section we consider the effect of this in more detail. There are a multitude of ways in which we can assume the hypergraph connections given the graph, and we focus on two in this section. We consider the hypergraph obtained by representing each maximal clique as a hyperedge and the simplicial hypergraph obtained by representing each clique by a hyperedge. For convenience, we refer to each of these hypergraphs as the maximal clique hypergraph and simplicial clique hypergraph, respectively. To investigate the effect of these choices on predictive inference, we fit the model detailed in Algorithm 1 to the hypergraph with $K = 4$, the projected graph with $K = 2$, the maximal clique hypergraph with $K = 4$ and the simplicial clique hypergraph with $K = 4$. As in Section 7.1, we constrain φ_k to be at most 20% of the observed density of order k hyperedges.

The posterior mean of the latent coordinates for each hypergraph are shown in Figure 18. We observe several similarities between the latent representations, such as the placement of the set of nodes $\{3, 4, 15, 12\}$, however the constraints imposed on the latent coordinates differ for each case. For instance, comparing Figures 18a and 18b, we see that the hyperedge $\{1, 8, 14\}$ is expressed differently due to the respective geometric constraints. Furthermore, there is a clear difference between the density of order 2 hyperedges in these cases which will have a significant impact on predictive inference. For this dataset, the maximal clique hypergraph is very similar to the observed hypergraph and we note that this will not be true in general. In fact there exist many possible hypergraph whose projected graph corresponds to the graph in Figure 18b. Finally, we comment that the simplicial clique hypergraph in Figure 18d imposes the strongest constraints on the latent representation due to the density of order 2 and order 3 hyperedges.
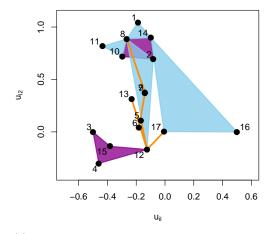
To further compare these fitted models we consider predictive inference for $N^* = 5$ newly simulated nodes. We focus on the motifs shown in Figure 6b and Figures 6a3, 6a4 and 6a5, and the predictive motif counts are shown in Figure 19 for each model fit. First, we will discuss in predictive inference for the motifs involving order 2 edges, namely m_1 (Figure 6b1), m_2 (Figure 6b2) and triangles (Figure 6b3). We see a clear similarity between the predictives for the graph and simplicial clique hypergraph, and the observed hypergraph and maximal clique hypergraph. As commented above, this can be explained by the particular characteristics of the observed hypergraph and this will not be the case in general. It is important to note that the edges in the graph and the order 2 hyperedges in the hypergraphs have a different interpretation, and so it may not be appropriate to make a direct comparison. However, it is clear that there is a large difference between the number of predicted motifs in for the observed hypergraph and the graph. We now consider the motifs involving order 3 edges, namely h_1 (Figure 6a3), h_2 (Figure 6a4) and h_3 (Figure 6a5). Since the graph contains no information about hyperedges, we cannot predict the occurrence of these motifs. There is a large difference between the predictives for the maximal clique hypergraph and the simplicial clique hypergraph. Generally speaking, it is not possible to make accurate inference about the observed hypergraph from these hypergraphs. However, we see a similarity between the maximal clique hypergraph and observed hypergraph
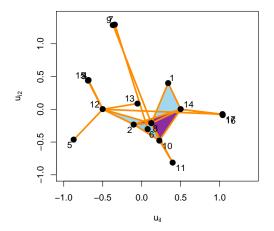
(a) Posterior mean of latent coordinates for observed hypergraph.

(b) Posterior mean of latent coordinates for pairwise projected graph.

(c) Posterior mean of latent coordinates for hypergraph induced from maximal cliques.

(d) Posterior mean of latent coordinates for simplicial hypergraph induced from cliques.

Figure 18: Posterior mean of the latent coordinates after 25000 post burn-in iterations. Figure 18a: observed hypergraph with $K = 4$. Figure 18b: graph obtained by connecting nodes if they are contained within the same observed hyperedge. Figure 18c: hypergraph obtained from representing maximal cliques in the graph by a hyperedge. Figure 18d: simplicial hypergraph obtained by representing cliques in the graph by a hyperedge. Connections in orange, blue and purple correspond to hyperedges of order $k = 2, 3$ and 4, respectively.

predictives in this case. Since our model expresses non-simplicial hypergraphs, we have non zero predictives for the motifs h_1 and h_2 for each hypergraph dataset. However, in the simplicial clique hypergraph, we find that the motif h_1 is relatively less likely to occur. This reflects a tendency for simplicial relationships.
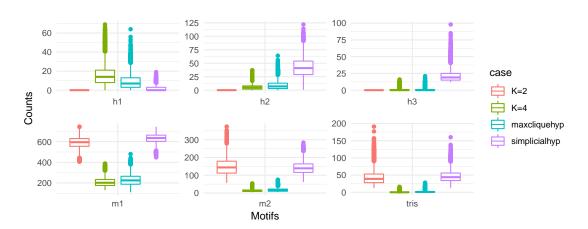
Figure 19: Predictive distributions for motif counts for $N^* = 5$ newly simulated nodes. Top row: predictives for the motifs shown in Figures 6a3, 6a4 and 6a5. Bottom row: predictives for the motifs shown in Figures 6b1, 6b2 and 6b3.