

# Advancements in Latent Space Network Modelling

Kathryn Ruth Turnbull, MMath (Hons), MRes



Submitted for the degree of Doctor of  
Philosophy at Lancaster University.

December 2019

# Abstract

The ubiquity of relational data has motivated an extensive literature on network analysis, and over the last two decades the latent space approach has become a popular network modelling framework. In this approach, the nodes of a network are represented in a low-dimensional latent space and the probability of interactions occurring are modelled as a function of the associated latent coordinates. This thesis focuses on computational and modelling aspects of the latent space approach, and we present two main contributions.

First, we consider estimation of temporally evolving latent space networks in which interactions among a fixed population are observed through time. The latent coordinates of each node evolve over time and this presents a natural setting for the application of sequential monte carlo (SMC) methods. This facilitates online inference which allows estimation for dynamic networks in which the number of observations in time is large. Since the performance of SMC methods degrades as the dimension of the latent state space increases, we explore the high-dimensional SMC literature to allow estimation of networks with a larger number of nodes.

Second, we develop a latent space model for network data in which the interactions occur between sets of the population and, as a motivating example, we consider a coauthorship network in which it is typical for more than two authors to contribute to an article. This type of data can be represented as a hypergraph, and we extend the latent space framework to this setting. Modelling the nodes in a latent space provides a convenient visualisation of the data and allows properties to be imposed on the

hypergraph relationships. We develop a parsimonious model with a computationally convenient likelihood. Furthermore, we theoretically consider the properties of the degree distribution of our model and further explore its properties via simulation.

# Acknowledgements

I would like to thank my supervisors Christopher Nemeth, Simón Lunagómez and Matthew Nunes for their support and guidance throughout my PhD. Your encouragement has helped me to grow as a researcher and has been instrumental to the completion of this thesis. Thank you also to Edoardo Airoidi who contributed to the work in Chapter 4 and my external supervisor Tyler McCormick.

My PhD has been funded through the EPSRC STOR-i CDT and I would like to thank all of those involved in STOR-i. I feel very privileged to have been able to complete my PhD in such a dynamic, friendly and supportive environment which has awarded me many exciting opportunities and experiences. I would especially like to thank the directors and support staff for making STOR-i possible.

I am also thankful for the support and kindness my friends have given me throughout my time at Lancaster. In particular, I would like to mention Lucy, Seán, Jake, Harry, Grace, Argenis, my STOR-i cohort and the friends I have made from LLC. Thank you especially to Wojtek for your seemingly endless supply of patience, support and encouragement. It would not have been the same without you.

Finally, thank you to my family. You have supported me in all my endeavours and I am eternally grateful for all that you have done for me.

# Declaration

I declare that the work in this thesis has been done by myself and has not been submitted elsewhere for the award of any other degree.

Kathryn Ruth Turnbull

# Contents

<b>Abstract</b>	<b>I</b>
<b>Acknowledgements</b>	<b>III</b>
<b>Declaration</b>	<b>IV</b>
<b>Contents</b>	<b>VIII</b>
<b>List of Figures</b>	<b>XIV</b>
<b>List of Tables</b>	<b>XV</b>
<b>List of Abbreviations</b>	<b>XVI</b>
<b>List of Symbols</b>	<b>XVII</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Notation and Definitions . . . . .	2
1.2 Statistical Network Modelling . . . . .	3
1.3 Contributions and Thesis Outline . . . . .	7
<b>2 Review of Latent Space Network Modelling</b>	<b>10</b>
2.1 Introduction . . . . .	10
2.2 Latent Space Network Modelling . . . . .	11
2.2.1 Generic Model . . . . .	11
2.2.2 Distance and Projection Model . . . . .	12

<i>CONTENTS</i>	VI
2.2.3 Inference and Computation . . . . .	14
2.2.4 Extensions . . . . .	15
2.3 Exploration of Properties . . . . .	20
2.3.1 Effect of Metric and $f_U$ . . . . .	20
2.3.2 Effect of Covariance for Normally Distributed $\mathbf{U}$ . . . . .	21
<b>3 Sequential Monte Carlo and Dynamic Latent Space Networks</b>	<b>23</b>
3.1 Introduction . . . . .	23
3.2 Sequential Monte Carlo . . . . .	24
3.2.1 State Space Model . . . . .	25
3.2.2 Particle Filter . . . . .	26
3.2.3 Parameter estimation . . . . .	29
3.2.4 High-Dimensional SMC . . . . .	32
3.3 Dynamic Latent Space Network Modelling . . . . .	33
3.3.1 Background . . . . .	33
3.3.2 SSM formulation . . . . .	34
3.3.3 Identifiability . . . . .	35
3.3.4 Model Fitting . . . . .	36
3.3.5 Extensions . . . . .	38
3.4 SMC and Dynamic Latent Space Networks . . . . .	38
3.4.1 State Estimation . . . . .	39
3.4.2 State and Parameter Estimation . . . . .	49
3.5 Simulations . . . . .	51
3.5.1 Alternative Scenarios . . . . .	51
3.5.2 Scalability . . . . .	53
3.6 Classroom Contact Dataset . . . . .	54
3.7 Discussion . . . . .	57
<b>4 Latent Space Modelling of Hypergraph Data</b>	<b>60</b>

4.1	Introduction . . . . .	60
4.2	Background . . . . .	67
4.2.1	Latent Space Network Modelling . . . . .	67
4.2.2	Random Geometric Graphs . . . . .	69
4.2.3	Random Geometric Hypergraphs . . . . .	70
4.3	Latent space hypergraphs . . . . .	72
4.3.1	Motivation . . . . .	73
4.3.2	Combining $k$ -skeletons . . . . .	74
4.3.3	Generative Model and Likelihood . . . . .	76
4.3.4	Extensions . . . . .	78
4.4	Identifiability . . . . .	80
4.5	Posterior Sampling . . . . .	81
4.5.1	MCMC scheme . . . . .	81
4.6	Theoretical Results . . . . .	83
4.6.1	Observations . . . . .	84
4.6.2	Properties of a nsRGH . . . . .	85
4.6.3	Degree Distribution for the $i^{\text{th}}$ Node . . . . .	87
4.7	Simulations . . . . .	89
4.7.1	Model depth comparisons . . . . .	89
4.7.2	Prior Predictive vs Posterior Predictive . . . . .	95
4.7.3	Misspecification . . . . .	99
4.8	Real data examples . . . . .	102
4.8.1	Star Wars: A New Hope . . . . .	102
4.8.2	Corporate Leadership . . . . .	105
4.8.3	Coauthorship for Statisticians . . . . .	109
4.9	Discussion . . . . .	111
<b>5</b>	<b>Conclusions and Further Work</b>	<b>116</b>



<b>A</b>	<b>Appendix for ‘Sequential Monte Carlo and Dynamic Latent Space Networks’</b>	<b>121</b>
A.1	Gradient derivation for Euclidean distance and binary observations . . . . .	121
A.2	Gradient derivation for parameter estimation . . . . .	122
A.3	Data simulation for Section 3.5.1 . . . . .	123
<b>B</b>	<b>Appendix for ‘Latent Space Modelling of Hypergraph Data’</b>	<b>125</b>
B.1	Bookstein coordinates . . . . .	125
B.1.1	Bookstein coordinates in $\mathbb{R}^2$ . . . . .	125
B.1.2	Bookstein coordinates in $\mathbb{R}^3$ . . . . .	126
B.2	Modifying the Hyperedge Indicators . . . . .	127
B.3	Conditional Posterior Distributions . . . . .	128
B.3.1	Conditional posterior for $\mu$ . . . . .	128
B.3.2	Conditional posterior for $\Sigma$ . . . . .	129
B.3.3	Conditional posterior for $\psi_k^{(0)}$ . . . . .	130
B.3.4	Conditional posterior for $\psi_k^{(1)}$ . . . . .	130
B.4	MCMC initialisation . . . . .	131
B.5	Practicalities . . . . .	132
B.5.1	Smallest Enclosing Ball . . . . .	133
B.5.2	Evaluating $\mathcal{L}(\mathbf{U}, \mathbf{r}, \boldsymbol{\psi}^{(1)}, \boldsymbol{\psi}^{(0)}; h_{N,K})$ . . . . .	135
B.6	Proofs for Section 4.6 . . . . .	137
B.6.1	Proof of Proposition 4.6.1 . . . . .	137
B.6.2	Proof of Lemma 4.6.1 . . . . .	137
B.6.3	Proof of Theorem 4.6.1 . . . . .	139
B.7	Prior and Posterior Predictive Degree Distributions . . . . .	139
	<b>Bibliography</b>	<b>142</b>

# List of Figures

1.1.1	Example of a graph with $V = \{1, 2, 3, 4, 5\}$ and $E = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{3, 4\}, \{3, 5\}, \{4, 5\}\}$ . . . . .	2
2.2.1	The likelihood conditional on the latent coordinates (2.2.8) is invariant to distance-preserving transformations of $\mathbf{U}$ . All latent configurations in this Figure have the same likelihood value. . . . .	15
2.3.1	Motifs considered in simulations in Section 2.3. . . . .	20
2.3.2	Summary of simulated networks for the cases described in Table 2.2.1	22
3.2.1	Depiction of the dependence structure in a SSM. Each observation is assumed independent conditional on latent variables which follow a first order Markov process. The dependence on model parameters $\theta$ is assumed throughout. . . . .	25
3.3.1	SSM for dynamic latent space networks. The observed adjacency matrices are modelled independently conditional on the latent coordinates, and the latent coordinates are modelled with a first order Markov process. $\theta$ is a vector of static parameters. . . . .	34
3.4.1	Performance of SIR and APF as $N$ increases. The ESS and average MSE in probability are shown in the left and right panels, respectively. For each filter, the number of particles was fixed at $M = 50000$ . . .	40

3.4.2 Performance of the independent approximation as  $N$  increases. The ESS and average MSE in probability are shown in the left and right panels, respectively. For each filter, the number of particles was fixed at  $M = 1000$ . . . . . 42

3.4.3 Effect of changing  $\gamma$  on the log-likelihood value for different sized networks. Left to right: network of size  $N = 5, 10, 20$  and  $30$ , all with  $d = 2$ . . . . . 45

3.4.4 Performance of nudging within an SIR filter as  $N$  increases for different values of  $\gamma$ . The ESS and average MSE in probability are shown in the left and right panels, respectively. The number of particles was fixed at  $M = 10000$ . . . . . 46

3.4.5 Intermediary steps  $\{x_{\tau_{t,s}}\}_{s=0}^S$  between observations  $y_t$  and  $y_{t+1}$ . . . . . 47

3.4.6 Performance of GIRF as  $N$  increases for varying number of intermediary states  $S$ . The ESS and average MSE in probability are shown in the left and right panels, respectively. The number of particles was fixed at  $M = 1000$ . . . . . 49

3.5.1 Summary of online and offline estimation, shown in blue and orange, respectively. Throughout we have left: (S1), middle: (S2), right: (S3). Figure 3.5.1a shows the effective sample size and Figure 3.5.1b shows the mean square error in probability. Figure 3.5.1c shows the ROC curves for observations 1 to  $T - 1$ , and the line  $y = x$  is shown in red. Figure 3.5.1d shows the ROC curve for the predicted probabilities at time  $T$ , and the ROC curve for the true probabilities is shown in grey. . . . . 52

3.5.2 Performance of online estimation as the dimension of the data increases. 55

3.5.3 Time taken for each filter for increasing  $N$  (left) and increasing  $T$  (right). . . . . 56

3.6.1 Fitted model for a selection of pairs of students. In both plots the blue line represents the mean, the 2.5<sup>th</sup> and 97.5<sup>th</sup> percentile are shown in purple, and the points correspond to the observations. The top plot shows the connection probabilities for binary data and the bottom plot shows the rate for count data. . . . . 57

3.6.2 Left: ROC curve for DLSN model fitted with a dot-product (teal) and Euclidean distance (orange) formulation. The line  $y = x$  is shown in red. Middle: average absolute error (AAE) for binary data simulated from predictive probabilities at time  $T$ . The colours correspond to those in the left plot. Right: average absolute error (AAE) for integer data simulated from predictive rates at time  $T$ . . . . . 58

4.1.1 Figures 4.1.1a and 4.1.1b depict two possible hypergraphs, where a node belongs to a hyperedge if it lies within the associated shaded region. Figure 4.1.1c is the graph obtained by replacing hyperedges in Figure 4.1.1a, or equivalently in Figure 4.1.1b, by cliques. The hypergraphs in 4.1.1a and 4.1.1b cannot be recovered from 4.1.1c. Figure 4.1.1d presents the hypergraph relationships in Figure 4.1.1a as a bipartite graph, where an edge from a population node to a hyperedge node indicates membership of a hyperedge. . . . . 62

4.2.1 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. . . . . 72

4.3.1 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:  $\cup_{k=2}^3 \mathcal{D}_{r_k}^{(k)}$ . . . . . 75

4.6.1 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 (a) and  $\Sigma = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$  in (b), and the theoretical probability for  $k = 2$  is plotted for each case (black, dash). 86

4.6.2 Comparison of theoretical (black, dashed) and simulated (red, solid) degree distribution for a hypergraph with  $N = 20$  and  $K = 3$ . Figures 4.6.2a and 4.6.2b show the degree distribution for hyperedges of order  $k = 2$  and  $k = 3$ , respectively. The theoretical degree distribution for  $k = 3$  is calculated using a monte carlo estimate of  $p_{e_3}$ . . . . . 88

4.7.1 Depiction of motifs considered in Sections 4.7 and 4.8. . . . . 92

4.7.2 Summary of hypergraphs simulated from each of the models considered in Section 4.7.1. The cases considered are summarised in Table 4.7.1. . . . . 96

4.7.3 Comparison of prior and posterior predictive degree distributions for  $N^* = 10$  newly simulated nodes. In each figure, the left panel shows the prior predictive degree distribution, and the right panel shows a qq-plot of the prior and posterior predictive degree distributions. Figures 4.7.3a and 4.7.3b show the degree distributions for hyperedges of order 2 and 3, respectively. . . . . 98

4.7.4 Summary of misspecification simulation study. Left to right: average degree distribution, number of triangles Figure 4.7.1b3, number of Figure 4.7.1a3, number of Figure 4.7.1a4, number of Figure 4.7.1a5 and density of order 3 hyperedges. Each row corresponds to the misspecification cases summarised in Table 4.7.2. The  $y$  and  $x$  axes show the quantiles of the posterior and prior predictives, respectively. The red lines correspond to  $y = x$ . . . . . 101

4.8.1 Posterior mean of latent coordinates for the Star Wars dataset for different upper limits on  $\boldsymbol{\varphi}$ . Connections in orange, blue and purple correspond to hyperedges of order  $k = 2, 3$  and  $4$ , respectively. . . . 103

4.8.2 Predicted degree distributions conditional on the fitted model. Given  $\hat{U}$ ,  $\hat{r}$  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 Figure 4.8.2a and  $1.5$  the hyperedge density in Figure 4.8.2b. The left plots show the degree distribution for “Luke” and the right plots show the degree distribution for “Darth Vader”. The order  $2, 3$ , and  $4$  hyperedges are shown in green, orange, and purple, respectively. . . . . 104

4.8.3 Posterior mean of the latent coordinates after 25000 post burn-in iterations. Figure 4.8.3a: observed hypergraph with  $K = 4$ . Figure 4.8.3b: graph obtained by connecting nodes if they are contained within the same observed hyperedge. Figure 4.8.3c: hypergraph obtained from representing maximal cliques in the graph by a hyperedge. Figure 4.8.3d: 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. . . . 107

4.8.4 Predictive distributions for motif counts for  $N^* = 5$  newly simulated nodes. Top row: predictives for the motifs shown in Figures 4.7.1a3, 4.7.1a4 and 4.7.1a5. Bottom row: predictives for the motifs shown in Figures 4.7.1b1, 4.7.1b2 and 4.7.1b3. . . . . 108

4.8.5 Fitted model for coauthorship example. . . . . 110

4.9.1  $N^* = 19$  predictive subgraph counts. Left to right: motifs depicted in Figure 4.7.1a3, 4.7.1a4, 4.7.1a5, 4.7.1b1, 4.7.1b2 and 4.7.1b3. The red dots correspond to the observed motif counts for the newly sampled nodes. . . . . 112

A.3.1	Summary of parameter estimates. . . . .	124
B.1.1	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 $(1/2, 0)$ . . . . .	126
B.5.1	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. . . . .	134
B.7.1	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. In each figure, the left panel shows the prior predictive degree distribution, and the right panel shows a qq-plot of the prior and posterior predictive degree distributions. Figures B.7.1a and B.7.1b show the degree distributions for hyperedges of order 2 and 3, respectively. . . . .	140
B.7.2	Comparison of prior and posterior predictive degree distributions for hyperedges occurring between the $N^* = 10$ newly simulated nodes only. In each figure, the left panel shows the prior predictive degree distribution, and the right panel shows a qq-plot of the prior and posterior predictive degree distributions. Figures B.7.2a and B.7.2b show the degree distributions for hyperedges of order 2 and 3, respectively. . . . .	141

# List of Tables

2.2.1 Cases considered in Section 2.3 simulation. $p_{ij}$ is given by (2.2.4) where $\eta_{ij} = \alpha + s(u_i, u_j)$ . When the latent coordinates are normally distributed we take $\Sigma_1 = 0.5 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ , $\Sigma_2 = 0.5 \begin{pmatrix} 1 & 0.9 & 0.9 \\ 0 & 0.9 & 0.9 \end{pmatrix}$ and $\Sigma_3 = 2 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ and for all cases we fix $\alpha = 1$ . . . . .	19
4.7.1 Cases for each hypergraph model considered in the model depth comparison study. The case numbers correspond to the labels in Figures 4.7.2a, 4.7.2b and 4.7.2c. For all cases set $N = 50$ and, where appropriate, $K = 4$ . . . . .	94
4.7.2 Types of misspecification. . . . .	99



# List of Abbreviations

<b>ER</b>	Erdős-Rényi
<b>ERGM</b>	Exponential Random Graphs
<b>SBM</b>	Stochastic Blockmodel
<b>LSM</b>	Latent Space Model
<b>MCMC</b>	Markov Chain Monte Carlo
<b>MH</b>	Metropolis-Hastings
<b>SMC</b>	Sequential Monte Carlo
<b>RDPG</b>	Random Dot-Product Graph
<b>SSM</b>	State Space Model
<b>PF</b>	Particle Filter
<b>SIR</b>	Sequential Importance Resampling
<b>APF</b>	Auxiliary Particle Filter
<b>GIRF</b>	Guided Intermediate Resampling Filter
<b>ESS</b>	Effective Sample Size
<b>MSE</b>	Mean Square Error
<b>RGG</b>	Random Geometric Graph
<b>RGH</b>	Random Geometric Hypergraph
<b>nsRGH</b>	Non-Simplicial Random Geometric Hypergraph

# List of Symbols

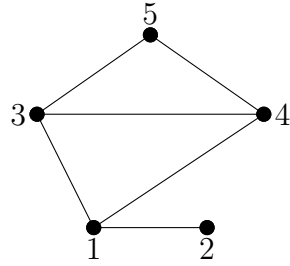
$\mathcal{G}$	Graph
$V$	Node set
$E$	Edge set
$N$	Number of nodes
$d$	Dimension of latent space
$y_{ij}$	Connection between nodes $i$ and $j$
$\mathcal{Y}$	$(N \times N)$ adjacency matrix
$u_i$	Latent coordinate of $i^{\text{th}}$ node in $\mathbb{R}^d$
$\mathbf{U}$	$N \times d$ matrix of latent coordinates
$T$	Number of time points
$\mathcal{Y}_t$	$(N \times N)$ adjacency matrix at time $t$
$\mathbf{U}_t$	$(N \times d)$ matrix of latent coordinates at time $t$
$\theta$	Static parameters
$M$	Hyperedge set
$e_k$	Hyperedge of order $k$
$K$	Maximum hyperedge order
$\mathcal{G}_{N,K}$	Space of hypergraphs on $N$ nodes with maximum hyperedge order $K$
$\mathcal{V}_N$	Set of $N$ nodes
$\mathcal{E}_{N,K}$	Set of possible hyperedges up to order $K$ on $N$ nodes

# Chapter 1

## Introduction

Data describing interactions among a population arise in a diverse range of disciplines, and examples include social relationships among a set of individuals (Zachary (1977)), cooperating regions of the brain (Biswal et al. (2010)) and documents citing one another (Ji and Jin (2016)). Data of this type can be analysed as a network, and the ubiquity of relational data (see Leskovec and Krevl (2014) and Kunegis (2013)) has motivated an extensive literature on network analysis. However, relational data present inferential challenges due to (i) the dependence inherent to the interactions, and (ii) the potentially large number of observations. There currently exists a diverse literature on network analysis which allows properties of an observed network to be characterised, and the exact nature of this depends on the application of interest. For example, we may be interested in the importance of certain members of the population, identifying subsets of the population which exhibit similar patterns of connectivity, or predicting future interactions.

This chapter will provide the necessary background for the remainder of this thesis. Section 1.1 contains basic notation and definitions for network analysis, Section 1.2 overviews the literature on statistical network analysis, and Section 1.3 contains an outline of the thesis and the highlights the contributions.

(a)  $V$  and  $E$ .

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

(b) Adjacency matrix.

Figure 1.1.1: Example of a graph with  $V = \{1, 2, 3, 4, 5\}$  and  $E = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{3, 4\}, \{3, 5\}, \{4, 5\}\}$ .

## 1.1 Notation and Definitions

Network data may be recorded in a number of forms. For example, we may observe binary interactions such as “friends” or “not friends” in the context of social networks, or weighted interactions describing the number of messages shared between two members of the population in the context of communications networks. Alternatively, we may observe interactions between two distinct groups where connections only occur across groups. Below we will provide the basic notation for the simplest case and comment here that these concepts can be adapted accordingly. For a comprehensive discussion, we refer to Newman (2010).

Typically, a network is represented as a graph  $\mathcal{G} = (V, E)$  comprised of a node set  $V$  and edge set  $E$ , where  $V$  indexes the population and  $E$  contains the edges that are present in  $\mathcal{G}$ . For a graph with  $N$  nodes, we have  $V = \{1, 2, \dots, N\}$  and  $E \subseteq \mathcal{E}_N$ , where  $\mathcal{E}_N = \{\{i, j\} | i, j \in V\}$  and  $|\mathcal{E}_N| = 2^N$ . When the pairs  $\{i, j\}$  are unordered, so that  $\{i, j\} \equiv \{j, i\}$ , the graph is *undirected* and when the pairs are ordered the graph is *directed*. An example of an undirected graph is given in Figure 1.1.1a. To express non-binary relationships, we can also associate a set of weights  $\omega = \{\omega_{ij} | \{i, j\} \in E\}$  with the graph  $\mathcal{G}$ . Finally, in a *simple* graph, the edges  $\{i, j\} \in E$  are unique and there are no self-ties so  $\{i, i\} \notin E$ . For the rest of this section we will restrict to simple, undirected graphs with binary connections.

A graph can also be represented by an  $N \times N$  adjacency matrix  $\mathcal{Y}$  whose  $\{i, j\}^{th}$  entry,  $y_{ij}$ , corresponds to the state of the  $(i, j)^{th}$  edge. For an undirected graph  $y_{ij} = y_{ji}$ , otherwise the graph is directed. When the connections are binary, we take  $y_{ij} = 1$  when  $\{i, j\} \in E$  and  $y_{ij} = 0$  otherwise and, since we assume no self ties, we have  $y_{ii} = 0 \forall i = 1, 2, \dots, N$ . The adjacency matrix representation of the graph in Figure 1.1.1a is given in Figure 1.1.1b.

There are a number of properties which may be of interest when analysing an observed network. In an undirected graph, the degree of the  $i^{th}$  node is the number of nodes which share an edge with the  $i^{th}$  node, namely  $Deg_i = \sum_{j \neq i} y_{ij}$ . Given an observed network, the *empirical degree distribution* is given by

$$P(Deg = k) = \frac{\#\{i \in V | Deg_i = k\}}{N}, \quad (1.1.1)$$

which simply denotes the proportion of nodes which have degree exactly equal to  $k$ . Conditional on a generative model, it is often of interest to compare (1.1.1) with the degree distribution derived under the model which describes the probability of observing a node with degree  $k$ .

The *density* of a graph is the proportion of edges which are present and is given by  $\sum_{i,j} y_{ij} / \binom{N}{2}$  and a *subgraph* refers to a smaller graph contained within  $\mathcal{G}$ . Often *motif counts*, in which the number of occurrences of a specific subgraph are considered, are of interest. As an example, there are two occurrences of the triangle subgraph  $\{\{i, j\}, \{j, k\}, \{i, k\}\}$  in Figure 1.1.1a. Note that the triangles  $\{1, 4, 3\}$  and  $\{1, 3, 4\}$  are not considered distinct.

## 1.2 Statistical Network Modelling

The prevalence of network data has motivated a broad literature on network analysis (see Kolaczyk (2009), Barabási and Pósfai (2016) and Newman (2010)). Given an observed network, we wish to characterise and understand the structure of the interactions, and the details of this depend on the application. It is common to take a

model-based approach in which a network can be understood in terms of a specific characteristic or property such as, for example, the generative mechanism from which an observed network arose or the degree distribution. In the statistical literature, inference can then be made for a given model via, for instance, estimation and prediction. In this section we will briefly overview the network modelling literature, and we refer to Salter-Townshend et al. (2012) and Goldenberg et al. (2010) for more detail and further references.

We begin with the Erdős-Rényi (ER) random graph model (Erdős and Rényi (1959)) in which a fixed number of edges are chosen randomly from the  $\binom{N}{2}$  possible edges, so that all graphs with exactly  $N_e$  edges are equally likely to occur. However, a closely related model introduced in Gilbert (1959), in which each edge in an  $N$  node graph occurs independently with probability  $p$ , is more often referred to as the ER random graph model in the modern literature. To sample a graph from this model, we must specify the success probability  $p$  and sample the observations as

$$y_{ij} \sim \text{Bern}(p) \tag{1.2.1}$$

for  $\{i, j\} \in \{1, 2, \dots, N | i < j\}$ . Though these models are well understood, they are typically insufficient for modelling networks which exhibit complex structures. Alternative random graph models were later introduced which allow finer control on certain aspects of a network. For example, the preferential attachment model of Barabási and Albert (1999) models the growth of a graph from a seed graph, and the mechanism which governs how additional nodes join the existing graph allows control over the degree distribution.

Exponential random graph models (ERGMs) (Frank and Strauss (1986) Frank (1991) Wasserman and Pattison (1996)) are a popular class of random graph models. In an ERGM the probability of a network is defined as a function of network statistics, such as the number of triangles, and the likelihood is specified as a member of the exponential family. ERGMs are also referred to as  $p^*$  models and they build upon the  $p_1$  model of Holland and Leinhardt (1981) and the  $p_2$  model of van Duijn et al. (2004).

In the  $p_1$  model, the probability of connection depends on parameters associated with the nodes and this model allows control over the degree distribution. A special case of this is the  $\beta$  model in which the  $i^{\text{th}}$  node is assigned a parameter  $\beta_i \in \mathbb{R}$  and the connections are modelled as

$$p_{ij} = \frac{e^{\beta_i + \beta_j}}{1 + e^{\beta_i + \beta_j}} \quad (1.2.2)$$

$$y_{ij} \sim \text{Bern}(p_{ij}) \quad (1.2.3)$$

for  $\{i, j\} \in \{1, 2, \dots, N | i < j\}$ . The  $p_2$  model extends the  $p_1$  to the setting where the node specific parameters are unknown and estimated as a realisation from an underlying distribution. ERGMs have become popular in the network modelling literature, and we refer to Section 3.6 of Goldenberg et al. (2010) for more details.

The stochastic blockmodel (SBM) of Nowicki and Snijders (2001) was introduced to model graphs which exhibit a community structure. In this model, it is assumed that nodes of the network can be partitioned into  $G$  distinct groups such that the nodes within each group have similar patterns of connectivity. More specifically, we let  $z_i = (z_{i1}, z_{i2}, \dots, z_{iG}) \in \{0, 1\}^G$  denote the community membership of the  $i^{\text{th}}$  node, for  $i = 1, 2, \dots, N$ , and we assume that  $z_i$  contains exactly one nonzero entry so that  $z_{ig} = 1$  indicates that the  $i^{\text{th}}$  node belongs to the  $g^{\text{th}}$  community. The symmetric matrix  $Q \in [0, 1]^{G \times G}$  then defines the probability of connections forming, where the diagonal and off-diagonal entries correspond to within and between community connections, respectively. The connection between nodes  $i$  and  $j$  is then modelled as

$$y_{ij} \sim \text{Bern}(z_i^T Q z_j), \quad (1.2.4)$$

for  $\{i, j\} \in \{1, 2, \dots, N | i < j\}$ , and a generative model can be obtained by placing a distribution on the community assignments. Important variants of the SBM include the degree corrected SBM (Karrer and Newman (2011)) where additional parameters are introduced to control the degree of each node, and the mixed membership SBM (Airoldi et al. (2009)) in which nodes may belong to multiple latent classes.

The latent space framework, as introduced in Hoff et al. (2002), has proven to be a popular network modelling approach. In this framework, low-dimensional latent coordinates are associated with each of the nodes and the probability of a connection forming is modelled as a function of these coordinates. The contributions of this thesis fall within this framework and so we refer to Chapter 2 for an in depth discussion of this approach and the surrounding literature.

Graphons are another important class of network models (Borgs and Chayes (2017), Lovász (2012)), and a graphon is characterised by a function  $W : [0, 1]^2 \rightarrow [0, 1]$  which determines the probability of a connection forming. To sample a graph with  $N$  nodes from a graphon we take

$$x_i \sim \mathcal{U}([0, 1]) \quad \text{for } i = 1, 2, \dots, N \quad (1.2.5)$$

$$y_{ij} \sim \text{Bern}(W(x_i, x_j)) \quad \text{for } \{i, j\} \in \{1, 2, \dots, N | i < j\} \quad (1.2.6)$$

A graphon generalises many existing random graph models and, as an example, the ER model can be specified by taking  $W(x_i, x_j) = p$ . Graphons have been considered in the context of identifying communities in Eldridge et al. (2016) and developing meaningful centrality measures for uncertain networks in Avella-Medina et al. (2018).

The models highlighted so far are vertex exchangeable, meaning that the probability of observing a given graph is invariant to relabelling of the nodes. As a consequence of the Aldous-Hoover theorem (Aldous (1981), Hoover (1979)), vertex exchangeable models have been shown to generate dense graphs, in which the number of edges grows quadratically with increasing  $N$ , or empty graphs with probability 1. In recent years an alternative class of models have been introduced in which the edges are exchangeable (see Dempsey et al. (2019), Crane and Dempsey (2018), Campbell et al. (2018) and Cai et al. (2016)) and these models have been shown to express sparse graphs in which the number of edges grows sub-quadratically with increasing  $N$ . This property is observed in many real world graphs and so models of this class present an important contribution to the literature.



## 1.3 Contributions and Thesis Outline

This thesis contains two new contributions to the network analysis literature, which are presented in Chapters 3 and 4. These contributions focus on computational and modelling aspects of the latent space approach for network data. A summary of the remainder of the chapters in this thesis are provided below, and the contributions are highlighted where appropriate.

### **Chapter 2: Review of Latent Space Network Modelling**

This chapter overviews the existing literature on latent space network models. We begin by introducing a generic latent space model, and view the initial models of Hoff et al. (2002) as a special case of this formulation. Then, we discuss inference for these models and detail the surrounding modelling literature which builds upon the approach of Hoff et al. (2002). Finally, we conclude with a simulation study which explores the properties of standard latent space network models.

### **Chapter 3: Sequential Monte Carlo and Dynamic Latent Space Networks**

This chapter focuses on temporally evolving networks in which interactions among a fixed population are observed through time. The latent space framework has been adapted to this setting by allowing the coordinates associated with the nodes to vary over time (Sarkar and Moore (2006), Sewell and Chen (2015b), Durante and Dunson (2014)). Typically, posterior samples are obtained via an MCMC scheme and in this chapter we explore the application of sequential monte carlo (SMC) in this setting. This has yet to be considered in the literature and has two important advantages: (i) learning the latent representation sequentially avoids the increased mixing times associated with MCMC as the number of observations in time increases, and (ii) SMC methods facilitate online estimation which requires a smaller computational cost. However, SMC methods typically do not perform well as the number of nodes in the network increases and in this chapter we explore the literature on high-dimensional

SMC to find a methodology that is appropriate for this setting. We begin with an overview of SMC and dynamic latent space network modelling. Then, we explore different approaches for state estimation and we conclude with simulations and real data examples.

#### **Chapter 4: Latent Space Modelling of Hypergraph Data**

This chapter extends the latent space approach of Hoff et al. (2002) to the setting in which interactions occur between sets of nodes. As a motivating example we consider a coauthorship network in which nodes correspond to authors and interactions indicate which authors have contributed to a given paper. Data of this type are most appropriately represented by a hypergraph, which extends the representation discussed in Section 1.1 beyond pairwise interactions. However, the literature on hypergraph modelling is less developed and hypergraphs are typically represented by a graph in which nodes are connected if they appear in the same interaction. This results in a loss of structural information, and we develop a latent space hypergraph model to partially address this gap in the literature. Our model provides a convenient visualisation of the data and allows exploration of predictive distributions. Furthermore, by representing the nodes in a latent space, we are able to take advantage of the underlying geometry to impose desirable properties on the interactions. By relying on tools from computational topology, we avoid the expensive likelihood calculation implied by the construction of Hoff et al. (2002) and develop a parsimonious model. We remove the effects of non-identifiability by drawing a novel connection with Bookstein coordinates from shape theory. Additionally, we theoretically examine the degree distribution of our model and explore the broader properties via simulations. Finally, we analyse three real world datasets using our framework.

**Chapter 5: Conclusions and Further Work**

In this chapter we conclude the thesis and describe three directions for future work. We consider improvements to the scalability of the methodology discussed in the thesis, the extension to changepoint and anomaly detection, and the exploration of alternative underlying geometries.

# Chapter 2

## Review of Latent Space Network Modelling

### 2.1 Introduction

The latent space approach for network modelling was first introduced in Hoff et al. (2002), and has since given rise to a rich literature. In this approach, nodes of a network are positioned in a low-dimensional latent space and the probability of connections forming are modelled as a function of the associated latent coordinates. This presents a convenient modelling framework in which the underlying geometry imposes properties on the networks. For example, we may specify that nodes which lie close in Euclidean distance are more likely to be connected and, in this case, the latent space will not only provide an intuitive visualisation of the network but also encourage transitive relationships. Additionally, latent space network models allow control of joint distributions on subgraph counts and facilitate exploration of predictive distributions of new nodes. Latent space models were first introduced for social networks in Hoff et al. (2002), however they have since been applied in a range of applications. For instance, they are used to study biological networks in Hoff (2008a), coreadership networks in Krivitsky et al. (2009) and financial networks in Ward et al. (2013).

In this chapter we review the existing literature on latent space network modelling. The details of the approach of Hoff et al. (2002) and its variants are given in Section 2.2, and an empirical investigation of the properties of latent space network models is presented in Section 2.3.

## 2.2 Latent Space Network Modelling

In Section 2.2.1 we introduce a generic latent space network model, and then in Section 2.2.2 we view the initial models of Hoff et al. (2002) as a special case of the generic model. Sections 2.2.3 and 2.2.4 then discuss inference and modelling extensions, respectively.

### 2.2.1 Generic Model

Here we outline a generic latent space model for a network on  $N$  nodes with binary connections  $\mathcal{Y} = \{y_{ij}\}_{i,j \in \{1,2,\dots,N\}}$ , where  $y_{ij} = 1$  indicates the presence of the edge  $\{i, j\}$ . We let  $u_i \in \mathbb{R}^d$  denote the  $d$ -dimensional latent coordinate associated with the  $i^{\text{th}}$  node, for  $i = 1, 2, \dots, N$  and we model the probability of edges forming as a function of  $\mathbf{U} = \{u_i\}_{i=1}^N$  and additional model parameters  $\theta$ . Furthermore, we assume that the edges are conditionally independent given  $\mathbf{U}$ . Let  $p_{ij} = P(y_{ij} = 1 | \mathbf{U}, \theta)$ , then, for  $i, j \in \{1, 2, \dots, N\}$ , the connections are given by

$$Y_{ij} \stackrel{iid}{\sim} \text{Bern}(p_{ij}), \quad (2.2.1)$$

$$p_{ij} = h(u_i, u_j, \theta). \quad (2.2.2)$$

It is common to take  $h(\cdot)$  as a function that is monotone decreasing in  $d_{ij} = d(u_i, u_j)$  where, for example,  $d(\cdot)$  may be chosen as the Euclidean distance between  $u_i$  and  $u_j$  so that nodes which are close in a Euclidean sense are more likely to be connected. Alternative examples will be provided in the remaining sections of this chapter.

For simplicity we will also assume there are no self-ties, so that  $y_{ii} = 0$  for  $i = 1, 2, \dots, N$ , though this is not a necessary assumption. Finally, to obtain a generative

model, we must specify a distribution on the latent coordinates. We let

$$u_i \stackrel{iid}{\sim} f_U(u|\theta_u), \quad \text{for } i = 1, 2, \dots, N, \quad (2.2.3)$$

where  $\theta_u$  define the distributional parameters on  $\mathcal{U}$ .

It is typical in the latent space network modelling literature to take the latent dimension  $d$  to be equal to 2 or 3. This allows for a convenient visualisation of the data, however the choice of  $d$  will affect the flexibility of the model and several authors have considered methods for choosing  $d$ . For example, Durante and Dunson (2014) specify a maximum number for  $d$  that is generally larger than 3 and use a prior to encourage only a small number of non-zero latent dimensions. This allows the data to suggest a larger number of latent dimensions when appropriate. Alternatively, Handcock et al. (2007) suggest that the choice of  $d$  may be thought of as a model selection problem.

### 2.2.2 Distance and Projection Model

Hoff et al. (2002) introduced two models for network data, namely the distance and projection model, and in this section we will specify these models as an instance of the framework given in Section 2.2.1. The models of Hoff et al. (2002) also provide the necessary context for the extensions discussed in Section 2.2.4.

In the distance and projection model, binary connections are modelled via logistic regression in which

$$p_{ij} = \frac{1}{1 + e^{-\eta_{ij}}}. \quad (2.2.4)$$

Then, to obtain the distance model we take

$$\eta_{ij} = \alpha - \|u_i - u_j\|, \quad (2.2.5)$$

$$f_U(u|\theta_u) = \mathcal{N}(\mu, \Sigma), \quad (2.2.6)$$

where  $\|u_i - u_j\|$  denotes the Euclidean distance between  $u_i$  and  $u_j$ .  $\alpha$  represents the global tendency for edges to form in the network whereas the latent coordinates

capture the node specific tendencies. (2.2.5) has an intuitive interpretation, since nodes which lie close together in the latent space are more likely to be connected.

Alternatively, to obtain the projection model of Hoff et al. (2002), we replace (2.2.5) by

$$\eta_{ij} = \alpha + \frac{u_i^T u_j}{|u_j|}. \quad (2.2.7)$$

Note that it is straightforward to incorporate covariate information into (2.2.5) and (2.2.7) through adding a term of the form  $\beta^T z_{ij}$ , where  $z_{ij}$  denotes a  $p$ -dimensional vector of covariates on the edge  $\{i, j\}$ .

Although the distance and projection model take a similar form, (2.2.5) and (2.2.7) impose different properties on the resulting networks. Since the Euclidean distance is a metric, (2.2.5) imposes stronger constraints on the connections. In particular, the Euclidean distance satisfies symmetry and the triangle inequality which suggest reciprocity and transitive relationships are likely, respectively. Reciprocity means that the edges  $\{i, j\}$  and  $\{j, i\}$  occur simultaneously, and transitivity refers to connections in which “friends of friends are also friends”. In contrast to this, (2.2.7) does not in general satisfy symmetry and the triangle inequality. Instead, (2.2.7) imposes that nodes which lie in a similar direction from the origin are more likely to be connected. To see this, note that  $u_i^T u_j$  will approximately be positive when  $u_i$  and  $u_j$  lie in the same quadrant and negative when  $u_i$  and  $u_j$  are in opposite quadrants. We also observe that (2.2.7) will suggest reciprocated connections when  $u_i$  and  $u_j$  are equidistant from the origin. Finally, we comment here that the dot-product is a metric when the latent space is appropriately constrained.

The choice of latent distribution (2.2.6) reflects the intuition that nodes will have varying levels of connectivity. For example, in the distance model, nodes which are positioned close to  $\mu$  are likely to share the greatest number of ties and these nodes are also likely to be connected to other nodes of high degree. These differences will be explored further in Section 2.3 via simulation.

### 2.2.3 Inference and Computation

Typically, latent space models are fitted via an MCMC scheme (see Hoff et al. (2002), Krivitsky et al. (2009) and Salter-Townshend and McCormick (2017)) which requires evaluation of the posterior at each iteration. Since the connections are modelled as conditionally independent, a general expression for the distribution of  $\mathcal{Y}$  conditional on  $\mathbf{U}$  and  $\theta$  is given by

$$P(\mathcal{Y}|\mathbf{U}, \theta) = \prod_{\{i,j \in \{1,2,\dots,N\} | i \neq j\}} p_{ij}^{y_{ij}} (1 - p_{ij})^{1-y_{ij}}, \quad (2.2.8)$$

where in the symmetric case the product is over  $\{i, j \in \{1, 2, \dots, N\} | i < j\}$ . (2.2.8) is the computational bottleneck when evaluating the posterior since the product contains  $O(N^2)$  terms and this scales poorly as  $N$  grows. Several authors have explored approximations of (2.2.8) to facilitate inference for increasing  $N$ . For example, Raftery et al. (2012) introduce a case-control approximation in which the connections are subsampled and Rastelli et al. (2018) develop an approximation by partitioning the latent space. For a dot-product based model Durante and Dunson (2014) rely on a Pólya-gamma data augmentation scheme (Polson et al. (2013)) which allows posterior samples to be obtained via Gibbs sampling. Alternatively, others have opted to perform approximate inference at a reduced computational cost via variational Bayes (for example, see Salter-Townshend and Murphy (2013) and Sewell and Chen (2017)).

Note that the conditional distribution (2.2.8) only depends on  $\mathbf{U}$  through a function of  $\mathbf{U}$  and, for the distance model (see (2.2.5)), (2.2.8) is invariant to distance preserving transformations of  $\mathbf{U}$  (see Figure 2.2.1). Typically, a Procrustes transform (Borg and Groenen (1997)) is applied to the posterior samples as a post-processing step (see Hoff et al. (2002)) to ensure the interpretability of the latent coordinates. This transform finds the coordinates which minimise the squared difference to a set of pre-specified reference coordinates. Some authors instead fix a subset of coordinates (see McCormick and Zheng (2015)) and others interpret the posterior samples on the probability space (see Durante and Dunson (2014)) and forgo the identifiability issues



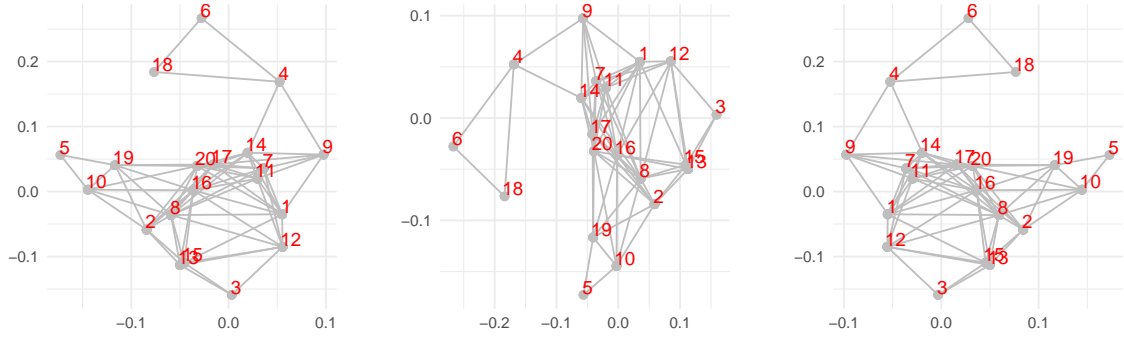


Figure 2.2.1: The likelihood conditional on the latent coordinates (2.2.8) is invariant to distance-preserving transformations of  $\mathbf{U}$ . All latent configurations in this Figure have the same likelihood value.

associated with  $\mathbf{U}$ .

## 2.2.4 Extensions

### Alternative link functions and non-binary connections

In Section 2.2.2 binary connections were modelled via logistic regression, however alternative forms of (2.2.2) have been considered in the literature. Hoff (2008a) model the connections via probit regression and Rastelli et al. (2016) study properties of latent variable network models in which the probability (2.2.2) is modelled via a Gaussian distribution.

It is also straightforward to adapt the generic model in Section 2.2.1 to express non-binary connections. For instance, to model integer connections where  $y_{ij} \in \mathbb{N}_0$ , we may replace (2.2.1) with a Poisson likelihood and model the  $(i, j)^{th}$  rate parameter as a function of  $\mathbf{U}$  and  $\theta$ . For a particular example of this see Sewell and Chen (2016).

### Node specific random effects

Krivitsky et al. (2009) introduce random effects into the linear predictor (2.2.5) to capture each individual's tendency to form connections. For directed edges, the linear

predictor takes the form

$$\eta_{ij} = \beta x_{ij} - \|u_i - u_j\| + \delta_i + \gamma_j, \tag{2.2.9}$$

where  $\delta_i$  and  $\gamma_i$  correspond to the propensity of the  $i^{th}$  node to send and receive connections, respectively. In the undirected case, the term  $\gamma_j$  in (2.2.9) is replaced by  $\delta_j$  and  $\delta_i$  represents the latent ‘sociality’ of the  $i^{th}$  node.

**Modelling community structures**

Several authors have considered modelling networks with community structure in the latent space framework. Handcock et al. (2007) assume that the coordinates are distributed according to a mixture of  $G$  Gaussian distributions, where  $G$  corresponds to the number of communities. Hence, we let (2.2.3) take the form

$$f_U(u_i|\theta_u) = \sum_{g=1}^G \lambda_g \mathcal{N}(\mu_g, \sigma_g^2 I_d) \tag{2.2.10}$$

where  $\theta_u = (\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\lambda})$  for  $\boldsymbol{\mu} = (\mu_1, \dots, \mu_G)$ ,  $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_G)$  and  $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_G)$ . The parameter  $\lambda_g$  represents the probability of each node belonging to the  $g^{th}$  group and, to ensure (2.2.10) is a valid probability distribution, we specify  $\boldsymbol{\lambda}$  so that  $\lambda_g \geq 0$  for  $g = 1, 2, \dots, G$  and  $\sum_{g=1}^G \lambda_g = 1$ . A similar approach was also taken in Krivitsky et al. (2009), and Gormley and Murphy (2010) introduce a mixture of experts model in which the mixture probabilities  $\boldsymbol{\lambda}$  are modelled as a function of covariates.

In many applications, networks exhibit homophily in which nodes that share similar characteristics are more likely to be connected. This property can be expressed within the models in Section 2.2.2 through including covariates in the linear predictor (2.2.4). Additionally, networks may also exhibit stochastic equivalence in which nodes can be divided into groups such that nodes in the same group display similar connectivity patterns. This is closely related to the community structures described in the previous paragraph. Hoff (2008a) introduce the eigenmodel which combines the approaches of Hoff et al. (2002) and Nowicki and Snijders (2001), which focus on

describing homophily and stochastic equivalence, respectively. In the eigenmodel, the linear predictor is given by

$$\eta_{ij} = \alpha + \beta^T z_{ij} + u_i^T \Lambda u_j, \quad (2.2.11)$$

where  $u_i$  denotes the  $d$ -dimensional of latent characteristics of the  $i^{\text{th}}$  node, and  $\Lambda$  is a  $d \times d$  diagonal matrix. The non-zero entries of  $\Lambda$  describe the positive or negative effect of the  $l^{\text{th}}$  latent characteristic on the connection  $y_{ij}$ , for  $l = 1, 2, \dots, d$ . (2.2.11) can also be adapted to model asymmetric ties as demonstrated in Hoff (2008b). Note that the eigenmodel bears similarity to the mixed membership stochastic blockmodel of Airoldi et al. (2009) in which connections are modelled as a function of latent group memberships.

An alternative approach to expressing community structures is presented in Fosdick et al. (2019) where the authors introduce multiresolution network models. In particular, they introduce the latent space stochastic blockmodel which mimics the stochastic blockmodel of Holland et al. (1983) where the connection probabilities are modelled as a function of latent group memberships. In this model, the within community connection probabilities are determined by a latent space distance model and the between community connection probabilities are modelled as an Erdős-Rényi random graph.

### Networks with multiple views

Multiview network data in which several types of relationships on the same set of nodes have also been considered in the latent space framework. In Salter-Townshend and McCormick (2017), each view is assigned a unique latent representation and the influences between different connection types are explicitly modelled through an association parameter. Similarly, Sweet et al. (2013) also assign a unique latent representation to each view, however they develop a hierarchical approach to modelling multiple views. Alternatively, Gollini and Murphy (2016) specify latent variables that

are common to each view and allow additional model parameters to capture the differences between views. This approach is also taken in D’Angelo et al. (2019), and D’Angelo et al. (2018) extend this to include additional sender and receiver effects for each network view.

### **Extensions to the dot-product formulation**

Variants of the projection model detailed in Section 2.2.2 have been considered in the literature. Nickel (2007) first introduced the random dot product graph (RDPG) in which the probability of a connection forming is a function of  $u_i^T u_j$ , and generalisations of the RDPG have also been studied in Young and Scheinerman (2007) and Ng and Murphy (2019). Furthermore, Athreya et al. (2017) present a survey of inference on RDPGs and Rubin-Delanchy et al. (2017) make the connection between the RDPG and the mixed membership SBM of Airoldi et al. (2009). Finally, Rubin-Delanchy et al. (2017) introduce a model called the generalised random dot product graph which extends the RDPG to allow for dissasortative connections where “opposites attract”.

### **Temporally evolving networks**

Another interesting extension arises when considering connections between a fixed population of size  $N$  over time. This setting was first considered within the latent space framework in Sarkar and Moore (2006), where the authors model the latent trajectories via a Gaussian random walk. Alternatively, Durante and Dunson (2014) have introduced a model in which the latent trajectories are modelled as a Gaussian process. There exist many variants of these models (see Sewell and Chen (2015b), Sewell and Chen (2017), Sewell and Chen (2016), Sewell and Chen (2015a), Friel et al. (2016), Durante et al. (2016) and Durante et al. (2017)), and a discussion of this literature will be provided in Chapter 3.

Case	$f_U$	Metric
1	$\mathcal{N}_2(\mathbf{0}, \Sigma_1)$	$s(u_i, u_j) = u_i^T u_j$
2	$\mathcal{N}_2(\mathbf{0}, \Sigma_1)$	$s(u_i, u_j) = -\ u_i - u_j\ $
3	$U([-1.7, 1.7]^2)$	$s(u_i, u_j) = u_i^T u_j$
4	$U([-1.7, 1.7]^2)$	$s(u_i, u_j) = -\ u_i - u_j\ $
5	$\mathcal{N}_2(\mathbf{0}, \Sigma_2)$	$s(u_i, u_j) = -\ u_i - u_j\ $
6	$\mathcal{N}_2(\mathbf{0}, \Sigma_3)$	$s(u_i, u_j) = -\ u_i - u_j\ $

Table 2.2.1: Cases considered in Section 2.3 simulation.  $p_{ij}$  is given by (2.2.4) where  $\eta_{ij} = \alpha + s(u_i, u_j)$ . When the latent coordinates are normally distributed we take  $\Sigma_1 = 0.5 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ ,  $\Sigma_2 = 0.5 \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}$  and  $\Sigma_3 = 2 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$  and for all cases we fix  $\alpha = 1$ .

### Alternative underlying geometries

In the literature discussed so far, the latent coordinates are assumed to lie in Euclidean space. This choice impacts the properties of networks generated in this framework, and in recent years several authors have explored the effect of modifying the underlying geometry. For example, McCormick and Zheng (2015) consider modelling aggregated relational data, in which each actor is asked “how many people with characteristic  $X$  do you know?”, using a latent space model where the latent coordinates are assumed to lie on the  $p$  dimensional sphere,  $\mathcal{S}^p$ . Alternatively Krioukov et al. (2010) investigate the effect of modelling the latent coordinates in hyperbolic space. They demonstrate that this approach can express networks with a power law degree sequence by taking advantage of varying density of the space. This idea has also been considered in the context of link prediction by Kitsak et al. (2019). Finally, Smith et al. (2017) provide an analysis of the effect of specifying the latent coordinates in Euclidean, Hyperbolic and Elliptic geometry. They empirically explore how the degree distributions and common network statistics change as a function of the latent geometry, and examine how graph Laplacians may be used to identify an appropriate geometry.

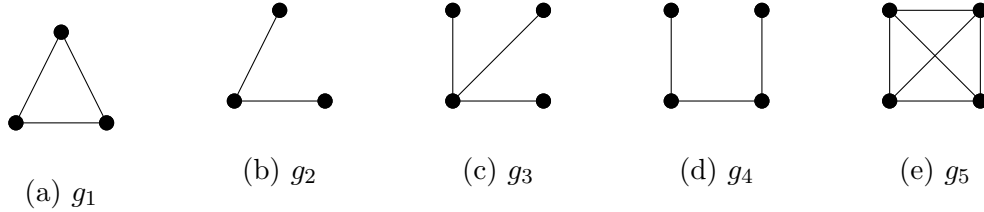


Figure 2.3.1: Motifs considered in simulations in Section 2.3.

## 2.3 Exploration of Properties

In this section we explore the properties of the distance and projection models from Section 2.2.2 via simulation where we will assume  $d = 2$  throughout. We will consider the effect of varying the distribution of the latent coordinates and the choice of  $s(u_i, u_j)$ , where  $p_{ij}$  is given by (2.2.4) and  $\eta_{ij} = \alpha + s(u_i, u_j)$ . Details of the choices considered in this section are given in Table 2.2.1, where the latent coordinate distributions have been specified so that, when comparing uniform and Gaussian, the coordinates lie within a box of roughly the same size. Throughout,  $\alpha$  remains fixed so that differences between networks can be attributed to choices of  $f_U$  and  $s(u_i, u_j)$ , and we set the number of nodes to be  $N = 40$ .

We focus on the degree distribution and the distribution of motif counts, and the motifs considered are depicted in Figure 2.3.1. In Section 2.3.1 we compare the Euclidean metric and dot-product when the latent coordinates are uniform and normally distributed. Then, in Section 2.3.2, we explore the effect of varying the parameters of normally distributed latent coordinates.

### 2.3.1 Effect of Metric and $f_U$

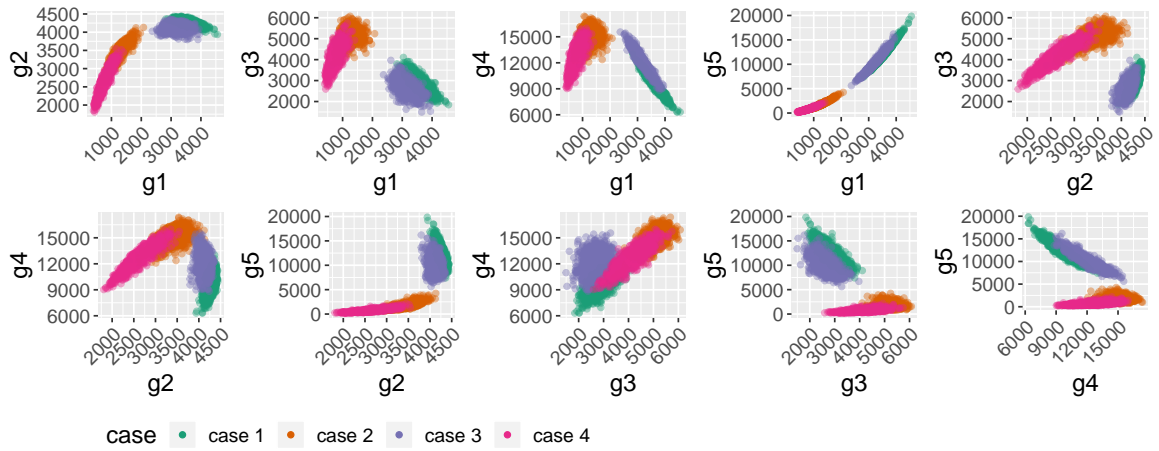
Here we compare cases 1, 2, 3 and 4 from Table 2.2.1 in which we vary the distribution of the latent coordinates and the form of  $s(u_i, u_j)$ . The joint distributions of the motif counts are shown in Figure 2.3.2a and the average degree distributions are shown in Figure 2.3.2c.

First, we notice that there is a distinct difference between the networks simulated using Euclidean distance and the dot-product. Most notably, we see differing patterns in the joint motif counts and, as an example, observe the correlations in the joint distribution of  $g_1$  and  $g_3$ . Additionally, the networks simulated with the dot-product are overall much denser. We also observe that, for each choice of  $s(u_i, u_j)$ , the networks are denser when the points are normally distributed. This is intuitive since the choice of parameters of the normal distribution implies that the coordinates are placed in a region of higher density when compared to coordinates simulated from the uniform distribution. Alternative choices of covariance can be made to alter the density of the networks. Finally, we see that the degree distributions are generally flatter and more skewed when  $s(u_i, u_j)$  is the Euclidean distance.

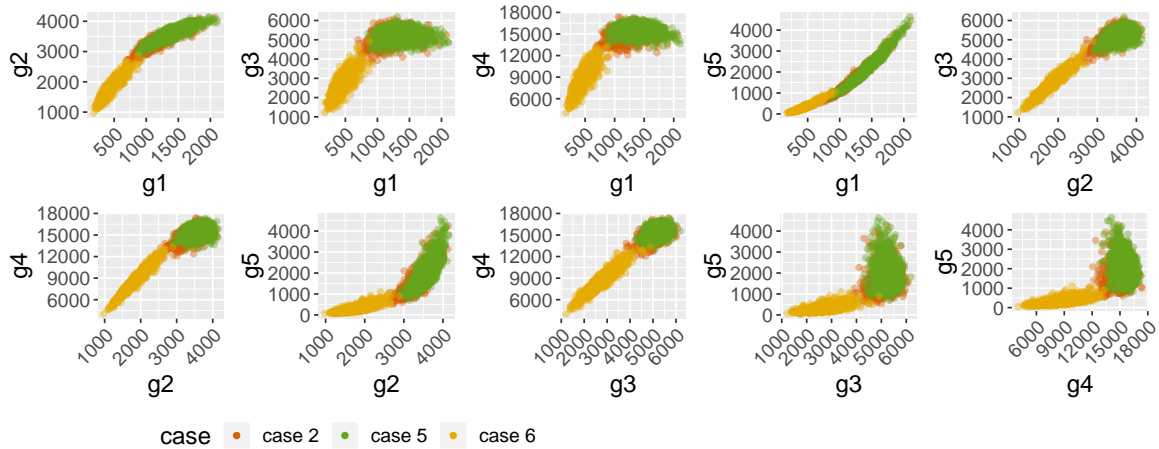
### 2.3.2 Effect of Covariance for Normally Distributed $U$

Here we consider the effect of changing the covariance of the latent coordinates when they are normally distributed. Throughout, we only consider the Euclidean distance. Case 2 from Table (2.2.1) is included for reference, and in addition to this we consider highly correlated and more dispersed latent coordinates in cases 3 and 4, respectively. The joint motif counts are shown in Figure 2.3.2b and the average degree distributions are shown in Figure 2.3.2d.

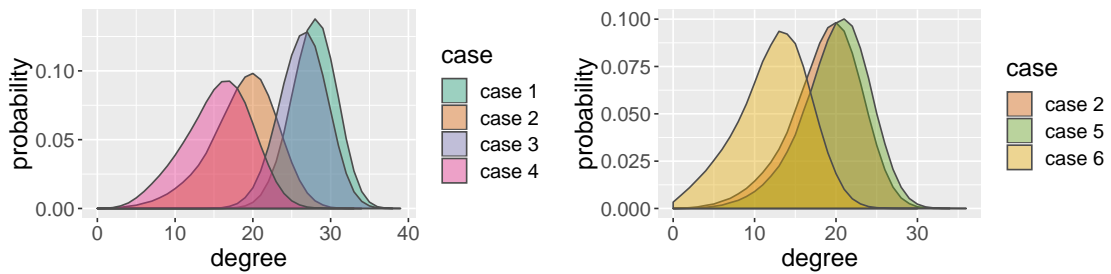
We observe similar patterns in Figure 2.3.2b, however in some cases we see less correlation in some of the cases. Cases 2 and 5 are most similar, and the the degree distribution is slightly flatter when the coordinates are more dispersed. For all cases we find that the degree distributions exhibit some negative skewness. The differences in the networks for these cases reflects the intuition offered by the Euclidean distance, since the connectivity patterns change depending on how densely positioned the latent coordinates are. We note here that a broad range of degree distributions can be expressed by taking mixture distributions on the latent coordinates, as in (2.2.10).



(a) Joint motif counts for cases 1,2,3 and 4.



(b) Joint motif counts for cases 2,5,6.



(c) Degree distribution for cases 1,2,3 and 4. (d) Degree distribution for cases 2,5 and 6.

Figure 2.3.2: Summary of simulated networks for the cases described in Table 2.2.1



# Chapter 3

## Sequential Monte Carlo and Dynamic Latent Space Networks

### 3.1 Introduction

Network data describing interactions among a population arise in a multitude of disciplines (see Leskovec and Krevl (2014) and Kunegis (2013)) including sociology, biology and finance, and a range of network models have been introduced to gain statistical insights to data of this type (see Goldenberg et al. (2010), Salter-Townshend et al. (2012) and Kolaczyk (2009)). In many applications interactions among a fixed population are observed over time, and this motivates the development of methodology for temporal network data. This type of data may present practical challenges as the number of nodes, the number of timestamps, or both, increase.

Over the past two decades, the latent space approach has proven to be a popular modelling framework for network data (see Kim et al. (2018)). In this approach, nodes of the network are associated with a low-dimensional latent coordinate that encodes the propensity for edges to form. This was first introduced for static networks in Hoff et al. (2002), and has since been extended to the temporal case (see Sarkar and Moore (2006), Durante and Dunson (2014) and Sewell and Chen (2015b)). The latent space

framework has a number of appealing properties. For example, the latent coordinates provide a parsimonious representation of complex data which can also be used for visualisation, and the latent space allows exploration of predictive distributions via simulation.

Bayesian inference for temporally evolving latent space network models is typically facilitated by Markov Chain Monte Carlo (MCMC) since the posterior is intractable. In this chapter, we instead consider Sequential Monte Carlo (SMC) as the inference mechanism and the state space model formulation of Sarkar and Moore (2006) and Sewell and Chen (2015b) provides a natural setting for SMC which has yet to be explored in the literature. As the number of time points increases, we comment that SMC will avoid mixing issues associated with MCMC. Furthermore, SMC facilitates online and recursive inference, meaning that the full inference procedure is not required to be updated given additional observations. We note that SMC has been considered in a different context for networks in Bloem-Reddy and Orbanz (2018).

The remainder of this chapter is organised as follows. An overview of SMC methodology is given in Section 3.2. Then, in Section 3.3, we review the existing literature on DLSNs and introduce the state space formulation of Sarkar and Moore (2006) and Sewell and Chen (2015b). The application of SMC to DLSNs is discussed in Section 3.4, where we pay particular attention to methodology which is appropriate for networks with a large numbers of nodes. Simulations and data examples are given in Sections 3.5 and 3.6, respectively. Finally, we conclude with a discussion in Section 3.7.

## 3.2 Sequential Monte Carlo

Sequential Monte Carlo (SMC) methods are a broad class of simulation-based algorithms designed to estimate a posterior distribution. Although they may be used in alternative settings, our focus will be on particle filtering for state space models

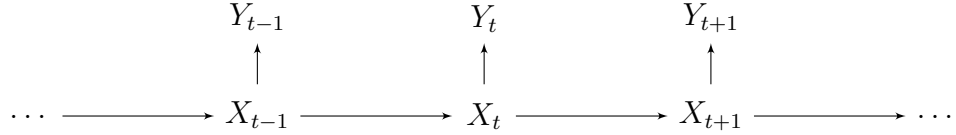


Figure 3.2.1: Depiction of the dependence structure in a SSM. Each observation is assumed independent conditional on latent variables which follow a first order Markov process. The dependence on model parameters  $\theta$  is assumed throughout.

(SSMs). For a more general overview of SMC methodology see Doucet and Johansen (2008), Lopes and Tsay (2011) and Doucet et al. (2001). We begin in Section 3.2.1 by reviewing SSMs and we then discuss state and parameter estimation in Sections 3.2.2 and 3.2.3, respectively. Finally, we briefly discuss the SMC literature for high-dimensional state spaces in Section 3.2.4.

### 3.2.1 State Space Model

SSMs are a general class of models for time series data. They model a sequence of observations  $\{y_t\}_{t=1}^T$  as conditionally independent given a sequence of latent variables  $\{x_t\}_{t=0}^T$  that are assumed to follow a first order Markov process. We let  $g_\theta(y_t|x_t)$  represent the likelihood of each observation conditional on its corresponding latent variable, and  $f_\theta(x_t|x_{t-1})$  represent the latent transition density. Note that both densities depend on additional model parameters  $\theta$ . A general SSM may be defined by the following equations, and a depiction of the dependence structure is given in Figure 3.2.1.

$$\begin{aligned}
 X_0 &\sim \mu_\theta(x_0). \\
 X_t|X_{t-1} = x_{t-1} &\sim f_\theta(x_t|x_{t-1}), & \text{for } t = 1, 2, \dots, T, \\
 Y_t|X_t = x_t &\sim g_\theta(y_t|x_t), & \text{for } t = 1, 2, \dots, T,
 \end{aligned} \tag{3.2.1}$$

### 3.2.2 Particle Filter

Given a sequence  $\{y_t\}_{t=1}^T$ , it is natural to consider which values of  $\{x_t\}_{t=0}^T$  are likely to have generated the observations. By targeting the filtering densities  $\{p_\theta(x_t|y_{1:t})\}_{t=1}^T$ , the particle filter (PF) describes a framework for estimating the latent sequence. In this section, we detail the particle filter and we comment here that there exists a related problem in which the objective is to target the smoothing densities  $\{p_\theta(x_t|y_{1:T})\}_{t=1}^T$ . This is referred to as particle smoothing, and is discussed in Doucet and Johansen (2008).

We begin by considering an expression for the filtering density at time  $t$ . By Bayes theorem, we obtain

$$\begin{aligned} p_\theta(x_t|y_{1:t}) &= \int \frac{g_\theta(y_t|x_t)f_\theta(x_t|x_{t-1})}{p(y_t|y_{1:t-1},\theta)} p_\theta(x_{t-1}|y_{1:t-1}) dx_{t-1} \\ &\propto \int g_\theta(y_t|x_t)f_\theta(x_t|x_{t-1})p_\theta(x_{t-1}|y_{1:t-1}) dx_{t-1}. \end{aligned} \quad (3.2.2)$$

The expression (3.2.2) makes explicit the dependence between the filtering density at time  $t$  and the filtering density at time  $t-1$ . When the transition density  $f_\theta(x_t|x_{t-1})$  and conditional likelihood  $g_\theta(y_t|x_t)$  are Gaussian, it is possible to derive an exact expression for (3.2.2) and this results in the well known Kalman filter (Kalman (1960)). However, in general, there is no analytic expression for (3.2.2) and we instead rely on approximations of the filtering densities.

A particle filtering scheme relies on importance sampling (IS) at each time step to sequentially approximate the filtering densities. IS is a monte carlo method for approximating a density of the form  $p(x) = \tilde{p}(x)/Z$ , where  $\tilde{p}(x)$  is the unnormalised density and  $Z = \int \tilde{p}(x) dx$  is the normalising constant. In IS, we introduce a proposal density  $q(x)$  and define the unnormalised weights as  $w(x) = \tilde{p}(x)/q(x)$ . Given this, we may write

$$p(x) = \frac{\tilde{p}(x)}{Z} = \frac{w(x)q(x)}{Z}, \quad (3.2.3)$$

$$\text{and} \quad Z = \int \tilde{p}(x) dx = \int w(x)q(x) dx. \quad (3.2.4)$$

Now suppose we have drawn  $M$  samples  $\{x^{(i)}\}_{i=1}^M$  from  $q(x)$ . Using (3.2.3) and (3.2.4) we can then approximate  $p(x)$  using

$$\hat{p}(x) = \sum_{i=1}^M W^{(i)} \delta_{x^{(i)}}(x), \quad (3.2.5)$$

$$W^{(i)} = \frac{w(x^{(i)})}{\sum_{i=1}^M w(x^{(i)})}. \quad (3.2.6)$$

The normalised weights  $W^{(i)}$  follow from approximating (3.2.4) by  $\hat{Z} = \sum_{i=1}^M w(x^{(i)})/M$  and the set  $\{x^{(i)}, w(x^{(i)})\}_{i=1}^M$  is then referred to as a particle approximation of  $p(x)$ .

Given (3.2.5) and (3.2.6), we may now outline a generic particle filter. The dependence highlighted in (3.2.2) suggests a recursive scheme in which an approximation of  $p_\theta(x_{t-1}|y_{1:t-1})$  is updated to generate an approximation of  $p_\theta(x_t|y_{1:t})$ . More precisely, suppose that we have a particle approximation to  $p_\theta(x_{t-1}|y_{1:t-1})$  which we denote by  $\{x_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^M$ , where  $x_{t-1}^{(i)}$  and  $w_{t-1}^{(i)}$  denote the particles and their associated weights, respectively. Then, to estimate  $p_\theta(x_t|y_{1:t})$ , we sample new particles from a proposal distribution  $q_\theta(\cdot|y_t, x_{t-1})$ , and adjust the weights accordingly. This procedure can then be repeated for each time step. For this to work in practice, we also need to introduce a resampling step into this scheme. Otherwise we will observe particle degeneracy in which the weights concentrate onto a single particle, resulting in poor quality approximations. Intuitively, the resampling step replicates more informative particles and forgets less informative particles.

Algorithm 1 details a generic particle filter in which particles are propagated according to  $q_\theta(x_t|y_t, x_{t-1})$ . To obtain the best performance in terms of the variance of the importance weights (Doucet et al. (2000)) we should take the proposal  $q_\theta(x_t|y_t, x_{t-1})$  to be equal to

$$p_\theta(x_t|y_t, x_{t-1}) = \frac{g_\theta(y_t|x_t) f_\theta(x_t|x_{t-1})}{p(y_t|x_{t-1})} = \frac{g_\theta(y_t|x_t) f_\theta(x_t|x_{t-1})}{\int g_\theta(y_t|x_t) f_\theta(x_t|x_{t-1}) dx_t}. \quad (3.2.7)$$

However, for many practical applications it is not possible to find an exact expression for (3.2.7). Instead, we may obtain the standard SIR filter of Gordon et al. (1993) from Algorithm 1 by taking  $q_\theta(\cdot|x_{t-1}, y_t) = f_\theta(\cdot|x_{t-1})$  and  $w_t = g_\theta(y_t|\cdot)$ . This scheme

**Algorithm 1** General Particle Filter

- *Iteration*  $t = 0$ :

Sample  $M$  particles  $\{x_0^{(i)}\}_{i=1}^M$  from  $\mu_\theta(\cdot)$  and assign weights  $w_0^{(i)} = 1/M$ .

- *Iteration*  $t = 1, \dots, T$ :

Assume particles  $\{x_{t-1}^{(i)}\}_{i=1}^M$  with weights  $\{w_{t-1}^{(i)}\}_{i=1}^M$  that approximate  $p_\theta(x_{t-1}|y_{1:t-1})$ .

a) Sample a parent index  $a_{t-1}^{(i)}$  for the  $i^{\text{th}}$  particle according to the weights  $W_{t-1} = (w_{t-1}^{(1)}, w_{t-1}^{(2)}, \dots, w_{t-1}^{(M)})$ . Denote the sampling mechanism by  $a_{t-1}^{(i)} \sim \mathcal{F}(\cdot|W_{t-1})$ .

b) Propagate the particles according to  $x_t^{(i)} \sim q_\theta(\cdot|y_t, x_{t-1}^{a_{t-1}^{(i)}})$ . Set  $x_{1:t}^{(i)} = (x_{1:t-1}^{a_{t-1}^{(i)}}, x_t^{(i)})$

c) Calculate the weights using  $w_t^{(i)} \propto \frac{p_\theta(x_{1:t}^{(i)}, y_{1:t})}{q_\theta(x_t^{(i)}|y_t, x_{1:t-1}^{a_{t-1}^{(i)}})p_\theta(x_{1:t-1}^{a_{t-1}^{(i)}}, y_{1:t-1})}$

and normalise.

propagates particles blindly according to the transition density and, whilst this will not perform optimally, it is straightforward to implement for a variety of models. Alternatively, we may approximate  $p(x_t|y_t, x_{t-1})$  as in the auxiliary particle filter of Pitt and Shephard (1999). Here, particles are first resampled according to the predictive density  $\zeta_t^{(i)} \propto w_{t-1}^{(i)} \int g_\theta(y_t|x_t)f_\theta(x_t|x_{t-1}^{(i)}) dx_t = w_{t-1}^{(i)}p(y_t|x_{t-1}^{(i)})$  and then propagated via the proposal  $q_\theta(x_t|y_t, x_{t-1})$ . Since the proposal incorporates information about the observation  $y_t$ , the APF is expected to improve upon the blind propagation in the SIR filter. Details of the APF are given in Algorithm 2 where  $\xi_t^{(i)}$  approximates  $\zeta_t^{(i)}$ . When  $f_\theta(x_t|x_{t-1})$  is Gaussian and  $g_\theta(y_t|x_t)$  is log-concave, Pitt and Shephard (1999) suggest approximating  $p(y_t|x_{t-1}^{(i)})$  using a Taylor expansion. Note that in Algorithm 2 we provide the implementation of Carpenter et al. (1999) who avoid an extra resampling step included in Pitt and Shephard (1999).

Implementation of a PF requires the user to choose a resampling scheme, denoted by  $\mathcal{F}(\cdot|W_{t-1})$  in Algorithm 1, and standard choices include multinomial, systematic, residual and stratified resampling (see Section 2 of Douc and Cappé (2005) for the algorithmic details). It has been shown that systematic resampling performs comparably to other schemes for a range of scenarios (Douc and Cappé (2005), Doucet and Johansen (2008)), and so we rely on this throughout. This scheme has the added

---

**Algorithm 2** APF filter (Pitt and Shephard (1999), Carpenter et al. (1999))

---

• *Iteration*  $t = 0$ :

Sample  $M$  particles  $\{x_0^{(i)}\}_{i=1}^M$  from  $\mu_\theta(\cdot)$  and assign weights  $w_1^{(i)} = 1/M$ .

• *Iteration*  $t = 1, 2, \dots, T$ :

Assume particles  $\{x_{t-1}^{(i)}\}_{i=1}^M$  with weights  $\{w_{t-1}^{(i)}\}_{i=1}^M$  that approximate  $p_\theta(x_{t-1}|y_{1:t-1})$ .

a) Sample indices  $\{k_1, k_2, \dots, k_M\}$  from  $\{1, 2, \dots, M\}$  according to probabilities  $\{\xi_t^{(i)}\}_{i=1}^M$ .

b) Propagate the particles according to  $x_t^{(i)} \sim q_\theta(\cdot|x_{t-1}^{(k_i)}, y_t)$ .

c) Weight particles according to  $w_t^{(i)} \propto \frac{w_{t-1}^{(k_i)} g_\theta(y_t|x_t^{(i)}) f_\theta(x_t^{(i)}|x_{t-1}^{(k_i)})}{\xi_t^{(k_i)} q_\theta(x_t^{(i)}|x_{t-1}^{(k_i)}, y_t)}$  and normalise.

---

benefit of being simple to implement, however there is currently no theoretical justification of the performance. For a discussion of resampling mechanisms see Doucet and Johansen (2008) and Douc and Cappé (2005).

The performance of a particle filter can be assessed by considering the distribution of the importance weights at each time point. In the optimal case, all particles have equal weight and therefore contribute the same amount of information to the likelihood. Since the weights are normalised, we may consider the variance of the weights as a measure of the quality of the particle set where a smaller variance indicates a better quality approximation. Alternatively, we can estimate the effective sample size (ESS) and an approximate expression for this at time  $t$  is given by

$$\widehat{ESS} = \frac{1}{\sum_{i=1}^M (w_t^{(i)})^2}. \quad (3.2.8)$$

Although (3.2.8) is widely used in the literature, we note that some authors suggest other choices may be preferable (see Elvira et al. (2018)).

### 3.2.3 Parameter estimation

The PF schemes discussed in the previous section are designed to estimate the latent states, and in this section we consider estimation of static parameters  $\theta$ . We describe

methodology for offline and online estimation in Sections 3.2.3 and 3.2.3, respectively, and we pay particular attention to the approaches relied upon in later sections. This discussion is not fully comprehensive, and we refer to Kantas et al. (2009), Gao and Zhang (2012), Kantas et al. (2015) and Lopes and Tsay (2011) for an overview of the literature.

### Offline estimation

In the Bayesian setting, Chopin et al. (2013) present a sequential but offline algorithm for joint estimation of the state vector and parameters  $\theta$  and Andrieu et al. (2010) introduce particle MCMC methods, which avoid the calculation of complex proposal distributions in MCMC by utilising an SMC approximation to the likelihood which leaves the target distribution invariant. In particular, Andrieu et al. (2010) introduce Particle Independent Metropolis Hastings (PIMH) for estimating the latent states only, and both Particle Marginal Metropolis Hastings (PMMH) and Particle Gibbs (PG) for estimating the latent states and the model parameters.

Alternatively, in the frequentist setting, many authors consider estimation  $\theta$  via maximum likelihood estimation and a survey of current methodology is given in Section 5.1 of Kantas et al. (2015). In this section we will focus on the approach of Nemeth et al. (2016) in which estimates of  $\theta$  are obtained via gradient ascent. We take

$$\theta_k = \theta_{k-1} + \gamma_k \nabla \log p(y_{1:T}|\theta)|_{\theta=\theta_{k-1}}, \quad (3.2.9)$$

where  $\gamma_k$  is a sequence of decreasing steps such that  $\sum_k \gamma_k = \infty$  and  $\sum_k \gamma_k^2 < \infty$ . A typical choice for this sequence is  $\gamma_k = k^{-\alpha}$  with  $0.5 < \alpha < 1$ .

To evaluate (3.2.9) we need an expression for the score  $S_T = \nabla \log p(y_{1:T}|\theta)|_{\theta=\theta_{k-1}}$ . Nemeth et al. (2016) rely on a particle approximation of the score and the details of their approach are given in Algorithm 3 where  $S_t = \nabla \log p(y_{1:t}|\theta)$ . In the offline setting a PF is implemented  $n_{it}$  times and, in the  $i^{\text{th}}$  iteration, the parameters are given by  $\theta_i$ .



---

**Algorithm 3** Rao-Blackwellised Score

---

**Initialise:** set  $m_0^{(i)} = 0$  for  $i = 1, 2, \dots, M$ ,  $S_0 = 0$ .

**For**  $t = 1, 2, \dots, T$

1) Run one iteration of a PF to obtain  $\{x_t^{(i)}\}_{i=1}^M$ ,  $\{a_{t-1}^{(i)}\}_{i=1}^M$  and  $\{w_t^{(i)}\}_{i=1}^M$

(see Algorithm 1)

2) Update the mean approximation

$$m_t^{(i)} = \lambda m_{t-1}^{(i)} + (1 - \lambda) S_{t-1} + \nabla \log g(y_t | x_t^{(i)}) + \nabla \log f(x_t^{(i)} | x_{t-1}^{(i)})$$

3) Update the score vector

$$S_t = \sum_{i=1}^M x_t^{(i)} m_t^{(i)}$$

**End**

---

Nemeth et al. (2016) also discuss estimation of the observed information matrix, though we do not detail this in Algorithm 3. Furthermore, Nemeth et al. (2016) demonstrate that their approach is more accurate than the particle learning (PL) scheme of Carvalho et al. (2010), particularly when  $T$  is large, and that PL is more sensitive to dependency in the parameters. Their approach has a linear computational cost, and generalises the work of Poyiadjis et al. (2011).

### Online estimation

A natural approach for estimating  $\theta$  in the online setting would be to find a particle approximation to the joint density  $p(x_t, \theta | y_{1:t})$ , similar to the procedure for estimating  $\{x_t\}_{t=1}^T$  outlined in Section 3.2.2. However, since  $\theta$  does not evolve in time, the particle set will degenerate under repeated resampling. Online estimation of  $\theta$  therefore presents a challenging task and remains an open problem in the literature. Here we discuss existing methodology, distinguishing between Bayesian and Frequentist approaches.

In the Bayesian framework, several methods for static parameter estimation have been proposed. For example, Gordon et al. (1993) include artificial dynamics to reduce the degeneracy and Gilks and Berzuini (2001) rely on MCMC kernels to add

diversity to the particle set. Other approaches include practical filtering (see Polson et al. (2008)), using kernel approximations (see Liu and West (2001)) and estimating  $\theta$  using sufficient statistics (see Storvik (2002), Fearnhead (2002) and Carvalho et al. (2010)).

In the frequentist setting,  $\theta$  can be estimated according to an expectation-maximisation procedure (for example see Cappé (2011)) or via likelihood maximisation (for example see Poyiadjis et al. (2011)). Here we again focus on the approach of Nemeth et al. (2016), and this can be implemented in an online manner when the following approximation is made.

$$\nabla \log p(y_t|y_{1:t-1}, \theta_t) = \nabla \log p(y_{1:t}|\theta_t) - \nabla \log p(y_{1:t-1}|\theta_{t-1}). \quad (3.2.10)$$

Then, using the procedure in Algorithm 3, we can approximate the score  $S_T$  by  $\nabla \log \hat{p}(y_t|y_{1:t-1}, \theta_t) = S_t - S_{t-1}$  at the  $t^{\text{th}}$ . This allows us to perform the parameter update given in (3.2.9).

### 3.2.4 High-Dimensional SMC

SMC methods have proven successful in a range of problems, however they typically do not perform well as the dimension of the state space increases (Bengtsson et al. (2008), Snyder et al. (2008)) and it has been shown that the number of particles must increase exponentially with the state dimension to avoid particle degeneracy (Snyder et al. (2015)). Developing methodology for this scenario remains an open problem in the literature, and in recent years a body of work has developed to address this issue. A review of recent methodology can be found in Septier and Peters (2015) where they discuss the bridging density approach of Godsill and Clapp (2001), the block particle filter of Rebeschini and van Handel (2015), and the space-time particle filter of Beskos et al. (2017). Finally, Septier and Peters (2015) end their discussion with the Sequential MCMC (SMCMC) approach (Septier et al. (2009), Khan et al. (2005), Brockwell et al. (2012)). Alternative approaches not included in this review include

the guided intermediary particle filter of Park and Ionides (2019), gradient nudging as detailed in Akyildiz and Míguez (2019), and the nested particle filter of Naesseth et al. (2015). In Section 3.4 we consider this literature in the context of dynamic latent space network models and discuss the relevant approaches in more detail.

### 3.3 Dynamic Latent Space Network Modelling

In this section we consider modelling a temporally evolving network on  $N$  nodes. We focus on the latent space approach, as discussed in Chapter 2. This framework provides an intuitive visualisation of the network and is able to express properties that are observed in many real world networks, such as transitivity. The properties of latent space network models are well understood (Rastelli et al. (2016)) and there is a broad modelling literature centred around this idea (for example see Handcock et al. (2007), Krivitsky et al. (2009) and Kim et al. (2018)).

#### 3.3.1 Background

We focus on the SSM approach of Sewell and Chen (2015b) and, prior to introducing this in Section 3.3.2, we first discuss the necessary background. Our focus is on the setting in which a network on  $N$  nodes is observed over time. Whilst we refer to time-varying networks as dynamic networks, note that this term may also refer to networks in which the population of nodes evolves or to processes on networks and we do not consider these cases here.

The latent space framework was first considered for dynamic networks in Sarkar and Moore (2006) and in this work the authors rely on a SSM to extend the model of Hoff et al. (2002) to the dynamic setting. A similar approach was taken in Sewell and Chen (2015b), however these works differ in their approach to model fitting. Sarkar and Moore (2006) take an optimisation based approach, whereas Sewell and Chen (2015b) rely on an MCMC scheme to obtain posterior samples. In this chapter, our

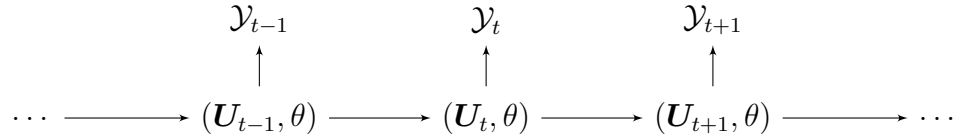


Figure 3.3.1: SSM for dynamic latent space networks. The observed adjacency matrices are modelled independently conditional on the latent coordinates, and the latent coordinates are modelled with a first order Markov process.  $\theta$  is a vector of static parameters.

focus will be on the SSM approach, which we discuss in Section 3.3.2. However, we also note the approach explored in Durante and Dunson (2014) and related works, where the authors model the latent trajectory of each node using a Gaussian process.

### 3.3.2 SSM formulation

Here we discuss the dynamic latent space network model of Sewell and Chen (2015b), and we begin by introducing some notation. In this setting, we assume that we have observed connections between  $N$  nodes at times  $t = 1, 2, \dots, T$  and the observed binary connections at time  $t$  will be denoted by the  $(N \times N)$  symmetric adjacency matrix  $\mathcal{Y}_t$ . This matrix contains entries  $y_{ijt} = 1$  if nodes  $i$  and  $j$  share a connection at time  $t$ , and  $y_{ijt} = 0$  otherwise. Additionally, we will assume that there are no self ties so that  $y_{iit} = 0 \forall t = 1, 2, \dots, T$  and  $i = 1, 2, \dots, N$ . We denote the  $d$ -dimensional latent coordinates of the  $i^{\text{th}}$  node at time  $t$  by  $u_{it}$ , and we let  $\mathbf{U}_t \in \mathbb{R}^{N \times d}$  be the  $N \times d$  matrix whose  $i^{\text{th}}$  row corresponds to  $u_{it}$ .

To model the temporal aspect of the data we assume that the latent coordinates follow a first order Markov process and, conditional on the latent trajectories, each observed adjacency matrix occurs independently. Additionally, we assume that the latent coordinates of each node are independent of one another. This corresponds to a SSM and a depiction of the dependence structure is given in Figure 3.3.1.

As in Hoff et al. (2002), the binary connections can be modelled using a logistic re-

gression model where the probability of connection depends on the latent coordinates.

A dynamic latent space network model can be specified as follows.

$$p(\mathbf{U}_0|\theta) = \prod_{i=1}^N \mathcal{N}_p(0, \tau^2 I_p) \quad (3.3.1)$$

$$p(\mathbf{U}_t|\mathbf{U}_{t-1}, \theta) = \prod_{i=1}^N \mathcal{N}_p(u_{i,t-1}, \sigma^2 I_p) \quad \text{for } t = 1, 2, \dots, T \quad (3.3.2)$$

$$p(\mathcal{Y}_t|\mathbf{U}_t, \theta) = \prod_{i<j} p_{ijt}^{y_{ijt}} (1 - p_{ijt})^{1-y_{ijt}} \quad \text{for } t = 1, 2, \dots, T \quad (3.3.3)$$

where

$$p_{ijt} = \frac{1}{1 + e^{-\eta_{ijt}}} \quad (3.3.4)$$

$$\eta_{ijt} = \alpha - \|u_{it} - u_{jt}\| \quad (3.3.5)$$

for  $(i, j) \in \{i, j \in \{1, 2, \dots, N\} | i < j\}$ .  $\theta = (\alpha, \sigma, \tau) \in \mathbb{R} \times \mathbb{R}_{>0} \times \mathbb{R}_{>0}$  denotes the model parameters;  $\alpha$  controls the global tendency for a connection to form between two nodes;  $\sigma$  defines the variance in the latent trajectories;  $\tau$  defines the variance in the initial latent coordinates. As in the static case, the latent representation captures the node specific tendencies for a connection to form between each pair of nodes. This model is the time-varying extension of the distance model of Hoff et al. (2002), and the projection model can be extended in a similar way. The model specified in (3.3.1) - (3.3.5) is simpler than the model introduced in Sewell and Chen (2015b), and we choose this specification since our focus will be on the inference mechanism.

### 3.3.3 Identifiability

Note that (3.3.3) is a function of the distance between the latent coordinates and  $\theta$ , and so  $p(\mathcal{Y}_t|\mathbf{U}_t, \theta)$  is invariant to distance-preserving transformations of  $\mathbf{U}_t$ . To resolve this, many authors rely on a Procrustes transformation which finds the coordinates  $\hat{\mathbf{U}}$  which minimise the sum of squared difference between  $\mathbf{U}$  and some reference coordinates  $\mathbf{U}_0$ . More precisely, the coordinates are given by  $\hat{\mathbf{U}} = \arg \min_{T\mathbf{U}} \text{tr}(\mathbf{U}_0 - T\mathbf{U})^T(\mathbf{U}_0 - T\mathbf{U})$ . Note that the reference coordinates  $\mathbf{U}_0$  are fixed and specified

beforehand. In the temporal setting, the Procrustes transformation can be applied as a post-processing step as in Sewell and Chen (2015b), or as part of the inference mechanism as in Sarkar and Moore (2006).

An alternative approach in the static case is to instead fix a set of latent coordinates. Fixing a single coordinate will remove the effect of translations and fixing a second will remove the effect of rotations. This approach was used by, for example, McCormick and Zheng (2015). It is important to note that it is not straightforward to directly apply this in the temporal setting. Other authors instead opt to examine the model fit on the probability space, which does not suffer from non-identifiability of  $\mathbf{U}$  (see, for example, Durante and Dunson (2014)).

### 3.3.4 Model Fitting

In the Bayesian setting, inference for dynamic latent space networks is typically carried out via a MH-within-Gibbs MCMC scheme (Sewell and Chen (2015b), Sewell and Chen (2016)) or variational Bayes (Sewell and Chen (2017)). Variational methods target a computationally cheaper approximation to the posterior, and this facilitates feasible approximate inference for higher-dimensional temporal networks. Posterior samples obtained via MCMC are from the true posterior, however this comes at a greater computational cost. Alternatively in the frequentist setting, Sarkar and Moore (2006) infer the parameters of their model using an optimisation scheme. Our focus will be on the Bayesian approach, and we aim to avoid the approximations introduced in variational methods.

Initialisation of the MCMC scheme requires careful consideration. Since there are  $N \times d \times T$  latent coordinates to estimate for a network with  $T$  time stamps, this initialisation is important to obtain good performance out of the MCMC. Typically, the latent coordinates are initialised using generalised multidimensional scaling (GMDS). Classical MDS (see Cox and Cox (2000)) takes as an input the Euclidean distance between all pairs of a collection of objects and returns a set of coordinates with the

specified distances. GMDS considers non-Euclidean measures of distance and, in the case of networks, we use the shortest path between nodes  $i$  and  $j$  as the distance measure. This idea was first introduced in Sarkar and Moore (2006).

To obtain posterior samples via MCMC we are required to evaluate the likelihood at each iteration. Note that evaluation of (3.3.3) involves a calculation over all  $\binom{N}{2}$  pairs. This scales poorly as  $N$  grows, and we comment that this is more costly for a directed network. Several authors have developed approximations of this likelihood to improve the scalability of the MCMC. For example Raftery et al. (2012) introduce the case-control approximation for latent space networks. This approach divides the connections into  $\{i, j\}$  pairs which share a tie and those that do not. Then, relying on an assumption of sparsity, they argue that the likelihood sum will be dominated by terms with  $y_{ij} = 0$ . To alleviate the computational burden of this calculation they approximate the likelihood by taking a subsample over these pairs. They note that since the model is built around distances, they should sample the node pairs for which  $y_{ij} = 0$  accordingly. Instead of a random sample, they stratify the sample according to the shortest path distances between each of the node pairs. This approximation was also used in the dynamic case in Sewell and Chen (2015b) and modified to account for missing data. An alternative approximation is introduced in Rastelli et al. (2018). This approach assumes that the latent coordinates can be represented in a finite box in  $\mathbb{R}^p$ . This box is then divided into smaller boxes which form the basis of the approximation. Since each latent coordinate lies in a single box, the authors note that the centre of the boxes can be used as an approximation to the coordinate. Building an approximation using this approach reduces the computational complexity when the number of boxes is much less than the number of nodes. This has yet to be explored for dynamic latent space networks.

To improve their MCMC scheme, Durante and Dunson (2014) rely on Pólya-gamma data augmentation (Polson et al. (2013)). This scheme was developed for fully Bayesian inference in models with binomial likelihoods, and allows the latent

coordinates to be updated via a Gibbs sampler. Whilst this is efficient, we note that this is not compatible for a dynamic latent space model based on Euclidean distance. In the model of Durante and Dunson (2014), the probability of a connection is determined by the dot product between the latent coordinates.

### 3.3.5 Extensions

The model detailed in (3.3.1) - (3.3.5) can be modified in many ways. For example, additional node-specific parameters which affect the tendency for individual nodes to form connections can be included as in Sewell and Chen (2015b). Other authors have developed models for non-binary connections and, in particular, Sewell and Chen (2016) considered weighted connections and Sewell and Chen (2015a) considered ranked connections.

In Friel et al. (2016) the authors have extended this framework to the bipartite setting in which there are two groups of nodes and connections only occur between nodes in different groups. Their model incorporates higher-order temporal dependency and they use this to model which company directors are associated with certain boards. Another extension is considered in Sewell and Chen (2017), where the authors include community structure by modelling the latent coordinates with a mixture of Gaussians. Finally, Durante et al. (2017) model multilayer networks in which different types of connections are observed for the same set of nodes, and Durante et al. (2016) model populations of networks.

## 3.4 SMC and Dynamic Latent Space Networks

The dynamic latent space network (DLSN) model discussed in Section 3.3.2 presents a natural setting for the application of SMC methods which has yet to be explored in the literature. As commented in Section 3.3.4, the latent coordinates are typically updated within an MCMC scheme (for example see Sewell and Chen (2015b)) and,



whilst this performs exact inference, the mixing times are expected to degenerate as the dimension of the state space increases. This could be due to an increase in  $T$  or  $N$ , and we expect SMC methods to perform favourably when  $T$  grows. Furthermore, the SMC framework facilitates both offline and online inference. In this section, we discuss the practicalities of SMC methods in this context and, since SMC methods perform poorly as the dimension of the latent state increases, we also focus on approaches which are appropriate for increasing values of  $N$ . In Sections 3.4.1 and 3.4.2 we consider state and parameter estimation, respectively.

### 3.4.1 State Estimation

Throughout this section we focus on estimation of the latent coordinates for the model detailed in (3.3.1) - (3.3.5). We first consider standard particle filtering algorithms and then examine a simplifying approximation which improves the performance of standard algorithms. Finally, we consider two approaches from the high-dimensional SMC methodology. Although there is a broad literature on particle filtering for high-dimensional state spaces (see Section 3.2.4), we only consider methodology that is appropriate for DLSNs.

We apply each method to networks simulated from the model detailed in (3.3.1) - (3.3.5) and, when the number of nodes is the same, each method is applied to the same simulated network. For all examples we take  $P = 2, T = 30, \sigma = 0.05, \tau = 0.1, \alpha = 0.2$ , and the number of nodes  $N$  is varied. For the simplifying approximation, we also consider networks generated with a dot-product so that (3.3.5) is replaced by  $\eta_{ijt} = \alpha + u_{it}^T u_{jt}$ , and for this formulation we take  $\alpha = 0$ . To assess the performance of each algorithm we consider the ESS and average MSE in the estimated connection probabilities.

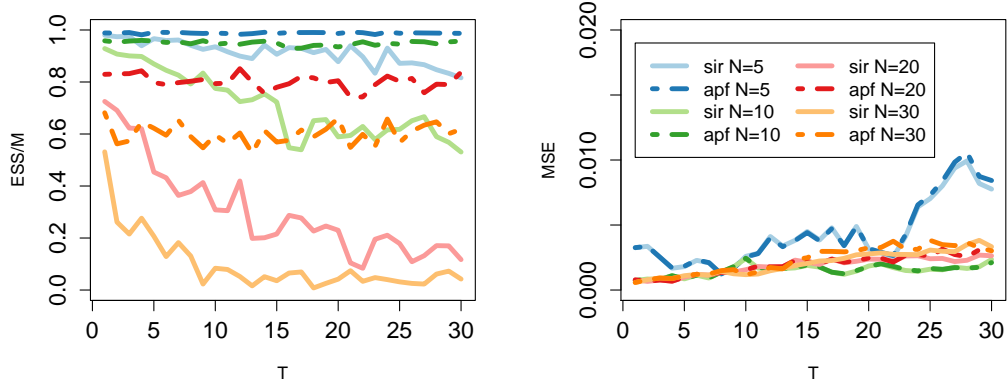


Figure 3.4.1: Performance of SIR and APF as  $N$  increases. The ESS and average MSE in probability are shown in the left and right panels, respectively. For each filter, the number of particles was fixed at  $M = 50000$ .

### SIR and APF

It is natural to begin by considering the standard particle filters and in this section we focus our attention on the SIR and APF filters. The SIR filter is obtained by setting  $q_\theta(\cdot|x_{t-1}, y_t) = f_\theta(\cdot|x_{t-1})$  and  $w_t = g_\theta(y_t|\cdot)$  in Algorithm 1 and the APF is detailed in Algorithm 2. We expect the APF to outperform the SIR since particles at time  $t$  are propagated according to the observation at time  $t + 1$ . The SIR has the advantage of being straightforward to implement, however the performance of this filter degrades as the dimension of the state space increases.

To implement the APF filter we must resample the particles at time  $t$  according to the predictive  $\zeta_t^{(i)} \propto q_{t-1}^{(i)} p(y_t|x_{t-1}^{(i)})$ . For our model of interest, there is no analytic expression for this and so we must rely on an approximation. Pitt and Shephard (1999) suggest relying on a Taylor expansion when the transition and observation densities are Gaussian and log-concave, respectively. Whilst our model satisfies the first condition, the observation density (3.3.3) is not log-concave (see Section 3 of Hoff et al. (2002)) and so this approximation is not appropriate. Instead we use an approximation of the form  $p_\theta(y_t|\mu_t)$  where  $\mu_t$  is a deterministic function of  $x_{t-1}$ . We take  $\mu_t = E(X_t|X_{t-1} = x_{t-1})$ , but comment that there are alternative choices.

When estimating the latent states we note that the dependence in (3.3.3) means that each particle must correspond to the entire set of latent coordinates. Therefore we obtain  $\{\mathbf{U}_t^{(i)}, w_t^{(i)}\}_{i=1}^M$  as the particle set at time  $t$ . The dimension of the latent state is given by  $N \times d$ , and we now consider the performance of the SIR and APF as the number of nodes  $N$  increases. It is important to note that these algorithms are theoretically appropriate for high-dimensional state spaces, but the number of particles required to achieve good performance comes with an associated computational cost which quickly becomes impractical. Under certain settings, Bickel et al. (2008) show that the number of particles should scale exponentially with the dimension of the state space.

Figure 3.4.1 compares the performance of the SIR and APF filters as  $N$  increases. Overall we see that the APF outperforms the SIR filter. Both filters suffer from a degradation in performance as  $N$  increases, though it is clear that the SIR filter degrades at a faster rate. The performance can be improved by increasing  $M$ , though this will quickly become infeasible. From this we conclude that it is only appropriate to rely on these filters when  $N$  is small.

### Independent Approximation

In Section 3.4.1 we observed that the performance of standard particle filtering algorithms degrade with an increase in the dimension of the state space. Recall that, due to the dependence in (3.3.3), each particle must correspond to the entire set of latent coordinates  $\mathbf{U}$ . In this section we examine an approximation which facilitates inference via standard particle filtering methods for networks of increasing  $N$ . More specifically, we consider the following approximations to (3.3.1), (3.3.2) and (3.3.4).

$$s_{ij0} \sim \mathcal{N}(0, \tau_s^2) \tag{3.4.1}$$

$$s_{ijt} \sim \mathcal{N}(s_{ij(t-1)}, \sigma_s^2) \tag{3.4.2}$$

$$\tilde{\eta}_{ijt} = \alpha + s_{ijt} \tag{3.4.3}$$

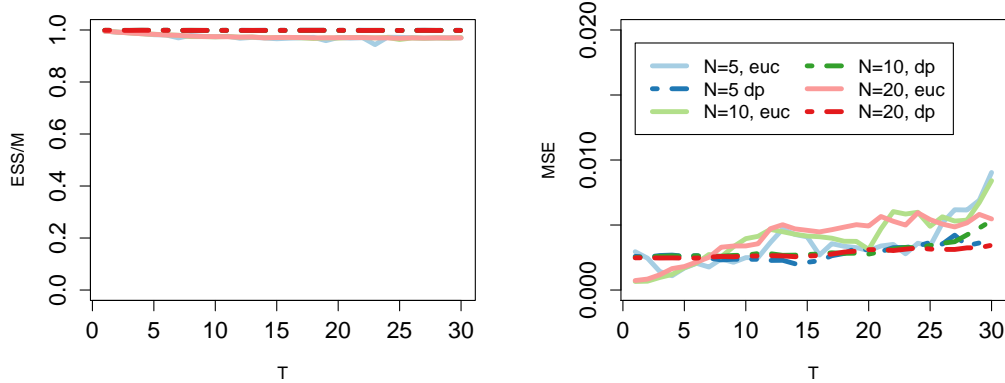


Figure 3.4.2: Performance of the independent approximation as  $N$  increases. The ESS and average MSE in probability are shown in the left and right panels, respectively. For each filter, the number of particles was fixed at  $M = 1000$ .

for  $t = 1, 2, \dots, N$ , and  $(i, j) \in \{i, j \in \{1, 2, \dots, N\} | i < j\}$ . In these equations,  $s_{ijt}$  represents the latent ‘similarity’ between nodes  $i$  and  $j$  at time  $t$ . Note that in (3.3.5) the linear predictor is specified as a function of the Euclidean distance between  $u_{it}$  and  $u_{jt}$ , though other choices can be made here. Since the Euclidean distance is a metric, it must satisfy positivity and the triangle inequality. Imposing these constraints onto the latent similarities is challenging and it is likely that this approximation will be poor. Alternatively, the dot product  $u_{it}^T u_{jt}$  is not a metric and does not satisfy these constraints.

This approximation avoids the dependence in (3.3.3) and so we may estimate each  $\{s_{ijt}\}_{t=1}^T$  independently. Note that we lose the latent representation when using this approximation, however the performance of the filters will not degrade with increasing  $N$ .

Figure 3.4.2 shows the performance of this filter as  $N$  increases for networks simulated using Euclidean distance and the dot product. The simulated data correspond to the data used in Figure 3.4.1 when connections are determined by the Euclidean distance. We estimate  $\sigma_s^2$  in (3.4.2) from the variance of the Euclidean distance and dot-product of the known latent trajectories. Since each similarity is estimated inde-

pendently, increasing  $N$  does not affect the ESS. The MSE reflects the intuition that this approximation is more appropriate for the dot product, since this imposes fewer constraints on the connections. Although this approximation appears comparable to the SIR and APF in Figure 3.4.1, it is not obvious how to characterise the scenarios in which this approach is appropriate.

## Nudging

Since standard particle filtering algorithms perform poorly as  $N$  increases, we must explore alternative SMC methodology. In this section we will focus on the approach of Akyildiz and Míguez (2019) where nudging steps are introduced to guide particles to regions of the state space with higher likelihoods. The authors develop this to improve the performance of particle filtering algorithms as the dimension of the state space increases. Here we will introduce the details of this approach and then consider this in the context of DLSNs.

Akyildiz and Míguez (2019) develop a procedure which can be applied to a general particle filter, but for simplicity we introduce their methodology using the SIR filter. For a particle filter with  $M$  particles, their algorithm specifies a number of particles  $M^*$  which are modified according to a nudging rule. Several nudging strategies are suggested, including using gradient information, along with different ways of choosing the particles to be nudged. A key contribution of their work is showing that, providing a sufficiently small number of particles are nudged, the potentially computationally costly step of correcting for the nudge can be avoided.

We focus on gradient based nudging in which a subset of particles are shifted according to  $\gamma \nabla g(y_t | x_t)$  for some  $\gamma \in \mathbb{R}_{>0}$ . Let  $\mathcal{I}$  indicate the set of particles which will be nudged, where  $\mathcal{I} \subset [M] = \{1, 2, \dots, M\}$  and  $|\mathcal{I}| = M^*$ . Akyildiz and Míguez (2019) show that, when  $\gamma M^* \leq \sqrt{M}$ , the correction for the nudge step can be omitted without affecting the convergence of the filter and we restrict to this case. The details of the procedure are presented in Algorithm 4 within an SIR filter, and we comment

---

**Algorithm 4** Nudged Bootstrap Particle Filter

---

For  $t \geq 1$ :

*Propagate:*

Sample  $M$  particles  $\{x_t^{(i)}\}_{i=1}^M$  from  $f(\cdot|x_{t-1})$

*Nudging:*

Calculate  $\tilde{x}_t^{(i)} = x_t^{(i)} + \gamma \nabla g(y_t|x_t)$  for  $i \in \mathcal{I}$ , where  $\mathcal{I} \in [M]$  and  $|\mathcal{I}| = M^*$

*Weight:*

Weight each particle according to  $w_t^{(i)} \propto g(y_t|\tilde{x}_t^{(i)})$

*Resample:*

Resample and store particles

---

that other choices for the nudging mechanisms and for obtaining  $\mathcal{I}$  are described in Akyildiz and Míguez (2019).

To implement gradient nudging for the DLSN model detailed in Section 3.3.2, we wish to find an expression for  $\nabla_{\mathbf{U}_t} \log p(\mathcal{Y}_t|\mathbf{U}_t)$ . Since the conditional likelihood is expressed in terms of  $u_{it}$ , we instead find  $\nabla_{u_{kt}} \log p(\mathcal{Y}_t|\mathbf{U}_t)$ . This is given by

$$\nabla_{u_{kt}} \log p(\mathcal{Y}_t|\mathbf{U}_t) = \sum_{j \in [N] \setminus k} \frac{u_{kt} - u_{jt}}{\|u_{kt} - u_{jt}\|} \left\{ \frac{e^{\alpha - \|u_{kt} - u_{jt}\|}}{1 + e^{\alpha - \|u_{kt} - u_{jt}\|}} - y_{kjt} \right\}, \quad (3.4.4)$$

and details of this calculation can be found in Appendix A.1.

The expression (3.4.4) has an intuitive interpretation. Firstly, consider the case when  $y_{kjt} = 1$  and  $\|u_{kt} - u_{jt}\|$  is large. In this case the the exponent term will be small and so the  $-y_{kjt}$  term dominates. This means that  $u_{kt}$  and  $u_{jt}$  will be moved closer together after the gradient shift. Similarly, when  $y_{kjt} = 0$  but  $\|u_{kt} - u_{jt}\|$  is small the gradient will move the coordinates further apart. Hence, the gradient will aim to move nodes that are connected closer together and nodes that are not connected further apart.

There are several ways in which we can apply the gradient nudge (3.4.4). Essentially, we must consider all node pairs and move them closer together or further apart as appropriate. For a given node pair we may opt to move a single coordinate

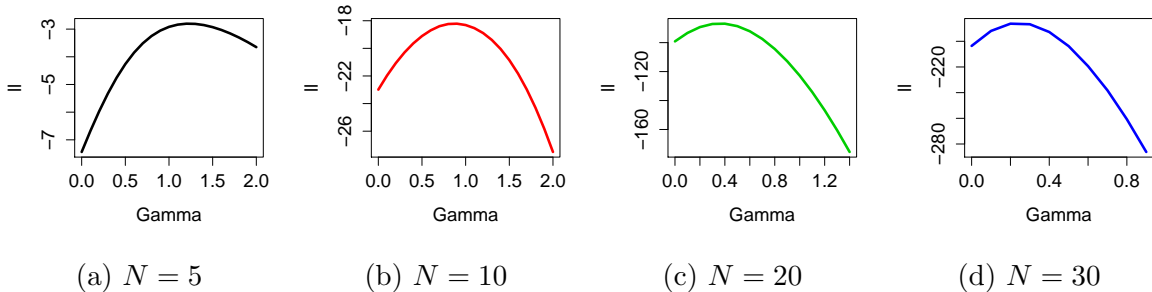


Figure 3.4.3: Effect of changing  $\gamma$  on the log-likelihood value for different sized networks. Left to right: network of size  $N = 5, 10, 20$  and  $30$ , all with  $d = 2$ .

proportional to  $\gamma$ . However, recall in our model that the latent trajectory of each node follows a Gaussian random walk. Therefore we should ensure that certain nodes are not shifted disproportionately more than others. To achieve this, we instead opt to move each node in a node pair proportional to  $\gamma/2$ .

The choice of  $\gamma$  plays an important role in the performance of this approach. To explore this, we examine the effect of nudging the states  $\mathbf{U}$  by  $\gamma \nabla \log p(\mathcal{Y}|\mathbf{U})$  for different values of  $\gamma$ . Figure 3.4.3 shows the conditional log-likelihood of the  $\gamma$  shifted coordinates,  $\mathbf{U}_\gamma = \mathbf{U} + \gamma \nabla \log p(\mathcal{Y}|\mathbf{U})$ , as a function of  $\gamma$  for networks of different sizes. For each plot, the starting value of  $\mathbf{U}$  was kept the same so that any change in log-likelihood is due to  $\gamma$ . From this figure, we see that if  $\gamma$  is too large then the coordinates are nudged beyond regions of higher likelihood, and that  $\gamma$  must scale according to  $N$ . For example, comparing Figures 3.4.3a and 3.4.3c we see that the same value of  $\gamma$  can correspond to an increase in log-likelihood in Figure 3.4.3a and a decrease in log-likelihood in Figure 3.4.3c. This is intuitive since each node is nudged according to all other  $N - 1$  nodes. Finally, we comment that if the initial value of  $\mathbf{U}$  is poor, it may be difficult to improve the log-likelihood through gradient nudging.

Figure 3.4.4 shows the performance of nudging within an SIR filter for increasing  $N$ . For each  $N$ , we apply nudging with  $\gamma = (0.1, 0.3, 0.5, 0.7, 0.9)$ . Although the performance is reasonable in terms of MSE, it is clear that nudging can drastically alter the distribution of the filtering weights. Ultimately, the performance of nudg-

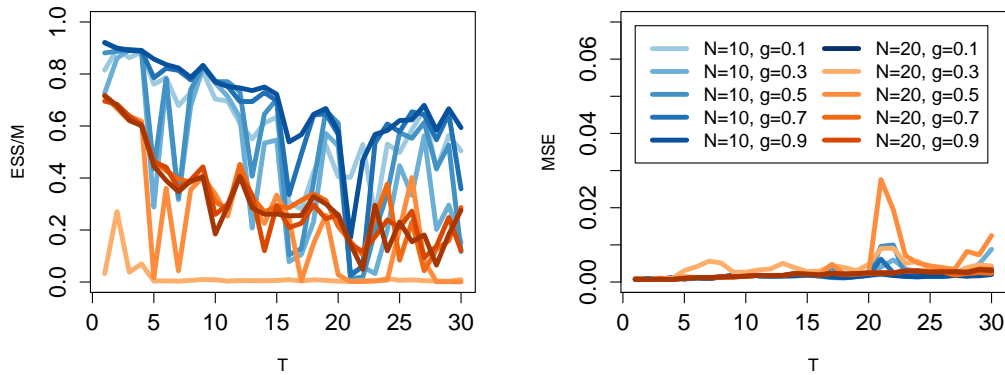


Figure 3.4.4: Performance of nudging within an SIR filter as  $N$  increases for different values of  $\gamma$ . The ESS and average MSE in probability are shown in the left and right panels, respectively. The number of particles was fixed at  $M = 10000$ .

ing will depend on the underlying filter and the plot analogous to Figure 3.4.4 for nudging within an APF looks largely similar. To improve the stability we also considered scaling the gradient according to its magnitude, however this did not offer a significant improvement. It is clear that the performance of nudging in this context is unpredictable.

## GIRF

We now consider the Guided Intermediate Resampling Filter (GIRF), as introduced in Park and Ionides (2019), for high-dimensional filtering. In this work artificial intermediary states are introduced between observations to guide particles to regions of the state space with high probability. Here we will overview the GIRF algorithm and consider its application to DLSNs.

Recall the general SSM (3.2.1) in which we let  $\{y_t\}_{t=1}^T$  represent a sequence of observations. The GIRF introduces  $S - 1$  intermediary time steps between each pair of observations  $\{y_t, y_{t+1}\}_{t=1}^{T-1}$  and we denote these by  $\{\tau_{t,s}\}_{s=0}^S$ . We assume that the latent transition density can be decomposed as

$$f_{\theta}(x_{t+1}|x_t) = f_{\theta}(x_{\tau_{t,1}}|x_t)f_{\theta}(x_{\tau_{t,2}}|x_{\tau_{t,1}})\dots f_{\theta}(x_{t+1}|x_{\tau_{t,S-1}}). \quad (3.4.5)$$



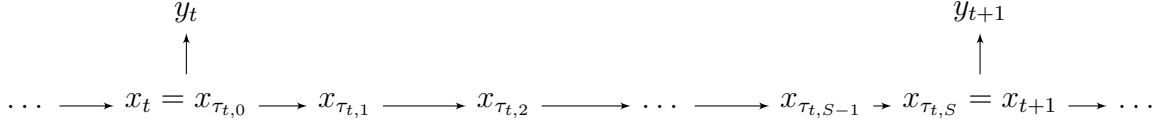


Figure 3.4.5: Intermediary steps  $\{x_{\tau_{t,s}}\}_{s=0}^S$  between observations  $y_t$  and  $y_{t+1}$ .

where  $\{\tau_{t,s}\}_{s=0}^S$  satisfy  $\tau_{t,0} := t < \tau_{t,1} < \dots < \tau_{t,S-1} < \tau_{t,S} := t + 1$ . Note that  $\tau_{t,0}$  and  $\tau_{t,S}$  correspond to the times  $t$  and  $t + 1$ , respectively, and for a depiction of the intermediary states see Figure 3.4.5.

At each intermediary time step the particles are weighted according to an assessment function  $u_{\tau_{t,s}}(x)$  which guides particles towards future observations. This function should be chosen so that  $u_{\tau_{t,0}}(x) = 1$  and  $u_{\tau_{t,S}}(x) = g(y_{t+1}|x)$ , and we will discuss particular choices below. Given this function, particles at time step  $\tau_{t,s}$  are then weighted according to

$$\omega_{\tau_{t,s}}(x_{\tau_{t,s}}, x_{\tau_{t,s-1}}) = \begin{cases} \frac{v_{\tau_{t,s}}(x_{\tau_{t,s}})}{v_{\tau_{t,s-1}}(x_{\tau_{t,s-1}})} & \text{if } \tau_{t,s-1} \notin 1:T \\ \frac{v_{\tau_{t,s}}(x_{\tau_{t,s}})}{v_{\tau_{t,s-1}}(x_{\tau_{t,s-1}})} g(y_t|x_{\tau_{t,s-1}}) & \text{if } \tau_{t,s-1} \in 1:T \end{cases}. \quad (3.4.6)$$

Using (3.4.6), we see that the likelihood  $\ell(y_{1:T}) = \mathbb{E} \left[ \prod_{t=1}^T g(y_t|x_t) \right]$  can be approximated by  $\hat{\ell} = \prod_{t=0}^T \prod_{s=1}^S \frac{1}{M} \sum_{i=1}^M \omega_{\tau_{t,s}}(x_{\tau_{t,s}}^{(i)}, \tilde{x}_{\tau_{t,s-1}}^{(i)})$ .

Algorithm 5 details the GIRF for a general SSM where  $L = \log \hat{\ell}$ . To implement this procedure we must specify  $v_{\tau_{t,s}}(x)$  and the number of intermediary states  $S$ . Park and Ionides (2019) suggest choosing  $v_{\tau_{t,s}}(x) \approx p(y_{t+1:t+B}|x_{\tau_{t,s}} = x)$  so that particles are guided towards  $B$  future observations. Additionally, they show that  $S$  should scale linearly according to the dimension of the latent states.

A connection can be drawn between the GIRF and several existing particle filtering schemes. For example, we obtain the SIR filter as a special case if  $S = 1$  and  $v_{\tau_{t,s}} = g(y_t|x_{\tau_{t,s}})$ . Additionally, the APF is also a special case when  $S = 1$  and  $v_{\tau_{t,s}} = g(y_t|x_{\tau_{t,s}})g(y_{t+1}|\mu_{t+1}(x_{\tau_{t,s}}))$ , where  $\mu_{t+1}(x_{\tau_{t,s}})$  represents a prediction of  $x_{t+1}$  given  $x_{\tau_{t,s}}$ . Alternatively, we may guide the particles via a sequence of bridging densities given

---

**Algorithm 5** Guided Intermediate Resampling Filter (GIRF)

---

**Initialise:**  $L = 0, \tilde{x}_{\tau_0,0}^{(i)} \sim \mu(\cdot)$  for  $i \in 1 : M$

**For**  $t = 0 : T - 1$

**For**  $s \in 1 : S$

$x_{\tau_t,s}^{(i)} \sim f(\cdot | \tilde{x}_{\tau_t,s-1}^{(i)})$  for  $i \in 1 : M$

$w_{\tau_t,s}^{(i)} = \omega_{\tau_t,s}(x_{\tau_t,s}^{(i)}, \tilde{x}_{\tau_t,s-1}^{(i)})$  for  $i \in 1 : M$

$L = L + \log(\sum_j w_{\tau_t,s}^{(j)})$

        Sample  $b^{(i)}$  such that  $\mathbb{P}(b^{(i)} = j) \propto w_{\tau_t,s}^{(j)}$  for  $i \in 1 : M$

        Set  $\tilde{x}_{\tau_t,s}^{(i)} = x_{\tau_t,s}^{(b^{(i)})}$

**End**

**End**

---

by, for example,  $v_{\tau_t,s}(x) = p(y_{t+1}|x)^{s/S}$ . This relates to annealed importance sampling (see Neal (2001)) and has been considered in the high-dimensional filtering context in Beskos et al. (2017), Beskos et al. (2014a) and Beskos et al. (2014b). Finally, Park and Ionides (2019) comment that their methodology is similar to that of Del Moral and Murray (2015), however Del Moral and Murray (2015) instead focus on the case where observations are highly informative.

To consider the application of the GIRF to DLSNs, we must specify the form of  $v_{\tau_t,s}(\mathbf{U})$ . A straightforward option is to take  $v_{\tau_t,s}(\mathbf{U}) = p(\mathcal{Y}_{t+1}|\mathbf{U})$ . Alternatively, following Section 5 of Park and Ionides (2019), we may incorporate  $B$  future observations by choosing

$$v_{\tau_t,s}(\mathbf{U}) = \prod_{b=1}^{\min\{B,T-t\}} (v_{\tau_t,s,\tau_t+b}(\mathbf{U}))^{\eta_{\tau_t,s,\tau_t+b}} \quad (3.4.7)$$

where  $v_{\tau_t,s,\tau_t+b}(\mathbf{U})$  approximates  $p_{\mathcal{Y}_{t+b}|\mathbf{U}_{\tau_t,s}}(\mathcal{Y}_{t+b}|\mathbf{U})$  and  $\eta_{\tau_t,s,\tau_t+b}$  controls the contribution of  $v_{\tau_t,s,\tau_t+b}(\mathbf{U})$  in the assessment function. We take

$$\eta_{\tau_t,s,\tau_t+b} = 1 - \frac{(bS - s)}{S[(t+b) - \max(t+b-B, 0)]} \quad (3.4.8)$$

so that the contribution of observations decreases as a function of distance from  $\tau_t,s$ . This ensures that the potentially less accurate approximations have a smaller

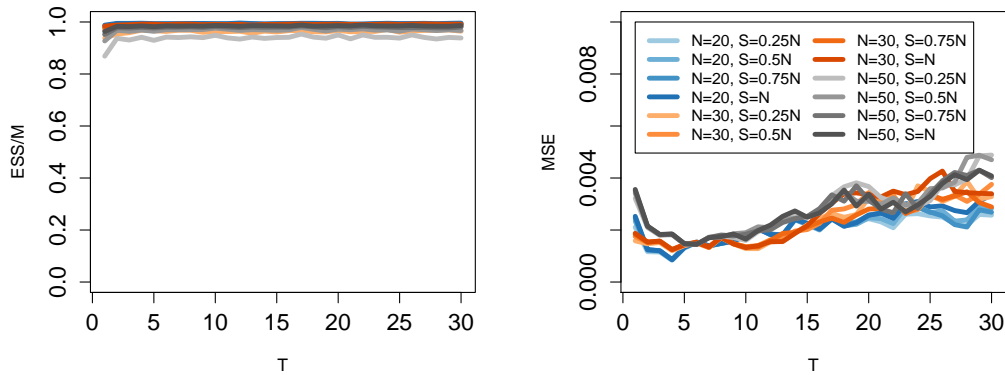


Figure 3.4.6: Performance of GIRF as  $N$  increases for varying number of intermediary states  $S$ . The ESS and average MSE in probability are shown in the left and right panels, respectively. The number of particles was fixed at  $M = 1000$ .

contribution to  $v_{\tau_{t,s}}(\mathbf{U})$ . It may be possible to approximate  $p_{\mathcal{Y}_{t+b}|\mathbf{U}_{\tau_{t,s}}}(\mathcal{Y}_{t+b}|\mathbf{U})$  via simulation, however this is likely to be computationally expensive. Instead we take  $v_{\tau_{t,s},\tau_{t+b}}(\mathbf{U}) = g(\mathcal{Y}_{t+b}|m_{\tau_{t,s}}(\mathbf{U}))$ , where  $m_{\tau_{t,s}}(\mathbf{U}) = \mathbb{E}[\mathbf{U}_{t+b}|\mathbf{U}_{\tau_{t,s}} = \mathbf{U}] = \mathbf{U}$ , which can be conveniently calculated.

Figure 3.4.6 shows the performance of the GIRF for different choices of  $S$  and varying  $N$  when the assessment function is taken as  $v_{\tau_{t,s}}(\mathbf{U}) = p(\mathcal{Y}_{t+1}|\mathbf{U})$ . There was little difference in the plots when the assessment function was given by (3.4.7), though it is not clear whether this is true in general. Overall, we see a much improved performance for networks with larger  $N$ . In this example there is little difference between the different choices of  $S$ , suggesting that just a few intermediary states vastly improves performance of the filter even when the number of particles is relatively small. By comparing the right plots of Figures 3.4.6 and 3.4.1, we see that the GIRF performs much better in terms of MSE also.

### 3.4.2 State and Parameter Estimation

Based on the discussion in Section 3.4.1, we only consider the GIRF for the remainder of this chapter. Here, we discuss the details of estimating static parameters via

---

**Algorithm 6** Online parameter estimation within the GIRF

---

**Initialise:** set  $S_0 = 0, L = 0, \theta_0, m_0^{(i)} = 0$  and  $\tilde{x}_{\tau_0,0}^{(i)} \sim \mu_{\theta_0}(\cdot)$  for  $i = 1, 2, \dots, M$ .

**For**  $t = 1, 2, \dots, T$

1) Run  $S$  intermediary steps of the GIRF to obtain  $\{x_t^{(i)}\}_{i=1}^M, \{a_{t-1}^{(i)}\}_{i=1}^M$  and  $\{w_t^{(i)}\}_{i=1}^M$

(innermost loop in Algorithm 5)

2) Update the mean approximation

$$m_t^{(i)} = \lambda m_{t-1}^{(a_{t-1}^{(i)})} + (1 - \lambda) S_{t-1} + \nabla \log g_{\theta_{t-1}}(y_t | x_t^{(i)}) + \nabla \log f_{\theta_{t-1}}(x_t^{(i)} | x_{t-1}^{(a_{t-1}^{(i)})})$$

3) Update the score vector

$$S_t = \sum_{i=1}^M x_t^{(i)} m_t^{(i)}$$

4) Update theta:  $\theta_t = \theta_{t-1} + \gamma_k (S_t - S_{t-1})$

**End**

---

gradient ascent within the GIRF. We rely on the approach discussed in Section 3.2.3 and the online  $\theta$  estimation scheme is outlined in Algorithm 6. For offline estimation the  $\theta$  update is applied after a full pass of the GIRF and  $S_T$  replaces the term  $(S_t - S_{t-1})$  in Algorithm 6. Expressions for the gradient for the model in (3.3.1) - (3.3.5) are give in Appendix A.2.

To set the hyperparameter  $\tau$  we estimate an initial set of latent coordinates via GMDS and set  $\tau$  to be the sample variance of these coordinates. The GMDS takes as input a distance matrix and returns a set of coordinates in  $\mathbb{R}^p$  with the corresponding distances. Following, Sarkar and Moore (2006) we take the path length between each pair of nodes as the distance.  $\sigma$  is initialised by considering the variance between the GMDS initialisation for the observations at  $t = 1$  and  $t = 2$ , and  $\alpha$  is initialised as the value which maximises (3.3.3) for  $u^*$  simulated according to  $\hat{\sigma}$  and  $\tau$ .

## 3.5 Simulations

In this section we explore the properties of the GIRF with parameter estimation, as discussed in Section 3.4. First, we will consider the performance of this approach under three different simulated data scenarios in Section 3.5.1. Then, we will consider the scalability of this approach as  $N$  and  $T$  increase in Section 3.5.2.

### 3.5.1 Alternative Scenarios

In this section our focus will be on the accuracy of the GIRF for networks simulated from three different scenarios. We consider simulated networks with the following characteristics.

- (S1) The nodes begin in a single group at time  $t = 1$ , evolve into two distinct groups and return to a single group and time  $T$ .
- (S2) The nodes begin in a single group at time  $t = 1$  and then evolve into two groups.
- (S3) The density of the networks change over time due to the latent representation and  $\alpha$  remains fixed.

To simulate from each of these scenarios, we model the latent trajectories as a random walk that is guided by a deterministic function, similarly to how the community memberships of the latent trajectories are modelled in Sewell and Chen (2017). Details of this are provided in Appendix A.3, and for all cases we set  $N = 30$ ,  $d = 2$  and  $T = 75$ .

We fit the model detailed in (3.3.1) - (3.3.5) with online and offline  $\theta$  estimation to the first  $T - 1$  observations and assess predictive accuracy for the  $T^{\text{th}}$  observation. Based on the simulations in Section 3.4.1 we set  $S = 15$ , and we consider the performance of the guide function given in (3.4.7) and (3.4.8) with  $B \in \{1, 2, 3\}$ , where  $B$  is the number of future observations incorporated into the guide function. When  $B = 1$ , we take the guide function to be  $p(\mathcal{Y}_{t+1}|\mathbf{U})$ .

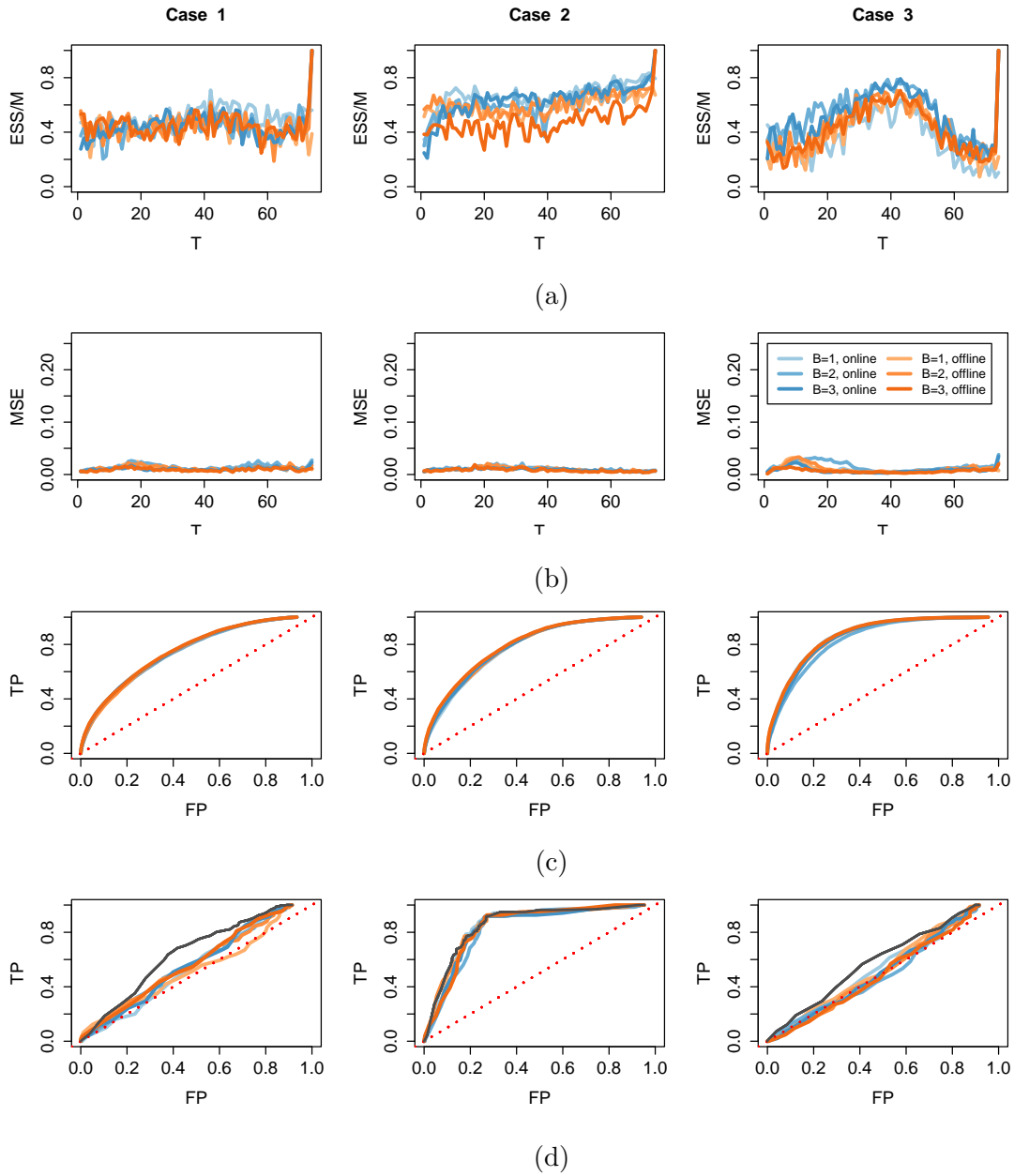


Figure 3.5.1: Summary of online and offline estimation, shown in blue and orange, respectively. Throughout we have left: (S1), middle: (S2), right: (S3). Figure 3.5.1a shows the effective sample size and Figure 3.5.1b shows the mean square error in probability. Figure 3.5.1c shows the ROC curves for observations 1 to  $T - 1$ , and the line  $y = x$  is shown in red. Figure 3.5.1d shows the ROC curve for the predicted probabilities at time  $T$ , and the ROC curve for the true probabilities is shown in grey.

Figure 3.5.1 summarises the performance of the filter for each case in terms of ESS, average MSE in probability and ROC curves. Each ROC curve compares the proportion of true and false positives for a classifier at various thresholds, and a classifier is most accurate when the ROC curve is close to the upper left hand corner of the plot. Compared to Figure 3.4.6, Figure 3.5.1 appears to show a poorer performance in terms of ESS and MSE. However, here we are also estimating static parameters meaning that the inference task is more challenging and so we expect to see a slight degradation in performance. We see that online estimation performs comparably to offline estimation for all cases, and there is no clear advantage of incorporating multiple future observations in the guide function. The performance in terms of ROC is comparable to the truth, and we comment that, in Figure 3.5.1d, we see a different behaviour in case (S2) since the data belong to two communities at the  $T^{\text{th}}$  observation. This differs to cases (S1) and (S3) where the data belong to a single group at the  $T^{\text{th}}$  observation.

Overall, we see that the filter performs well in a range of scenarios, and additional improvement may be made by incorporating other structures into the model. For example, scenarios (S1) and (S2) may be modelled with community structure and scenario (S3) may be modelled with a temporally evolving base rate  $\alpha$ . We also find that estimates for  $\sigma$  and  $\alpha$  are consistent between filters, though there is more variability in the online cases. The estimates are given in Figure A.3.1 in Appendix A.3.

### 3.5.2 Scalability

A key motivation for considering SMC in the context of DLSNs was the improved scalability as the number of observations in time  $T$  increases. In this section we explore the scalability of our choice of SMC algorithm for online estimation, and comment that each iteration of the offline estimation procedure will scale similarly to the online case. We consider the effects of increasing  $N$  and  $T$  separately, and

simulate data from model detailed in (3.3.1) - (3.3.5) for each case. In particular, we simulate data according to

1. **Increasing  $T$**

$$T \in \{50, 100, 500, 1000\}, N = 20, P = 2, \alpha = 1.5, \sigma = 0.125, \tau = 0.075$$

2. **Increasing  $N$**

$$T = 25, N \in \{50, 75, 100\}, P = 2, \alpha = 1.5, \sigma = 0.125, \tau = 0.075$$

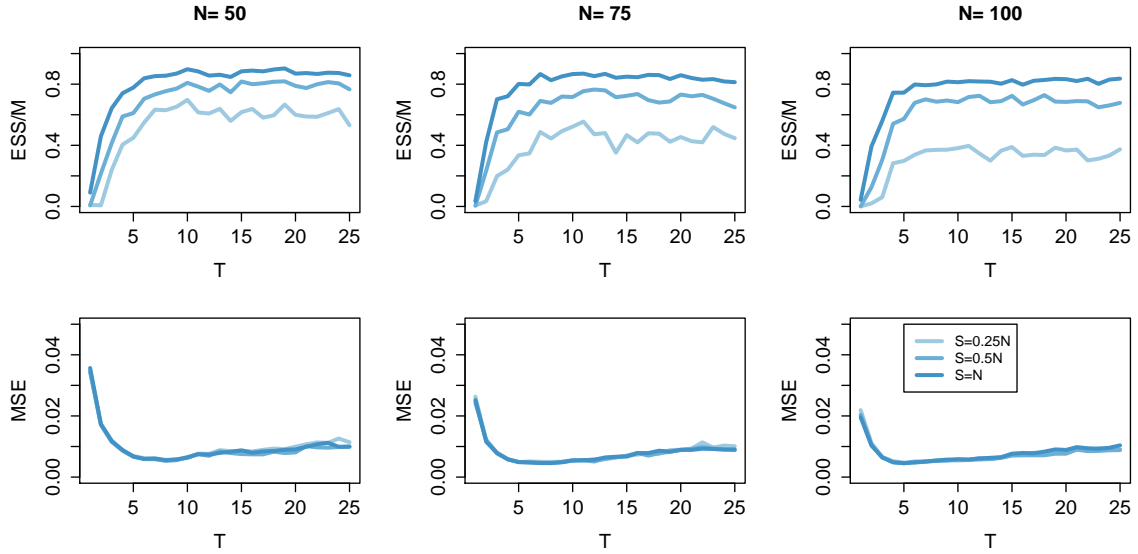
where, for each filter, we set  $S = N$ . For increasing  $T$ , we simulate a single dataset with  $T = 1000$  and apply the filter to the first  $t$  observations, where  $t \in \{50, 100, 500, 1000\}$ .

The performance of online estimation for each case is summarised in Figure 3.5.2. Overall we see a good performance and, as expected, decreasing the number of intermediary states reduces the performance. Similarly to the example in Section 3.5.1, the performance is slightly poorer than in Section 3.4.1 due to added difficulty from estimation of  $\theta$ . The time to run each filter is given in Figure 3.5.3 and we see that the scaling in terms of  $N$  is much worse than the scaling in terms of  $T$ . This is due to the additional terms in the likelihood and increased number of intermediary states required to obtain a good performance from the filter. We may reduce the computation cost by reducing  $S$ , however from Figure 3.5.2 we see that this will degrade the performance of the filter.

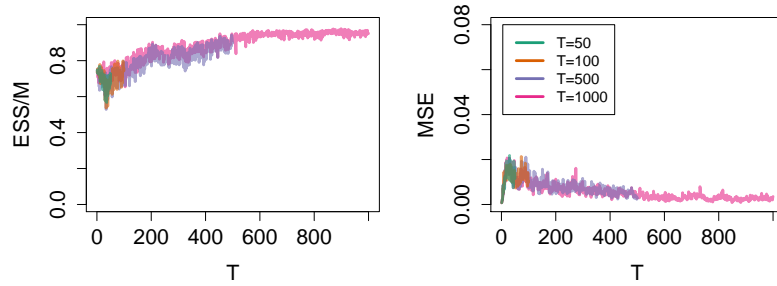
## 3.6 Classroom Contact Dataset

In this section we consider a dataset describing face-to-face contact among primary school children. The data record a connection if two students face each other within a 20 second interval, and the data is available from [www.sociopatterns.org](http://www.sociopatterns.org) and was published in Stehlé et al. (2011) and Gemmetto et al. (2014). We analyse interactions among a class of 25 school children on an aggregate level, where we record whether





(a) Summary of fit for increasing  $N$  and  $T = 25$ . Top: ESS for varying  $S$ . Bottom: MSE for varying  $S$ .



(b) Summary of fit for increasing  $T$  with  $N = 20$  and  $S = 20$ .

Figure 3.5.2: Performance of online estimation as the dimension of the data increases.

each pair of students interacted within a 5 minute interval. If there is an interaction between students  $i$  and  $j$  within the  $t^{th}$  interval we set  $y_{ijt} = 1$ , otherwise we take  $y_{ijt} = 0$ .

Similarly to Durante and Dunson (2014) we opt to assess the model fit on the connection probabilities, though we may also obtain a visualisation of the dataset after a procrustes transform has been applied to remove the effects of distance-preserving transformations on the latent coordinates. We will compare the fit for the model given in (3.3.1) - (3.3.5) with the dot-product formulation in which (3.3.5) is replaced by

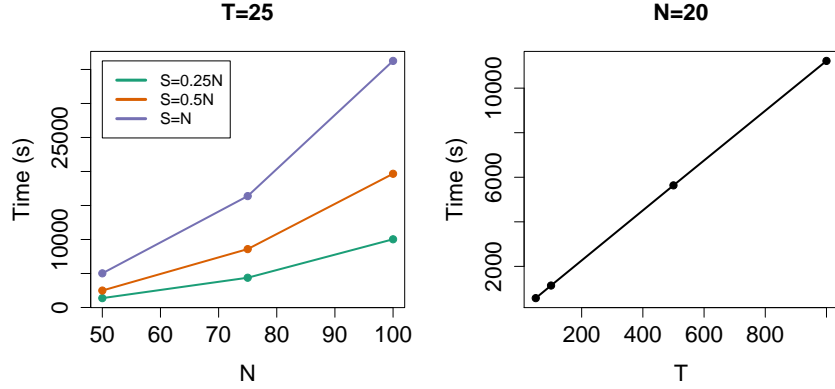
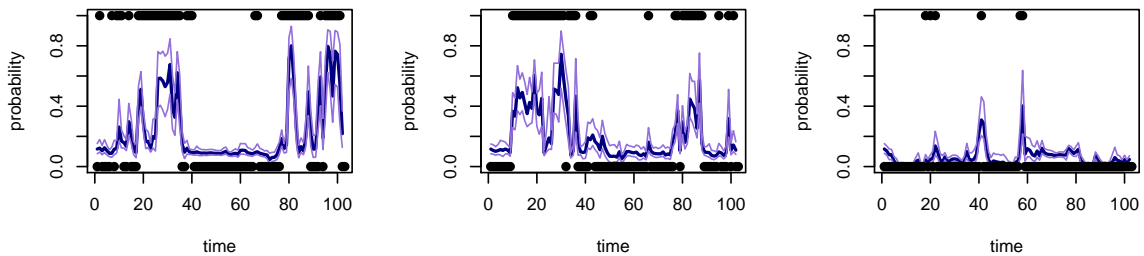


Figure 3.5.3: Time taken for each filter for increasing  $N$  (left) and increasing  $T$  (right).

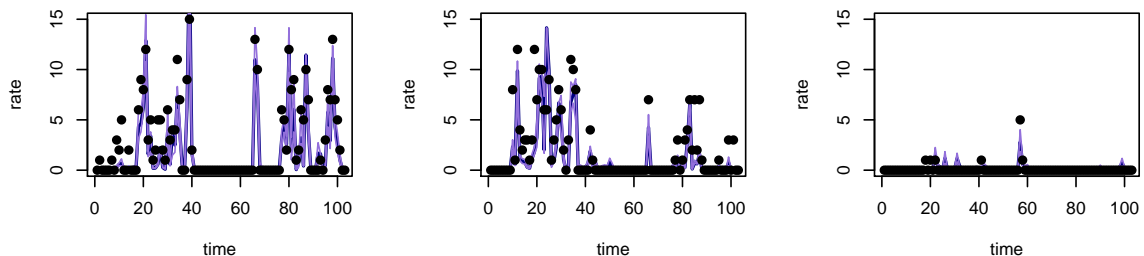
$\eta_{ijt} = \alpha + u_{it}^T u_{jt}$ . For each model, we fit the data to the first  $T - 1$  observations and examine the predictives for the  $T^{\text{th}}$  observation. Figure (3.6.1a) shows a subset of the estimated connection probabilities for the dot-product formulation, and we see that the connection probabilities reflect the observations well.

To further assess the fit we examine the ROC curve for the dot-product and Euclidean distance formulation. This is shown in Figure 3.6.2a and we see that the models perform similarly. We also assess the predictive probabilities for the  $T^{\text{th}}$  observation by simulating  $\hat{y}_{ijT}$  according to  $\hat{p}_{ijT}$  and calculating the average absolute error (AAE) between the estimates and the observations. This is given in Figure 3.6.2b, and we see that the  $T^{\text{th}}$  observations are predicted reasonably well.

An advantage of our approach is that it can be easily adapted to variants of the model. We will now consider the dataset obtained by counting the number of times students  $i$  and  $j$  interact within the  $t^{\text{th}}$  time period, so that  $y_{ijt} \in \mathbb{N}_0$ . To model this data we let  $p_{ijt} = e^{-\lambda_{ijt}} \lambda_{ijt}^{y_{ijt}} / y_{ijt}!$  and  $\lambda_{ijt} = \exp\{\alpha + u_{it}^T u_{jt}\}$ , similarly to the approach in Sewell and Chen (2016). As above, we may also consider a Euclidean formulation in which  $\lambda_{ijt} = \exp\{\alpha - \|u_{it} - u_{jt}\|\}$ . The estimated rates for a subset of the data are shown in Figure 3.6.1b and the average absolute error is given in Figure 3.6.2c. Overall, we see that the model fits the data well and performs reasonably in terms of predictive inference. By analysing the data in this way, we are able to obtain a finer scaled understanding of the interactions.



(a) Connection probabilities for binary data.



(b) Connection rates for integer data.

Figure 3.6.1: Fitted model for a selection of pairs of students. In both plots the blue line represents the mean, the 2.5<sup>th</sup> and 97.5<sup>th</sup> percentile are shown in purple, and the points correspond to the observations. The top plot shows the connection probabilities for binary data and the bottom plot shows the rate for count data.

### 3.7 Discussion

In this chapter we have considered estimation of DLSNs via SMC and, since standard SMC algorithms do not perform well as the dimension of the state space increases, we have explored high-dimensional SMC in this context. Although there is a large literature on this topic, many proposed algorithms are not appropriate for our application of interest. This is typically due to the dependence in the likelihood, since many high-dimensional SMC algorithms aim to take advantage of regions of independence, or approximate independence, within the state space (for example see Rebeschini and van Handel (2015)). SMC methods have so far not been explored for temporally evolving networks in the literature, however Bloem-Reddy and Orbanz (2018) rely on

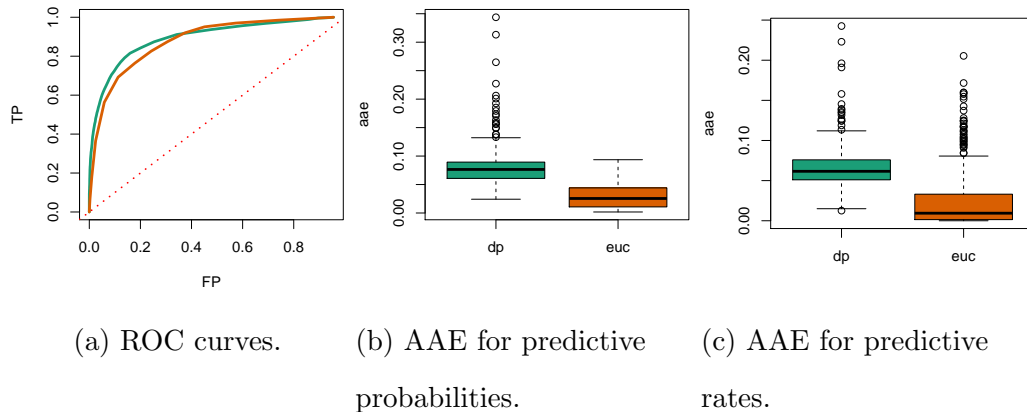


Figure 3.6.2: Left: ROC curve for DLSN model fitted with a dot-product (teal) and Euclidean distance (orange) formulation. The line  $y = x$  is shown in red. Middle: average absolute error (AAE) for binary data simulated from predictive probabilities at time  $T$ . The colours correspond to those in the left plot. Right: average absolute error (AAE) for integer data simulated from predictive rates at time  $T$ .

SMC for estimation of static networks which evolve through the addition of nodes. This differs from our application since we focus on relationships between a fixed group of individuals that are observed over time.

The algorithms considered in Section 3.4 can be adapted to variants of the DLSN model given in Section 3.3.2, and we note that this is not the case for all applicable SMC algorithms. Although a clear connection between the nudging approach of Akyildiz and Míguez (2019) and the GIRF of Park and Ionides (2019) can be drawn, since they both aim to improve performance by shifting particles to regions of the state space with high likelihood, we find that the GIRF has a much more stable performance in our application. Furthermore, the GIRF can be viewed as a generalisation of the SIR filter and so it is reasonable to use this approach with  $S = 1$  when the network of interest is sufficiently small as suggested by Section 3.4.1. It is also interesting to note that, although the context differs, Bloem-Reddy and Orbanz (2018) rely on similar SMC methodology. A range of modifications to the algorithms investigated

in Section 3.5 can also be explored and, as an example, we may incorporate nudging steps into the GIRF. Whatever the modification, the computational cost must also be taken into account.

In Section 3.4.1, we found that the independent approximation performed well in terms of ESS and MSE in probability. However, it is important to stress that this only considered state estimation and, when  $\theta$  is also estimated, we find that the variance in the ‘similarities’ is overestimated. This is a consequence of the approximation, since the likelihood for each latent similarity no longer accounts for the effect of neighbours of each node involved. Adapting this approximation presents an interesting direction for further work, and this approach may also be combined with other SMC methodology such as that of Fasano et al. (2019).

Since the increased computational cost when  $N$  increases is partly due to the additional terms in the likelihood, it is common to rely on likelihood approximations when the number of nodes becomes too large (see Raftery et al. (2012) and Rastelli et al. (2018)). It would be interesting to explore this within the context of the SMC algorithms considered in this chapter, since a direct application of these approximations may not be appropriate. An alternative direction to improve scalability may be to consider a model similar to Fosdick et al. (2019) in which nodes are partitioned into communities and the connection probabilities within each community are modelled via a latent space. This will reduce the computational cost associated with the likelihood.

Finally, this work may also be extended to changepoint and anomaly detection in which the inference task is to determine the point at which the generative mechanism of the data changes and spurious observations, respectively. It is interesting to consider what constitutes a change in the context of network data, and how the framework presented here can be utilised. The latent space approach was recently considered in the problem of anomaly detection in Lee et al. (2019), however the authors rely on approximation inference via variational methods.

# Chapter 4

## Latent Space Modelling of Hypergraph Data

### 4.1 Introduction

In this chapter 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 (Erdős and Rényi (1959), Barabási and Albert (1999)), and latent space network models (Hoff et al. (2002)). However, representing a hypergraph as a graph results in a loss of information (see Figure 4.1.1)

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 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 4.1.1a shows a possible hypergraph relationship between three authors, where a shaded region indicates which authors contributed to an article. Whilst a range of inferential 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. Once we have fitted a model to our data, this question translates into a prediction problem. The model we introduce represents the nodes of the hypergraph in a latent space, and this allows us to explore predictive distributions by simulating new nodes in the latent space and

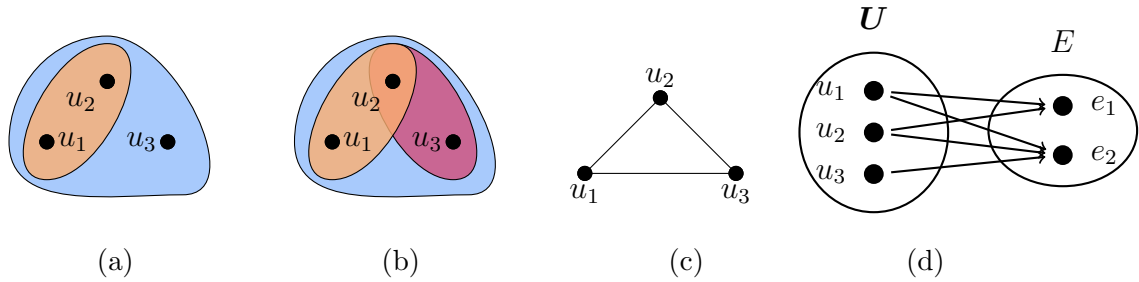


Figure 4.1.1: Figures 4.1.1a and 4.1.1b depict two possible hypergraphs, where a node belongs to a hyperedge if it lies within the associated shaded region. Figure 4.1.1c is the graph obtained by replacing hyperedges in Figure 4.1.1a, or equivalently in Figure 4.1.1b, by cliques. The hypergraphs in 4.1.1a and 4.1.1b cannot be recovered from 4.1.1c. Figure 4.1.1d presents the hypergraph relationships in Figure 4.1.1a as a bipartite graph, where an edge from a population node to a hyperedge node indicates membership of a hyperedge.

examining their connections.

(Q2) “Which authors have a greater importance in the hypergraph?”. By associating each node with a latent coordinate, we are able to determine a visualisation of the hypergraph from our model. In latent space models, it is typical for the nodes with greater degree to be placed more centrally in the latent representation. Hence, we can determine the importance of a node by examining its latent coordinate with respect to the full latent representation. This point will be discussed further in Section 4.8.

To address the inferential questions (Q1) and (Q2) we develop a model for hypergraph data that builds upon the latent space framework for graphs introduced in Hoff et al. (2002). 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. (2018)) and, although properties of these models are well understood (see Rastelli et al. (2016)), their 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, this suggests transitive relationships in which “friends of friends are likely to be friends” are likely to occur. We wish to take advantage of properties analogous to this in the hypergraph setting, and this leads us to consider (Q3). In general, it is unclear how to impose properties on a hypergraph when a bipartite representation is used (see Figure 4.1.1d). Hence, when developing our model, we rely on the representation shown in Figure 4.1.1a.

(Q3) “*How do we formulate a latent space model for hypergraphs?*”. Extending the distance model of Hoff et al. (2002) to the hypergraph setting is non-trivial and we wish to develop a model which uses constraints implied by the latent space to impose properties on the hypergraphs. For instance, recall the motivating example of coauthorship and consider the hypergraph depicted in Figure 4.1.1b. In this example, it is intuitive that the hyperedge  $\{u_1, u_2, u_3\}$  is likely to occur given the presence of  $\{u_1, u_2\}$  and  $\{u_2, u_3\}$ . We wish to develop a model which expresses this type of ‘higher-order transitivity’, where a set of authors to be more likely to collaborate if a subset of them have already written a paper together. Note that this notion of transitivity differs from other definitions of hypergraph transitivity in the literature (for example, see Mansilla and Serra (2008)).

We now review the existing literature on hypergraph analysis, and we begin with work related to community detection. Several authors have considered this problem for hypergraphs (see Chien et al. (2018), Lin et al. (2017), Kim et al. (2018), and Ghoshdastidar and Dukkipati (2014)) where the interest is in determining the community membership of each node. To facilitate this, Ghoshdastidar and Dukkipati (2014) have extended the stochastic blockmodel (SBM) of Holland et al. (1983) for graphs to the  $k$ -uniform SBM in which all hyperedges are assumed to be of the same size  $k$ . Alternatively, Zhou et al. (2006) have considered spectral clustering in the

hypergraph setting and Ke et al. (2019) have developed a tensor decomposition based approach to determine community membership. Related to these works is the approach of Ng and Murphy (2018) who develop a model to capture clustering in the hyperedges by extending methodology from latent class analysis (see Lazarsfeld and Henry (1968), Goodman (1974)). Note that this differs to community detection since the focus is on clustering structures in the hyperedges, not the nodes. Furthermore, in the bipartite setting, Aksoy et al. (2016) consider a model for bipartite graphs which exhibit community structure.

Link prediction for hypergraphs has also been explored in the literature by Benson et al. (2018) and Sharma et al. (2014). These works consider predicting future hyperedge connections given a sequence of time-indexed hyperedges. Benson et al. (2018) focus on prediction of simplicial closure events in which a set of nodes appear within the same hyperedge. Their work does not rely on a formal statistical model, but instead aims to identify which features of the hypergraph are indicative of a simplicial closure event. Alternatively, Sharma et al. (2014) consider predicting reoccurrence of previously observed hyperedges and they refer to this as ‘old edge’ prediction. They develop a tensor based approach to address this problem. We note here that the prediction task we consider in (Q1) differs from these works since our focus is on predicting connections for new nodes, and not future connections between the observed set of nodes.

Other authors have developed models for hypergraphs by considering the extension of existing graph models. For example, Stasi et al. (2014) and Liu et al. (2013) have developed models which allow control over the degree distribution by extending the  $\beta$ -model of Holland and Leinhardt (1981) and the preferential attachment model of Barabási and Albert (1999), respectively. In the model of Stasi et al. (2014) each node is assigned a parameter which controls its tendency to form connections so that a hypergraph with certain degree distribution can be expressed. Alternatively, the model of Liu et al. (2013) describes a generative process in which the hypergraph

is grown from a seed hypergraph. This allows control over the degree distribution through the mechanism in which new nodes are added to the hypergraph. Additionally, whilst the latent space framework has not been considered for the representation in Figure 4.1.1a, Friel et al. (2016) have developed a latent space model for temporally evolving bipartite graphs. In this work the authors examine company directors and boards they associate with.

There has also been a recent interest in edge-exchangeable graph models which, unlike node-exchangeable graph models, are able to express sparse graphs (see Caron and Fox (2017), Dempsey et al. (2019), Crane and Dempsey (2018), Campbell et al. (2018) and Cai et al. (2016)). Many of these models are able to express bipartite graphs, which can be used to represent a hypergraph (see Figure 4.1.1d). For example, Caron and Fox (2017) present an edge-exchangeable model for network data and, in Section 3.4, they consider how their framework may be applied to bipartite networks. Related to this is the approach of Dempsey et al. (2019) who model structured interaction processes with an edge-exchangeable framework. This work considers interactions between sets of nodes, which includes hypergraphs.

Finally, in the probability literature, several authors have studied properties of random hypergraphs (see Cooley et al. (2018), Élie de Panafieu (2015) and Karoński and Łuczak (2002)) including phase transitions. A particular class of hypergraphs, known as simplicial hypergraphs, are considered in Kahle (2017). In a simplicial hypergraph the presence of a hyperedge indicates the presence of all subsets of the hyperedge. Simplicial 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, they are used within topological data analysis (see Salnikov et al. (2018)) and to determine distances between distributions (see Pronzato et al. (2018) and Pronzato et al. (2019)). Additionally, they are used in Lunagómez et al. (2017) to propose simplicial priors for graphical models. Although our work is related to Lunagómez et al. (2017), we focus on modelling non-simplicial hypergraph data.

The main contributions of this chapter are as follows. First, using the representation shown in Figure 4.1.1a, we develop a latent space model for non-simplicial hypergraph data by extending the distance model of Hoff et al. (2002) to the hypergraph setting. We present a specific instance of our model, and comment that alternative modelling choices can be explored in future work. Second, our model avoids the computationally expensive full conditional implied by the construction of Hoff et al. (2002) by relying on tools from computational geometry (see Edelsbrunner and Harer (2010)). Furthermore, by representing the nodes of the hypergraph in a low-dimensional latent space, we develop a parsimonious model that is able to express complex data structures. Third, we draw upon the previously exploited connection between latent space network models and shape theory to remove non-identifiability present in our model. More specifically, we infer the latent representation on the space of Bookstein coordinates which have so far not been explored in this context. Fourth, by simulating new nodes from the latent representation, we demonstrate how our model facilitates exploration of the predictive distributions. We also comment that the latent representation provides a convenient visualisation of the hypergraph in which more centrally placed nodes have a larger degree. Fifth, we investigate the theoretical properties of our model and, in particular, we present a framework for examining the degree distribution. Whilst this proves challenging for our model, our discussion provides an outline which can be explored for other modelling choices.

The rest of this chapter is organised as follows. In Section 4.2 we provide the necessary background to introduce our hypergraph model in Section 4.3. Then, in Section 4.4 we discuss identifiability of our hypergraph model and, in Section 4.5, we describe our procedure for obtaining posterior samples. The simulation studies and real data examples are presented in Section 4.7 and 4.8, respectively. Finally, we conclude with a discussion in Section 4.9.

## 4.2 Background

In this section we will review the latent space network modelling literature that is relevant to the model for hypergraph data we introduce in Section 4.3. First, in Section 4.2.1, 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 4.2.2, 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 4.2.3 we demonstrate how RGGs can be extended from the pairwise case to model a restricted class of hypergraphs.

### 4.2.1 Latent Space Network Modelling

Latent space models were introduced for network data in Hoff et al. (2002) and, since their introduction, have given rise to a rich modelling literature. 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. Furthermore, conditional on the iid latent coordinates, the edge between a given pair of nodes is modelled as independent of all other edges. The dependence in the network is captured by the latent representation, and this can be made clear through marginalising over the latent space.

We will first describe a general latent space modelling framework for a network with  $N$  nodes. Let  $\mathbf{Y} = \{y_{ij}\}_{i,j=1,2,\dots,N}$  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  $i^{\text{th}}$  node, for

$i = 1, 2, \dots, N$ . The presence of an edge is then given by the following model.

$$\begin{aligned}
 Y_{ij} &\sim \text{Bernoulli}(p_{ij}), \\
 p_{ij} &= P(y_{ij} = 1 | u_i, u_j, \theta) = \frac{1}{1 + \exp\{-f(u_i, u_j, \theta)\}},
 \end{aligned} \tag{4.2.1}$$

where  $\theta$  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

$$f(u_i, u_j, \theta) = \alpha - \|u_i - u_j\|, \tag{4.2.2}$$

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

We note that the choice of similarity measure will impose characteristics on graphs generated under this model. If the similarity measure is chosen to be a metric, for example the Euclidean distance, we know that it satisfies the triangle inequality. If connections exist between the pairs  $\{i, j\}$  and  $\{i, k\}$ , we know that their latent coordinates are close in terms of the similarity measure. The triangle inequality suggests that the node pair  $\{i, k\}$  is also likely to be connected, and so transitive relationships are likely.

Both asymmetric and symmetric adjacency matrices can be represented in this framework. Suppose the connections are symmetric and that there are no self ties, so that we have  $y_{ij} = y_{ji}$  and  $y_{ii} = 0$  for  $i, j = 1, 2, \dots, N$ . In this case, the likelihood, conditional on  $\mathbf{U}$  and  $\theta$ , is given by

$$\mathcal{L}(\mathbf{U}, \theta; \mathbf{Y}) \propto \prod_{i < j} P(y_{ij} = 1 | u_i, u_j, \theta)^{y_{ij}} [1 - P(y_{ij} = 1 | u_i, u_j, \theta)]^{1 - y_{ij}}, \tag{4.2.3}$$

where  $\mathbf{U}$  is the  $(N \times d)$  matrix of latent coordinates such that the  $i^{\text{th}}$  row of  $\mathbf{U}$  corresponds to  $u_i$ .

The model specified in (4.2.1) and (4.2.3) can be modified in many different ways. For example, we can model the connection probabilities using the probit link function instead of the logit link function. Properties of modifications of this type are discussed in Rastelli et al. (2016). We may also model non-binary connections such as integer weighted edges. For example, a Poisson likelihood allows us to model edges which represent the number of interactions between a pair of nodes. Note that this will require specification of a rate parameter which has a different interpretation to  $p_{ij}$ .

## 4.2.2 Random Geometric Graphs

In Section 4.2.1 we outlined the latent space modelling approach for network data. This framework specifies the probability of an edge forming between a pair of nodes as a function of their latent coordinates. 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. RGGs can be viewed as a special case of the latent space framework where, conditional on the latent representation, there is no uncertainty in the connections. For an in-depth discussion of RGGs see Penrose (2003).

As in the previous section, we will assume that the  $i^{\text{th}}$  node can be represented by  $u_i = (u_{i1}, u_{i2}, \dots, u_{id}) \in \mathbb{R}^d$ . 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 an RGG we choose the closed ball in  $\mathbb{R}^d$  with centre  $u_i$  and radius  $r$ . This set is represented  $B_r(u_i) = \{u \in \mathbb{R}^d \mid \|u - u_i\| \leq r\} = \left\{u \in \mathbb{R}^d \mid \sqrt{\sum_{j=1}^d (u_j - u_{ij})^2} \leq r\right\}$ , and a graph is constructed by connecting each pair of nodes  $\{i, j\}$  for which  $B_r(u_i) \cap B_r(u_j) \neq \emptyset$ . Generating a graph in this way is equivalent to connecting pairs of nodes for which  $\|u_i - u_j\| \leq 2r$ . An example of this construction is given in the left and middle panel of Figure 4.2.1.

We now express the likelihood for this model as a function of the latent coordinates, keeping the notation from Section 4.2.1. The likelihood conditional on  $\mathbf{U}$  and  $r$  is

given by

$$\mathcal{L}(\mathbf{U}, r; \mathbf{Y}) \propto \prod_{i < j} \mathbb{1}(\|u_i - u_j\| \leq 2r)^{y_{ij}} [1 - \mathbb{1}(\|u_i - u_j\| \leq 2r)]^{1-y_{ij}}. \quad (4.2.4)$$

By comparing (4.2.3) and (4.2.4), we see that a RGG can be viewed as a latent space network model where the probability of a connection is expressed as a step function. More specifically, we have  $P(y_{ij} = 1 | u_i, u_j, \theta) = \mathbb{1}(\|u_i - u_j\| \leq 2r)$  where  $\theta = r$ . It is clear from this that, conditional on a set of latent coordinates, a RGG is deterministic. Note that (4.2.4) is equal to 1 if there is a perfect correspondence between the observed connections  $\mathbf{Y}$  and the connections induced by the latent coordinates  $\mathbf{U}$  and the radius  $r$ . If there are any connections which do not correspond to each other, then (4.2.4) is equal to 0.

To specify a more general construction, we define  $A_i$  to be a convex set for which  $u_i \in A_i$ . In the construction above, we let  $A_i = B_r(u_i)$ , for  $i = 1, 2, \dots, N$ . This choice of convex set is convenient since, given the radius  $r$  and coordinates  $\mathbf{U}$ , we are able to generate the graph by considering the distance between pairs of latent coordinates. 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. An alternative choice for  $A_i$  that is common in the literature is the Voronoi cell, where  $A_i = \{x \in \mathbb{R}^d | \|x - u_i\| \leq \|x - u_j\|, \forall u_j \text{ st } i \neq j\}$  (see Section 3.3 of Edelsbrunner and Harer (2010)).

### 4.2.3 Random Geometric Hypergraphs

The graph generating procedure described in Section 4.2.2 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)). This represents the set of indices for which their corresponding convex regions have a



non-empty intersection and it is given in Definition 4.2.1.

**Definition 4.2.1.** (*Nerve*) Let  $\mathcal{A} = \{A_i\}_{i=1}^N$  represent a collection of non-empty convex sets. The nerve of  $\mathcal{A}$  is given by

$$Nrv(\mathcal{A}) = \left\{ \sigma \subseteq \{1, 2, \dots, N\} \mid \bigcap_{j \in \sigma} A_j \neq \emptyset \right\}. \quad (4.2.5)$$

Note that the sets  $\{1\}, \{2\}, \dots, \{N\}$  are included in  $Nrv(\mathcal{A})$  and that  $|\sigma| \leq N$  for  $\sigma \in Nrv(\mathcal{A})$ , where  $|\sigma|$  is the order, or dimension, of the set.

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. Kahle (2017) overview properties of simplicial random hypergraphs along with more general constructions.

In Section 4.2.2, 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 4.2.2.

**Definition 4.2.2.** (*Čech Complex*) For a set of coordinates  $\mathbf{U} = \{u_i\}_{i=1}^N$  and a radius  $r$ , the Čech complex  $\mathcal{C}_r$  is given by

$$\mathcal{C}_r = Nrv(\{B_r(u_i)\}_{i=1}^N). \quad (4.2.6)$$

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)) where  $A_i$  is specified as the Voronoi cell for  $u_i$  (see Section 4.2.2) as the intersection of the Voronoi cell for  $u_i$  and  $B_r(u_i)$ , respectively.

We now introduce a subset of the Čech complex known as the  $k$ -skeleton, which is given in Definition 4.2.3. This will be revisited in Section 4.3.2.

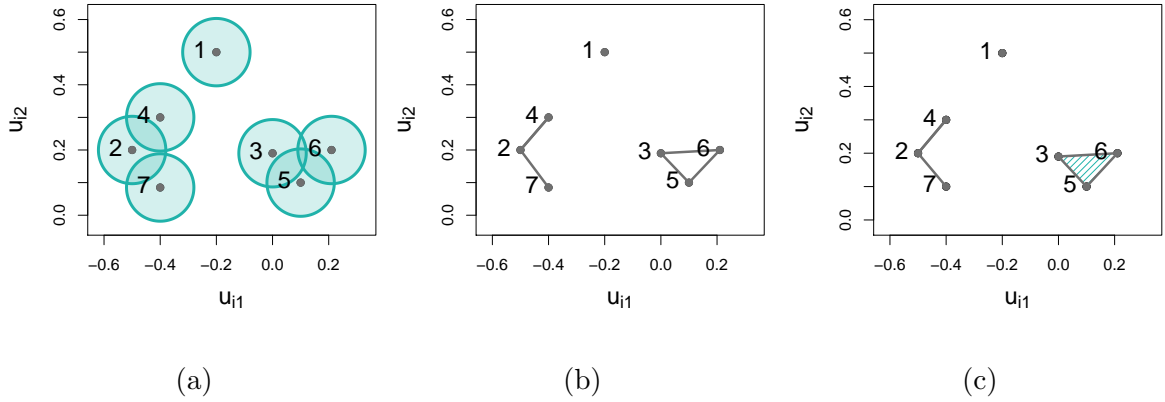


Figure 4.2.1: Example of a Čech complex. Left:  $B_r(u_i)$  for  $\{u_i = (u_{i1}, u_{i2})\}_{i=1}^7$  in  $\mathbb{R}^2$ . Middle: the graph obtained by taking pairwise intersections. Right: the hypergraph obtained by taking intersections of arbitrary order. The shaded region between nodes 3, 5 and 6 indicates a hyperedge of order 3.

**Definition 4.2.3.** (*k-skeleton of the Čech complex*) Let  $\mathcal{C}_r$  denote the Čech complex, as given in Definition 4.2.2. The  $k$ -skeleton of  $\mathcal{C}_r$  is given by

$$\mathcal{C}_r^{(k)} = \{\sigma \in \mathcal{C}_r \mid |\sigma| \leq k\}. \quad (4.2.7)$$

$\mathcal{C}_r^{(k)}$  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 4.2.1:** Figure 4.2.1 depicts an example of a Čech complex, where  $\mathcal{C}_r = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}, \{2, 4\}, \{2, 7\}, \{3, 5\}, \{3, 6\}, \{5, 6\}, \{3, 5, 6\}\}$ . The  $k$ -skeletons are given by  $\mathcal{C}_r^{(1)} = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}\}$ ,  $\mathcal{C}_r^{(2)} = \mathcal{C}_r^{(1)} \cup \{\{2, 4\}, \{2, 7\}, \{3, 5\}, \{3, 6\}, \{5, 6\}\}$  and  $\mathcal{C}_r^{(3)} = \mathcal{C}_r^{(2)} \cup \{3, 5, 6\}$ .

### 4.3 Latent space hypergraphs

In this section we will introduce a model for hypergraph data which builds upon the models discussed in Section 4.2. Our model will balance the computational aspects of

latent space network modelling (Section 4.2.1) with the approach of RGGs (Section 4.2.2) and RGHs (Section 4.2.3). The aims of our modelling framework are given in Section 4.3.1, notation and set-up are given in Section 4.3.2, and the generative model and likelihood are given in Section 4.3.3. Finally, in Section 4.3.4 we extend the model to a more flexible scenario.

### 4.3.1 Motivation

Consider a co-authorship network where nodes represent authors and edges indicate which authors contributed to a given paper. In this context, papers that have been written by more than two authors are naturally represented by a hyperedge. The hypergraph model discussed in Section 4.2.3 is likely not appropriate for these data since a set of authors having worked on a paper does not imply that all subsets of those authors have also written papers together. This motivates us to develop a model for non-simplicial hypergraphs.

In this section we will build upon the pairwise graph and hypergraph models introduced in Section 4.2. We introduce a model for hypergraph data which represents the nodes of the network in a low-dimensional space and, unlike the approach in Section 4.2.3, is able to express a broad class of hypergraphs. The model will be developed with the following objectives.

#### 1. Convenient likelihood

The model of Hoff et al. (2002) specifies the connection probabilities for all node pairs and so the likelihood, conditional on the latent coordinates, for this model is a function of a 2D tensor. In the hypergraph analogue of this framework, the conditional likelihood would be a function of a  $k$ D tensor, where  $k$  is the order of the hyperedges. Evaluation over a  $k$ D tensor is an order  $O(N^k)$  computation, which becomes increasingly computationally expensive as the number of nodes  $N$  and the hyperedge order  $k$  grow. In contrast to this, graphs and hypergraphs generated in the RGG (Section 4.2.2) and RGH (Section 4.2.3) framework are

a deterministic function of the latent space. RGHS can be computed efficiently without considering all possible hyperedges, however the conditional likelihood is equal to either 0 or 1 and this may hinder model fitting. Our aim is to develop a model which draws on the advantages of each of these approaches. We present a model with a likelihood that is amenable to Bayesian computation and whose evaluation does not rely on a calculation over a tensor.

## 2. Simple to describe support

Since the edges in the model of Hoff et al. (2002) exhibit uncertainty after conditioning on the latent coordinates, it is clear that the model can express the entire space of graphs on  $N$  nodes with pairwise interactions. In the RGG framework (Section 4.2.2), connections are instead modelled deterministically through the intersection of convex sets. It is not clear in general how to characterise the space of graphs that represent the support for this generative model. Furthermore, when this framework is extended to model non-simplicial hypergraphs, the support for the generative model is complicated further. Based on the approach of RGHS (Section 4.2.3), we develop a model for non-simplicial hypergraph data whose support is straightforward to describe.

Throughout this section we will describe our model for a hypergraph on  $N$  nodes with a maximal hyperedge order  $K$ , where  $2 \leq K \leq N$ . We let  $e \subseteq \{1, 2, \dots, N\}$  denote a hyperedge and the presence of  $e$  is denoted by  $y_e = 1$ . Otherwise, we let  $y_e = 0$ . Where necessary, we will denote a hyperedge of order  $k$  by  $e_k$  so that  $|e_k| = k$ .

### 4.3.2 Combining $k$ -skeletons

To extend the model in Section 4.2.3 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  $\mathbf{U}$ , 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

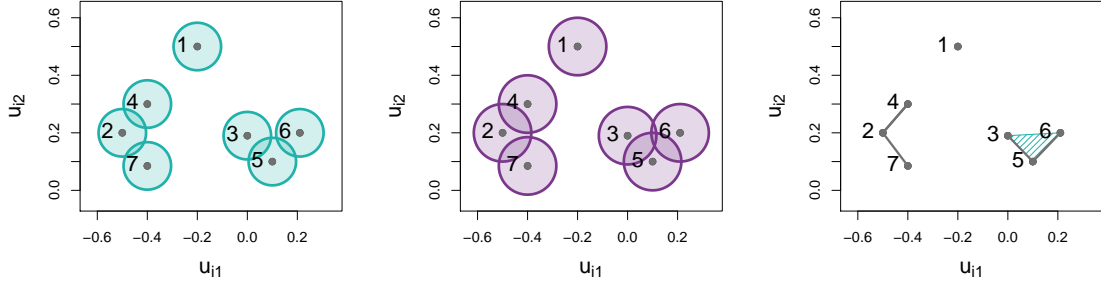


Figure 4.3.1: Example of a Čech complex. Left:  $B_{r_2}(u)$  for each of 6 points in  $\mathbb{R}^2$ . Middle:  $B_{r_3}(u)$  for each of 6 points in  $\mathbb{R}^2$ . Right:  $\cup_{k=2}^3 \mathcal{D}_{r_k}^{(k)}$ .

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 4.3.1. We refer to hypergraphs constructed in this way as non-simplicial random geometric hypergraphs (nsRGH), and Definition 4.3.1 details this construction for a hypergraph on  $N$  nodes with maximal hyperedge order  $K$ .

**Definition 4.3.1.** (*non-simplicial RGH*) Consider  $\mathbf{r} = (r_2, r_3, \dots, r_K)$  which satisfy  $r_k > r_{k-1}$  for  $k = 3, 4, \dots, K$ . We define the non-simplicial RGH (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 4.2.3.

**Example 4.3.1:** Consider the non-simplicial hypergraph shown in the right panel of Figure 4.3.1. We have  $\mathcal{D}_{r_2}^{(2)} = \{\{2, 4\}, \{3, 5\}, \{5, 6\}\}$  and  $\mathcal{D}_{r_3}^{(3)} = \{3, 5, 6\}$ .

In Definition 4.3.1, constraints are imposed on the radii  $\mathbf{r} = (r_2, r_3, \dots, r_K)$  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 comment here that the Čech complex is also used in Lunagómez et al. (2017) to propose simplicial priors for graphical models, however our setting differs from this since we focus on modelling non-simplicial hypergraph data.

### 4.3.3 Generative Model and Likelihood

The procedure of generating non-simplicial hypergraphs given in Definition 4.3.1 is deterministic conditional on  $\mathbf{U}$ . Therefore, similarly to the RGG in Section 4.2.2, 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 4.3.1.

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  $\mathcal{V}_N = \{1, 2, \dots, N\}$  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  $k = 2, 3, \dots, K$ . Then, for  $e_k \in \mathcal{E}_{N,k}$ , we sample a variable  $S_k \sim \text{Bernoulli}(\varphi_k)$  and let

$$y_{e_k}^{(g^*)} = \begin{cases} 0, & \text{if } s_k = 1 \text{ and } y_{e_k}^{(g)} = 1 \\ 1, & \text{if } s_k = 1 \text{ and } y_{e_k}^{(g)} = 0 \\ y_{e_k}, & \text{if } s_k = 0 \end{cases} \quad (4.3.1)$$

where  $g_{N,K}(\mathbf{U}, \mathbf{r})$  denotes the nsRGH induced from  $\mathbf{U}$  and  $\mathbf{r}$  as in Definition 4.3.1, and  $y_{e_k}^{(g)}$  denotes the state of the hyperedge  $e_k$  in  $g_{N,K}(\mathbf{U}, \mathbf{r})$ . We let  $g_{N,K}^*(\mathbf{U}, \mathbf{r})$  denote the hypergraph sampled from our model. From (4.3.1), we see that  $\varphi_k$  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

---

**Algorithm 7** Sample a hypergraph  $g_{N,K}^*$  given  $N, K, \mathbf{r}, \boldsymbol{\varphi}, \mu$  and  $\Sigma$ .

---

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

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

a) Given  $\mathbf{U}$  and  $r_k$ , check which  $e_k = \{i_1, i_2, \dots, i_k\} \in \mathcal{E}_{N,k}$  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  $S_k \sim \text{Bernoulli}(\boldsymbol{\varphi}_k)$ .

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

---

the latent coordinates  $\mathbf{U}$ , and we assume that  $u_i \stackrel{iid}{\sim} \mathcal{N}(\mu, \Sigma)$ , for  $i = 1, 2, \dots, N$ . 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 7.

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

$$d_k(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K}) = \sum_{e_k \in \mathcal{E}_{N,k}} |y_{e_k}^{(g)} - y_{e_k}^{(h)}| \quad (4.3.2)$$

denote the distance between the order  $k$  hyperedges in  $g_{N,K}(\mathbf{U}, \mathbf{r})$  and  $h_{N,K}$ , where  $y_{e_k}^{(g)}$  and  $y_{e_k}^{(h)}$  represent an order  $k$  hyperedge in  $g_{N,K}(\mathbf{U}, \mathbf{r})$  and  $h_{N,K}$ , respectively. Note that no modifications have been made to the hypergraph  $g_{N,K}(\mathbf{U}, \mathbf{r})$ . The measure of distance (4.3.2) corresponds to the number of hyperedges which differ between  $h_{N,K}$  and  $g_{N,K}(\mathbf{U}, \mathbf{r})$ . This is equivalent to the Hamming distance and is 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 (4.3.2), and we can instead evaluate the discrepancy by only considering hyperedges that are present in  $g_{N,K}(\mathbf{U}, \mathbf{r})$  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 B.5.2.

Given this notion of hypergraph distance the likelihood of observing  $h_{N,K}$ , condi-

tional on  $\mathbf{U}$ ,  $\mathbf{r}$  and  $\boldsymbol{\varphi}$ , can be written as

$$\mathcal{L}(\mathbf{U}, \mathbf{r}, \boldsymbol{\varphi}; h_{N,K}) \propto \prod_{k=2}^K \varphi_k^{d_k(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K})} (1 - \varphi_k)^{\binom{N}{k} - d_k(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K})}. \quad (4.3.3)$$

We obtain (4.3.3) by considering which hyperedges in  $g_{N,K}(\mathbf{U}, \mathbf{r})$  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  $\varphi_k$ . 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 (4.3.3) behaves in an intuitive way.

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

$$\mu \sim \mathcal{N}(m_\mu, \Sigma_\mu) \quad \Sigma_\mu \sim \mathcal{W}^{-1}(\Phi, \nu), \quad r_k \sim \exp(\lambda_k), \quad \text{and} \quad \varphi_k \sim \text{Beta}(a_k, b_k), \quad (4.3.4)$$

where the priors in (4.3.4) are chosen for computational convenience.

### 4.3.4 Extensions

In Algorithm 7 the number of order  $k$  edges is controlled by varying the parameter  $r_k$ . However, the constraint  $r_{k+1} > r_k$ , for  $k = 2, 4, \dots, K - 1$ , 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 7 the noise  $\varphi_k$  is applied independently across all hyperedges of order  $k$ . Alternatively, we can modify each hyperedge depending on its state in  $g_{N,K}(\mathbf{U}, \mathbf{r})$ . For  $k = 2, 3, \dots, K$ , let  $\boldsymbol{\psi}^{(0)} = (\psi_2^{(0)}, \psi_3^{(0)}, \dots, \psi_K^{(0)}) \in [0, 1]$  denote the probability of modifying the state of a hyperedge in  $g_{N,K}(\mathbf{U}, \mathbf{r})$  from absent to present, and let  $\boldsymbol{\psi}^{(1)} = (\psi_2^{(1)}, \psi_3^{(1)}, \dots, \psi_K^{(1)}) \in [0, 1]$  denote the probability of modifying the state of a hyperedge in  $g_{N,K}(\mathbf{U}, \mathbf{r})$  from present to absent. Suppose our observed hypergraph suggests that there are many hyperedges of order 2



---

**Algorithm 8** Sample a hypergraph  $g_{N,K}^*$  given  $N, K, \mathbf{r}, \boldsymbol{\psi}^{(0)}, \boldsymbol{\psi}^{(1)}, \mu$  and  $\Sigma$ .

---

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

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

a) Given  $\mathbf{U}$  and  $r_k$ , check which  $e_k = \{i_1, i_2, \dots, i_k\} \in \mathcal{E}_{N,k}$  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  $\psi_k^{(0)}$ .

---

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 8, and as commented in Section 4.3.3, Algorithm 8 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  $g_{N,K}(\mathbf{U}, \mathbf{r})$  are discussed in Appendix B.5.1 with details of the hyperedge modifications given in Appendix B.2.

As in Section 4.3.3, the likelihood of observing  $h_{N,K}$  is based on a distance metric,

$$d_k^{(ab)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K}) = \#\{e_k \in \mathcal{E}_{N,k} | y_{e_k}^{(g)} = a \cap y_{e_k}^{(h)} = b\}, \quad (4.3.5)$$

which records the number of hyperedges that have state  $a \in \{0, 1\}$  in  $g_{N,K}(\mathbf{U}, \mathbf{r})$  and state  $b \in \{0, 1\}$  in  $h_{N,K}$ . For example,  $d_k^{(01)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K})$  represents the number of hyperedges absent in  $g_{N,K}(\mathbf{U}, \mathbf{r})$  and present in  $h_{N,K}$ . Efficient evaluation of (4.3.5) is discussed in Appendix (B.5.2) and the likelihood conditional on  $\mathbf{U}, \mathbf{r}, \boldsymbol{\psi}^{(1)}$  and  $\boldsymbol{\psi}^{(0)}$  is given by

$$\begin{aligned} \mathcal{L}(\mathbf{U}, \mathbf{r}, \boldsymbol{\psi}^{(1)}, \boldsymbol{\psi}^{(0)}; h_{N,K}) &\propto \prod_{k=2}^K \left[ \left( \psi_k^{(1)} \right)^{d_k^{(10)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K})} \left( 1 - \psi_k^{(1)} \right)^{d_k^{(11)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K})} \right. \\ &\quad \left. \times \left( \psi_k^{(0)} \right)^{d_k^{(01)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K})} \left( 1 - \psi_k^{(0)} \right)^{d_k^{(00)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K})} \right]. \end{aligned} \quad (4.3.6)$$

We obtain (4.3.6) in a similar way to (4.3.3), where we distinguish between hyperedges that are present and absent in the induced hypergraph. Note that (4.3.6) is equivalent to (4.3.3) when  $\psi_k^{(1)} = \psi_k^{(0)}$ , for  $k = 2, 3, \dots, K$ .

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

$$\begin{aligned} \mu &\sim \mathcal{N}(m_\mu, \Sigma_\mu), \quad \Sigma \sim \mathcal{W}^{-1}(\Phi, \nu), \quad r_k \sim \exp(\lambda_k), \\ \psi_k^{(0)} &\sim \text{Beta}(a_k^{(0)}, b_k^{(0)}), \quad \text{and} \quad \psi_k^{(1)} \sim \text{Beta}(a_k^{(1)}, b_k^{(1)}). \end{aligned} \quad (4.3.7)$$

where priors in (4.3.7) are chosen for computational convenience.

## 4.4 Identifiability

Let  $g_{N,K}(\mathbf{U}, \mathbf{r})$  denote the nsRGH obtained from  $\mathbf{U}$  and  $\mathbf{r}$ , and let  $g_{N,K}^*$  be the hypergraph obtained by modifying the hyperedges in  $g_{N,K}(\mathbf{U}, \mathbf{r})$  according to  $\boldsymbol{\varphi}$  (see Algorithm 7). By conditioning on  $g_{N,K}(\mathbf{U}, \mathbf{r})$ , we can decompose the conditional distribution for  $g_{N,K}^*$  in the following way.

$$p(g_{N,K}^*, g_{N,K}(\mathbf{U}, \mathbf{r}) | \mu, \Sigma, \boldsymbol{\varphi}, \mathbf{r}) = p(g_{N,K}^* | g_{N,K}(\mathbf{U}, \mathbf{r}), \boldsymbol{\varphi}) p(g_{N,K}(\mathbf{U}, \mathbf{r}) | \mu, \Sigma, \mathbf{r}). \quad (4.4.1)$$

An equivalent decomposition for the model outlined in Algorithm 8 can also be expressed.

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

To remove these sources of non-identifiability, we define  $\mathbf{U}$  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  $\mathbf{U}$  with respect to a set of anchor points and, since these anchor points remain fixed

throughout model fitting, the radii  $\mathbf{r}$  are also appropriately re-scaled. For details of the Bookstein transformation see Appendix B.1.

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  $\mathbf{U}$ . Due to the non-identifiability associated with scaling  $\mathbf{U}$  and  $\mathbf{r}$ , we note that this approach is not sufficient for removing all sources of non-identifiability in our model.

From (4.4.1), we see that hyperedges can either arise from  $g_{N,K}(\mathbf{U}, \mathbf{r})$  or the hyper-edge modification. To maintain the properties imposed on the hypergraph from the construction of  $g_{N,K}(\mathbf{U}, \mathbf{r})$ , we wish to keep the parameters  $\boldsymbol{\phi}$  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.

## 4.5 Posterior Sampling

To sample from the models specified in Sections 4.3.3 and 4.3.4 we implement an MCMC scheme. In Section 4.5.1 we provide a high-level description of the posterior sampling procedure for the model detailed in Algorithm 8 and comment that this can easily be modified to the model detailed in Algorithm 7. Where appropriate we refer the reader to the relevant sections of the appendix for more detail.

### 4.5.1 MCMC scheme

To obtain posterior samples from the model presented in Algorithm 8 we implement a Metropolis-Hastings-within-Gibbs MCMC scheme (see Section 6.4.2 of Gamerman and Lopes (2006)). We update the latent coordinates  $\mathbf{U}$  and radii  $\mathbf{r}$  with a Metropolis Hastings (MH) step, and the remaining parameters are updated via Gibbs samplers. The priors for the model are specified in (4.3.7).

When updating the latent coordinates we use a random walk MH. As discussed in Section 4.4, we define  $\mathbf{U}$  on the Bookstein space of coordinates and so a set of anchor points will remain fixed throughout the MCMC scheme. For  $u_i \in \mathbb{R}^d$ , let the anchor points be denoted by  $u_1$  and  $u_2$ . For  $i = 3, 4, \dots, N$ , 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  $\mathbf{U}^* = \{u_i^*\}_{i=1}^N$  as a sample from  $p(\mathbf{U}|\mu, \Sigma, \mathbf{r}, \boldsymbol{\psi}^{(0)}, \boldsymbol{\psi}^{(1)}, h_{N,K})$  with probability

$$\min \left\{ 1, \frac{\mathcal{L}(\mathbf{U}^*, \mathbf{r}, \boldsymbol{\psi}^{(1)}, \boldsymbol{\psi}^{(0)}; h_{N,K})p(\mathbf{U}^*|\mu, \Sigma)}{\mathcal{L}(\mathbf{U}, \mathbf{r}, \boldsymbol{\psi}^{(1)}, \boldsymbol{\psi}^{(0)}; h_{N,K})p(\mathbf{U}|\mu, \Sigma)} \right\}, \quad (4.5.1)$$

where  $p(\mathbf{U}|\mu, \Sigma) = \prod_{i=1}^N p(u_i|\mu, \Sigma)$ . Since the proposal mechanism is symmetric in terms of  $\mathbf{U}$  and  $\mathbf{U}^*$ , the term associated with this does not appear in (4.5.1).

The acceptance ratio (4.5.1) is for the entire  $(N \times d)$  latent representation  $\mathbf{U}$ . As  $N$  grows, jointly proposing all latent coordinates will become increasingly inefficient due to the dimension of  $\mathbf{U}$ . Alternatively we can partition the latent coordinates into disjoint sets  $\{U_l\}_{l=1}^L$  such that  $\cup_{l=1}^L U_l = \{1, 2, \dots, N\}$ , and perform the MH update for each of these  $L$  sets separately. This approach will be used in the examples in Sections 4.7 and 4.8, and details of this are given in Algorithm 9.

To update  $\mathbf{r}$ , we let  $\mathbf{r}^* = (r_2^*, r_3^*, \dots, r_K^*)$  where  $r_k^* = r_k + \epsilon_r$  and  $\epsilon_r \sim \mathcal{N}(0, \sigma_r)$ . Then, if  $r_{k+1}^* > r_k^*$  for  $k = 2, 3, \dots, K-1$ , we accept  $\mathbf{r}^*$  as a sample from  $p(\mathbf{r}|\mathbf{U}, \boldsymbol{\psi}^{(0)}, \boldsymbol{\psi}^{(1)}, \mu, \Sigma)$  with probability

$$\min \left\{ 1, \frac{\mathcal{L}(\mathbf{U}, \mathbf{r}^*, \boldsymbol{\psi}^{(1)}, \boldsymbol{\psi}^{(0)}; h_{N,K})p(\mathbf{r}^*|\boldsymbol{\lambda})}{\mathcal{L}(\mathbf{U}, \mathbf{r}, \boldsymbol{\psi}^{(1)}, \boldsymbol{\psi}^{(0)}; h_{N,K})p(\mathbf{r}|\boldsymbol{\lambda})} \right\}, \quad (4.5.2)$$

where  $p(\mathbf{r}|\boldsymbol{\lambda}) = \prod_{k=2}^K p(r_k|\lambda_k)$ . Otherwise, we reject  $\mathbf{r}^*$ .

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 9. Initialisation for the MCMC is non-trivial, and a discussion of this can be found in Appendix B.4.

To implement the MCMC scheme, there are a number of computational considerations we must address (see Appendix B.5). Firstly, to evaluate the likelihood we need to determine the hyperedges in the hypergraph generated from  $\mathbf{U}$  and  $\mathbf{r}$ ,  $g(\mathbf{U}, \mathbf{r})$ . We rely on methodology from the computational topology literature to do this, and

---

**Algorithm 9** MCMC scheme to obtain posterior samples of  $\mathbf{U}, \mathbf{r}, \boldsymbol{\psi}^{(0)}, \boldsymbol{\psi}^{(1)}, \Sigma$  and  $\mu$ .

---

Specify  $N, K, L, i_{max} \in \mathbb{N}$ .

**Initialise**

Determine initial values for  $\mathbf{U}, \mathbf{r}, \boldsymbol{\psi}^{(0)}, \boldsymbol{\psi}^{(1)}, \Sigma$  and  $\mu$  using Algorithm 12 (see Appendix B.4).

**For**  $it$  **in**  $1, 2, \dots, i_{max}$

- 1) Sample  $\mu^{(i)}$  from  $p(\mu|\mathbf{U}, \Sigma, m_\mu, \Sigma_\mu)$  (see Appendix B.3.1).
- 2) Sample  $\Sigma^{(i)}$  from  $p(\Sigma|\mathbf{U}, \mu, \Phi, \nu)$  (see Appendix B.3.2).
- 3) Partition  $\{u_3, u_4, \dots, u_N\}$  into  $L$  sets  $U_l$ .

**For**  $l = 1, 2, \dots, L$

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 (4.5.1).

- 4) **For**  $k = 2, 3, \dots, K$ , propose  $\mathbf{r}^*$ , where  $r_k^* = r_k + \epsilon_r$  and  $\epsilon_r \sim \mathcal{N}(0, \sigma_r)$

Accept proposal  $\mathbf{r}^*$  with probability (4.5.2).

- 5) **For**  $k = 2, 3, \dots, K$

Sample  $\psi_k^{(0)}$  from  $p\left(\psi_k^{(0)}|\mathbf{U}, \mathbf{r}, h_{N,K}, a_k^{(0)}, b_k^{(0)}\right)$  (see Appendix B.3.3).

Sample  $\psi_k^{(1)}$  from  $p\left(\psi_k^{(1)}|\mathbf{U}, \mathbf{r}, h_{N,K}, a_k^{(1)}, b_k^{(1)}\right)$  (see Appendix B.3.4).

---

details of the approach used are given in Appendix B.5.1. Secondly, to calculate the likelihood we note that we can avoid the summation over the entire set of hyperedges as suggested by (4.3.5) and, in Appendix B.5.2, we discuss this in more detail.

## 4.6 Theoretical Results

In this section we study the behaviour of the node degree in the hypergraph model detailed in Algorithm 7. We begin by making observations on the structure of our model in Section 4.6.1, and then consider the probability of a hyperedge occurring in a nsRGH in Section 4.6.2. Finally, we present our results for the degree distribution

in Section 4.6.3. We comment here that it is straightforward to extend the results in this section to the model detailed in Algorithm 8.

### 4.6.1 Observations

To study the properties of the node degree we first observe that the nodes in our hypergraph model are exchangeable, and so we focus on obtaining results for the  $i^{\text{th}}$  node. To guide the structure of our proofs we make the following observations.

**(O1) A hypergraph generated from our model is a modification of a nsRGH**

To generate a hypergraph from our model, we first determine hyperedges through the intersection of sets  $B_r(u_i)$  for  $i = 1, 2, \dots, N$ . The indicators for the presence and absence of hyperedges are then modified according to the noise parameters  $\boldsymbol{\varphi}$ . Since the modifications are applied independently, we can view our hypergraph model as an Erdős-Rényi modification of a nsRGH generated from  $\mathbf{U}$  and  $\mathbf{r}$ .

**(O2) Hyperedges of different orders occur independently in our model**

Conditional on the latent coordinates  $\mathbf{U}$  and radii  $\mathbf{r}$ , the hyperedges of order  $k$  occur independently of hyperedges of order  $k' \neq k$ . Therefore, we can consider the degree distribution as a sum over the degree distributions for hyperedges of exactly order  $k$ .

Throughout this section, we will assume that the number of nodes  $N$  and maximum hyperedge order  $K$  are fixed. We let  $g(\mathbf{U}, \mathbf{r}) = g \in \mathcal{G}_{N,K}$  denote the nsRGH generated from the coordinates  $\mathbf{U}$  and radii  $\mathbf{r}$ . A hypergraph is generated from our model by modifying  $g$  with noise  $\boldsymbol{\varphi}$ , and we denote this hypergraph by  $g^* \in \mathcal{G}_{N,K}$ . Additionally, we denote the degree of order  $k$  hyperedges and the overall degree of the  $i^{\text{th}}$  node in  $g$  by  $\text{Deg}_{(i,k)}^g = \sum_{\{e_k \in \mathcal{E}_{N,k} | i \in e_k\}} y_{e_k}^{(g)}$  and  $\text{Deg}_{(i)}^g = \sum_{k=2}^K \text{Deg}_{(i,k)}$ , respectively.

Using this notation, we recall the decomposition for  $h_{N,K}$  in (4.4.1) and comment that this follows from (O1). Finally, we will assume that the covariance matrix for the

latent coordinates  $\Sigma$  is diagonal so that  $\Sigma_{ll} = \sigma_l^2$  for  $l = 1, 2, \dots, d$  and  $\Sigma_{lm} = 0$  for  $l \neq m$ . Note that this assumption 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.

## 4.6.2 Properties of a nsRGH

In this section we consider the probability an order  $k$  hyperedge occurring in a nsRGH generated from  $\mathbf{U}$  and  $\mathbf{r}$ , and we denote this by  $p_{e_k} = P(y_{e_k}^{(g)} = 1 | \mu, \Sigma, r_k)$ . We present results for  $k = 2$  in Section 4.6.2 and discuss the connection probability for  $k \geq 3$  in Section 4.6.2.

### Connection Probabilities for $k = 2$

Recall that an edge  $e_2 = \{i, j\}$  is present in  $g(\mathbf{U}, \mathbf{r})$  if  $B_{r_2}(u_i) \cap B_{r_2}(u_j) \neq \emptyset$ . Hence, to obtain an expression for the occurrence probability  $p_{e_2}$ , we consider the probability of the coordinates  $u_i$  and  $u_j$  lying within distance  $2r_2$  of each other. This probability is given in Proposition 4.6.1 and follows by considering the distribution of a squared Normal random variable.

**Proposition 4.6.1:** Let  $U_i \sim \mathcal{N}(\mu, \Sigma)$ , for  $i = 1, 2, \dots, N$ , and  $\Sigma = \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_d^2)$ .

The probability of an edge  $e_2 = \{i, j\}$  occurring in  $g(\mathbf{U}, \mathbf{r})$  is given by

$$p_{e_2} = P(\|U_i - U_j\| \leq 2r_2 | \Sigma) = \int_0^{(2r_2)^2} \sum_{l=1}^d f\left(z; \frac{1}{2}, 4\sigma_l^2\right) dz, \quad (4.6.1)$$

where  $f(z; a, b) = \frac{b^a}{\Gamma(a)} z^{a-1} e^{-bz}$  is the pdf of a  $\Gamma(a, b)$  random variable.

*Proof.* See Appendix B.6.1. □

To check the validity of this result, Figure 4.6.1 shows an empirical estimate of  $p_{e_2}$  compared to the result in Proposition 4.6.1 for two choices of  $\Sigma$ . Next, we will consider the connection probability  $p_{e_k}$  for  $k \geq 3$ .

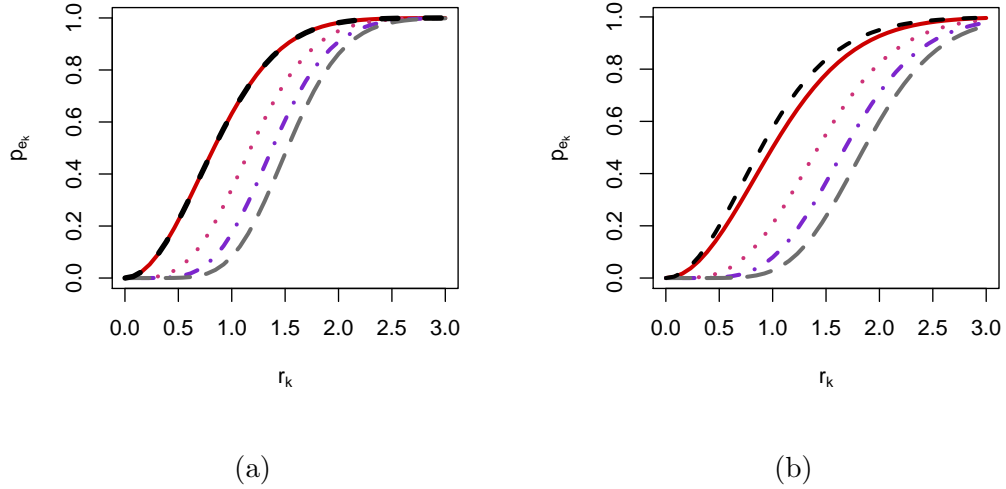


Figure 4.6.1: Estimate of probability of a hyperedge occurring for  $k = 2$  (red, solid),  $k = 3$  (pink, dot),  $k = 4$  (purple, dash-dot),  $k = 5$  (grey, dash) as a function of  $r_k$ . Points were simulated from a Normal distribution with  $\mu = (0, 0)$ , and  $\Sigma = I_2$  in (a) and  $\Sigma = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$  in (b), and the theoretical probability for  $k = 2$  is plotted for each case (black, dash).

### Connection Probabilities for $k \geq 3$

To determine  $p_{e_k}$  for  $k \geq 3$ , we first note that the edge  $e_k = \{i_1, i_2, \dots, i_k\}$  is present in  $g(\mathbf{U}, \mathbf{r})$  if  $\bigcap_{l=1}^k B_{r_k}(u_{i_l}) \neq \emptyset$ . This condition is equivalent to the coordinates  $\{u_{i_l}\}_{l=1}^k$  being contained within a ball of radius  $r_k$  (see section 3.2 of Edelsbrunner and Harer (2010)). This is depicted in Figure B.5.1, and for more details of this see Appendix B.5.1. Given this observation, we can determine  $p_{e_k}$  by finding the probability of exactly  $k$  points falling within a ball of radius  $r_k$ .

For normally distributed coordinates, the probability of a point falling within a ball of radius  $r_k$  and centre  $c$  is given by

$$P(u \in B_{r_k}(c) | \mu, \Sigma) = \int_{B_{r_k}(c)} p(u | \mu, \Sigma) du. \tag{4.6.2}$$

(4.6.2) presents a challenging computation and results for this integral are pro-



vided in Gilliland (1962) when  $d = 2$ . It is therefore not possible to obtain an exact expression for  $p_{e_k}$  when  $k \geq 3$  and we instead rely on empirical approximations.

Figure 4.6.1 shows Monte Carlo 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. From the figure, we see 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 4.6.1, we see that, as the elements of  $\Sigma$  increase, the radii  $r_k$  must also increase to obtain the same probability of a connection.

### 4.6.3 Degree Distribution for the $i^{\text{th}}$ Node

In this section we investigate the degree distribution for the  $i^{\text{th}}$  node. This is presented in Theorem 4.6.1 for hyperedges of order  $k = 2$  and  $k = 3$  and, from this result, we obtain Lemma 4.6.1 which describes the expected degree of the  $i^{\text{th}}$  node. In Theorem 4.6.1 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 4.6.1.** *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  $i^{\text{th}}$  node is given by*

$$\mathbb{E} \left[ \text{Deg}_{(i)}^{g^*} | \boldsymbol{\varphi}, \Sigma, \mathbf{r} \right] = \sum_{k=2}^K \binom{N-1}{k-1} [(1 - \varphi_k)p_{e_k} + \varphi_k(1 - p_{e_k})]. \quad (4.6.3)$$

*Proof.* See Appendix B.6.2. □

**Theorem 4.6.1:** Let  $g$  represent the nsRGH generated from  $\mathbf{U}$  and  $\mathbf{r}$ , and let  $g^*$  be the hypergraph obtained by modifying  $g$  with noise  $\boldsymbol{\varphi}$ . It follows that

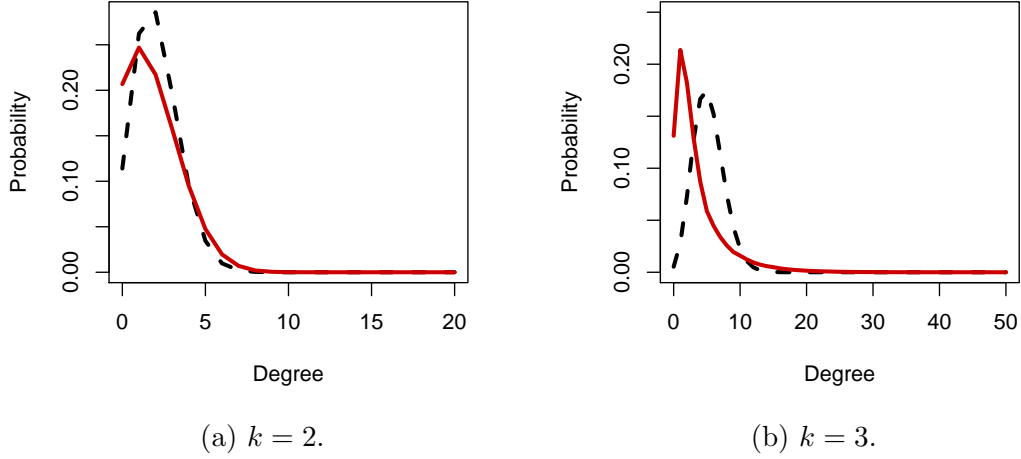


Figure 4.6.2: Comparison of theoretical (black, dashed) and simulated (red, solid) degree distribution for a hypergraph with  $N = 20$  and  $K = 3$ . Figures 4.6.2a and 4.6.2b show the degree distribution for hyperedges of order  $k = 2$  and  $k = 3$ , respectively. The theoretical degree distribution for  $k = 3$  is calculated using a monte carlo estimate of  $p_{e_3}$ .

1. For  $k=2$ :

$$\text{Deg}_{(i,2)}^{g*} | \boldsymbol{\varphi}, \Sigma, \mathbf{r} \sim \text{Binomial}(N - 1, (1 - \varphi_2)p_{e_2} + \varphi_2(1 - p_{e_2})), \quad (4.6.4)$$

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

2. For  $k=3$ :

$$\text{Deg}_{(i,3)}^{g*} | \boldsymbol{\varphi}, \Sigma, \mathbf{r} \sim \text{Poisson} \left( \sum_{\{e_3 \in \mathcal{E}_{N,3} | i \in e_3\}} \binom{N-1}{2} [(1 - \varphi_3)p_{e_3} + \varphi_3(1 - p_{e_3})] \right), \quad (4.6.5)$$

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

*Proof.* See Appendix B.6.3. □

Theorem 4.6.1 and the corresponding Lemma describe how the model parameters affect the degree distribution. The probability of connection  $p_k = (1 - \varphi_k)p_{e_k} + \varphi_k(1 - p_{e_k})$  reflects observation (O1), and we note that equivalent degree distributions can be obtained from different choices for  $\varphi_k$  and  $p_{e_k}$ . For example, small  $p_{e_k}$  and large  $\varphi_k$  will behave similarly to large  $p_{e_k}$  and small  $\varphi_k$ . However, the characteristics of the resulting hypergraphs will differ significantly in each of these cases. The Poisson approximation in Theorem 4.6.1 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). As a sanity check, Figure 4.6.2 compares the theoretical degree distribution with the simulated degree distribution for a hypergraph with  $N = 20$  and  $K = 3$ . We also compare the theoretical and observed degree distributions for the data example presented in Section 4.8.3.

## 4.7 Simulations

In this section we describe three different simulation studies. We begin in Section 4.7.1 with an investigation of the flexibility of our modelling approach in comparison with two other hypergraph models from the literature. Then, in Section 4.7.2, we examine the predictive degree distribution conditional on an observed hypergraph, and in Section 4.7.3 we consider the robustness of our model with respect to different types of misspecification.

### 4.7.1 Model depth comparisons

In this study we explore the range of hypergraphs that can be expressed in our modelling framework. We compare this with two other statistical models from the literature, which have been designed according to different modelling aims. For each model, we specify several cases which are designed to highlight particular aspects of the model. Then, we simulate hypergraphs for each case and record summary statis-

tics to characterise the simulated hypergraphs. We will begin by outlining the models of Stasi et al. (2014) and Ng and Murphy (2018) for a hypergraph with  $N$  nodes, and then describe and justify the choice of cases and summary statistics. Finally, we discuss the results of the simulations.

We first describe an extension of the  $\beta$ -model for random graphs (see Holland and Leinhardt (1981)), introduced by Stasi et al. (2014). 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  $\beta_i$ , for  $i = 1, 2, \dots, N$ . Let  $y_{e_k} = 1$  denote the presence of the hyperedge  $e_k = \{i_1, i_2, \dots, i_k\} \subseteq \{1, 2, \dots, N\}$  for  $k \geq 2$ . The probability of the hyperedge  $e_k$  occurring is then given by

$$p(y_{i_1 i_2 \dots i_k} = 1) = \frac{\exp\{\beta_{i_1} + \beta_{i_2} + \dots + \beta_{i_k}\}}{1 + \exp\{\beta_{i_1} + \beta_{i_2} + \dots + \beta_{i_k}\}}. \quad (4.7.1)$$

Since the hyperedges are assumed to occur independently conditional on  $\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_N)$ , 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 our study.

Next, we overview the model introduced in Ng and Murphy (2018) which 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  $i^{\text{th}}$  node belongs to a hyperedge with size label  $s$  and topic label  $t$  with probability  $\alpha_s \phi_{it}$ , so that  $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_S)$  controls the size clusters and  $\boldsymbol{\phi} = \{\phi_{it}\}_{i=1,2,\dots,N,t=1,2,\dots,T}$  controls the topic clusters.

Additionally, we let  $\pi = (\pi_1, \pi_2, \dots, \pi_T)$  and  $\tau = (\tau_1, \tau_2, \dots, \tau_S)$  denote the prior topic and size assignment probabilities, respectively. To write down the likelihood, we let  $x_{ij} = 1$  indicate that the  $i^{\text{th}}$  node belongs to the  $j^{\text{th}}$  hyperedge,  $z_{jt}^{(1)} = 1$  indicate that the  $j^{\text{th}}$  hyperedge has topic label  $t$ , and  $z_{js}^{(2)} = 1$  indicate that the  $j^{\text{th}}$  hyperedge has size label  $s$ . The likelihood is then given by

$$\mathcal{L}(x, z^{(1)}, z^{(2)}; \theta) = \prod_{j=1}^M \prod_{t=1}^T \prod_{s=1}^S \left[ \pi_t \tau_s \prod_{i=1}^N (\alpha_s \phi_{it})^{x_{ij}} (1 - \alpha_s \phi_{it})^{1-x_{ij}} \right]^{z_{jt}^{(1)} z_{js}^{(2)}}. \quad (4.7.2)$$

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 this simulation study we consider the models detailed above and the hypergraph model described in Algorithm 8. Before describing the set up, we note that each of these models has been designed for a different purpose. The  $\beta$ -model of Stasi et al. (2014) allows fine control over the degree distribution through the parameters  $\beta$ , and the model of Ng and Murphy (2018) is designed to describe hyperedges which exhibit a clustering structure. The modelling choices in our approach impose different characteristics on the resulting simulated hypergraphs. By determining the hyperedges from the Čech complex, we expect that the resulting hypergraphs will exhibit transitivity since, if  $\{i, j\}$  and  $\{i, k\}$  lie sufficiently close to be connected, then  $\{j, k\}$  is likely to also be present in the hypergraph. Additionally, the presence of hyperedges  $\{i, j\}$ ,  $\{i, k\}$  and  $\{j, k\}$  suggests that the hyperedge  $\{i, j, k\}$  is more likely to be present. To model hypergraphs with different characteristics within our framework, we can investigate the effect of changing the assumed distribution of the latent coordinates and using an alternative simplicial complex. This point will be discussed in more detail in Section 4.9.

We now describe the set of metrics used to capture the above model behaviours. To capture the connectivity patterns we expect from our model, we record counts for the subgraphs depicted in Figure 4.7.1a. To measure the degree distribution and

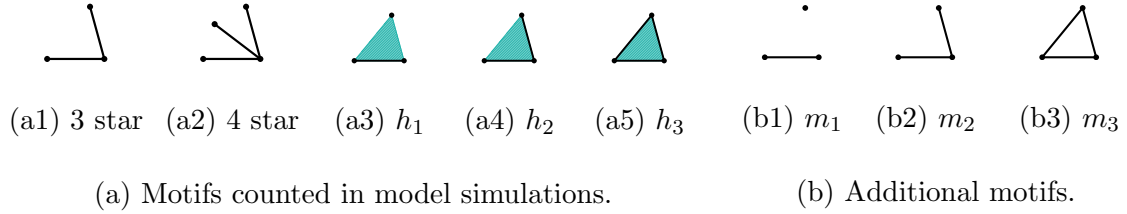


Figure 4.7.1: Depiction of motifs considered in Sections 4.7 and 4.8.

spread of hyperedge orders we record the percentiles of the node degrees and edge sizes, respectively. Additionally, we record the density of the hyperedges of order  $k = 2$  and  $k = 3$ . Note that, since the number of possible hyperedges of order  $k$  is  $\binom{N}{k}$ , we expect that the density of order  $k$  edges will decrease as  $k$  grows. Finally, to determine the clustering in the hypergraph, we project the hypergraph onto a pairwise graph such that the edge  $\{i, j\}$  exists if  $i$  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. The modularity 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. For each model, we have specified several cases which showcase the features of the model and a summary of these cases can be found in Table 4.7.1.

Figures 4.7.2a, 4.7.2b and 4.7.2c show the results of study after 10,000 simulations from each of the models. Clockwise from the top left, these plots show subgraph counts for the motifs depicted in Figure 4.7.1a, percentiles of the hyperedge order, the number of clusters and modularity, the density of hyperedges of order 2 and 3, and percentiles of the node degree. We are able to demonstrate the strengths of each model by comparing these metrics. For each model, the cases correspond to those described in Table 4.7.1.

In Figure 4.7.2a we see the summary measures for the  $\beta$  hypergraph model. From this bottom-left plot of this figure, we observe that each case demonstrates a very

different behaviour in the degree distribution as we would expect. In case 1, all nodes have a similar degree and, in case 2, a portion of nodes have either a very large or small degree. It is also apparent that, since case 2 generates denser hypergraphs, larger motif counts are observed. The maximum hyperedge order was set to 4, and so no hyperedges for  $k \geq 5$  are generated.

Figure 4.7.2b shows the equivalent plot for the model of Ng and Murphy (2018). Firstly, we note that the topic clustering is consistently captured for all simulated hypergraphs. We see that the degree distributions are largely the same, however by comparing case 1 and 4 we observe that we can express different levels of connectivity. When simulating from this model, we are unable to explicitly control the order of the hyperedges and we note that this is controlled by the probabilities  $\alpha$  and  $\phi$ . 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, Figure 4.7.2c shows our results for the latent space hypergraph model. Overall, we observe that there are a greater number of motifs observed than in the previous models. Additionally, we see that we are able to demonstrate more control over the motif counts. For example, consider case 3 where the order  $k = 2$  hyperedges are denser and in case 4 where the order  $k = 3$  hyperedges are denser. The counts for triangles and  $h_1$  subgraphs clearly reflect the number of hyperedges of each order. In this model we also observe some control over the degree distribution and density. As expected, 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. As commented previously, we may alter aspects of our model to incorporate community structure or to vary the degree distribution. For example, we may model the latent coordinates as a mixture of Gaussians.

Model	Case	Parameters
Stasi et al. (2014)	1) All nodes equally likely to form connections	$\beta_i = -1.4$ for $i = 1, 2, \dots, N$
	2) Some nodes more likely to form connections	$\beta = (-0.5, -0.53, \dots, -1.97, -2)$
Ng and Murphy (2018)	1) Single hyperedge cluster	$G = K = 1, a = 1, \phi_{i1} = 0.075,$ $\pi = b\mathbf{1}, \tau = 1$
	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 $\Sigma$	$\mathbf{r} = (0.18, 0.3, 0.35), \mu = (0, 0),$ $\Sigma = 0.25 \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix},$ $\boldsymbol{\psi}_0 = (0.01, 0.01, 0.01),$ $\boldsymbol{\psi}_1 = (0.01, 0.01, 0.01)$
	2) No correlation in $\Sigma$	$\mathbf{r} = (0.18, 0.3, 0.35), \mu = (0, 0),$ $\Sigma = 0.25 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$ $\boldsymbol{\psi}_0 = (0.01, 0.01, 0.01),$ $\boldsymbol{\psi}_1 = (0.01, 0.01, 0.01)$
	3) Dense in $e_2$ , sparse in $e_3, e_4$	$\mathbf{r} = (0.2, 0.3, 0.35), \mu = (0, 0),$ $\Sigma = 0.25 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$ $\boldsymbol{\psi}_0 = (0.01, 0.01, 0.01),$ $\boldsymbol{\psi}_1 = (0.01, 0.5, 0.01)$
	4) Sparse $e_2, e_4$ , dense in $e_3$	$\mathbf{r} = (0.1, 0.35, 0.4), \mu = (0, 0),$ $\Sigma = 0.25 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$ $\boldsymbol{\psi}_0 = (0.01, 0.01, 0.01),$ $\boldsymbol{\psi}_1 = (0.01, 0.01, 0.01)$
	5) Increase latent dimension from $d = 2$ to $d = 3$	$\mathbf{r} = (0.18, 0.3, 0.35), \mu = (0, 0),$ $\Sigma = 0.25 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$ $\boldsymbol{\psi}_0 = (0.01, 0.01, 0.01),$ $\boldsymbol{\psi}_1 = (0.01, 0.01, 0.01)$

Table 4.7.1: Cases for each hypergraph model considered in the model depth comparison study. The case numbers correspond to the labels in Figures 4.7.2a, 4.7.2b and 4.7.2c. For all cases set  $N = 50$  and, where appropriate,  $K = 4$ .



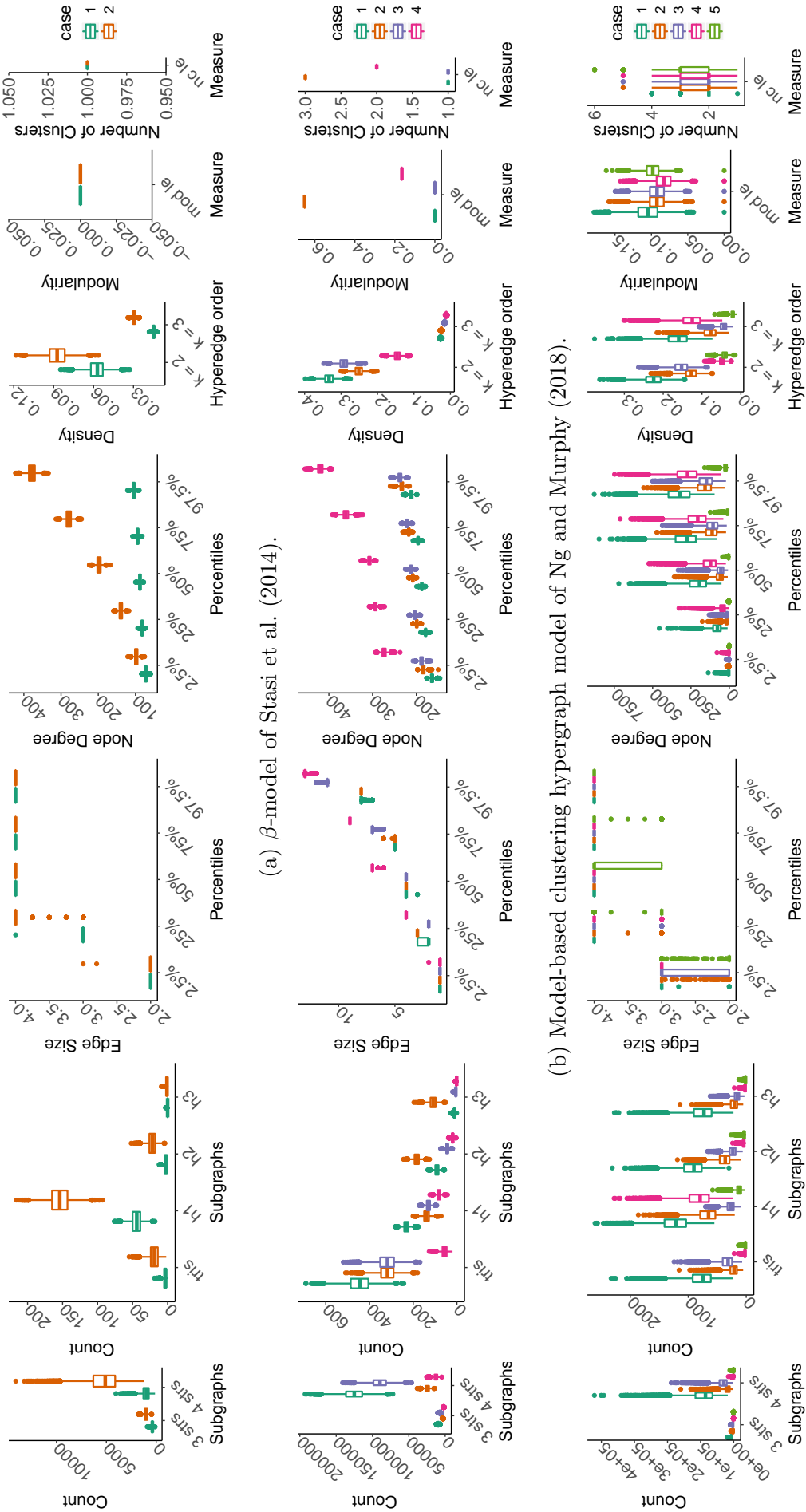
Currently, there do not exist theoretical results for subgraph counts within our framework. The most relevant work can be found in Coulson et al. (2016), who present results for subgraph counts in the SBM and graphons. However, it is not obvious how to develop similar results for our framework.

In this study, we have clearly demonstrated 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. However we may make alternative choices for the distribution of the latent coordinates and the Čech complex to adapt our framework to express hypergraphs with different characteristics. We note that, whilst simulating these graphs may be straightforward, fitting them may be much more challenging. This point will be discussed further in Section 4.9.

### 4.7.2 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 8 to a hypergraph  $h_{obs}$  and we obtain the parameter estimates  $\hat{\mu}$ ,  $\hat{\Sigma}$ ,  $\hat{r}$ ,  $\hat{\boldsymbol{\phi}}^{(0)}$  and  $\hat{\boldsymbol{\phi}}^{(1)}$ . Conditional on these estimates, we may simulate new nodes and determine the hyperedges induced from these additional nodes. Through repeated simulation we can then empirically estimate the predictive degree distribution of the newly simulated nodes and  $h_{obs}$ .



(c) Latent space hypergraph model detailed in Algorithm 8.

Figure 4.7.2: Summary of hypergraphs simulated from each of the models considered in Section 4.7.1. The cases considered are summarised in Table 4.7.1.

To implement this procedure, we begin by simulating a hypergraph  $h_{sim}$  according to Algorithm 8 with  $N = 50, d = 2, K = 3, \mathbf{r} = (0.32, 0.4), \psi_k^{(0)} = \psi_k^{(1)} = 0.001, \mu = (0.16, 1.24)$  and  $\Sigma = \begin{pmatrix} 0.58 & 0 \\ 0 & 0.58 \end{pmatrix}$ . We then estimate the model parameters for this hypergraph and, after 10,000 post burn-in iterations, we obtain  $\hat{\mathbf{r}} = (0.13, 0.16), \hat{\boldsymbol{\psi}}^{(0)} = (0.0058, 0.0014), \hat{\boldsymbol{\psi}}^{(1)} = (0.0057, 0.0035), \hat{\boldsymbol{\mu}} = (-0.13, 0.44)$  and  $\hat{\Sigma} = \begin{pmatrix} 0.14 & -0.0039 \\ -0.0039 & 0.078 \end{pmatrix}$ .

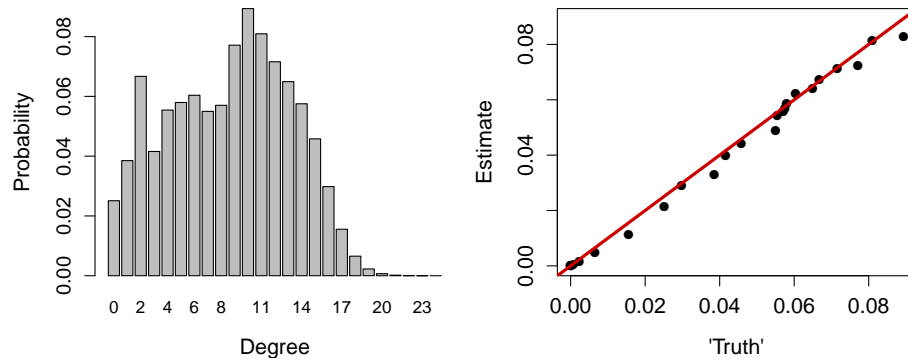
To estimate the posterior predictive degree distribution we apply the following procedure  $n_{rep}$  times.

1. Set  $u_i^* = \hat{u}_i$  for  $i = 1, 2, \dots, N$ .
2. Simulate coordinates  $u_i^* \sim \mathcal{N}(\hat{\boldsymbol{\mu}}, \hat{\Sigma})$ , for  $i = N + 1, N + 2, \dots, N + N^*$ .
3. Determine the hypergraph  $h_{sim}^*$  obtained by taking the union of  $h_{obs}$  and the additional hyperedges induced from  $\mathbf{U}^* = \{u_i^*\}_{i=N+1}^{N+N^*}, \hat{\mathbf{r}}, \hat{\boldsymbol{\psi}}^{(0)}$  and  $\hat{\boldsymbol{\psi}}^{(1)}$ .
4. Calculate the degree distribution of  $h_{sim}^*$ .

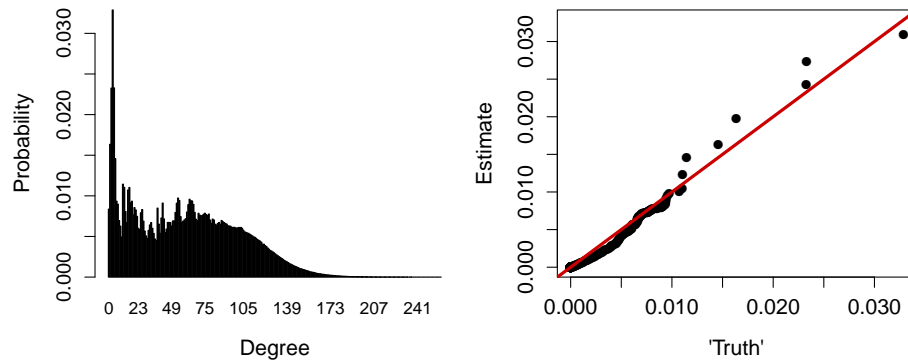
By averaging over the  $n_{rep} = 100,000$  simulated degree distributions, we then obtain an estimate of the degree distribution. Since we know the true model parameters, we also estimate the true predictive degree distribution. We refer to this distribution as the prior predictive below.

Recall that, for  $u_i \in \mathbb{R}^2$ , two coordinates are specified as anchor points throughout posterior sampling. The fixing of these points determines the scaling and hence affects the magnitude of  $\Sigma$  and  $\mathbf{r}$ . It is therefore not appropriate to compare parameter estimates with the truth directly. However, by comparing the prior and posterior predictive degree distributions, we consider a fair comparison between the true and estimated model parameters. We expect that these two distributions are similar if the model has been fitted well.

The full estimated prior and posterior predictive degree distributions are presented in Figure 4.7.3, where the left panel shows the estimated prior predictive and the right panel shows a qq-plot of the prior and posterior predictive degree distributions. We



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



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

Figure 4.7.3: Comparison of prior and posterior predictive degree distributions for  $N^* = 10$  newly simulated nodes. In each figure, the left panel shows the prior predictive degree distribution, and the right panel shows a qq-plot of the prior and posterior predictive degree distributions. Figures 4.7.3a and 4.7.3b show the degree distributions for hyperedges of order 2 and 3, respectively.

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.

We may also examine the distributions for hyperedges occurring between newly simulated nodes only, and the hyperedges occurring between the nodes of  $h_{sim}$  and the

Index	$\mathbf{U}$ misspecification	$\mathbf{r}$ 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 4.7.2: Types of misspecification.

newly simulated nodes. Plots for these cases are presented in Appendix B.7 and, for both cases, we see a close correspondence between the prior and posterior predictive degree distributions.

### 4.7.3 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 4.7.2. 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 4.7.2, 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 4.7.2 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 4.7.1b3, 4.7.1a3, 4.7.1a4 and 4.7.1a5.
- The density of order 3 hyperedges.

The results of our study are presented in Figure 4.7.4, where the  $i^{th}$  row corresponds to the  $i^{th}$  case detailed in Table 4.7.2. 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 constraints implied by the latent representation. 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.

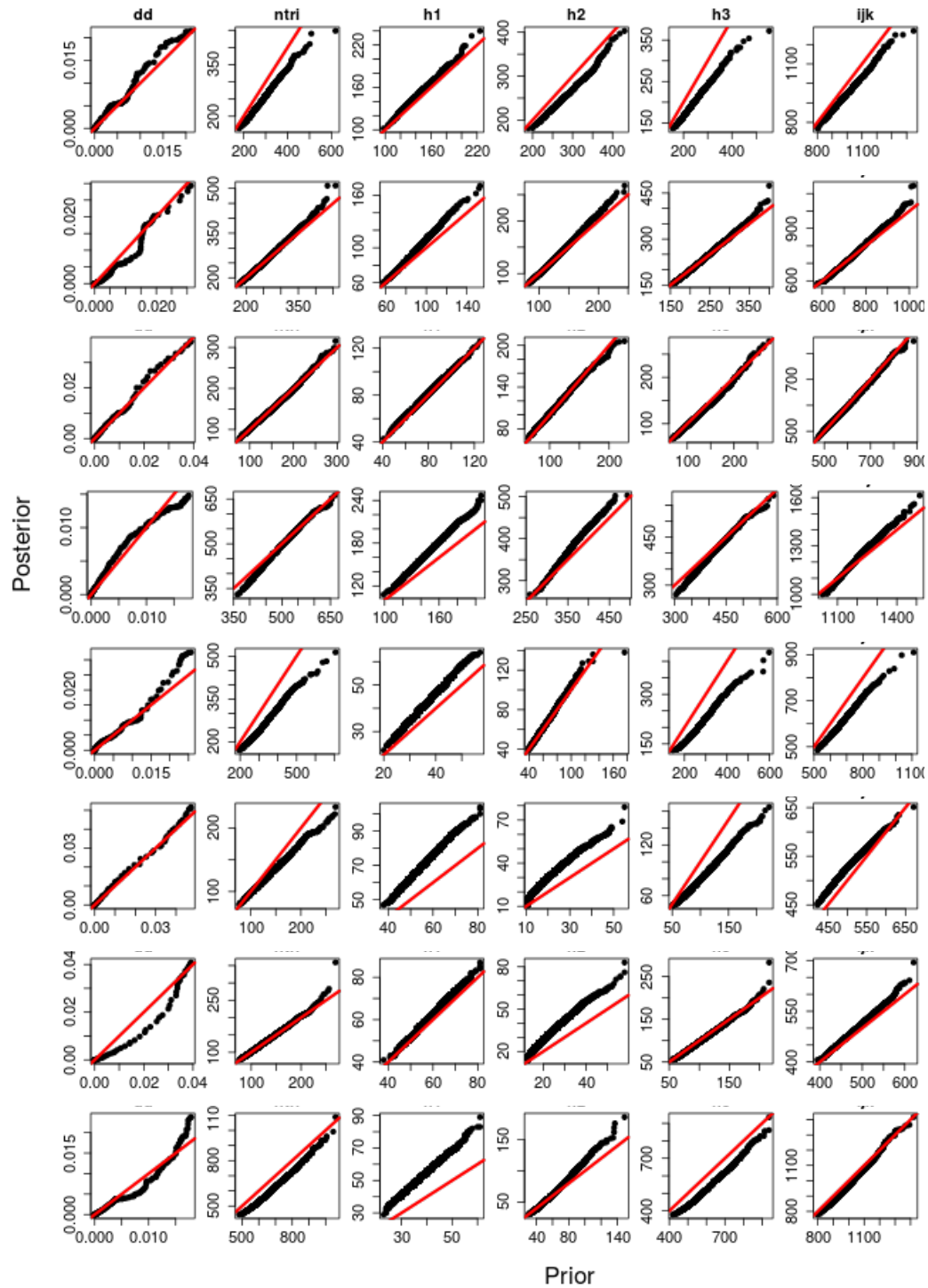


Figure 4.7.4: Summary of misspecification simulation study. Left to right: average degree distribution, number of triangles Figure 4.7.1b3, number of Figure 4.7.1a3, number of Figure 4.7.1a4, number of Figure 4.7.1a5 and density of order 3 hyperedges. Each row corresponds to the misspecification cases summarised in Table 4.7.2. The  $y$  and  $x$  axes show the quantiles of the posterior and prior predictives, respectively. The red lines correspond to  $y = x$ .

## 4.8 Real data examples

In this section we analyse three real world hypergraph datasets using the model we have introduced. To begin, 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). Next, we consider a dataset describing company leadership and investigate predictive inference on the observed hypergraph compared to hypergraphs imputed from the pairwise projected graph. Finally, we consider a coauthorship dataset and assess predictive inference by comparing to the observed hypergraph.

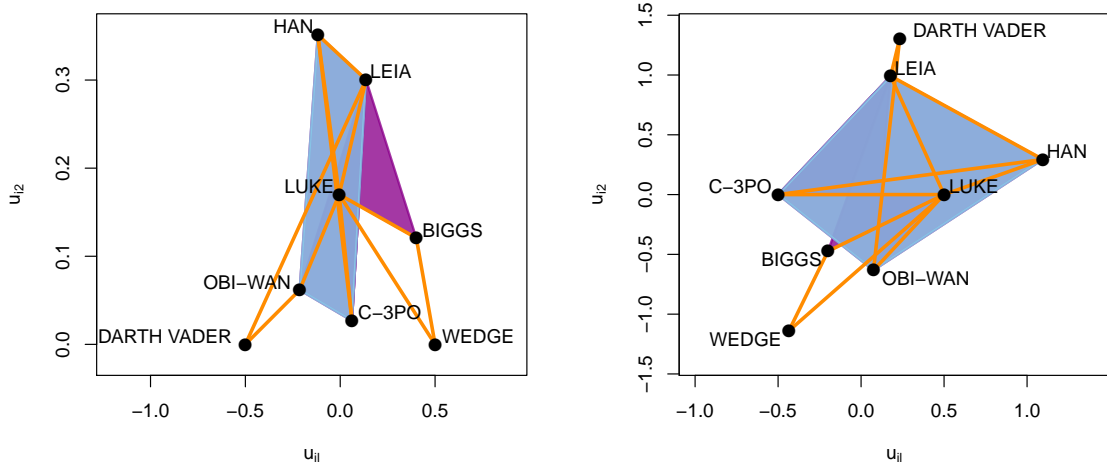
### 4.8.1 Star Wars: A New Hope

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 and contrast 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 7 and set the upper limit for  $\varphi_k$  to be  $0.75 \times$  the density of order  $k$  hyperedges, for  $k = 3, 4$ .

The posterior mean of the latent coordinates after 37500 post burn-in iterations is given in Figure 4.8.1a. 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



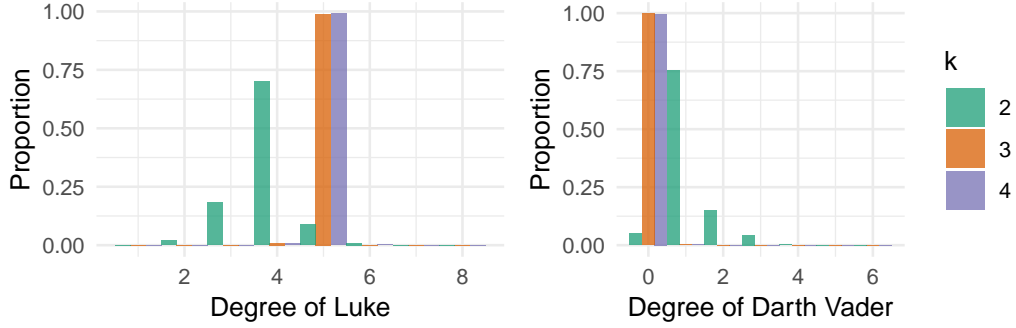


(a) Upper limit of  $\varphi_k$  is set to  $0.75 \times$  the observed order  $k$  hyperedge density. (b) Upper limit of  $\varphi_k$  is set to  $1.5 \times$  the observed order  $k$  hyperedge density.

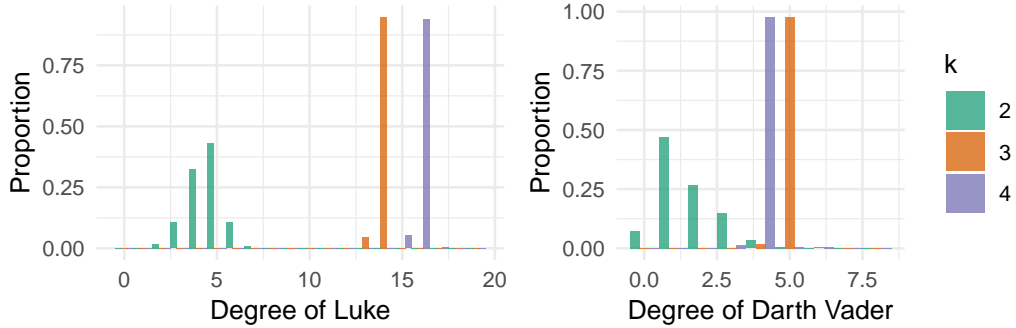
Figure 4.8.1: Posterior mean of latent coordinates for the Star Wars dataset for different upper limits on  $\varphi$ . Connections in orange, blue and purple correspond to hyperedges of order  $k = 2, 3$  and  $4$ , respectively.

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. Note that 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 4.8.1a is largely determined by the order 3 and 4 hyperedges between “Leia”, “C-3PO”, “Luke”, “Obi-Wan” and “Han”. We note that the characters “Wedge” and “Darth Vader” are less connected, and so we see them located on the periphery of the latent representation.

We also consider setting the upper limit for  $\varphi_k$  to be  $1.5 \times$  the density of order  $k$  hyperedges, for  $k = 3, 4$ , and the posterior mean of the latent coordinates for this case is given in Figure 4.8.1b. Here we also see the importance of the character “Luke” reflected in the latent coordinates, however the increased noise parameter



(a) Upper limit of  $\varphi_k$  is set to  $0.75 \times$  the observed order  $k$  hyperedge density.



(b) Upper limit of  $\varphi_k$  is set to  $1.5 \times$  the observed order  $k$  hyperedge density.

Figure 4.8.2: Predicted degree distributions conditional on the fitted model. Given  $\hat{U}$ ,  $\hat{r}$  and  $\hat{\varphi}$  we simulate the connections in the hypergraph to estimate the degree distribution, and the upper limit for  $\varphi$  is 0.75 the hyperedge density in Figure 4.8.2a and 1.5 the hyperedge density in Figure 4.8.2b. The left plots show the degree distribution for “Luke” and the right plots show the degree distribution for “Darth Vader”. The order 2, 3, and 4 hyperedges are shown in green, orange, and purple, respectively.

means that the latent coordinates are less constrained. To make this point further, we now consider the variability in the observed connections.

As mentioned in Section 4.7, our modelling framework allows us to explore predictive distributions. Given the fitted model, we can simulate new connections to examine how variable the degree of specific nodes are expected to be. More specifically, we consider the degree distributions for the characters “Luke” and “Darth Vader” by repeatedly simulating their connections given  $\hat{U}$ ,  $\hat{r}$  and  $\hat{\varphi}$ . We do this for

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 Figures 4.8.2a and 4.8.2b, respectively. In both Figures, we see a clear difference between the levels of connectivity for each of these characters. Since “Luke” is more centrally located with respect to the latent coordinates, we observe that this character 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. Comparing Figures 4.8.2a and 4.8.2b, we see more variability and a higher level of connectivity in Figure 4.8.2b. This is due to the larger upper limit on  $\boldsymbol{\varphi}$ . Since the noise is estimated to be larger, the latent representation explains fewer of the observed hyperedges. Therefore, when we fix the coordinates to  $\hat{\boldsymbol{U}}$ , our estimates may not reflect the observed hypergraph well.

Although we are able to draw parallels between our observations and the observations of Ng and Murphy (2018), our approach differs considerably. We now comment on the advantages of each approach. Firstly, in Ng and Murphy (2018) the authors incorporate multiple occurrences of a hyperedge and hyperedges containing a single node into their analysis. Our methodology does not facilitate this, and so the dataset was reduced accordingly. Secondly, our model provides a visualisation of the hypergraph and the approach of Ng and Murphy (2018) does not. When the parameters  $\boldsymbol{\varphi}$  are appropriately constrained, this visualisation reflects many observations of the analysis in Ng and Murphy (2018). Finally, our framework allows us to investigate the predictive distributions, and this has allowed us to comment on the expected variability of the degree of certain nodes.

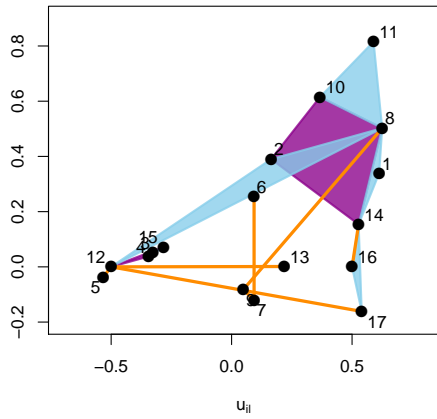
## 4.8.2 Corporate Leadership

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

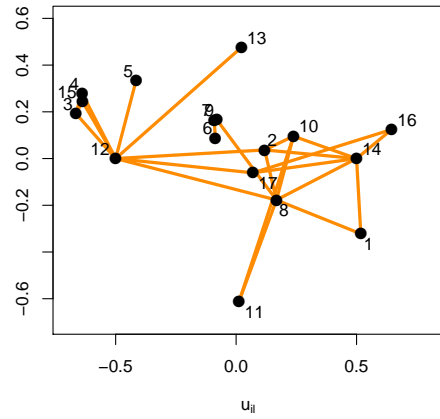
KONECT (2017), and to analyse this dataset we remove hyperedges of order larger than 4. We comment that our model can in theory express hyperedges of arbitrary order, however larger values of  $K$  become increasingly computationally challenging in our framework. Finally, we remove nodes with degree 0 to obtain a hypergraph with  $N = 17$  nodes.

As noted in Section 4.1, 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 7 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 4.8.1, we constrain  $\varphi_k$  to be at most 20% of the observed density of order  $k$  hyperedges.

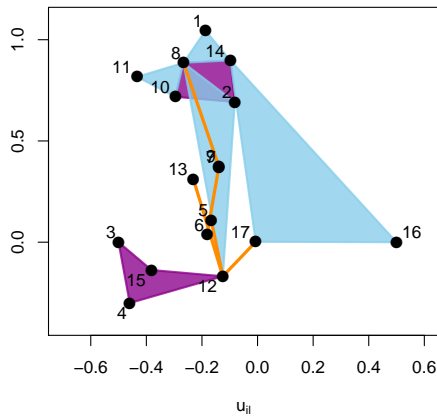
The posterior mean of the latent coordinates for each hypergraph are shown in Figure 4.8.3. We observe several similarities between the latent representations, such as the placement of the set of nodes  $\{3, 4, 15, 12\}$ , however the constraints imposed on the latent coordinates differ for each case. For instance, comparing Figures 4.8.3a and 4.8.3b, 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



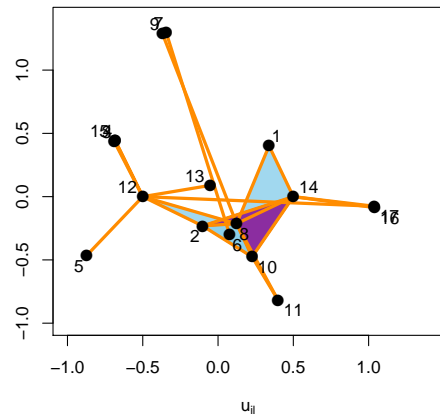
(a) Observed hypergraph.



(b) Pairwise projected graph.



(c) Maximal clique hypergraph.



(d) Simplicial hypergraph.

Figure 4.8.3: Posterior mean of the latent coordinates after 25000 post burn-in iterations. Figure 4.8.3a: observed hypergraph with  $K = 4$ . Figure 4.8.3b: graph obtained by connecting nodes if they are contained within the same observed hyperedge. Figure 4.8.3c: hypergraph obtained from representing maximal cliques in the graph by a hyperedge. Figure 4.8.3d: 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.

graph in Figure 4.8.3b. Finally, we comment that the simplicial clique hypergraph in Figure 4.8.3d imposes the strongest constraints on the latent representation due to the density of order 2 and order 3 hyperedges.

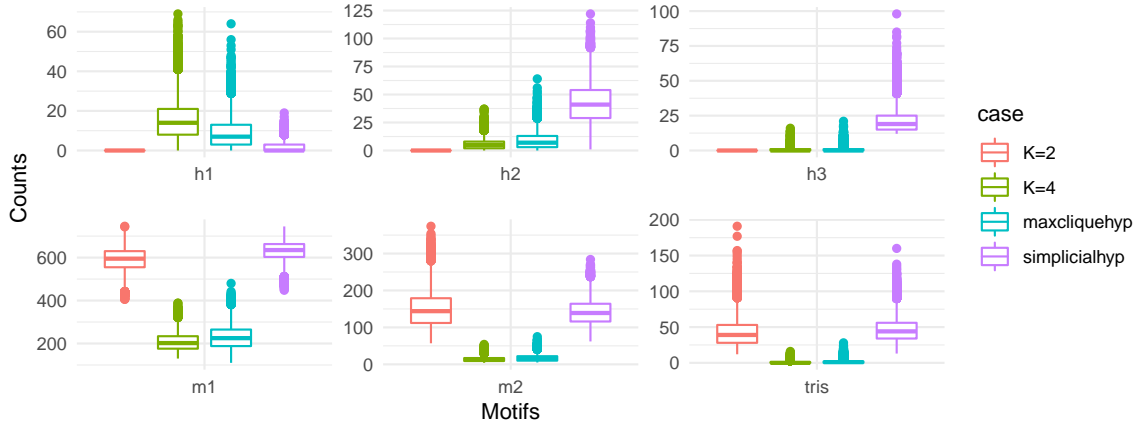


Figure 4.8.4: Predictive distributions for motif counts for  $N^* = 5$  newly simulated nodes. Top row: predictives for the motifs shown in Figures 4.7.1a3, 4.7.1a4 and 4.7.1a5. Bottom row: predictives for the motifs shown in Figures 4.7.1b1, 4.7.1b2 and 4.7.1b3.

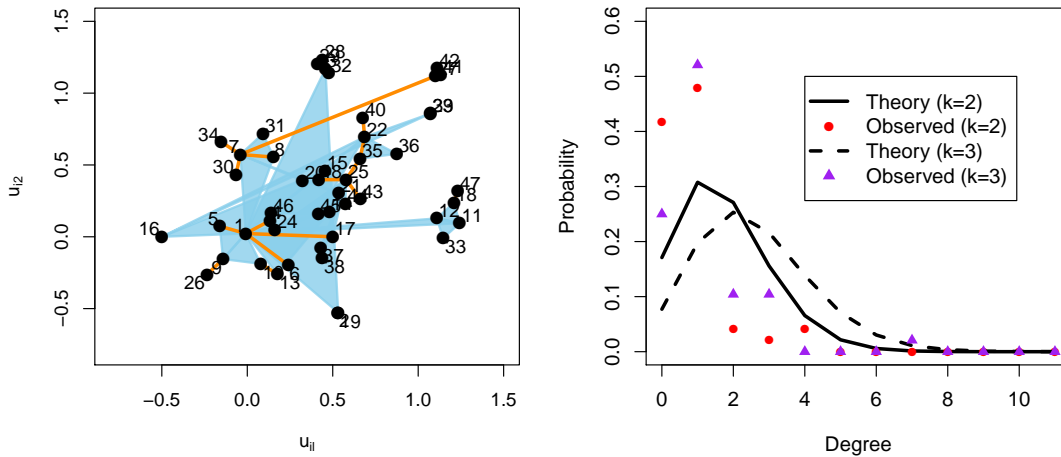
To further compare these fitted models we consider predictive inference for  $N^* = 5$  newly simulated nodes. We focus on the motifs shown in Figure 4.7.1b and Figures 4.7.1a3, 4.7.1a4 and 4.7.1a5, and the predictive motif counts are shown in Figure 4.8.4 for each model fit. First, we will discuss in predictive inference for the motifs involving order 2 edges, namely  $m_1$  (Figure 4.7.1b1),  $m_2$  (Figure 4.7.1b2) and triangles (Figure 4.7.1b3). 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 4.7.1a3),  $h_2$  (Figure 4.7.1a4) and  $h_3$  (Figure 4.7.1a5). Since the graph contains no information about hyperedges, we cannot pre-

dict 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 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.

### 4.8.3 Coauthorship for Statisticians

In this section we 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  $K = 3$ . To obtain the subset we first restrict to hyperedges of order less than 4. Then 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 7 and the posterior mean of the latent coordinates is shown in Figure 4.8.5a. 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  $g_{N,K}(\hat{\mathbf{U}}, \hat{\mathbf{r}})$  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. Hence, the hyperedge modification explains



(a) Posterior mean of latent coordinates after 20000 post burn-in iterations. Hyperedges of order 2 and 3 are shown in orange and blue, respectively. (b) Observed degree distribution compared to theoretical degree distribution calculated using the fitted model parameters. We obtain  $\hat{p}_{e_3}$  through simulation.

Figure 4.8.5: Fitted model for coauthorship example.

only a small proportion of the observed hyperedges. We note that the nodes 1, 7 and 25 have the highest degree, and we see these nodes placed close to the estimated mean  $\hat{\mu}$  of the distribution of the latent coordinates. For this example, we also consider the theoretical degree distributions discussed in Section 4.6. In Figure 4.8.5b we compare the observed degree distribution with the theoretical degree distribution calculated with the fitted model parameters. Since we cannot find an exact expression for  $p_{e_3}$ , we obtain an estimate of this using simulation. We observe a strong correspondence between the observed degree distribution and the degree distribution predicted by the theory.

To further assess the fitted model, we now compare subgraph counts for the next wave of sampling with the predictive distributions. We estimate the predictive distribution for the subgraphs shown in Figures 4.7.1a3, 4.7.1a4, 4.7.1a5, 4.7.1b1, 4.7.1b2 and 4.7.1b3, and compare this with the counts observed in the next wave of subsampling. The predictive distributions for the addition  $N^* = 19$  nodes are shown in Figure



4.9.1, 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 4.9.1, 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.

## 4.9 Discussion

In this chapter we have introduced a latent space model for hypergraph data in which the nodes are represented by coordinates in  $\mathbb{R}^d$ . To extend the framework introduced in Hoff et al. (2002), we have relied on 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 imposes properties on the hypergraphs generated from our model, including a type of ‘higher-order transitivity’. This property, in which a 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 4.7.1. In particular, we see a greater presence of certain subgraph counts in comparison to the two other models considered. 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.

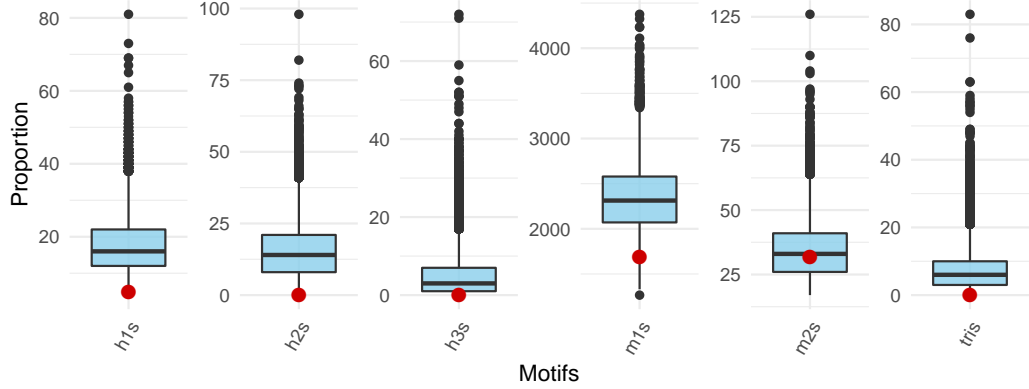


Figure 4.9.1:  $N^* = 19$  predictive subgraph counts. Left to right: motifs depicted in Figure 4.7.1a3, 4.7.1a4, 4.7.1a5, 4.7.1b1, 4.7.1b2 and 4.7.1b3. The red dots correspond to the observed motif counts for the newly sampled nodes.

Firstly, without this modification, the conditional distribution  $\mathcal{L}(\mathbf{U}, \mathbf{r}; h_{N,K})$  would be equal to one only when there is a perfect correspondence between the observed and estimated hyperedges. Hence model fitting may be difficult. Secondly, the modification extends the support of the model to the space of all hypergraphs and, without the modification, it is unclear whether an observed hypergraph is expressible within our framework which greatly limits the applicability of our model. 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 still remains. From (4.4.1) we observed that a hypergraph generated from our model is a modification of a nsRGH, as the probability of modification grows, the generated hypergraphs will behave more like the hypergraph analogue to an Erdős-Rényi random graph. Therefore, to maintain the hyperedge properties inherited from the latent space, we impose that the probability of modification is small. For a hypergraph on  $N$  nodes, the density of order  $k$  hyperedges is given by the fraction of possible order  $k$  hyperedges that are present in the hypergraph. It is clear that, to obtain a similar density of hyperedges for different  $k$ , the probability of modification must scale according to number of possible hyperedges of each order. This means that the magnitude of the modification probability

will decrease as  $k$  grows. The hyperedge modification has a connection with measurement error models in which the observations are assumed to be measured with some noise. 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 observation on the magnitude of the probability of hyperedge modification.

To obtain posterior samples we rely on a Metropolis-Hastings-within-Gibbs MCMC scheme in which each parameter is sampled conditionally on the remaining parameters. In Section 4.4, we observed that the conditional distribution  $p(g_{N,K}(\mathbf{U}, \mathbf{r}) | \mu, \Sigma, \mathbf{r})$  is invariant to rotations, translations and reflections of the latent coordinates  $\mathbf{U}$ . However, since samples are obtained from the conditionals, these sources of non-identifiability can be removed in a post-processing step using a Procrustes transform. This approach is typically used for latent space network models and ensures that the samples have a clear interpretation. We instead infer the latent representation on the Bookstein space of coordinates, which avoids the need for post-processing and further removes the source of non-identifiability from joint rescaling of  $\mathbf{U}$  and  $\mathbf{r}$ . To initialise the MCMC, we rely on techniques commonly used in the latent space network literature. Since random initialisation of  $\mathbf{U}$  and  $\mathbf{r}$  performs poorly, we use generalised multidimensional scaling (GMDS) to determine initial values of  $\mathbf{U}$  and the radii are then scaled accordingly. GMDS was used in this context by Sarkar and Moore (2006), and details of the MCMC initialisation are given in Appendix B.4. We also exploit the connection with computational topology in our MCMC scheme. By relying on existing tools we are able to sample from our model and, in particular, to calculate the Čech complex we use the GUDHI C++ library (see The GUDHI Project (2015)).

Although the model we present can in theory be extended to hypergraphs with arbitrary maximum edge size  $K$ , this proves computationally challenging in practice. In our examples, we have predominantly restricted to cases in which  $K$  is equal to 3

or 4. This allows insight on the hypergraph relationships, but this restriction of  $K$  is currently a large limitation of our methodology. In the next paragraph we comment on a possible direction for a more scalable model. As previously mentioned, there are some motifs, for example stars, which may be difficult to express in our framework. Whilst the hyperedge modification can aid fitting here, increasing the probability of modification may have undesirable effects on the other connections. The number of leaves that can be expressed in a star is also related with the choice of latent dimension (see Helly’s Theorem, Section 3.2 of Edelsbrunner and Harer (2010)). For interpretation, we have assumed that  $d$  is small and choosing  $d$  in a more principled manner requires careful consideration. In this chapter we have presented two different models (detailed in Algorithms 7 and 8) for hypergraph. The model given in Algorithm 7 has a single modification parameter for each order hyperedge, and this model has a clear interpretation. From Proposition 3.1 in Lunagómez et al. (2019), it follows that hypergraphs with a greater number of modifications are less likely to occur. However, the interpretation for the model in Algorithm 8 is less straightforward. Since there are competing sources of modification, the above argument only applies when the density of the hypergraphs remain fixed. A related framework is considered in Le et al. (2018), where the authors consider recovery of a true network given a set of noisy realisations. In this work, the sources of edge noise are divided into false positives and false negatives. By viewing the underlying nsRGH  $g_{N,K}(\mathbf{U}, \mathbf{r})$  as the truth, we see that the probabilities of hyperedge modification in Algorithm 8 have an analogous interpretation.

There are a number of extensions to the model we have presented that can be explored. For instance, the choice of underlying distribution on  $\mathbf{U}$  will affect the characteristics of the hypergraphs expressed by the model. This can be examined from a theoretical perspective by adapting the arguments discussed in Section 4.6, or practically by simulating from the generative model. Exploring this would further explain which aspects of the model depth simulations in Section 4.7.1 are an artefact

of our modelling choices. In Spencer and Rohilla Shalizi (2017) the authors assume that the latent coordinates in a latent position are generated according to a Poisson process, and show that this is able express graphs with various levels of sparsity. It would be interesting to examine this within the hypergraph setting, and we leave this to future work. Another extension would be to consider modelling hypergraphs which exhibit community structure. One approach would be to assume that the latent coordinates are distributed according to a mixture of Gaussians, an idea which has been explored in the latent space network literature (for example, see Handcock et al. (2007)). Alternatively, Rubin-Delanchy et al. (2017) show that a generalisation of the latent position framework in which connections are determined via a dot product is able to express graphs with community structure. This idea may also be considered in the hypergraph setting, though this extension is likely to be more involved. There also exist several other interesting extensions for which the adaptation of our model is less clear. In many real world examples, the hyperedges may occur multiple times. This motivates developing a model for non-binary hypergraphs, though it is not clear how to extend our model accordingly. Another line of future work stems from considering alternatives to the Čech complex. One choice to consider is the Vietoris-Rips complex (Vietoris (1927), Gromov (1987)) in which an order  $k$  hyperedge exists if each of the  $\binom{k}{2}$  balls of radius  $r$  intersect. Evaluating this complex only requires considering the pairwise intersections, and so is likely more scalable in terms of  $K$ . Finally, in this chapter we have not explored the addition of covariate information. Typically covariate information is included through an autoregressive term and a similar approach may be explored here.

# Chapter 5

## Conclusions and Further Work

The contributions of this thesis focus on computational and modelling aspects of the latent space approach for network data. Chapter 3 considered estimation of temporally evolving latent space networks via SMC and Chapter 4 introduced a latent space model for hypergraph data in which interactions occur between sets of nodes. The methodology explored in this thesis can be extended in a variety of ways, some of which were outlined in the conclusions of Chapters 3 and 4, and in this chapter we highlight three possible directions for future work.

### **Improving computational scalability**

A key computational challenge associated with the DLSN model considered in Chapter 3 is the evaluation of the  $O(N^2)$  terms in the likelihood. Estimating the latent representation sequentially via SMC improves the scalability in terms of the number of observations in time  $T$ , however this reduces the scalability in terms of the number of nodes  $N$ . This is primarily due to characteristics of SMC methodology, but this issue is further compounded by the computational cost of evaluating the likelihood. To improve the scalability in terms of  $N$  we explored the high-dimensional SMC literature, and further improvements to the scalability may also be made by considering likelihood approximations to reduce the  $O(N^2)$  cost. In the existing literature, it is

typical to rely on likelihood approximations (see Raftery et al. (2012) and Rastelli et al. (2018)) to facilitate estimation of networks with larger  $N$  within an MCMC scheme. Usually, each latent coordinate is sampled conditional on all other latent coordinates via a MH-within-Gibbs update and, since the entire latent representation is estimated jointly within an SMC scheme, a direct application of existing likelihood approximations may not be appropriate. It is also important to highlight that, due to mechanics the of SMC, employing a likelihood approximation may present additional challenges that are not found within the MCMC context. Approximations within SMC have been considered for static models within Gunawan et al. (2018) and the adaption to the time varying case may not be straightforward.

Alternatively, we may explore a model-based approach to improving the scalability. For example, Fosdick et al. (2019) introduce the latent space SBM in which the within and between community connection probabilities are modelled via a latent space model and an Erdős-Rényi random graph, respectively. By partitioning the latent coordinates into distinct communities, the likelihood can be decomposed and the distance calculation between all  $\binom{N}{2}$  pairs can be avoided. Additionally, it may be possible to take advantage of the model structure to design a more efficient particle filter which accounts for the regions of independence.

In the hypergraph model introduced in Chapter 4, we avoided the  $O(N^k)$  likelihood terms implied by a construction analogous to that of Hoff et al. (2002). This was achieved by viewing the hypergraph as an Erdős-Rényi modification of a construction based on the Čech complex. This reduced the computational cost associated with the likelihood calculation, however evaluating the likelihood relies on calculating the Čech complex. The Vietoris-Rips (VR) complex (Vietoris (1927), Gromov (1987)) likely presents a more scalable alternative to the Čech complex, and this would be an interesting extension to explore. In the VR complex an order  $k$  hyperedge is present when each of the  $\binom{k}{2}$  balls of radius  $r$  intersect, meaning that no additional computational cost is required to determine all hyperedges once the order 2 hyperedges

are known. However, altering the underlying complex affects the properties of the resulting hypergraphs and an analysis of this should also be considered.

The hypergraph model presented in Chapter 4 can be adapted in many ways, similar to the extensions of the initial models of Hoff et al. (2002) discussed in Chapter 1. However, it is important that certain properties of the model are preserved when developing these extensions. For example, it is natural to include covariate information into the model so that nodes which share similar covariates are more likely to be connected. In the construction of Hoff et al. (2002), this can be achieved by including covariate information in the linear predictor. The analogous approach in the hypergraph setting would require a calculation over all order  $k$  sets, and this is computationally expensive. Therefore, this must be included in a way which maintains the computationally appealing properties of the likelihood.

### **Changepoint and anomaly detection**

When a series of interactions are observed through time, it is natural to ask whether or not the patterns of connectivity change. In the statistical literature this is referred to as changepoint detection, where the objective is to determine the points at which the characteristics of a time series change. This literature is well developed for general time series data (for example see Truong et al. (2018), Aminikhanghahi and Cook (2017) and references therein) and particular examples of changepoint detection for temporally evolving networks include Wang et al. (2017), De Ridder et al. (2016) and Ludkin et al. (2018). In the context of network data, changes may, for instance, relate to the density of the network, the tendency for specific nodes to form connections or the community memberships of each node. It would be interesting to consider this problem in the context of latent space network models. Park and Sohn (2018) develop a changepoint approach for networks using the tensor regression framework of Hoff (2011), and this is the most relevant existing methodology. We also note here that SMC presents a natural setting for online changepoint detection which has been



discussed in Heard and Turcotte (2017) which also presents an interesting direction for future work.

The related problem of anomaly detection (see Patcha and Park (2007)), in which the goal is to determine which observations are uncharacteristic of the overall time series, may also be explored. Recently, this has been considered in the latent space framework by Lee et al. (2019), where the authors develop a sequential variational Bayes approach to inference. Variational methods consider a computationally cheaper approximation to the target, whereas SMC methods do not. Hence, SMC presents an interesting alternative which facilitates online inference.

Here we have focused on the DLSN model from Chapter 3, but it is worth noting that the hypergraph model from Chapter 4 may also be extended to the dynamic setting and considered within the problem of changepoint and anomaly detection. However, it is important to note that this extension may not be entirely without practical challenges.

### **Modifying the underlying geometry**

The models considered in this thesis assume that the nodes can be appropriately represented in Euclidean space. This imposes properties on the resulting networks and, depending on the application, this may be a major limitation of this work. Several authors have explored the effect of modifying the underlying geometry within latent space network models (see Krioukov et al. (2010), Smith et al. (2017) and McCormick and Zheng (2015)). A particularly appealing choice is hyperbolic geometry since Krioukov et al. (2010) demonstrate that this naturally represents networks with power law degree distributions. Adapting the DLSN from Chapter 3 and the latent space hypergraph model from Chapter 4 to the hyperbolic setting both present interesting directions for future work. In the case of pairwise interactions, it is understood empirically that many real world networks exhibit power law degree distributions (Barabási and Pósfai (2016)) and it is interesting to explore whether an equivalent

behaviour can be found for interactions of arbitrary order. Adapting either of these models to the hyperbolic setting will present many practical challenges. For example, an approach for calculating the Čech complex with points in hyperbolic space must be considered when adapting the methodology from Chapter 4. Finally, developing an approach for posterior sampling in the Bayesian setting would also present an important contribution to the literature.

# Appendix A

## Appendix for ‘Sequential Monte Carlo and Dynamic Latent Space Networks’

### A.1 Gradient derivation for Euclidean distance and binary observations

Here we derive the gradient for the DLSN model given in (3.3.1) - (3.3.5) for binary data with a Euclidean metric. We wish to evaluate

$$\nabla_{\mathbf{U}_t} \log p(\mathcal{Y}_t | \mathbf{U}_t) = \nabla_{\mathbf{U}_t} \sum_{i < j} \{\eta_{ijt} y_{ijt} - \log(1 + e^{\eta_{ijt}})\} \quad (\text{A.1.1})$$

Since it is not possible to consider  $\nabla_{\mathbf{U}_t}$ , we focus on  $\nabla_{X_{kt}}$  so that each node is moved conditional on all other nodes in a gradient nudge step.

$$\nabla_{X_{it}} \log p(\mathcal{Y}_t | \mathbf{U}_t) = \sum_{i < j} \nabla_{X_{kt}} \{\eta_{ijt} y_{ijt} - \log(1 + e^{\eta_{ijt}})\} \quad (\text{A.1.2})$$

$$= \sum_{i < j} \nabla_{X_{kt}} \{(\alpha - \|X_{it} - X_{jt}\|) y_{ijt} - \log(1 + e^{\alpha - \|X_{it} - X_{jt}\|})\} \quad (\text{A.1.3})$$

To evaluate (A.1.3) we require

$$\nabla_{X_{it}} \|X_{it} - X_{jt}\| = \frac{(X_{it} - X_{jt})}{\|X_{it} - X_{jt}\|} \quad (\text{A.1.4})$$

and similarly

$$\nabla_{X_{jt}} \|X_{it} - X_{jt}\| = -\frac{(X_{it} - X_{jt})}{\|X_{it} - X_{jt}\|} = \frac{(X_{jt} - X_{it})}{\|X_{it} - X_{jt}\|} \quad (\text{A.1.5})$$

Now, we find

$$\nabla_{X_{kt}} \{(\alpha - \|X_{it} - X_{jt}\|)y_{ijt} - \log(1 + e^{\alpha - \|X_{it} - X_{jt}\|})\} \quad (\text{A.1.6})$$

$$\begin{aligned} &= \sum_{k < j} \left\{ -y_{kjt} \frac{X_{kt} - X_{jt}}{\|X_{kt} - X_{jt}\|} - \frac{-\frac{X_{kt} - X_{jt}}{\|X_{kt} - X_{jt}\|} e^{\alpha - \|X_{kt} - X_{jt}\|}}{1 + e^{\alpha - \|X_{kt} - X_{jt}\|}} \right\} \\ &\quad + \sum_{i < k} \left\{ y_{ikt} \frac{X_{it} - X_{kt}}{\|X_{it} - X_{kt}\|} - \frac{\frac{X_{it} - X_{kt}}{\|X_{it} - X_{kt}\|} e^{\alpha - \|X_{it} - X_{kt}\|}}{1 + e^{\alpha - \|X_{it} - X_{kt}\|}} \right\} \quad (\text{A.1.7}) \end{aligned}$$

$$\begin{aligned} &= \sum_{k < j} \frac{X_{kt} - X_{jt}}{\|X_{kt} - X_{jt}\|} \left\{ \frac{e^{\alpha - \|X_{kt} - X_{jt}\|}}{1 + e^{\alpha - \|X_{kt} - X_{jt}\|}} - y_{kjt} \right\} \\ &\quad - \sum_{i < k} \frac{X_{it} - X_{kt}}{\|X_{it} - X_{kt}\|} \left\{ \frac{e^{\alpha - \|X_{it} - X_{kt}\|}}{1 + e^{\alpha - \|X_{it} - X_{kt}\|}} - y_{ikt} \right\} \quad (\text{A.1.8}) \end{aligned}$$

$$= \sum_{j \neq k} \frac{X_{kt} - X_{jt}}{\|X_{kt} - X_{jt}\|} \left\{ \frac{e^{\alpha - \|X_{kt} - X_{jt}\|}}{1 + e^{\alpha - \|X_{kt} - X_{jt}\|}} - y_{kjt} \right\} \quad (\text{A.1.9})$$

where, in the last equality, we have used  $y_{ijt} = y_{jit}$ ,  $\|X_{it} - X_{jt}\| = \|X_{jt} - X_{it}\|$  and (A.1.5).

Hence, we obtain the following expression for the gradient.

$$\nabla_{X_{kt}} \log p(\mathcal{Y}_t | \mathbf{U}_t) = \sum_{j \neq k} \frac{X_{kt} - X_{jt}}{\|X_{kt} - X_{jt}\|} \left\{ \frac{e^{\alpha - \|X_{kt} - X_{jt}\|}}{1 + e^{\alpha - \|X_{kt} - X_{jt}\|}} - y_{kjt} \right\} \quad (\text{A.1.10})$$

Similar calculations can be made for a dot-product formulation.

## A.2 Gradient derivation for parameter estimation

To estimate  $\theta = (\alpha, \sigma)$  using the scheme detailed in Section 3.4.2, we must find expressions for  $\frac{d}{d\theta} \log p(\mathbf{U}_t | \mathbf{U}_{t-1}, \theta)$  and  $\frac{d}{d\theta} \log p(\mathcal{Y}_t | \mathbf{U}_t, \theta)$ . For the model given in

(3.3.1) - (3.3.5), we obtain

$$\frac{\partial}{\partial \alpha} \log p(\mathcal{Y}_t | \mathbf{U}_t, \theta) = \frac{\partial}{\partial \alpha} \sum_{i < j} \{(\alpha - \|u_{it} - u_{jt}\|)y_{ijt} - \log(1 + \exp(\alpha - \|u_{it} - u_{jt}\|))\} \quad (\text{A.2.1})$$

$$= \sum_{i < j} \left\{ y_{ijt} - \frac{1}{1 + \exp(\|u_{it} - u_{jt}\| - \alpha)} \right\} \quad (\text{A.2.2})$$

$$= \sum_{i < j} (y_{ijt} - p_{ijt}) \quad (\text{A.2.3})$$

$$\frac{\partial}{\partial \sigma} \log p(\mathcal{Y}_t | \mathbf{U}_t, \theta) = 0 \quad (\text{A.2.4})$$

$$\frac{\partial}{\partial \alpha} \log p(\mathbf{U}_t | \mathbf{U}_{t-1}, \theta) = 0 \quad (\text{A.2.5})$$

$$\begin{aligned} \frac{\partial}{\partial \sigma} \log p(\mathbf{U}_t | \mathbf{U}_{t-1}, \theta) &= \frac{\partial}{\partial \sigma} \sum_{i=1}^N \left\{ \frac{d}{2} \log(2\pi) - \frac{1}{2} \log(|\sigma^2 I_d|) \right. \\ &\quad \left. - \frac{1}{2} (u_{it} - u_{i,t-1})^T (\sigma^2 I_d)^{-1} (u_{it} - u_{i,t-1}) \right\} \end{aligned} \quad (\text{A.2.6})$$

$$= \sum_{i=1}^N \left\{ -\frac{d}{\sigma} + \frac{1}{\sigma^3} (u_{it} - u_{i,t-1})^T (u_{it} - u_{i,t-1}) \right\} \quad (\text{A.2.7})$$

Since  $\sigma > 0$ , we opt to estimate  $\log(\sigma)$ . We obtain  $\frac{\partial}{\partial \log(\sigma)} \log p(\mathbf{U}_t | \mathbf{U}_{t-1}, \theta)$  by multiplying (A.2.7) by  $\sigma$ . This follows from the chain rule.

### A.3 Data simulation for Section 3.5.1

The data were simulated according to scenarios (S1), (S2) and (S3) by simulating the latent trajectories according to

$$p(x_t) = \lambda p(x_{t-1}) + (1 - \lambda)g_t \quad (\text{A.3.1})$$

where  $g_t$  is the  $t^{\text{th}}$  time step of a deterministic function. We specify these functions so that they exhibit the behaviours described in scenarios (S1), (S2) and (S3).

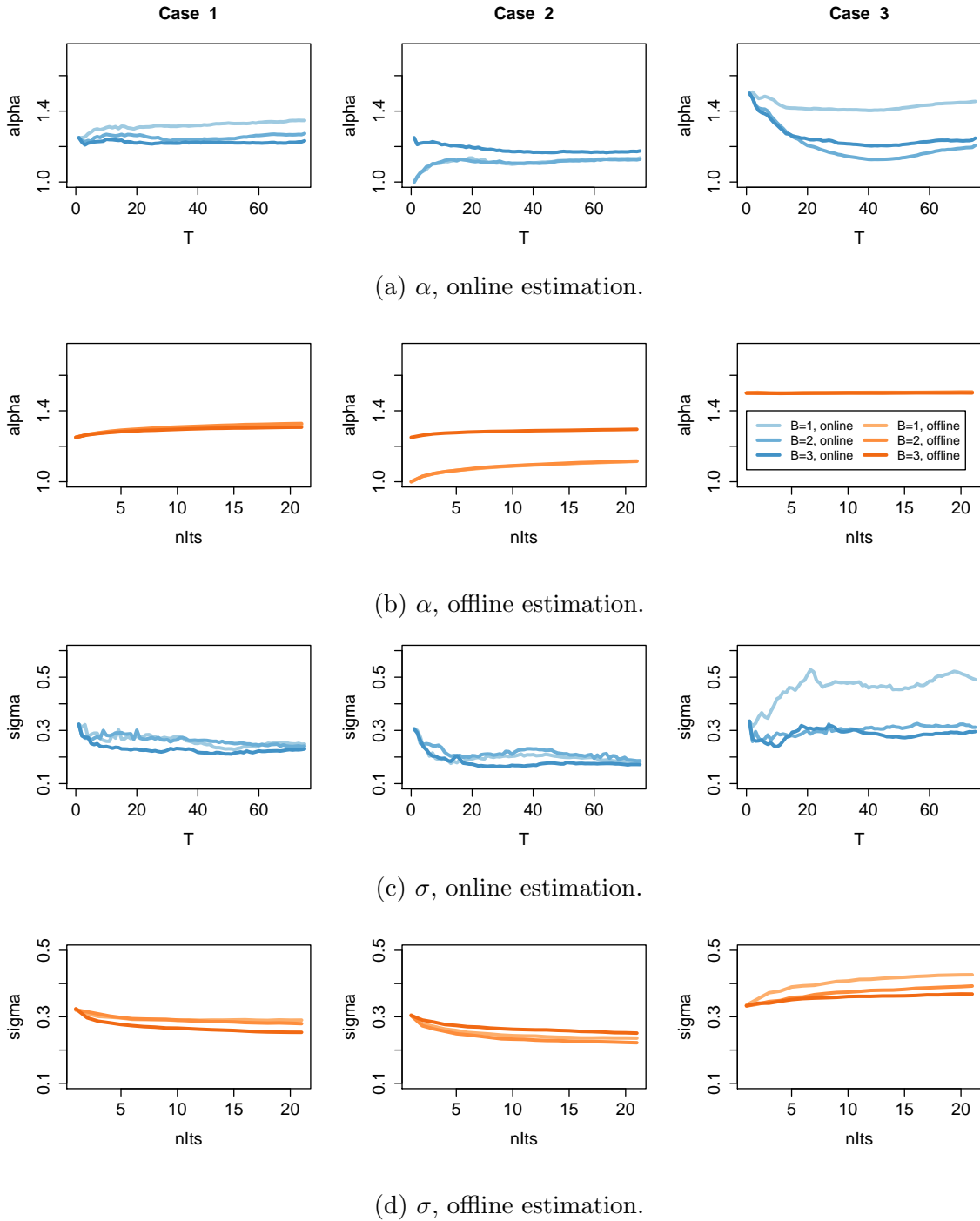


Figure A.3.1: Summary of parameter estimates.

# Appendix B

## Appendix for ‘Latent Space Modelling of Hypergraph Data’

### B.1 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 B.1.1 we describe the Bookstein coordinates in  $\mathbb{R}^2$ , and in Appendix B.1.2 we describe the Bookstein coordinates in  $\mathbb{R}^3$ .

#### B.1.1 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  $\mathbf{U}^B$  denote the Bookstein coordinates and  $\mathbf{U}$  denote the untransformed coordinates. Then  $\mathbf{U}^B$  is given by

$$\begin{aligned} \mathbf{U}^B &= cR(\mathbf{U} - \mathbf{b}) \\ &= \frac{1}{\sqrt{(u_{21}^B - u_{11}^B)^2 + (u_{22}^B - u_{12}^B)^2}} \begin{bmatrix} \cos(a) & \sin(a) \\ -\sin(a) & \cos(a) \end{bmatrix} \left( \mathbf{U} - \frac{1}{2} \begin{bmatrix} u_{11}^B + u_{21}^B \\ u_{12}^B + u_{22}^B \end{bmatrix} \right), \end{aligned} \tag{B.1.1}$$

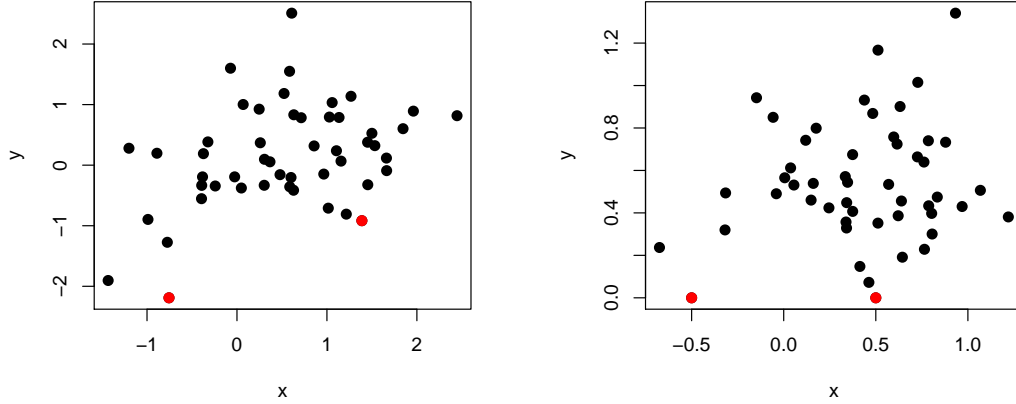


Figure B.1.1: Bookstein transformation in  $\mathbb{R}^2$ . Left: original coordinates. Right: transformed Bookstein coordinates. The points highlighted in red are mapped to  $(-1/2, 0)$  and  $(1/2, 0)$ .

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 rescaling of the coordinates  $\mathbf{U}$ . Figure B.1.1 shows an example of the Bookstein transformation in  $\mathbb{R}^2$ .

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

$$\mu^B = cR(\mu - b), \quad (\text{B.1.2})$$

$$\Sigma^B = c^2R\Sigma R^T. \quad (\text{B.1.3})$$

### B.1.2 Bookstein coordinates in $\mathbb{R}^3$

Section 4.3.3 of Dryden and Mardia (1998) gives the Bookstein transformation for  $\mathbf{U}$  where  $u_i \in \mathbb{R}^3$ . Following from (B.1.1) 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  $i = 4, 5, \dots, N$  and  $l = 1, 2, 3$  we calculate

$$w_{il} = u_{il} - \frac{1}{2}(u_{1l}^B + u_{2l}^B). \quad (\text{B.1.4})$$

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

$$u_i^B = R_1 R_2 R_3 (w_{i1}, w_{i2}, w_{i3}) / D_{12} \quad (\text{B.1.5})$$



where

$$R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix}, \quad R_2 = \begin{bmatrix} \cos(\omega) & 0 & \sin(\omega) \\ 0 & 1 & 0 \\ -\sin(\omega) & 0 & \cos(\omega) \end{bmatrix} \quad (\text{B.1.6})$$

$$R_3 = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad D_{12} = 2(w_{21}^2 + w_{22}^2 + w_{23}^2)^{1/2}. \quad (\text{B.1.7})$$

Furthermore, we have

$$\theta = \arctan(w_{22}/w_{21}) \quad (\text{B.1.8})$$

$$\omega = \arctan(w_{23}/(w_{21}^2 + w_{22}^2)^{1/2}) \quad (\text{B.1.9})$$

$$\phi = \arctan\left(\frac{(w_{21}^2 + w_{22}^2)w_{33} - (w_{21}w_{31} + w_{22}w_{32})w_{23}}{(w_{21}^2 + w_{22}^2 + w_{23}^2)^{1/2}(w_{21}w_{32} - w_{31}w_{22})}\right). \quad (\text{B.1.10})$$

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.2 Modifying the Hyperedge Indicators

In the generative model detailed in Algorithm 7, the indicators for all order  $k$  hyperedges are modified with probability  $\varphi_k$ . Since the probability  $\varphi_k$  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( $\varphi_k$ ) random variable for each hyperedge is 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 8.

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  $\{i_1 < i_2 < \dots < i_k | i_1, i_2, \dots, i_k \in \{1, 2, \dots, N\}\}$  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  $p(i_1) = \frac{N - i_1}{\sum_{i=1}^{N-1} (N - i)}$ , for  $i_1 = 1, \dots, (N - 1)$ .
2. Sample  $i_2 | i_1$  with probability  $p(i_2 | i_1) = \frac{1}{N - i_1}$ , for  $i_2 = (i_1 + 1), \dots, N$ .

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.

## B.3 Conditional Posterior Distributions

### B.3.1 Conditional posterior for $\mu$

The conditional posterior for  $\mu$  is given by

$$p(\mu | \mathbf{U}, \Sigma, m_\mu, \Sigma_\mu) \propto p(\mathbf{U} | \mu, \Sigma) p(\mu | m_\mu, \Sigma_\mu) \quad (\text{B.3.1})$$

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

We have

$$p(\mu | \mathbf{U}, \Sigma, m_\mu, \Sigma_\mu) \propto \exp \left\{ -\frac{1}{2} \sum_{i=1}^N (u_i - \mu)^T \Sigma^{-1} (u_i - \mu) - \frac{1}{2} (\mu - m_\mu)^T \Sigma_\mu^{-1} (\mu - m_\mu) \right\} \quad (\text{B.3.2})$$

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

$$p(\mu|\mathbf{U}, \Sigma, m_\mu, \Sigma_\mu) = \mathcal{N} \left( (N\Sigma^{-1} + \Sigma_\mu^{-1})^{-1} \left( \Sigma^{-1} \sum_{i=1}^N u_i + \Sigma_\mu^{-1} m_\mu \right), (N\Sigma^{-1} + \Sigma_\mu^{-1})^{-1} \right). \quad (\text{B.3.3})$$

### B.3.2 Conditional posterior for $\Sigma$

The conditional posterior for  $\Sigma$  is given by

$$p(\Sigma|\mathbf{U}, \mu, \Phi, \nu) \propto p(\mathbf{U}|\mu, \Sigma)p(\Sigma|\Phi, \nu) \quad (\text{B.3.4})$$

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

We have

$$p(\Sigma|\mathbf{U}, \mu, \Phi, \nu) \propto \prod_{i=1}^N \frac{1}{|\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} (u_i - \mu)^t \Sigma^{-1} (u_i - \mu) \right\} |\Sigma|^{-(\nu+d+1)/2} \exp \left\{ -\frac{1}{2} \text{tr} (\Phi \Sigma^{-1}) \right\} \quad (\text{B.3.5})$$

$$\propto |\Sigma|^{-(\nu+d+N+1)/2} \exp \left\{ -\frac{1}{2} \left[ \text{tr} \left( \Sigma^{-1} \sum_{i=1}^N (u_i - \mu)(u_i - \mu)^T \right) + \text{tr}(\Phi \Sigma^{-1}) \right] \right\} \quad (\text{B.3.6})$$

$$\propto |\Sigma|^{-(\nu+d+N+1)/2} \exp \left\{ -\frac{1}{2} \text{tr} \left( \left[ \sum_{i=1}^N (u_i - \mu)(u_i - \mu)^T + \Phi \right] \Sigma^{-1} \right) \right\}. \quad (\text{B.3.7})$$

Line (B.3.7) follows from the symmetry of  $\Sigma$  and  $\sum_{i=1}^N (v_i - \mu)(v_i - \mu)^T$ , and properties of the trace operator.

Hence, we obtain

$$p(\Sigma|\mathbf{U}, \mu, \Phi, \nu) = \mathcal{W}^{-1} \left( \Phi + \sum_{i=1}^N (u_i - \mu)(u_i - \mu)^T, \nu + N \right). \quad (\text{B.3.8})$$

### B.3.3 Conditional posterior for $\psi_k^{(0)}$

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

$$p\left(\psi_k^{(0)}|\mathbf{U}, \mathbf{r}, h_{N,K}, a_k^{(0)}, b_k^{(0)}\right) \propto \mathcal{L}\left(\mathbf{U}, \mathbf{r}, \boldsymbol{\psi}^{(1)}, \boldsymbol{\psi}^{(0)}; h_{N,K}\right) p\left(\psi_k^{(0)}|a_k^{(0)}, b_k^{(0)}\right) \quad (\text{B.3.9})$$

where  $\mathcal{L}\left(\mathbf{U}, \mathbf{r}, \boldsymbol{\psi}^{(1)}, \boldsymbol{\psi}^{(0)}; h_{N,K}\right)$  is as in (4.3.6) 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

$$\begin{aligned} & p(\psi_k^{(0)}|\mathbf{U}, \mathbf{r}, h_{N,K}, a_k^{(0)}, b_k^{(0)}) \\ & \propto \left(\psi_k^{(0)}\right)^{d_k^{(01)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K})} \left(1 - \psi_k^{(0)}\right)^{d_k^{(00)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K})} \left(\psi_k^{(0)}\right)^{a_k^{(0)}-1} \left(1 - \psi_k^{(0)}\right)^{b_k^{(0)}-1} \end{aligned} \quad (\text{B.3.10})$$

$$\propto \left(\psi_k^{(0)}\right)^{d_k^{(01)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K})+a_k^{(0)}-1} \left(1 - \psi_k^{(0)}\right)^{d_k^{(00)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K})+b_k^{(0)}-1}. \quad (\text{B.3.11})$$

Hence, we obtain

$$\begin{aligned} & p(\psi_k^{(0)}|\mathbf{U}, \mathbf{r}, h_{N,K}, a_k^{(0)}, b_k^{(0)}) \\ & = \text{Beta}\left(d_k^{(01)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K}) + a_k^{(0)}, d_k^{(00)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K}) + b_k^{(0)}\right). \end{aligned} \quad (\text{B.3.12})$$

### B.3.4 Conditional posterior for $\psi_k^{(1)}$

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

$$p\left(\psi_k^{(1)}|\mathbf{U}, \mathbf{r}, h_{N,K}, a_k^{(1)}, b_k^{(1)}\right) \propto \mathcal{L}\left(\mathbf{U}, \mathbf{r}, \boldsymbol{\psi}^{(1)}, \boldsymbol{\psi}^{(0)}; h_{N,K}\right) p\left(\psi_k^{(1)}|a_k^{(1)}, b_k^{(1)}\right) \quad (\text{B.3.13})$$

where  $\mathcal{L}\left(\mathbf{U}, \mathbf{r}, \boldsymbol{\psi}^{(1)}, \boldsymbol{\psi}^{(0)}; h_{N,K}\right)$  is as in (4.3.6) 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

$$\begin{aligned} & p(\psi_k^{(1)}|\mathbf{U}, \mathbf{r}, h_{N,K}, a_k^{(1)}, b_k^{(1)}) \\ & \propto \left(\psi_k^{(1)}\right)^{d_k^{(10)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K})} \left(1 - \psi_k^{(1)}\right)^{d_k^{(11)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K})} \left(\psi_k^{(1)}\right)^{a_k^{(1)}-1} \left(1 - \psi_k^{(1)}\right)^{b_k^{(1)}-1} \end{aligned} \quad (\text{B.3.14})$$

$$\propto \left(\psi_k^{(1)}\right)^{d_k^{(10)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K})+a_k^{(1)}-1} \left(1 - \psi_k^{(1)}\right)^{d_k^{(11)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K})+b_k^{(1)}-1}. \quad (\text{B.3.15})$$

Hence, we obtain

$$\begin{aligned} p(\psi_k^{(1)} | \mathbf{U}, \mathbf{r}, h_{N,K}, a_k^{(1)}, b_k^{(1)}) \\ = \text{Beta} \left( d_k^{(10)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K}) + a_k^{(1)}, d_k^{(11)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K}) + b_k^{(1)} \right). \end{aligned} \tag{B.3.16}$$

## B.4 MCMC initialisation

For the MCMC scheme in Algorithm 9, 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  $\mathbf{U}$ .

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 that 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  $\mathbf{U}$  we then transform these coordinates onto the Bookstein space of coordinates. Our initialisation procedure for  $\mathbf{U}$  is given in Algorithm 10.

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

To initialise the parameters  $\mu$  and  $\Sigma$  we use an ABC scheme (see Marin et al. (2012) for an overview). In this scheme we first sample  $\mu$  and  $\Sigma$  from their priors.

---

**Algorithm 10** Initialise  $\mathbf{U}$ .

---

**Input:** Observed hypergraph  $h_{N,K}$

1) Let  $A \in \mathbb{R}^{N \times N}$  denote a weighted adjacency matrix.

For  $i, j \in \{1, 2, \dots, N\}$ , 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  $\mathbf{U}_0 \in \mathbb{R}^{N \times d}$ .

4) Specify the index of the anchor points, and transform  $\mathbf{U}_0$  onto Bookstein coordinates (see Appendix B.1).

---

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  $\mu$  and  $\Sigma$ . 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 11 and the full initialisation scheme is given in Algorithm 12.

## B.5 Practicalities

To implement the MCMC scheme given in Algorithm 9 there are a number of practical considerations we must address. In this section we comment on these where, in B.5.1 we discuss an approach for determining the presence of a hyperedge of arbitrary order and in B.5.2 we discuss efficient evaluation of the likelihood.

---

**Algorithm 11** Initialise  $\mu$  and  $\Sigma$ .

---

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

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  $i = 1, 2, \dots, N$ .

Let  $\mathbf{U}^*$  be the  $N \times d$  matrix whose  $i^{\text{th}}$  row is  $u_i^*$ .

-Given initial  $\mathbf{r}_0$ , determine the hypergraph  $g_{N,K}(\mathbf{U}^*, \mathbf{r}_0)$ .

-Let  $g_{N,K}^*$  be the hypergraph obtained by modifying  $g_{N,K}(\mathbf{U}^*, \mathbf{r}_0)$  with noise  $\boldsymbol{\psi}_0^{(0)}$  and  $\boldsymbol{\psi}_0^{(1)}$

-Calculate  $T(g_{N,K}^*)$ .

-If  $|T(h_{N,K}) - T(g_{N,K}^*)| < \epsilon$

Accept samples  $\mu^*$  and  $\Sigma^*$ .

3) Let  $\mu_0$  and  $\Sigma_0$  be the average of  $N_{smp}$  samples.

---

### B.5.1 Smallest Enclosing Ball

Here we discuss an approach for determining the presence of a hyperedge conditional on  $\mathbf{U}$  and  $\mathbf{r}$ . Recall that a hyperedge  $e_k = \{i_1, i_2, \dots, i_k\}$  is present if  $B_{r_k}(u_{i_1}) \cap B_{r_k}(u_{i_2}) \cap \dots \cap B_{r_k}(u_{i_k}) \neq \emptyset$ . 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  $\{u_{i_1}, u_{i_2}, \dots, u_{i_k}\}$  are contained within a ball of radius  $r_k$  (see section 3.2 of Edelsbrunner and Harer (2010)). Figure B.5.1 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  $\{u_{i_1}, u_{i_2}, \dots, u_{i_k}\}$ .

---

**Algorithm 12** Procedure for initialising MCMC scheme in Algorithm 9.

---

**Input** Observed hypergraph  $h_{N,K}$

- 1) Determine  $\mathbf{U}_0$  by applying Algorithm 10.
  - 2) Let initial radii  $\mathbf{r}_0$  be the smallest radii which induce all hyperedges observed in  $h_{N,K}$ , conditional on  $\mathbf{U}_0$ .
  - 3) Sample  $\psi_0^{(0)}$  and  $\psi_0^{(1)}$  from their prior distributions.
  - 4) Sample  $\mu_0$  and  $\Sigma_0$  by applying Algorithm 11.
- 

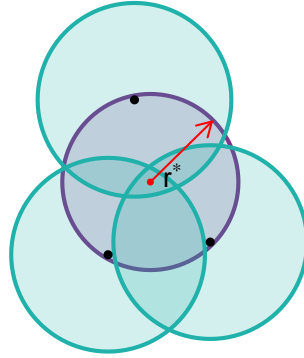


Figure B.5.1: The blue shaded regions correspond to  $B_r(u_i)$ , for  $i = 1, 2, 3$ , and the purple shaded region is the smallest enclosing ball of the points. The statements  $r^* < r$  and  $B_r(u_1) \cap B_r(u_2) \cap B_r(u_3) \neq \emptyset$  are equivalent.

2. If the radius of  $B$  is less than  $r_k$ , the hyperedge  $e_k = \{i_1, i_2, \dots, i_k\}$  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  $u_{i_1 \dots i_k} = \{u_{i_1}, u_{i_2}, \dots, u_{i_k}\}$ . If  $u$  lies within the smallest enclosing ball of  $u_{i_1 \dots i_k} \setminus u$ , 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 4.2.3). 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  $e_k = \{i_1, i_2, \dots, i_k\}$ , we let  $\sigma_1 \subseteq e_k$  and  $\sigma_2 \subseteq e_k$  denote subsets which partition  $e_k$  so that  $\sigma_1 \cap \sigma_2 = \emptyset$ . After a pass of the algorithm, the set  $\sigma_2$  will contain the index of nodes which lie on the boundary of the smallest enclosing ball  $B$ . Hence,  $\sigma_2$  represents the nodes which determine  $B$ . We initialise the miniball algorithm with  $\sigma_1 = e_k$  and  $\sigma_2$ , and iteratively identify which nodes from  $\sigma_1$  belong in  $\sigma_2$ . The procedure is given in Algorithm 13.

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)).

### B.5.2 Evaluating $\mathcal{L}(\mathbf{U}, \mathbf{r}, \boldsymbol{\psi}^{(1)}, \boldsymbol{\psi}^{(0)}; h_{N,K})$

The likelihood given in (4.3.6) requires the enumeration of hyperedge discrepancies between the observed hypergraph  $h_{N,K}$  and the induced hypergraph  $g_{N,K}(\mathbf{U}, \mathbf{r})$ . In this section we note that this does not require a summation over all possible hyper-

**Algorithm 13** Miniball

- 
- 1) Set  $\sigma_1 = e_k$  and  $\sigma_2 = \emptyset$
  - 2) **if**  $\sigma_1 = \emptyset$ , compute the miniball  $B$  of  $\sigma_2$ 
    - else** choose  $u \in \sigma_1$ 
      - Calculate the miniball  $B$  which contains the points  $\sigma_1 \setminus u$  in its interior and the points  $\sigma_2$  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
- 

edges, and so can be computed far more efficiently than (4.3.5) suggests. We first discuss evaluation of (4.3.6), and then observe that (4.3.3) 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  $g_{N,K}(\mathbf{U}, \mathbf{r})$ . In practice, as the data examples from Section 4.8 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  $g_{N,K}(\mathbf{U}, \mathbf{r})$ , respectively. To evaluate the likelihood, we begin by enumerating the number of order  $k$  hyperedges which are present in both hypergraphs to obtain  $d_k^{(11)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K})$ . This can easily be computed by evaluating the number of intersection between the hyperedges in  $h_{N,K}$  and  $g_{N,K}(\mathbf{U}, \mathbf{r})$ . Then, for  $k = 2, 3, \dots, K$ , it follows that

$$d_k^{(10)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K}) = n_k^{(g)} - d_k^{(11)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K}), \quad (\text{B.5.1})$$

$$d_k^{(01)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K}) = n_k^{(h)} - d_k^{(11)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K}), \quad (\text{B.5.2})$$

$$d_k^{(00)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K}) = \binom{N}{k} - \left[ d_k^{(11)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K}) + d_k^{(10)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K}) + d_k^{(01)}(g_{N,K}(\mathbf{U}, \mathbf{r}), h_{N,K}) \right]. \quad (\text{B.5.3})$$

Hence, we are able to avoid summation over all possible hyperedges. We can easily

calculate the distance specified in (4.3.2) from the above, by observing that it is given by the sum of (B.5.1) and (B.5.2).

## B.6 Proofs for Section 4.6

### B.6.1 Proof of Proposition 4.6.1

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

$$p_{e_2} = P(\|U_i - U_j\| \leq 2r_2 | \mu, \Sigma) = P((U_i - U_j)^T (U_i - U_j) \leq 4(r_2)^2 | \mu, \Sigma). \quad (\text{B.6.1})$$

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

$$X_{ij}^T X_{ij} | \Sigma = \sum_{l=1}^d \lambda_l \chi_1^2 \quad (\text{B.6.2})$$

where  $\lambda_l$  is the  $l^{\text{th}}$  eigenvalue of  $2\Sigma$ . Since  $\Sigma$  is diagonal, the eigenvalues of  $2\Sigma$  are given by  $\lambda_l = 2\sigma_l^2$ . Furthermore, since a  $\chi_1^2$  distribution is equivalent to a  $\Gamma(1/2, 2)$  distribution, we have

$$Z_{ij} | \Sigma = X_{ij}^T X_{ij} | \Sigma \sim \sum_{l=1}^d \Gamma\left(\frac{1}{2}, 2(2\sigma_l^2)\right). \quad (\text{B.6.3})$$

Hence, we have the result.

### B.6.2 Proof of Lemma 4.6.1

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

$$P(y_{e_k} = 1^{(g^*)} | \Sigma) = (1 - \varphi_k)P(y_{e_k}^{(g)} = 1 | \Sigma) + \varphi_k(1 - P(y_{e_k}^{(g)} = 1 | \Sigma)). \quad (\text{B.6.4})$$

The degree of the  $i^{\text{th}}$  node with respect to order  $k$  hyperedges is obtained by summing over all possible hyperedges  $e_k$  that are incident to  $i$ . We have

$$\text{Deg}_{(i,k)}^g = \sum_{\{e_k \in \mathcal{E}_{N,k} | i \in e_k\}} y_{e_k}^{(g)} \quad (\text{B.6.5})$$

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

$$\mathbb{E} \left[ \text{Deg}_{(i,k)}^g | \Sigma \right] = \binom{N-1}{k-1} P(y_{e_k} = 1^{(g)} | \Sigma). \quad (\text{B.6.6})$$

Note that (B.6.6) is valid for dependent probabilities and we rely on this for hyperedges of order  $k \geq 3$ . 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  $\text{Deg}_{(i,k)}^g$ .

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

$$\mathbb{E} \left[ \text{Deg}_{(i,k)}^{g^*} | \varphi_k, \Sigma \right] = \mathbb{E} \left[ \mathbb{E} \left[ \text{Deg}_{(i,k)}^{g^*} | \text{Deg}_{(i,k)}^g \right] | \varphi_k, \Sigma \right] \quad (\text{B.6.7})$$

From (B.6.4), it follows that

$$\mathbb{E} \left[ \text{Deg}_{(i,k)}^{g^*} | \varphi_k, \Sigma \right] = \mathbb{E} \left[ (1 - \varphi_k) \text{Deg}_{(i,k)}^g + \varphi_k \left( \binom{N-1}{k-1} - \text{Deg}_{(i,k)}^g \right) | \varphi_k, \Sigma \right] \quad (\text{B.6.8})$$

$$= (1 - \varphi_k) \mathbb{E} \left[ \text{Deg}_{(i,k)}^g | \varphi_k, \Sigma \right] + \varphi_k \left( \binom{N-1}{k-1} - \mathbb{E} \left[ \text{Deg}_{(i,k)}^g | \varphi_k, \Sigma \right] \right). \quad (\text{B.6.9})$$

Using (B.6.5), we obtain

$$\mathbb{E} \left[ \text{Deg}_{(i,k)}^{g^*} | \varphi_k, \mu, \Sigma \right] = \binom{N-1}{k-1} \left[ (1 - \varphi_k) P(y_{e_k} = 1^{(g)} | \mu, \Sigma) + \varphi_k (1 - P(y_{e_k}^{(g)} = 1 | \mu, \Sigma)) \right]. \quad (\text{B.6.10})$$

The final result then follows directly from Observation (O2).

### B.6.3 Proof of Theorem 4.6.1

To begin we will consider the degree distribution of hyperedges of order  $k = 2$ . From (B.6.4) we have that the probability of an edge  $e_2$  being present in  $g^*$  is given by

$$P(y_{e_2}^{(g^*)} = 1 | \Sigma, r_2, \varphi_2) = (1 - \varphi_2)p_{e_2} + \varphi_2(1 - p_{e_2}), \quad (\text{B.6.11})$$

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

The degree distribution of the  $i^{\text{th}}$  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

$$\text{Deg}_{(i,2)}^{g^*} | r_2, \varphi_2, \Sigma \sim \text{Binomial}(N - 1, (1 - \varphi_2)p_{e_2} + \varphi_2(1 - p_{e_2})). \quad (\text{B.6.12})$$

The degree of the  $i^{\text{th}}$  node for order  $k$  hyperedges in  $g^*$  is given by

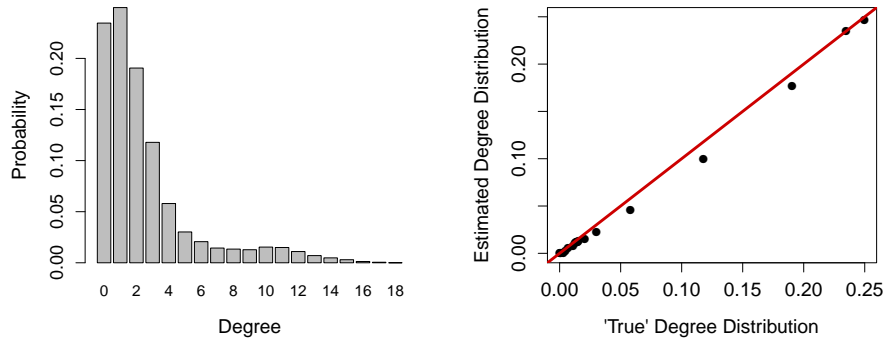
$$\text{Deg}_{(i,k)}^{g^*} = \sum_{\{e_k \in \mathcal{E}_{N,k} | i \in e_k\}} y_{e_k}^{g^*}. \quad (\text{B.6.13})$$

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  $P(y_{ijk}^{(g)} = 1)$  and  $P(y_{ijl}^{(g)} = 1)$  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  $\lambda = \sum_{\{e_k \in \mathcal{E}_{N,3} | i \in e_k\}} y_{e_k}^{(g^*)}$ . However, we note that this result can be extended using results presented in Teerapabolarn (2014).

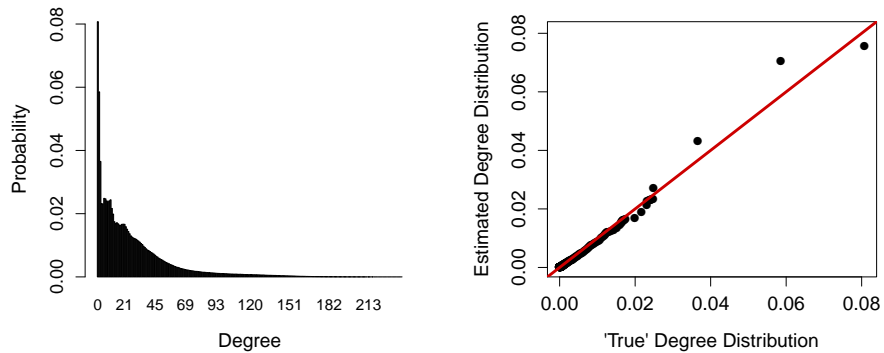
## B.7 Prior and Posterior Predictive Degree Distributions

Here we present additional plots for the simulation study detailed in Section 4.7.2. Figure B.7.1 compares the predictives for the connections between the observed nodes

and the newly simulated nodes, and Figure B.7.2 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 B.7.1: 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. In each figure, the left panel shows the prior predictive degree distribution, and the right panel shows a qq-plot of the prior and posterior predictive degree distributions. Figures B.7.1a and B.7.1b show the degree distributions for hyperedges of order 2 and 3, respectively.

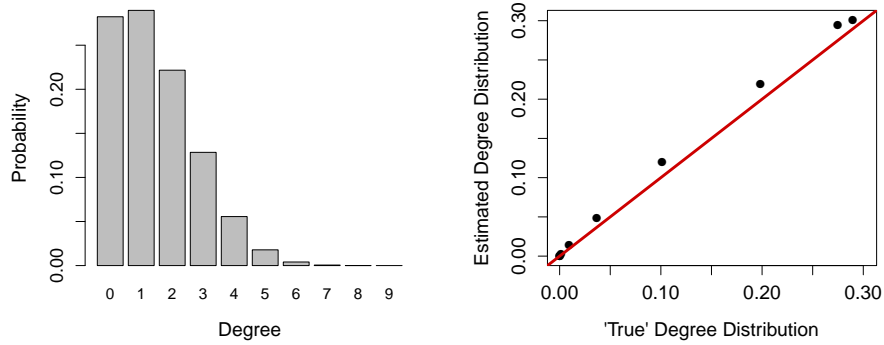
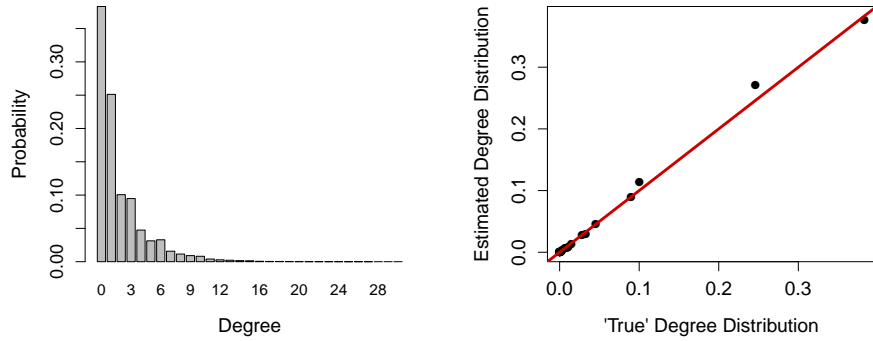
(a) Predictive degree distributions for hyperedges of order  $k = 2$ .(b) Predictive degree distributions for hyperedges of order  $k = 3$ .

Figure B.7.2: Comparison of prior and posterior predictive degree distributions for hyperedges occurring between the  $N^* = 10$  newly simulated nodes only. In each figure, the left panel shows the prior predictive degree distribution, and the right panel shows a qq-plot of the prior and posterior predictive degree distributions. Figures B.7.2a and B.7.2b show the degree distributions for hyperedges of order 2 and 3, respectively.

# Bibliography

- Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. (2009). Mixed membership stochastic blockmodels. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 21*, pages 33–40. Curran Associates, Inc.
- Aksoy, S., Kolda, T. G., and Pinar, A. (2016). Measuring and Modeling Bipartite Graphs with Community Structure. *arXiv e-prints*, page arXiv:1607.08673.
- Akyildiz, Ö. D. and Míguez, J. (2019). Nudging the particle filter. *Statistics and Computing*.
- Aldous, D. J. (1981). Representations for partially exchangeable arrays of random variables. *Journal of Multivariate Analysis*, 11(4):581 – 598.
- Aminikhanghahi, S. and Cook, D. J. (2017). A survey of methods for time series change point detection. *Knowledge and Information Systems*, 51(2):339–367.
- Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342.
- Athreya, A., Fishkind, D. E., Tang, M., Priebe, C. E., Park, Y., Vogelstein, J. T., Levin, K., Lyzinski, V., Qin, Y., and Sussman, D. L. (2017). Statistical inference on random dot product graphs: a survey. *Journal of Machine Learning Research*, 18:226:1–226:92.



- Avella-Medina, M., Parise, F., Schaub, M., and Segarra, S. (2018). Centrality measures for graphons: accounting for uncertainty in networks. *IEEE Transactions on Network Science and Engineering*.
- 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.
- Bengtsson, T., Bickel, P., Li, B., et al. (2008). Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. In *Probability and statistics: Essays in honor of David A. Freedman*, pages 316–334. Institute of Mathematical Statistics.
- Benson, A. R., Abebe, R., Schaub, M. T., Jadbabaie, A., and Kleinber, J. (2018). Simplicial closure and higher-order link prediction. *Proceedings of the National Academy of Sciences*, 115(48):E11221–E11230.
- 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.
- Beskos, A., Crisan, D., and Jasra, A. (2014a). On the stability of sequential monte carlo methods in high dimensions. *The Annals of Applied Probability*, 24(4):1396–1445.
- Beskos, A., Crisan, D., Jasra, A., Kamatani, K., and Zhou, Y. (2017). A stable particle filter for a class of high-dimensional state-space models. *Advances in Applied Probability*, 49(1):2448.
- Beskos, A., Crisan, D. O., Jasra, A., and Whiteley, N. (2014b). Error bounds and

- normalising constants for sequential monte carlo samplers in high dimensions. *Advances in Applied Probability*, 46(1):279–306.
- Bickel, P., Li, B., and Bengtsson, T. (2008). *Sharp failure rates for the bootstrap particle filter in high dimensions*, volume Volume 3 of *Collections*, pages 318–329. Institute of Mathematical Statistics, Beachwood, Ohio, USA.
- Biswal, B. B., Mennes, M., Zuo, X.-N., Gohel, S., Kelly, C., Smith, S. M., Beckmann, C. F., Adelstein, J. S., Buckner, R. L., Colcombe, S., Dogonowski, A.-M., Ernst, M., Fair, D., Hampson, M., Hoptman, M. J., Hyde, J. S., Kiviniemi, V. J., Kötter, R., Li, S.-J., Lin, C.-P., Lowe, M. J., Mackay, C., Madden, D. J., Madsen, K. H., Margulies, D. S., Mayberg, H. S., McMahon, K., Monk, C. S., Mostofsky, S. H., Nagel, B. J., Pekar, J. J., Peltier, S. J., Petersen, S. E., Riedl, V., Rombouts, S. A. R. B., Rypma, B., Schlaggar, B. L., Schmidt, S., Seidler, R. D., Siegle, G. J., Sorg, C., Teng, G.-J., Veijola, J., Villringer, A., Walter, M., Wang, L., Weng, X.-C., Whitfield-Gabrieli, S., Williamson, P., Windischberger, C., Zang, Y.-F., Zhang, H.-Y., Castellanos, F. X., and Milham, M. P. (2010). Toward discovery science of human brain function. *Proceedings of the National Academy of Sciences*, 107(10):4734–4739.
- Bloem-Reddy, B. and Orbanz, P. (2018). Random-walk models of network formation and sequential monte carlo methods for graphs. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(5):871–898.
- Bookstein, F. L. (1986). Size and shape spaces for landmark data in two dimensions. *Statist. Sci.*, 1(2):181–222.
- Borg, I. and Groenen, P. (1997). *Modern Multidimensional Scaling: Theory and Applications*. Springer series in statistics. Springer.
- Borgs, C. and Chayes, J. (2017). Graphons: A nonparametric method to model, estimate, and design algorithms for massive networks. In *Proceedings of the 2017*

- ACM Conference on Economics and Computation*, EC '17, pages 665–672, New York, NY, USA. ACM.
- Brockwell, A., Del Moral, P., and Doucet, A. (2012). Sequentially interacting Markov chain Monte Carlo methods. *ArXiv e-prints*.
- Cai, D., Campbell, T., and Broderick, T. (2016). Edge-exchangeable graphs and sparsity. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 4249–4257. Curran Associates, Inc.
- Campbell, T., Cai, D., and Broderick, T. (2018). Exchangeable trait allocations. *Electron. J. Statist.*, 12(2):2290–2322.
- Cappé, O. (2011). Online em algorithm for hidden markov models. *Journal of Computational and Graphical Statistics*, 20(3):728–749.
- Caron, F. and Fox, E. B. (2017). Sparse graphs using exchangeable random measures. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(5):1295–1366.
- Carpenter, J., P.Clifford, and Fearnhead, P. (1999). Improved particle filter for non-linear problems. *IEE Proceedings - Radar, Sonar and Navigation*, 146:2–7(5).
- Carvalho, C. M., Johannes, M. S., Lopes, H. F., and Polson, N. G. (2010). Particle learning and smoothing. *Statistical Science*, 25(1):88–106.
- Chien, I., Lin, C.-Y., and Wang, I.-H. (2018). Community detection in hypergraphs: Optimal statistical limit and efficient algorithms. In Storkey, A. and Perez-Cruz, F., editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 871–879, Playa Blanca, Lanzarote, Canary Islands. PMLR.

- Chopin, N., Jacob, P. E., and Papaspiliopoulos, O. (2013). Smc2: an efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(3):397–426.
- Cooley, O., Fang, W., Del Giudice, N., and Kang, M. (2018). Subcritical random hypergraphs, high-order components, and hypertrees. *arXiv e-prints*, page arXiv:1810.08107.
- Coulson, M., Gaunt, R., and Reinert, G. (2016). Poisson approximation of subgraph counts in stochastic block models and a graphon model. *ESAIM: Probability and Statistics*.
- 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.
- D’Angelo, S., Alfò, M., and Murphy, T. B. (2018). Modelling heterogeneity in latent space models for multidimensional networks.
- D’Angelo, S., Murphy, T. B., and Alfò, M. (2019). Latent space modelling of multidimensional networks with application to the exchange of votes in eurovision song contest. *Ann. Appl. Stat.*, 13(2):900–930.
- De Ridder, S., Vandermarliere, B., and Ryckebusch, J. (2016). Detection and localization of change points in temporal networks with the aid of stochastic block models. *Journal of Statistical Mechanics: Theory and Experiment*, 2016(11):113302.
- Del Moral, P. and Murray, L. M. (2015). Sequential monte carlo with highly informative observations. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):969–997.

- Delaunay, B. (1934). Sur la sphère 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.
- Douc, R. and Cappé, O. (2005). Comparison of resampling schemes for particle filtering. In *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, pages 64–69.
- Doucet, A., De Freitas, N., and Gordon, N., editors (2001). *Sequential Monte Carlo methods in practice*.
- Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10(3):197–208.
- Doucet, A. and Johansen, A. (2008). A tutorial on particle filtering and smoothing: Fifteen years later. Technical report.
- 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.
- Durante, D. and Dunson, D. B. (2014). Nonparametric Bayes dynamic modelling of relational data. *Biometrika*, 101(4):883–898.
- Durante, D., Dunson, D. B., and Vogelstein, J. T. (2016). Nonparametric bayes modeling of populations of networks. *Journal of the American Statistical Association*.
- Durante, D., Mukherjee, N., and Steorts, R. C. (2017). Bayesian learning of dynamic multilayer networks. *J. Mach. Learn. Res.*, 18(1):1414–1442.

- 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.
- Eldridge, J., Belkin, M., and Wang, Y. (2016). Graphons, mergeons, and so on! In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS’16, pages 2315–2323, USA. Curran Associates Inc.
- Élie de Panafieu (2015). Phase transition of random non-uniform hypergraphs. *Journal of Discrete Algorithms*, 31:26 – 39. 24th International Workshop on Combinatorial Algorithms (IWOCA 2013).
- Elvira, V., Martino, L., and Robert, C. P. (2018). Rethinking the Effective Sample Size. *ArXiv e-prints*.
- Erdős, P. and Rényi, A. (1959). On random graphs, I. *Publicationes Mathematicae (Debrecen)*, 6:290–297.
- Fasano, A., Rebaudo, G., Durante, D., and Petrone, S. (2019). A Closed-Form Filter for Binary Time Series. *arXiv e-prints*, page arXiv:1902.06994.
- Fearnhead, P. (2002). Markov chain monte carlo, sufficient statistics and particle filters. *Journal of Computational and Graphical Statistics*, 11(4):848–862.
- Fosdick, B. K., McCormick, T. H., Murphy, T. B., Ng, T. L. J., and Westling, T. (2019). Multiresolution network models. *Journal of Computational and Graphical Statistics*, 28(1):185–196.
- Frank, O. (1991). Statistical analysis of change in networks. *Statistica Neerlandica*, 45(3):283–293.
- Frank, O. and Strauss, D. (1986). Markov graphs. *Journal of the American Statistical Association*, 81(395):832–842.

- 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.
- Gao, M. and Zhang, H. (2012). Sequential monte carlo methods for parameter estimation in nonlinear state-space models. *Computers and Geosciences*, 44:70 – 77.
- Gemmetto, V., Barrat, A., and Cattuto, C. (2014). Mitigation of infectious disease at school: targeted class closure vs school closure. In *BMC Infectious Diseases*.
- 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.
- Gilbert, E. N. (1959). Random graphs. *Ann. Math. Statist.*, 30(4):1141–1144.
- Gilks, W. R. and Berzuini, C. (2001). Following a moving target monte carlo inference for dynamic bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1):127–146.
- Gilliland, D. C. (1962). Integral of the bivariate normal distribution over an offset circle. *Journal of the American Statistical Association*, 57(300):758–768.
- Godsill, S. and Clapp, T. (2001). *Improvement Strategies for Monte Carlo Particle Filters*, pages 139–158. Springer New York, New York, NY.
- Goldenberg, A., Zheng, A. X., Fienberg, S. E., and Airoldi, E. M. (2010). A survey of statistical network models. *Found. Trends Mach. Learn.*, 2(2):129–233.
- Gollini, I. and Murphy, T. B. (2016). Joint modeling of multiple network views. *Journal of Computational and Graphical Statistics*, 25(1):246–265.

- Goodman, L. A. (1974). Exploratory latent structure analysis using both identifiable and unidentifiable models. *Biometrika*, 61(2):215–231.
- Gordon, N., Salmond, D., and Smith, A. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEEE Proceedings F, Radar and Signal Processing*, 140(2):107–113.
- Gormley, I. C. and Murphy, T. B. (2010). A mixture of experts latent position cluster model for social network data. *Statistical Methodology*, 7(3):385 – 405. Special Issue on Statistical Methods for the Social Sciences. Honoring the 10th Anniversary of the Center for Statistics and the Social Sciences at the University of Washington.
- Gromov, M. (1987). Hyperbolic groups. *Essays in Group Theory, Mathematical Sciences Research Institute Publications*.
- Gunawan, D., Dang, K.-D., Quiroz, M., Kohn, R., and Tran, M.-N. (2018). Subsampling sequential monte carlo for static bayesian models.
- 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.
- Heard, N. A. and Turcotte, M. J. M. (2017). Adaptive sequential monte carlo for multiple changepoint analysis. *Journal of Computational and Graphical Statistics*, 26(2):414–423.
- Hoff, P. (2008a). Modeling homophily and stochastic equivalence in symmetric relational data. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T., editors, *Advances in Neural Information Processing Systems 20*, pages 657–664. Curran Associates, Inc.
- Hoff, P. D. (2008b). Multiplicative latent factor models for description and predic-



- tion of social networks. *Computational and Mathematical Organization Theory*, 15(4):261.
- Hoff, P. D. (2011). Hierarchical multilinear models for multiway data. *Computational Statistics & Data Analysis*, 55(1):530–543.
- 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.
- Hoover, D. (1979). Relations on probability spaces and arrays of random variables.
- Ji, P. and Jin, J. (2016). Coauthorship and citation networks for statisticians. *Ann. Appl. Stat.*, 10(4):1779–1812.
- Kahle, M. (2017). Random simplicial complexes. In *Handbook of Discrete and Computational Geometry*, chapter 22, pages 581–603. Chapman and Hall/CRC.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45.
- Kantas, N., Doucet, A., Singh, S., and Maciejowski, J. (2009). An overview of sequential monte carlo methods for parameter estimation in general state-space models. *IFAC Proceedings Volumes*, 42(10):774 – 785. 15th IFAC Symposium on System Identification.

- Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J., and Chopin, N. (2015). On particle methods for parameter estimation in state-space models. *Statist. Sci.*, 30(3):328–351.
- Karoński, M. and Łuczak, T. (2002). The phase transition in a random hypergraph. *Journal of Computational and Applied Mathematics*, 142(1):125 – 135. Probabilistic Methods in Combinatorics and Combinatorial Optimization.
- Karrer, B. and Newman, M. E. J. (2011). Stochastic blockmodels and community structure in networks. *Phys. Rev. E*, 83:016107.
- 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.
- Khan, Z., Balch, T., and Dellaert, F. (2005). Mcmc-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1805–1819.
- Kim, B., Lee, K. H., Xue, L., and Niu, X. (2018). A review of dynamic network models with latent variables. *Statist. Surv.*, 12:105–135.
- Kim, C., Bandeira, A. S., and Goemans, M. X. (2018). Stochastic Block Model for Hypergraphs: Statistical limits and a semidefinite programming approach. *arXiv e-prints*, page arXiv:1807.02884.
- Kitsak, M., Voitalov, I., and Krioukov, D. (2019). Link prediction with hyperbolic geometry. *arXiv e-prints*, page arXiv:1903.08810.
- 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](http://konect.uni-koblenz.de/networks/brunson_corporate-leadership).

- Krioukov, D., Papadopoulos, F., Kitsak, M., Vahdat, A., and Boguñá, M. (2010). Hyperbolic geometry of complex networks. *Phys. Rev. E*, 82:036106.
- 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.
- Lazarsfeld, P. and Henry, N. (1968). *Latent structure analysis*. Houghton, Mifflin.
- Le, C. M., Levin, K., and Levina, E. (2018). Estimating a network from multiple noisy realizations. *Electron. J. Statist.*, 12(2):4697–4740.
- Lee, W., McCormick, T. H., Neil, J., and Sodja, C. (2019). Anomaly detection in large scale networks with latent space models.
- Leskovec, J. and Krevl, A. (2014). SNAP Datasets: Stanford large network dataset collection. <http://snap.stanford.edu/data>.
- Lin, C., Chien, I. E., and Wang, I. (2017). On the fundamental statistical limit of community detection in random hypergraphs. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 2178–2182.
- Liu, D., Blenn, N., and Miegheem, P. V. (2013). Characterising and modelling social networks with overlapping communities. *Int. J. Web Based Communities*, 9(3):371–391.
- Liu, J. and West, M. (2001). *Combined Parameter and State Estimation in Simulation-Based Filtering*, pages 197–223. Springer New York, New York, NY.
- Lopes, H. F. and Tsay, R. S. (2011). Particle filters and bayesian inference in financial econometrics. *Journal of Forecasting*, 30(1):168–209.

- Lovász, L. (2012). *Large Networks and Graph Limits.*, volume 60 of *Colloquium Publications*. American Mathematical Society.
- Ludkin, M., Eckley, I., and Neal, P. (2018). Dynamic stochastic block models: parameter estimation and detection of changes in community structure. *Statistics and Computing*, 28(6):1201–1213.
- 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.
- Mansilla, S. P. and Serra, O. (2008). On  $s$ -arc transitive hypergraphs. *European Journal of Combinatorics*, 29(4):1003 – 1011. Homomorphisms: Structure and Highlights.
- Marin, J.-M., Pudlo, P., Robert, C. P., and Ryder, R. J. (2012). Approximate bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180.
- McCormick, T. H. and Zheng, T. (2015). Latent surface models for networks using aggregated relational data. *Journal of the American Statistical Association*, 110(512):1684–1695.
- Naesseth, C. A., Lindsten, F., and Schön, T. B. (2015). Nested sequential monte carlo methods. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15*, pages 1292–1301. JMLR.org.
- Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing*, 11(2):125–139.

- Nemeth, C., Fearnhead, P., and Mihaylova, L. (2016). Particle approximations of the score and observed information matrix for parameter estimation in statespace models with linear computational cost. *Journal of Computational and Graphical Statistics*, 25(4):1138–1157.
- Newman, M. E. J. (2010). *Networks: an introduction*. Oxford University Press, Oxford; New York.
- Ng, T. L. J. and Murphy, T. B. (2018). Model-based clustering for random hypergraphs. *ArXiv e-prints*.
- Ng, T. L. J. and Murphy, T. B. (2019). Generalized random dot product graph.
- Nickel, C. (2007). *Random Dot Product Graphs: A Model for Social Networks*. PhD thesis, Department of Applied Mathematics and Statistics, The Johns Hopkins University.
- Nowicki, K. and Snijders, T. A. B. (2001). Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087.
- Park, J. and Ionides, E. L. (2019). Inference on high-dimensional implicit dynamic models using a guided intermediate resampling filter. *arXiv e-prints*, page arXiv:1708.08543.
- Park, J. H. and Sohn, Y. (2018). Detecting structural changes in longitudinal network data. *Bayesian Anal.* Advance publication.
- Patcha, A. and Park, J.-M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, 51(12):3448 – 3470.
- Penrose, M. (2003). *Random Geometric Graphs*. Oxford Studies in Probability.

- Petersen, K. B. and Pedersen, M. S. (2012). The matrix cookbook. Version 20121115.
- Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599.
- Polson, N. G., Scott, J. G., and Windle, J. (2013). Bayesian inference for logistic models using polyá-gamma latent variables. *Journal of the American Statistical Association*, 108(504):1339–1349.
- Polson, N. G., Stroud, J. R., and Mller, P. (2008). Practical filtering with sequential parameter learning. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(2):413–428.
- Poyiadjis, G., Doucet, A., and Singh, S. S. (2011). Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika*, 98(1):65–80.
- 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.
- Pronzato, L., Wynn, H. P., and Zhigljavsky, A. A. (2018). Simplicial variances, potentials and mahalanobis distances. *Journal of Multivariate Analysis*, 168:276 – 289.
- Raftery, A. E., Niu, X., Hoff, P. D., and Yeung, K. Y. (2012). Fast inference for the latent space network model using a case-control approximate likelihood. *Journal of Computational and Graphical Statistics*, 21(4):901–919.
- Rastelli, R., Friel, N., and Raftery, A. E. (2016). Properties of latent variable network models. *Network Science*, 4(4):407432.
- Rastelli, R., Maire, F., and Friel, N. (2018). Computationally efficient inference for latent position network models. *arXiv e-prints*, page arXiv:1804.02274.

- Rebeschini, P. and van Handel, R. (2015). Can local particle filters beat the curse of dimensionality? *Ann. Appl. Probab.*, 25(5):2809–2866.
- Rubin-Delanchy, P., Priebe, C. E., and Tang, M. (2017). Consistency of adjacency spectral embedding for the mixed membership stochastic blockmodel.
- Rubin-Delanchy, P., Priebe, C. E., Tang, M., and Cape, J. (2017). A statistical interpretation of spectral embedding: the generalised random dot product graph. *arXiv e-prints*, page arXiv:1709.05506.
- Salnikov, V., Cassese, D., and Lambiotte, R. (2018). Simplicial complexes and complex systems. *European Journal of Physics*, 40(1):014001.
- Salter-Townshend, M. and McCormick, T. H. (2017). Latent space models for multi-view network data. *Ann. Appl. Stat.*, 11(3):1217–1244.
- Salter-Townshend, M. and Murphy, T. B. (2013). Variational bayesian inference for the latent position cluster model for network data. *Computational Statistics and Data Analysis*, 57(1):661 – 671.
- 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.
- Septier, F., Pang, S. K., Carmi, A., and Godsill, S. (2009). On mcmc-based particle methods for bayesian filtering: Application to multitarget tracking. In *2009 3rd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 360–363.

- Septier, F. and Peters, G. W. (2015). An Overview of Recent Advances in Monte-Carlo Methods for Bayesian Filtering in High-Dimensional Spaces. In Peters, G. W. and Matsui, T., editors, *Theoretical Aspects of Spatial-Temporal Modeling*. Springer-Briefs - JSS Research Series in Statistics.
- Sewell, D. K. and Chen, Y. (2015a). Analysis of the formation of the structure of social networks by using latent space models for ranked dynamic networks. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 64(4):611–633.
- Sewell, D. K. and Chen, Y. (2015b). Latent space models for dynamic networks. *Journal of the American Statistical Association*, 110(512):1646–1657.
- Sewell, D. K. and Chen, Y. (2016). Latent space models for dynamic networks with weighted edges. *Social Networks*, 44:105 – 116.
- Sewell, D. K. and Chen, Y. (2017). Latent space approaches to community detection in dynamic networks. *Bayesian Anal.*, 12(2):351–377.
- Sharma, A., Srivastava, J., and Chandra, A. (2014). Predicting Multi-actor collaborations using Hypergraphs. *arXiv e-prints*, page arXiv:1401.6404.
- Smith, A. L., Asta, D. M., and Calder, C. A. (2017). The Geometry of Continuous Latent Space Models for Network Data. *ArXiv e-prints*.
- Snyder, C., Bengtsson, T., Bickel, P., and Anderson, J. (2008). Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12):4629–4640.
- Snyder, C., Bengtsson, T., and Morzfeld, M. (2015). Performance bounds for particle filters using the optimal proposal. *Monthly Weather Review*, 143(11):4750–4761.
- Spencer, N. A. and Rohilla Shalizi, C. (2017). Projective, Sparse, and Learnable Latent Position Network Models. *arXiv e-prints*, page arXiv:1709.09702.
- Stasi, D., Sadeghi, K., Rinaldo, A., Petrović, S., and Fienberg, S. E. (2014).  $\beta$  models for random hypergraphs with a given degree sequence. *ArXiv e-prints*.



- Stehlé, J., Voirin, N., Barrat, A., Cattuto, C., Isella, L., Pinton, J.-F., Quaggiotto, M., Van den Broeck, W., Régis, C., Lina, B., and Vanhems, P. (2011). High-resolution measurements of face-to-face contact patterns in a primary school. *PLOS ONE*, 6(8):1–13.
- Storvik, G. (2002). Particle filters for state-space models with the presence of unknown static parameters. *IEEE Transactions on Signal Processing*, 50(2):281–289.
- Sweet, T. M., Thomas, A. C., and Junker, B. W. (2013). Hierarchical network models for education research: Hierarchical latent space models. *Journal of Educational and Behavioral Statistics*, 38(3):295–318.
- 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.
- Truong, C., Oudre, L., and Vayatis, N. (2018). Selective review of offline change point detection methods.
- van Duijn, M. A. J., Snijders, T. A. B., and Zijlstra, B. J. H. (2004). p2: a random effects model with covariates for directed graphs. *Statistica Neerlandica*, 58(2):234–254.
- Vietoris, L. (1927). Über den hheren zusammenhang kompakter räume und eine klasse von zusammenhangstreuen abbildungen. *Mathematische Annalen*, pages 454–472.
- Wang, Y., Chakrabarti, A., Sivakoff, D., and Parthasarathy, S. (2017). Hierarchical change point detection on dynamic networks. In *Proceedings of the 2017 ACM on Web Science Conference*, WebSci '17, pages 171–179, New York, NY, USA. ACM.

- Ward, M. D., Ahlquist, J. S., and Rozenas, A. (2013). Gravity’s rainbow: A dynamic latent space model for the world trade network. *Network Science*, 1(1):95118.
- Wasserman, S. and Pattison, P. (1996). Logit models and logistic regressions for social networks: I. an introduction to markov graphs andp. *Psychometrika*, 61(3):401–425.
- Young, S. J. and Scheinerman, E. R. (2007). Random dot product graph models for social networks. In Bonato, A. and Chung, F. R. K., editors, *Algorithms and Models for the Web-Graph*, pages 138–149, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Zachary, W. (1977). An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473.
- 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.