INFERRING NETWORK STRUCTURES USING HIERARCHICAL EXPONENTIAL RANDOM GRAPH MODELS

a thesis submitted to

The University of Kent at Canterbury

in the subject of statistics

for the degree

of doctor of philosophy

133 pages

By

Sa Ren

August 2022

Inferring Network Structures using Hierarchical Exponential Random Graph Models

Sa Ren

Declaration

The work in this thesis is based on research completed at the School of Mathematics, Statistics and Actuarial Science at the University of Kent. This thesis, nor any part of it, has been submitted elsewhere for any other degree or qualification. Chapter 2 contains necessary background material, for which no originality is claimed. All other material is believed to be original unless stated otherwise.

Acknowledgments

First of all, I would express my sincere gratitude to Dr Xue Wang for doing her utmost as my supervisor, especially that after she left the university, she still managed to join every supervision meeting for more than one and half years. She did not only teach me how to become an independent researcher but also help me with many things that are outside of work. I also want to thank Dr Peng Liu and Prof Jian Zhang for taking over my supervision and offering me continuous guidance, support and encouragement. I am very grateful to all my three supervisors for making a responsible, diverse and open supervision team. I could not imagine a better supervision team.

I would also like to thank all my friends for the laughs and company during my PhD. Special thanks goes to Fenia and Mariza for all the time (delicious meals, nice talks and Pamela time) we spent together at Northgate, our warm and cosy home. The chats with Li always made me feel peaceful and gave me a sense of belonging. The sports activities with Alex also gave me a lot of unique experiences, like playing badminton in a car park. The time spent together with Ian, Helder, Jose, Aniketh, Joe and Fabian is also unforgettable.

I also want to thank all my badminton partners, Lehu, Siyu, Floris and many others for all the fun we had together. Last but not least, I would also like to thank Peter for his companionship and understanding in these years. Moreover, I also want to thank my parents for their unconditional love.

Finally, I would like to thank the School of Mathematics, Statistics and Actuarial Science at the University of Kent for offering me a scholarship to make this PhD possible.

Abstract

Networks are broadly used to represent the interaction relationships between entities in a wide range of scientific fields. Ensembles of networks are employed to provide multiple network observations from the same set of entities. These observations may capture different features of the relationships: some ensembles exhibit group structures; some ensembles are collected over time; other ensembles have more individual differences. Statistical models for ensembles of networks should describe not only the dependency structure within each network, but also the variations of the structural patterns across networks.

Exponential random graph models (ERGMs) provide a highly flexible way to study the complex dependency structures within networks. We aim to develop novel methodologies that utilise ERGMs to infer the underlying structures of ensembles of networks from the following three aspects: (1) identifying and characterising groups of networks that are similar with respect to the effects of local connectivity patterns and covariates of interest on shaping the global structure of networks; (2) modelling the evolution of networks over time by representing the associated parameters using a piecewise linear function; (3) analysing the individual characteristics of each network and the population structures of the whole ensemble in terms of the block structure, homophily, transitivity and other local structural properties.

For identifying the group structure of ensembles and the block structure of networks, we employ a Bayesian nonparametric prior on an infinite sample space, instead of requiring a fixed number of groups in advance as in the existing models. In this way, the number of mixture models can grow along with the data size.

This appealing property enables our models to fit the data better. Moreover, for the ensembles of networks with a time order, we utilise a fused lasso penalty to encourage similarities on the parameter estimation of the consecutive networks as they tend to share similar connectivity patterns.

The inference of ERGMs under a Bayesian nonparametric framework is very challenging due to the fact that we have an infinite number of intractable ERGM likelihood functions in the model. Besides, the dependency among edges within the same block and the unknown number of blocks also significantly increase the difficulty of recovering the unknown block structure. What's more, the correlation between dynamic networks also requires us to work on all the possible edges of the ensemble simultaneously, posing a big challenge for the algorithm.

To solve these issues, we develop five algorithms for the model estimation: (1) a novel Metropolis-Hastings algorithm to sample from the intractable posterior distribution of ERGMs with multiple networks using an intermediate importance sampling technique; (2) a new Metropolis-within-slice sampling algorithm to perform full Bayesian inference on infinite mixtures of ERGMs; (3) a pseudo likelihood based Metropolis-within-slice sampling algorithm to learn the group structure of ensembles fast and adaptively; (4) an alternating direction method of multipliers (ADMM) algorithm for the fast estimation of dynamic ensembles using a matrix decomposition technique; (5) a Metropolis-within-Gibbs sampling algorithm for the population analysis of structural patterns with an approximated stick-breaking prior.

List of Abbreviations

ADMM alternating direction method of multipliers

BHB-ERGM Bayesian hierarchical block exponential random graph model

DPM-ERGM Dirichlet process mixture of exponential random graph models

ERGM exponential random graph model

FL-ERGM fused lasso exponential random graph model

gwdsp geometrically weighted dyadwise shared partners

gwesp geometrically weighted edgewise shared partners

IMS intermediate importance Metropolis-within-slice

lasso least absolute shrinkage and selection operator

LU lower-upper

MCMC Markov Chain Monte Carlo

MCMH Monte Carlo Metropolis-Hastings

MLE maximum likelihood estimation

MMCMH Multi-network Monte Carlo Metropolis-Hastings

PMS pseudo likelihood based Metropolis-within-slice

PMG pseudo likelihood based Metropolis-within-Gibbs

SBM stochastic block model

VCERGM varying coefficient exponential random graph model

Contents

1 Introduction **1**

1.1 Problem Statement 2

1.2 Challenges 4

1.3 Contributions 5

1.4 Outline of the Thesis 6

2 Literature Review **8**

2.1 Network Data Representation 8

2.2 Exponential Random Graph Models 9

2.2.1 Pseudo Likelihood Estimation 10

2.2.2 Maximum Likelihood Estimation 11

2.2.3 Bayesian Estimation 13

2.3 Dirichlet Process Mixture Models 16

2.3.1 Mixture Models 16

2.3.2 Dirichlet Process Mixture Models 17

2.3.3 Slice Sampling Algorithm 19

2.4 Stochastic Block Models 22

2.4.1 Basic SBMs 22

2.4.2 Mixed Membership SBMs 23

2.5 Fused Lasso Regression 24

3 Dirichlet Process Mixtures of Exponential Random Graph Models **27**

3.1 Introduction 27
3.2 Dirichlet Process Mixtures of ERGMs 30
3.3 Posterior Computation 32
3.3.1 Full Likelihood Based Algorithm 34
3.3.2 Pseudo Likelihood Based Algorithm 38
3.4 Empirical Results 39
3.4.1 Intermediate Importance Sampling 39
3.4.2 Simulated Networks 41
3.4.3 Krackhardt's Advice Networks 47
3.4.4 International Trade Networks 53
3.5 Discussion 59

4 Fused Lasso Exponential Random Graph Models 60
4.1 Introduction 60
4.2 Fused Lasso ERGMs 63
4.2.1 Model Formulation 63
4.2.2 Pseudo Likelihood Approximation 64
4.2.3 Choice of Tuning Parameters 66
4.3 Model Estimation 67
4.3.1 ADMM Algorithm 68
4.3.2 LU Decomposition 71
4.3.3 Theoretical Guarantee 74
4.4 Empirical Results 76
4.4.1 Simulated Networks 76
4.4.2 International Trade Networks 84
4.4.3 Political Co-voting Networks 88
4.5 Discussion 93

5 Bayesian Hierarchical Block Exponential Random Graph Models 94
5.1 Introduction 94
5.2 Model Formulation 97

5.2.1 Block ERGMs 97
5.2.2 Bayesian Hierarchical Frameworks 99
5.2.3 Bayesian Hierarchical Block ERGMs 100
5.3 Posterior Computation 101
5.4 Empirical Results 105
5.4.1 Simulated Networks 105
5.4.2 International Trade Networks 109
5.5 Discussion 113

6 Conclusions and Future Work **114**
6.1 Conclusions 114
6.2 Future Work 115

A Appendix **117**
A.1 Taylor Expansion 117
A.2 Proof of Lemma 1 119
A.3 Proof of Theorem 2 120

Bibliography **127**

Chapter 1

Introduction

Networks, consisting of nodes and edges, are widely used to describe the interactions among different entities. The analysis of such data is an emerging area in statistics. Over the last few decades, a significant number of models have been proposed to infer the underlying structures of network data, including latent space models (Hoff et al., 2002; Sewell and Chen, 2015), stochastic block models (Holland et al., 1983; Lee and Wilkinson, 2019), and exponential random graph models (ERGMs) (Frank and Strauss, 1986; Schweinberger et al., 2020).

An ensemble of networks, also known as a population of networks or multiplex networks, represents multiple network observations obtained on the same or similar set of nodes across different subjects or time points. With the recent development of technology, such data have become commonplace in many fields. Examples include social networks from different schools (Sweet et al., 2019), brain networks from different participants (Simpson et al., 2013), and trade networks from different years (Lee et al., 2020). They provide multiple instances of the complex interactions among the same set of nodes, increasing the complexity of the data structure significantly.

In ensembles of networks, multilevel structures occur naturally within and across networks. To be clear, we use an ensemble of trade networks for an example. The ensemble of trade networks includes 16 networks indicating the trade relationships among 60 countries from year 2001 to 2016. Within each network,

the trade relationship between two countries may be affected by other relationships, exhibiting a wide range of dependency structures. Across networks, some structures remain the same, for example when the associated countries have a long-standing trade agreement. Other connection patterns may change continuously with the time. There is a need to understand the similarities and dissimilarities of network structures that determine the formation of the ensemble.

ERGMs are a class of versatile network models that capture various types of topological structures through exponential family distributions. The ability of incorporating complex connectivity patterns makes it an attractive model. There is a large and growing body of literature on the developments and inference of ERGMs, as reviewed by Schweinberger et al. (2020). Applying ERGMs to trade networks can identify the significant structural patterns or exogenous factors associated with the formation of trade connections between countries within each network. In this thesis, we focus on developing three new methodologies to model the variations of network structures across networks based on ERGMs.

1.1 Problem Statement

The mixture model framework is a flexible method to capture the variations within the ensemble using a group representation. With a mixture model, the ensemble is divided into different groups. Each group contains networks generated from the same distribution. There are a number of models for network clustering, including Durante et al. (2017), Signorelli and Wit (2020), and Yin et al. (2022). However, the models developed by Durante et al. (2017) and Signorelli and Wit (2020) can not capture the complex connectivity patterns within networks. Although the finite mixtures of ERGMs proposed by Yin et al. (2022) can represent various topological features, this model requires a given number of clusters and struggles to adapt to the dataset with increasing size. In this thesis, we propose a Bayesian nonparametric mixture of ERGMs to analyse the ensemble of networks, which

allows us to divide the multiple networks into several clusters based on the similarities of the connectivity patterns without requiring a fixed number of clusters in advance.

In some cases the ensembles of networks are observed over time. In the ensemble of trade networks mentioned before, the networks have a time order as they are collected from year 2001 to 2016. Related work that employs ERGMs to model the temporal variations of dynamic ensembles includes Hanneke et al. (2010), Krivitsky and Handcock (2014), Lee et al. (2020), Ren (2019) and Lee et al. (2020). Hanneke et al. (2010) proposed a temporal ERGM by expressing the one-step transition probability between two consecutive networks using an exponential family distribution. Krivitsky and Handcock (2014) and Lee et al. (2020) focused on extensions to temporal ERGMs (Hanneke et al., 2010). But the temporal ERGM framework ignores the heterogeneity of the differences between networks. The model developed by Lee et al. (2020) exploits a varying coefficient framework to capture the dynamic pattern of the network structure using a smooth function. However, this model can not perform well when the underlying network structure does not change smoothly with the time. For example, in the trade network ensemble, the trade relationships were greatly affected by the financial crisis of 2008, resulting in an abrupt change on the network structure. Such a change can not be captured by a smooth function but a piecewise linear function. We propose to model the parameters associated with different features through a piecewise linear function.

The Bayesian hierarchical framework provides a way to capture the individual variations among networks as well as the common structure of the ensemble. Slaughter and Koehly (2016) and Lehmann and White (2021) studied the individual and population characteristics of ensembles of networks using a Bayesian hierarchical framework, but they did not include block structures in their model. The block structure, also called the community structure, is the structure that nodes are divided into different groups with similar features. Incorporating block structures into ERGMs would allow us to learn the various connectivity patterns

between nodes with certain features. Such an extension of applying ERGMs to networks with block structures has been studied by Schweinberger and Handcock (2015), Babkin et al. (2020) and Schweinberger and Stewart (2020). However, these methods focus on a singe network observation. The variations of the within-block connectivity patterns have not been studied in the ensembles of networks. In this thesis, we incorporate the block structure into the Bayesian hierarchical framework to study the individual variations among various local connectivity patterns.

1.2 Challenges

The probability function of ERGMs includes a normalising constant. The normalising constant depends on model parameters and is only computable in trivial cases. This increases the difficulty of handling ERGMs from a statistical viewpoint. In this thesis, we encounter three main challenges in applying ERGMs for ensembles of networks.

Firstly, the inference of Bayesian nonparametric mixtures of ERGMs is extremely challenging since an infinite number of intractable ERGM likelihood functions have to be evaluated and compared. The posterior distribution of ERGMs is doubly intractable due to the intractability of sampling directly from the posterior distribution, and the intractability of the likelihood model within the posterior (Caimo and Friel, 2011). Besides, sampling from the Bayesian nonparametric mixture model is also complicated due to the countably infiniteness of the discrete masses from the random distribution functions chosen from the Dirichlet process prior (Walker, 2007).

Estimating the dynamic evolution of networks is the second challenge as we want to present the correlation of ERGM parameter estimates using a piecewise linear function. This requires us to work on all the possible edges of the ensemble at the same time. In other words, for an ensemble with T directed networks and n nodes, $n(n-1)T$ edges have to be estimated simultaneously at every iteration of

the algorithm. Also, the inverse of a $dT \times dT$ matrix has to be calculated at each iteration step, with d denoting the number of characteristics of interest. This is particular challenging with large T.

Inferring the unknown block structure of ensembles of networks while describing the structural dependency within blocks using ERGMs is the third challenge. Existing network models usually implicitly assume the edges within the same block are independent of each other (Lee and Wilkinson, 2019). The within-block independence assumption facilitates model inference, but fails to capture the true connectivity patterns among nodes from the same group. In our framework, we model the dependency structures within each block using ERGMs. This enables us to learn the true underlying characteristics of each network in detail although increases the difficulty of model estimation. In addition, the joint estimation of individual and population characteristics of the ensemble also makes the problem even more complicated.

1.3 Contributions

The main contribution of this thesis is that we extend ERGMs using a Bayesian nonparametric framework to two different situations. To be specific, we first use Bayesian nonparametric ERGMs to detect and compare the groups of networks that have same generative processes without requiring the number of groups in advance. Then we develop another Bayesian nonparametric ERGMs to identify the subnetworks formed by the groups of nodes with similar connectivity patterns and to model the dependency within subnetworks using ERGMs. The Bayesian nonparametric analysis of ERGMs retains the advantage of both Bayesian nonparametric models and ERGMs. It provides a flexible and general framework that adapts its complexity automatically to the observed ensembles of networks, and has the ability to capture various dependency structural patterns among nodes.

Another major contribution of this thesis is that we analyse the temporal variations of ensembles by describing the parameters associated with each network

as a piecewise linear function. The parameters describe the effects of different structural patterns on the formation of networks, and a piecewise linear function of parameters provides an effective way to show the changing trend of networks, whether it has abrupt changes or it is invariant for some period of time. Moreover, the neighbouring networks tend to share similar characteristics as they are observed at consecutive time points. The use of fused lasso penalty encourages similar estimates between neighbouring networks.

Our contributions also include the development of effective algorithms for our proposed methods. To be specific, we develop two novel MCMC algorithms to sample from the proposed Bayesian nonparametric mixtures of ERGMs which tackle the issues of sampling from an infinite mixture of doubly intractable distributions. We also develop a fast algorithm, based on the alternating direction method of multipliers (ADMM), to estimate all the ERGM parameters simultaneously. Besides, we develop a Metropolis-within-Gibbs algorithm to infer the block structure of the ensemble adaptively and describe the various dependency structures within and across blocks using a multilevel framework.

1.4 Outline of the Thesis

In this thesis, we develop three novel statistical models for inferring the unknown structures of ensembles of networks.

Chapter 2 gives the background knowledge on ERGMs, Bayesian nonparametric models, block models and fused lasso regression.

In Chapter 3, we propose a Dirichlet process mixture of exponential random graph models (DPM-ERGM) to study the group variations of networks. This is achieved by detecting and comparing the groups of networks that are generated from the same distribution. Moreover, we also develop two MCMC algorithms to sample from the posterior distributions of the infinite intractable ERGM distributions.

Next, we develop a fused lasso exponential random graph model (FL-ERGM)

to model the temporal variations of networks in Chapter 4. The model is estimated by the ADMM algorithm, thereby providing a fast approach for statistical inference of complex dynamic networks.

In Chapter 5, we develop a Bayesian hierarchical block exponential random graph model (BHB-ERGM) and a corresponding algorithm to study the individual variations among networks. Specifically, the block structure is incorporated into ERGMs to capture the dependency structure within blocks.

Chapter 6 concludes the thesis and discusses the potential future work. It summarises the methodologies developed in this thesis for ensembles of networks and suggests some potential extensions of the methods in several different directions.

Chapter 2

Literature Review

In this chapter, we first give a literature review on ERGMs with a focus on the existing estimation methods. Then we review the Dirichlet process mixture models. Afterwards, we review the stochastic block models from a mixture model perspective. Finally, we review the fused lasso regression and the corresponding estimation algorithm.

2.1 Network Data Representation

Networks, also called graphs, are representations of relational data. A network is made up of nodes and edges, where nodes represent individuals or organisations, and edges are placed between nodes if the corresponding nodes have interactions. Mathematically, a network with n nodes is expressed using an adjacency matrix Y, an $n \times n$ square matrix, with entry Y_{ij} indicating the relationships between nodes i and j. The realisation of Y is denoted by y while the set of all possible outcomes of Y is denoted by $\mathcal{Y}$. The covariate information regarding the nodal or network attribute that affects the connections are denoted by $X \in \mathcal{X}$.

In this thesis, we focus on binary networks without loops. In other words, the adjacency matrix has binary values, 0 or 1, on the off-diagonal elements. $Y_{ij} = 1$ indicates an edge between nodes i and j, and $Y_{ij} = 0$ indicates no connection between nodes i and j. The diagonal entries of the adjacency matrix are 0,

$Y_{ii} = 0$. The adjacency matrices of undirected networks are symmetric while those of directed networks might not be symmetric. We consider both undirected and directed networks here.

2.2 Exponential Random Graph Models

ERGMs describe the generating process of networks through exponential family distributions with summary statistics showing various connecting patterns as explanatory variables. For a network observation y with covariate information X, the network structures of interest are expressed using a summary statistics vector, $S(y, X) : \mathcal{Y} \times \mathcal{X} \to \mathbb{R}^d$. It represents the characteristics of the network, such as the number of edges, triangles, etc., which are crucial to the formation and dissolution of networks. The general ERGM has the following form,

$$P(Y = y \,|\, \theta, X) = \frac{\exp\{\theta^\top S(y, X)\}}{k(\theta)}. \tag{2.2.1}$$

$\theta \in \mathbb{R}^d$ is the vector of model parameters, and $S(y, X)$ is the summary statistics (Morris et al., 2008). The normalising constant $k(\theta) = \sum_{y \in \mathcal{Y}} \exp\left\{\theta^\top S(y, X)\right\}$ is the sum over all potential graphs in the sample space, which is usually intractable except for very small networks. Given a realisation of network y, the aim of statistical inference is to find which value of θ provides best description for the data under ERGM framework. The intractability of the normalising constant is a strong barrier to the estimation of ERGMs as the likelihood function can only be specified up to a parameter dependent constant.

There is significant amount of literature on the estimation of ERGMs, including work on pseudo likelihood methods (Strauss and Ikeda, 1990; van Duijn et al., 2009; Bouranis et al., 2017), Monte Carlo maximum likelihood methods (Robbins and Monro, 1951; Geyer and Thompson, 1992; Snijders, 2002; Hunter and Handcock, 2006), and Bayesian inference methods (Møller et al., 2006; Murray et al., 2006; Koskinen, 2008; Caimo and Friel, 2011; Liang and Jin, 2013). We

will review these methods in detail here.

2.2.1 Pseudo Likelihood Estimation

Pseudo likelihood methods (Strauss and Ikeda, 1990) approximate the ERGM likelihood function using the product of conditional probabilities of different edges,

$$PL(\theta) = \prod_{i \neq j} P(y_{ij}|C_{ij}, X) = \prod_{i \neq j} P(y_{ij} = 1|C_{ij}, X)^{y_{ij}} \left[1 - P(y_{ij} = 1|C_{ij}, X)\right]^{1-y_{ij}}, \tag{2.2.2}$$

where $C_{ij} = \{y_{kl}, (k,l) \neq (i,j)\}$ denotes all the dyads of the graph excluding y_{ij}.

From the likelihood of ERGMs in (2.2.1), we have

$$\begin{aligned} \text{logit } P(y_{ij} = 1|C_{ij}, X) &= \log \frac{P(y_{ij} = 1|C_{ij}, X)}{P(y_{ij} = 0|C_{ij}, X)} \\ &= \log \frac{P(y_{ij} = 1, C_{ij}, X)}{P(y_{ij} = 0, C_{ij}, X)} \\ &= \theta^\top \Delta S_{ij}, \end{aligned} \tag{2.2.3}$$

where $\Delta S_{ij} = S(y_{ij} = 1, C_{ij}, X) - S(y_{ij} = 0, C_{ij}, X)$ is called change statistics, measuring the change in the summary statistics $S(y, X)$ when one edge is changed from 0 to 1, holding the rest of the network fixed.

Based on (2.2.3), we can write the edge connecting probability in terms of the change statistics as

$$P(y_{ij} = 1|C_{ij}, X) = \frac{\exp\left(\theta^\top \Delta S_{ij}\right)}{1 + \exp\left(\theta^\top \Delta S_{ij}\right)}.$$

Therefore, log pseudo likelihood function of ERGM is

$$\begin{aligned} pl(\theta) &= \sum_{i \neq j} y_{ij} \log\left\{P(y_{ij} = 1|C_{ij}, X)\right\} + (1 - y_{ij}) \log\left\{1 - P(y_{ij} = 1|C_{ij}, X)\right\} \\ &= \sum_{i \neq j} \left\{y_{ij} \theta^\top \Delta S_{ij} - \log\left\{1 + \exp\left(\theta^\top \Delta S_{ij}\right)\right\}\right\}. \end{aligned} \tag{2.2.4}$$

With such a tractable pseudo likelihood function, the maximum pseudo likelihood estimators can be obtained by finding the maximum point of (2.2.4). The simplicity and computational tractability of the pseudo likelihood function make it an appealing alternative to the likelihood function, regardless of the fact that the properties of such estimators are unknown. It is particular helpful in estimating the ensembles of networks with large data size and complicated structures.

2.2.2 Maximum Likelihood Estimation

The maximum likelihood estimation (MLE) of ERGMs is challenging due to the intractable normalising constants in ERGMs. Here, we review two algorithms that utilise network samples from Markov Chain Monte Carlo (MCMC) methods to estimate ERGMs.

Robbins-Monro Algorithm

The log likelihood function of model (2.2.1) is

$$l(\theta) = \theta^\top S(y, X) - \log k(\theta) = \theta^\top S(y, X) - \log \left\{ \sum_{y \in \mathcal{Y}} \exp \left\{ \theta^\top S\left(y, X\right) \right\} \right\}.$$

The derivative of the log likelihood function is

$$\begin{aligned}
\nabla l(\theta) =& \nabla \theta^\top S(y, X) - \frac{\sum_{y \in \mathcal{Y}} \nabla \theta^\top S(y, X) \exp \left\{ \theta^\top S\left(y, X\right) \right\}}{\sum_{y \in \mathcal{Y}} \exp \left\{ \theta^\top S\left(y, X\right) \right\}} \\
=& \nabla \theta^\top \left\{ S(y, X) - \frac{\sum_{y \in \mathcal{Y}} S(y, X) \exp \left\{ \theta^\top S\left(y, X\right) \right\}}{\sum_{y \in \mathcal{Y}} \exp \left\{ \theta^\top S\left(y, X\right) \right\}} \right\} \\
=& \nabla \theta^\top \left\{ S(y, X) - E_\theta S(y, X) \right\}.
\end{aligned}$$

The idea of Robbins-Monro algorithm (Snijders, 2002; Robbins and Monro, 1951) is to search for an approximate solution to the moment equation $E_\theta S(y, X) - S(y, X) = 0$ using the Newton-Raphson method. During the estimation, $E_\theta S(y, X)$ is approximated using network samples generated from ERGM distributions.

Importance Sampling

The normalising constant is intractable, but the normalising constant ratio can be rewritten with respect to expectation. Specifically, for the normalising constant $k(\theta) = \sum_{y \in \mathcal{Y}} q(y;\theta) = \sum_{y \in \mathcal{Y}} \exp\left\{\theta^\top S(y)\right\}$, the normalising constant ratio

$$\frac{k(\theta_2)}{k(\theta_1)} = \frac{\sum_{y \in \mathcal{Y}} q(y;\theta_2)}{k(\theta_1)} = \sum_{y \in \mathcal{Y}} \frac{q(y;\theta_2)}{k(\theta_1)} = \sum_{y \in \mathcal{Y}} \frac{q(y;\theta_2)}{q(y;\theta_1)} \frac{q(y;\theta_1)}{k(\theta_1)} = E_{\theta_1} \frac{q(y;\theta_2)}{q(y;\theta_1)}. \tag{2.2.5}$$

Then the importance sampling estimation to the normalising constant ratio (2.2.5) is

$$\frac{1}{m_1} \sum_{i=1}^{m_1} \frac{q(y_i;\theta_2)}{q(y_i;\theta_1)} = \frac{1}{m_1} \sum_{i=1}^{m_1} \exp\left\{(\theta_2 - \theta_1)^\top S(y_i)\right\}, \tag{2.2.6}$$

where $y_1, \ldots, y_{m_1}$ is a sequence of independent networks sampled from the distribution defined by θ_1.

Geyer-Thompson Algorithm

Given a constant θ_0, searching for the MLE of ERGMs is equivalent to finding a point at which the gradient of $r(\theta, \theta_0) = l(\theta) - l(\theta_0)$ is zero, since $l(\theta) - l(\theta_0)$ differs from $l(\theta)$ only by a constant,

$$\begin{aligned} l(\theta) - l(\theta_0) =& (\theta - \theta_0)^\top S(y, X) - \log k(\theta) + \log k(\theta_0) \\ =& (\theta - \theta_0)^\top S(y, X) - \log \frac{k(\theta)}{k(\theta_0)}. \end{aligned}$$

Geyer and Thompson (1992) proposed to replace the normalising constant ratio using importance sampling estimation (2.2.6), and obtained the MLE by maximising

$$(\theta - \theta_0)^\top S(y, X) - \log \frac{1}{m_1} \sum_{i=1}^{m_1} \exp[(\theta - \theta_0)^\top S(y_i, X)]. \tag{2.2.7}$$

Hunter and Handcock (2006) extended the Geyer-Thompson algorithm to curved ERGMs with complicated network statistics $S(y, X)$. The choice of θ_0 is very crucial to the estimation. Koskinen (2008) pointed out one drawback of the importance sampling is that the approximation to the likelihood ratio $k(\theta)/k(\theta_0)$ is inaccurate when θ is far from θ_0.

2.2.3 Bayesian Estimation

Given the prior distribution $g_0(\theta)$, the posterior distribution of $f(y|\theta)$ is

$$f(\theta|y) = \frac{g_0(\theta)f(y|\theta)}{f(y)}, \tag{2.2.8}$$

where $f(y) = \int_{\mathbb{R}^d} g_0(\theta)f(y|\theta)d\theta$.

In the case of ERGMs,

$$f(y|\theta) = \frac{q(y;\theta)}{k(\theta)} = \frac{\exp\{\theta^\top S(y, X)\}}{k(\theta)}. \tag{2.2.9}$$

The posterior distribution of ERGMs is doubly intractable since both $f(\theta|y)$ and $f(y|\theta)$ are intractable. It is impossible to sample from such a distribution directly.

Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm is widely used to sample from complicated distributions. To sample from $f(\theta|y)$, the acceptance probability for moving from the current value θ^k to the new proposal θ' is

$$\frac{g_0(\theta')f(y|\theta')}{g_0(\theta^k)f(y|\theta^k)}\frac{h(\theta^k|\theta')}{h(\theta'|\theta^k)} = \frac{g_0(\theta')}{g_0(\theta^k)}\frac{q(y;\theta')}{q(y;\theta^k)}\frac{k(\theta^k)}{k(\theta')}\frac{h(\theta^k|\theta')}{h(\theta'|\theta^k)}, \tag{2.2.10}$$

with the proposal density represented by h. The continuation of the algorithm requires the calculation of normalising constant ratio $k(\theta^k)/k(\theta')$ at every iteration step, which makes the direct Metropolis-Hastings algorithm impractical.

There are different algorithms dealing with this problem. Møller et al. (2006)

proposed to set up the Metropolis-Hastings algorithm with an auxiliary variable where the normalising constant in the proposal density of the auxiliary variable cancels with the normalising constant in the likelihood model. The auxiliary variable method is a clever way to sample from doubly intractable distributions, but it also suffers from slow mixing rate and low acceptance ratio. Further developments of this method have been proposed by Koskinen (2008), Murray et al. (2006) and among others. Instead of sampling one sample at each iteration as in auxiliary variable method, Koskinen (2008) proposed a linked importance sampler auxiliary variable Metropolis-Hastings algorithm where multiple Markov Chains are constructed at each iteration based on the idea of bridge sampling. This method by Koskinen (2008) is more stable compared to the auxiliary variable method, but it is also more time consuming. Next, we describe the details of the exchange algorithm.

Exchange algorithm

The exchange algorithm proposed by Murray et al. (2006) is an ingenious way to sample from the doubly intractable distributions. The details of the exchange algorithm are as follows:

(1) Draw θ' from a proposal distribution $h(\cdot|\theta^k)$ with the current value θ^k.

(2) Generate an auxiliary variable w from $f(\cdot|\theta')$ using a perfect sampler (Propp and Wilson, 1996).

(3) Update θ^{k+1} with θ' with probability $\min(1, r)$,

$$\begin{aligned} r &= \frac{f(\theta'|y)}{f(\theta^k|y)} \frac{h(\theta^k|\theta')}{h(\theta'|\theta^k)} \frac{f(w|\theta^k)}{f(w|\theta')} \\ &= \frac{g_0(\theta')}{g_0(\theta^k)} \frac{q(y;\theta')}{q(y;\theta^k)} \frac{k(\theta^k)}{k(\theta')} \frac{h(\theta^k|\theta')}{h(\theta'|\theta^k)} \frac{q(w;\theta^k)}{q(w;\theta')} \frac{k(\theta')}{k(\theta^k)} \\ &= \frac{g_0(\theta')}{g_0(\theta^k)} \frac{q(y;\theta')}{q(y;\theta^k)} \frac{h(\theta^k|\theta')}{h(\theta'|\theta^k)} \frac{q(w;\theta^k)}{q(w;\theta')}. \end{aligned} \tag{2.2.11}$$

As we can see, the normalising constants are cancelled out in formula (2.2.11). The exchange algorithm offered a swap between (y, θ^k) and (w, θ') and obtains

a higher acceptance probability compared to the auxiliary variable method by Møller et al. (2006). Furthermore, Caimo and Friel (2011) applied the exchange algorithm to sample from the posterior ERGMs. As the perfect sampling method is very time consuming, MCMC methods are used in practice to sample from ERGMs.

Monte Carlo Metropolis-Hastings Algorithm

Liang and Jin (2013) proposed a Monte Carlo Metropolis-Hastings (MCMH) algorithm to sample from the doubly intractable distributions. The idea is to replace the unknown normalising constant ratio with an importance sampling estimation. It has been proved that under mild conditions, the MCMH algorithm converges to the target distribution. The steps of the MCMH algorithm for sampling from the posterior ERGM distributions are as follows:

(1) Draw θ' from a proposal distribution $h(\cdot|\theta^k)$.

(2) Generate m samples $z_1, z_2, \ldots, z_m$ independently from $f(\cdot|\theta^k)$ using MCMC.

(3) Calculate the approximation to the normalising constant ratio $k(\theta')/k(\theta^k)$

$$\hat{\beta} = \frac{1}{m}\sum_{i=1}^{m}\frac{q(y_i;\theta')}{q(y_i;\theta^k)} = \frac{1}{m}\sum_{i=1}^{m}\exp\left\{(\theta'-\theta^k)^\top S(z_i,X)\right\}. \tag{2.2.12}$$

(4) Update θ^{k+1} with θ' with probability

$$\min\left(1, \frac{g_0(\theta')}{g_0(\theta^k)}\frac{h(\theta^k|\theta')}{h(\theta'|\theta^k)}\frac{q(y;\theta')}{q(y;\theta^k)}\frac{1}{\hat{\beta}}\right). \tag{2.2.13}$$

In the exchange algorithm, a perfect sampler is required at each iteration. A pragmatic alternative is to use samples generated from the MCMC approach, as discussed in Caimo and Friel (2011). In this case, the exchange algorithm for ERGMs can be explained as approximating the unknown normalising constant ratio with one sample, which is a special case of MCMH algorithm when $m = 1$.

2.3 Dirichlet Process Mixture Models

2.3.1 Mixture Models

The mixture model assumes that the data point y_i is drawn from a mixture of multiple distributions,

$$y_i \sim \sum_{k=1}^{K} w_k f(y_i|\theta_k),$$

where k is the group label, K is the total number of groups, w_k is the mixing proportion, standing for the probability that y_i is from group k, θ_k is the parameter of group k, and $f(\cdot|\theta_k)$ is the component distribution parametrised by θ_k.

The mixture model likelihood of data $y = \{y_1, \ldots, y_n\}$ is written in a product-of-sums form,

$$f(y|\theta) = \prod_{i=1}^{n} \sum_{k=1}^{K} w_k f(y_i|\theta_k),$$

which is usually challenging to handle.

An alternative way of explaining a mixture model is to use a latent variable z to indicate the group membership of each data point. The indicator variable $z = \{z_1, \ldots, z_n\}$ takes values $(1, 2, \ldots, K)$ with probability $P(z_i = k) = w_k$. Given z,

$$y_i|z_i, \theta \sim f(\cdot|\theta_{z_i}).$$

A Bayesian mixture model consisting of K components is

$$\begin{aligned} w|\beta &\sim \text{Dirichlet}(\beta), \\ z_i|w &\sim \text{Multinomial}(w), \\ \theta_k|H &\sim H, \\ y_i|z_i, \theta &\sim f(\cdot|\theta_{z_i}), \end{aligned}$$

where $\beta = \{\beta_1, \ldots, \beta_K\}$ is the hyperparameter, and H is the prior distribution of $\theta = \{\theta_1, \ldots, \theta_K\}$.

2.3.2 Dirichlet Process Mixture Models

The Dirichlet process is a stochastic process that is widely used in the Bayesian nonparametric field, particularly in the mixture model framework. It is a distribution over distributions (Ferguson, 1973), i.e. each draw from a Dirichlet process is a distribution. Let H be a distribution over Θ and α be a positive real number. Then for any finite measurable partition $A_1, \ldots, A_r$ of Θ the vector $(G(A_1), \ldots, G(A_r))$ is random since G is random. We say G is a Dirichlet process with concentration parameter α and base distribution H, written $G \sim DP(\alpha, H)$, if

$$(G(A_1), \ldots, G(A_r)) \sim \text{Dirichlet}(\alpha H(A_1), \ldots, \alpha H(A_r)) \tag{2.3.1}$$

for every finite measurable partition $A_1, \ldots, A_r$ of Θ.

Next, we model a set of observations $\{y_1, \ldots, y_n\}$ using a set of latent parameters $\{\theta_1^*, \ldots, \theta_n^*\}$. Each θ_i^* is drawn independently and identically from G, while each y_i has distribution $f(\cdot|\theta_i^*)$ parametrized by θ_i^*,

$$\begin{aligned} y_i|\theta_i^* &\sim f(\cdot|\theta_i^*), \\ \theta_i^*|G &\sim G, \\ G|\alpha, H &\sim DP(\alpha, H). \end{aligned} \tag{2.3.2}$$

Because G is discrete, multiple θ_i^* can take the same value simultaneously, and the above model can be seen as a mixture model, where y_i with the same value of θ_i^* belong to the same cluster. Model (2.3.2) can be reformulated using the stick-breaking representation.

Stick-Breaking Representation

We first write down the stick-breaking representation (Sethuraman, 1994) for the Dirichlet process $G \sim DP(\alpha, H)$,

$$G = \sum_{k=1}^{\infty} w_k \delta_{\theta_k}, \tag{2.3.3}$$

where

$$\begin{aligned} & v_k \sim \text{Beta}(1, \alpha), \\ & w_1 = v_1, w_k = v_k \prod_{l=1}^{k-1} (1 - v_l), \\ & \theta_k \sim H. \end{aligned} \tag{2.3.4}$$

δ_{θ_k} is a measure that assigns mass 1 at θ_k. $v_1, v_2, \ldots$ are independent and identically distributed from a Beta distribution with parameters 1 and α. $\theta_1, \theta_2, \ldots$ are independent and identically distributed from the base distribution H. The construction of w can be understood metaphorically as follows. Starting with a stick of length 1, we first break it at v_1. Then we obtain the first stick with length $w_1 = v_1$ and the remaining length is $1 - v_1$. Secondly, we break the remaining stick $1 - v_1$ with a ratio v_2. Then we obtain a stick with length $w_2 = v_2(1 - v_1)$ and the remaining length is $(1 - v_2)(1 - v_1)$. Next we recursively break the remaining stick to obtain w_3, w_4 and so forth.

Let z_i be a cluster assignment variable, which takes value k with probability

w_k. Then (2.3.2) can be equivalently expressed as

$$
\begin{aligned}
& v_k \sim \text{Beta}(1, \alpha), \\
& w_1 = v_1, w_k = v_k \prod_{l=1}^{k-1} (1 - v_l), \\
& z_i | w \sim \text{Multinomial}(w), \\
& \theta_k | H \sim H, \\
& y_i | z_i, \theta_k \sim f(\cdot | \theta_{z_i}),
\end{aligned}
\tag{2.3.5}
$$

with $G = \sum_{k=1}^{\infty} w_k \delta_{\theta_k}$ and $\theta_i^* = \theta_{z_i}$. In the mixture model terminology, w is the mixing proportion, θ_k is the model parameters, $f(\theta_{z_i})$ is the distribution over data in cluster k, and H the prior over cluster parameters.

The Dirichlet process mixture model is a mixture model with a countably infinite number of clusters. However, because w_k decreases exponentially, only a small number of clusters will be used to model the data a priori. In the Dirichlet process mixture model, the actual number of clusters used to model data is not fixed, and can be automatically inferred from data using the usual Bayesian posterior inference framework.

2.3.3 Slice Sampling Algorithm

Next, we review the slice sampling algorithm (Walker, 2007; Kalli et al., 2011) for sampling from the Dirichlet mixture of Gaussian distributions. The idea is to introduce a latent variable u to the model to identify a finite and sufficient set of variables that is required to produce a valid Markov chain with the correct stationary distributions.

An infinite mixture model is

$$
f(y_i | w, \theta) = \sum_{k=1}^{\infty} w_k N(y_i | \theta_k). \tag{2.3.6}
$$

Walker (2007) introduced a latent variable u_i for each data point y_i,

$$f(y_i, u_i|w, \theta) = \sum_{k=1}^{\infty} \mathbf{1}(u_i < w_k) N(y_i|\theta_k). \tag{2.3.7}$$

As the sum of the mixing portion w is 1, there are only finite numbers of k satisfying $w_k > u_i$. In other words, the inference can be performed by sampling from the finite set $\{k : w_k > u_i\}$, which simplifies the problem dramatically.

Furthermore, Kalli et al. (2011) introduced a positive sequence $\xi = \{\xi_1, \xi_2, \dots\}$ into (2.3.7), and obtain

$$f(y_i, u_i|w, \theta, \xi) = \sum_{k=1}^{\infty} \frac{w_k}{\xi_k} \mathbf{1}(u_i < \xi_k) N(y_i|\theta_k). \tag{2.3.8}$$

Typically, the sequence is a deterministic, decreasing sequence but a random sequence could also be considered. The choice of ξ is a delicate issue and any choice has to balance efficiency and computational time. See Kalli et al. (2011) for more details and choices of ξ.

(2.3.8) can also be expressed using the indicator variable z,

$$f(y_i, u_i, z_i|w, \theta, \xi) = \frac{w_{z_i}}{\xi_{z_i}} \mathbf{1}(u_i < \xi_{z_i}) N(y_i|\theta_{z_i}). \tag{2.3.9}$$

Therefore, the likelihood function of the infinite mixture model is

$$\prod_{i=1}^{n} \frac{w_{z_i}}{\xi_{z_i}} \mathbf{1}(u_i < \xi_{z_i}) N(y_i|\theta_{z_i}). \tag{2.3.10}$$

With the prior $\pi(v)$ for v and prior $g_0(\theta)$ for θ, the joint distribution of (u, v, θ, y, z) is

$$\pi(v) g_0(\theta) \prod_{i=1}^{n} \frac{w_{z_i}}{\xi_{z_i}} \mathbf{1}(u_i < \xi_{z_i}) N(y_i|\theta_{z_i}). \tag{2.3.11}$$

Next, we specify the full conditional distributions for each variable and derive the slice sampling algorithm as follows:

Step 1. For u_i, we have

$$f(u_i|\cdots) \propto \mathbf{1}(u_i < \xi_{z_i}).$$

Then u_i can be sampled from a uniform distribution

$$u_i \sim U(0, \xi_{z_i}). \tag{2.3.12}$$

Step 2. We sample v_k from

$$v_k \sim \text{Beta}(1 + a_k, \alpha + b_k) \quad (k = 1, 2, \ldots, K^*). \tag{2.3.13}$$

Here, $a_k = \sum_{i=1}^{n} \mathbf{1}(z_i = k)$ denotes the number of samples in group k and $b_k = \sum_{i=1}^{n} \mathbf{1}(z_i > k)$ corresponds to the number of samples in the groups where the labels are bigger than k. $K_i = \sum_k \mathbf{1}\{\xi_k > u_i\}$ is the number of components that are used for y_i. $K^* = \max\{\sum_k \mathbf{1}\{\xi_k > u_i\}\}$ is the current number of clusters.

Then update w_k with

$$w_1 = v_1, w_k = v_k \prod_{l=1}^{k-1}(1 - v_l) \quad (k = 2, \ldots, K^*). \tag{2.3.14}$$

Step 3. We sample θ_k from the full conditional distribution

$$f(\theta_k|\cdots) \propto g_0(\theta_k) \prod_{z_i=k} N(y_i|\theta_k). \tag{2.3.15}$$

If there are no samples allocated to group k, update θ_k using the prior distribution $g_0(\theta_k)$.

Step 4. We sample z_i from the full conditional distribution

$$P(z_i = k|\cdots) \propto \mathbf{1}(\xi_k > u_i)\frac{w_k}{\xi_k} N(y_i|\theta_k). \tag{2.3.16}$$

2.4 Stochastic Block Models

Block structures are commonly observed in many real network datasets. The adjacency matrices exhibit block structures after permuting their rows and columns. Some blocks have more zeros, while some blocks consist of more ones. Stochastic block models (SBMs) are widely used to infer the block structures of networks. The model was firstly formalised by Holland et al. (1983), based on the idea of Breiger et al. (1975).

From a mixture model perspective, a SBM is in fact a mixture model where each mixture component is specified by the group membership of nodes (Nicola et al., 2022). The node membership is typically identified by their connectivity behaviour, with nodes behaving similarly belonging to the same group. The problem of inferring block structures is also called node clustering or community detection.

2.4.1 Basic SBMs

We start with the basic version of SBMs. Assume that nodes belong to K different groups, and the latent variable $Z = \{Z_1, \ldots, Z_n\}$ denotes the group membership of each node. The connecting probability between any group is denoted by a $K \times K$ matrix C, where each entry C_{kl} represents the probability between groups k and l. Given the group membership of nodes i and j, $Z_i = k$, $Z_j = l$, nodes i and j are connected with probability C_{kl},

$$P(Y_{ij} = y_{ij} | Z_i = k, Z_j = l, C) \sim \text{Bernoulli}(C_{kl}).$$

The above model implies that the edges within the same block are connected with the same probability. In fact, the block structure may describe clusters of nodes that behave similarly from a connectivity standpoint without necessarily being more densely connected, thus allowing for other types of structures, such as disassortative communities and core-periphery.

SBMs assume that edges are conditionally independent given the group membership Z. Therefore, the likelihood of the network Y is

$$P(Y|Z,C) = \prod_{i,j} P(Y_{ij} = y_{ij}|Z,C).$$

When applying SBMs to real world data, Z is usually unknown, and has to be inferred. One common assumption is that Z_i follows a multinomial distribution with parameters $w = \{w_1, \ldots, w_K\}$,

$$P(Z_i = z_i|w) \sim \text{Multinomial}(w_1, \ldots, w_K).$$

The joint probability of Y and Z is,

$$\begin{aligned} P(Y,Z|w,C) &= P(Y|Z,C)P(Z|w) \\ &= \prod_{i,j} P(Y_{ij} = y_{ij}|Z_i = k, Z_j = l, C) \prod_i P(Z_i = k|w) \\ &= \prod_{i,j} C_{kl}^{y_{ij}}(1 - C_{kl})^{1-y_{ij}} \prod_{k=1}^{K} w_k^{\sum_{i=1}^{n} 1(z_i=k)}. \end{aligned}$$

2.4.2 Mixed Membership SBMs

In the basic SBMs described above, each node belongs to one group. Airoldi et al. (2008) argued that nodes can have multiple memberships depending on who they are interacting with. They extended the single membership model to a mixed membership model.

Let s_{ij} denote the membership of node i when interacting with node j, and r_{ij} denote the membership of node j when interacting with node i. The connecting probability of pair (i,j) is determined by the corresponding entry of C and the node-specific membership,

$$P(y_{ij} = 1|s_{ij} = k, r_{ij} = l, C) = C_{kl}.$$

The generating process of mixed membership SBMs can be described as follows:

1. For each node i, sample w_i from a Dirichlet distribution with parameter vector β,

$$P(w_i|\beta) \sim \text{Dirichlet}(\beta).$$

2. For each pair (i, j),

1) sample s_{ij}, the membership indicator variable of sender node i, from a Multinomial distribution with parameter w_i,

$$P(s_{ij}|w_i) \sim \text{Multinomial}(w_i);$$

2) sample r_{ij}, the membership indicator variable of receiver node j, from a Multinomial distribution with parameter w_j,

$$P(r_{ij}|w_j) \sim \text{Multinomial}(w_j);$$

3) sample y_{ij}, the value of the edge, from a Bernoulli distribution with parameter C_{kl},

$$P(Y_{ij} = y_{ij}|s_{ij} = k, r_{ij} = l, C) \sim \text{Bernoulli}(C_{kl}).$$

2.5 Fused Lasso Regression

Suppose that we observe $(y_1, x_1), \ldots, (y_n, x_n)$, where y_i is the response variable and $x_i = (x_{i1}, \ldots, x_{ip})^\top$ is a p-dimensional predictor. We consider a regression model for the observations

$$y_i = \sum_{j=1}^{p} \theta_j x_{ij} + \epsilon_i, \text{ for } i = 1, \ldots, n, \tag{2.5.1}$$

with ϵ_i standing for the errors.

The lasso (least absolute shrinkage and selection operator) method (Tibshirani,

1996) is a regularised regression method with an L_1 penalty. It imposes a bound on the sum of the absolute value of the coefficients. The lasso finds the coefficients $\hat{\theta} = (\hat{\theta}_1, \ldots, \hat{\theta}_p)$ satisfying

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \theta_j x_{ij} \right)^2 \quad \text{subject to } \sum_{j=1}^{p} |\theta_j| \leq s_1. \tag{2.5.2}$$

The bound s_1 is a tuning parameter. For sufficiently large s_1 we obtain the least squares solution. For smaller values of s_1, we obtain spare solutions, i.e. some components are exactly 0. This is an attractive property as it estimates and selects variables that are important and discards the rest.

The fused lasso (Tibshirani et al., 2005) is proposed to consider the ordering of the predictors in addition to the sparsity,

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \theta_j x_{ij} \right)^2 \quad \text{subject to } \sum_{j=1}^{p} |\theta_j| \leq s_1 \text{ and } \sum_{j=2}^{p} |\theta_j - \theta_{j-1}| \leq s_2. \tag{2.5.3}$$

The first constraint encourages sparsity in the coefficients, and the second encourages sparsity in the differences. For sufficiently large values of s_2, we obtain the same results as lasso. For smaller values of s_2, the solutions are piecewise constant.

The literature on algorithms for the estimation of fused lasso regression is extensive. Friedman et al. (2007) derived a coordinate descent algorithm for the fused lasso regression when the design matrix is an identity matrix. Tibshirani and Taylor (2011) presented the path algorithm for the generalised lasso in the case of Gaussian regression. Friedman et al. (2010) explored coordinate descent regularisation paths for generalised linear models with convex penalties, where they estimated the logistic regression using iteratively reweighted least squares. Boyd et al. (2010) presented the theory and applications of the ADMM algorithm to the large scale optimisation problem, including fused lasso estimation.

The regularisation method has also been used in networks. On the one hand,

it is used to estimate networks, the inverse covariance matrices of the data. Hallac et al. (2017) introduced a time-varying graphical lasso model to estimate the time-varying networks from data. On the other hand, it is used to impose constraints on the differences of network model parameters. Betancourt et al. (2017) developed a Bayesian fused lasso regression model for link prediction in a time series of directed networks. They imposed an L_1 penalty on the differences of coefficients in continuous time points based on a p_1 model.

Chapter 3

Dirichlet Process Mixtures of Exponential Random Graph Models

In this chapter, we focus on the group variations exhibited in ensembles of networks. To be specific, we propose a novel Dirichlet process mixture of exponential random graph models (DPM-ERGM). It divides the ensemble into different groups and models each group of networks using a separate ERGM. By employing a Dirichlet process mixture, the number of clusters can be determined automatically and changed adaptively with the data provided. The model estimation is challenging because it requires the calculation of an infinite number of intractable ERGM likelihood. To address this issue, we develop an accurate full Bayesian inference method using the intermediate importance sampling technique, and a fast pseudo likelihood based MCMC algorithm.

3.1 Introduction

Within an ensemble, some networks are similar, generated from the same distribution, while others exhibit distinct features, sampled from different distributions.

The group representation is a powerful tool to capture the similarities and differences of network structures in the same ensemble. Durante and Dunson (2018) introduced a Bayesian method to test the differences between two given groups of networks. Lehmann and White (2021) developed a multilevel network model to compare networks from different groups. In most cases, the underlying group structure is unknown and it is therefore necessary to develop a methodology that identifies the group membership and compares groups of networks simultaneously. Signorelli and Wit (2020) introduced a model-based clustering method based on mixtures of generalized linear models for populations of networks. Yin et al. (2022) proposed a finite mixture of exponential random graph models to model the ensemble of networks based on the pseudo likelihood method. Durante et al. (2017) extended the latent space models using a Bayesian nonparametric approach.

In this chapter, we propose a Dirichlet process mixture of exponential random graph models (DPM-ERGM) for ensembles of networks. The Dirichlet process mixture model uses the Dirichlet process as a prior over an infinite mixture model, where the number of mixtures can grow adaptively with the data. This enables the model to determine the group structure of the ensemble automatically, or more precisely, to compare different networks without prior knowledge of the number of clusters. Moreover, the Dirichlet process provides a large sample space and tractable posterior distributions, facilitating inference on the infinite sample space (Ferguson, 1973). On the other hand, the versatile ERGM is employed to model networks for its ability to represent various types of topological features. Thus, The DPM-ERGM is capable of determining the group structure and describing the group characteristics of an ensemble simultaneously.

Under the Bayesian nonparametric framework, performing Bayesian inference of the proposed model involves the evaluation and comparison of an infinite number of ERGMs. The intractability of ERGMs increases the difficulty of the estimation dramatically. To sample from the infinite sample space, we introduce a latent variable to the model, which helps us to find a finite set of components

required to produce the correct Markov chain, borrowing the idea of slice sampling algorithm (Walker, 2007). The inference can then be performed by sampling from the full conditional distributions of all variables on a finite space. However, for each network sample, the sampling of the membership variable requires model comparison between different group representations. This is challenging because the current method to estimating ERGM likelihood relies on approximating the intractable normalising constant ratio of two parameters that are close to each other. Parameters from different clusters can be quite far away from each other because they represent networks of different characteristics, and the parameters of empty groups can be even further from the rest.

One way to sample from the posterior distributions of ERGMs is to use Metropolis-Hastings algorithms. Standard Metropolis-Hastings algorithms are not applicable since the acceptance probability depends on the intractable normalising constants. To address this issue, Caimo and Friel (2011) applied the exchange algorithm (Murray et al., 2006), where a perfect sampler (Propp and Wilson, 1996) is employed to facilitate the Metropolis-Hastings algorithm, avoiding the calculation of the intractable normalising constant. As the perfect sampler for ERGMs is computational expensive, a sample from the MCMC method is used in practice. Liang and Jin (2013) developed a Monte Carlo Metropolis-Hastings (MCMH) algorithm to sample from the intractable posterior distributions. The algorithm is implemented by approximating the unknown normalising constant ratio in the acceptance probability using a Monte Carlo estimate and is proved to converge to the desired target distribution under some conditions. The exchange algorithm can be seen as a special case of the MCMH algorithm. However, most of the literature on ERGMs only deals with the single network situation. In our proposed DPM-ERGMs, networks from the same group are multiple samples from the same ERGM distribution. This requires the Bayesian inference to have the ability of incorporating multiple network samples.

To sample from the posterior distributions of DPM-ERGMs, we develop a

Metropolis-within-slice sampling algorithm that employs Metropolis-Hastings inside the slice sampling algorithm. Specifically, we extend the MCMH algorithm to a multi-network MCMH (MMCMH) algorithm in order to update the ERGM parameters that represent multiple networks from the same group. An importance sampling estimator with intermediate values is used in MMCMH to approximate the normalising constant ratio in the acceptance probability to ensure the accuracy of the estimation. In this way, the characteristics of the whole group can be captured by pooling information across networks. Posterior samples of membership variables also suffer from the issue of intractability. We express the membership variable distributions in such a way that a ratio of normalising constants is obtained, and employ an intermediate importance sampling estimator to approximate the constructed ratio. We refer to the combined algorithm as intermediate importance Metropolis-within-slice (IMS) algorithm. The IMS algorithm allows the full Bayesian inference to be performed based on the full likelihood, and is capable of modelling complex dependency structures beyond dyad independence. Moreover, we can replace the full likelihood with the pseudo likelihood function in the Metropolis-within-slice scheme to achieve a faster, approximate estimation.

The rest of chapter is organised as follows. In Section 3.2, we describe how the DPM-ERGMs are formulated. Section 3.3 provides the sampling methodology. Section 3.4 presents the simulation studies. We summarise this chapter in Section 3.5.

3.2 Dirichlet Process Mixtures of ERGMs

Ensembles of networks include multiple network observations. In addition to the complex structures within each network, we are also interested in studying the variations across different networks using a group representation. A mixture model is a natural approach to describe such a population as it can detect and characterise subpopulations that share common structures and distinguish networks that are different automatically. In particular, the infinite mixture model

is applied here because the number of mixture components can be adjusted to the data. Here, we propose to model the ensemble of networks through an infinite mixture of ERGMs, where each component represents a network cluster using a cluster-specific ERGM.

An ensemble with N network samples is denoted by $\{Y_i\}_{i=1}^N$, and the corresponding covariate information is $\{X_i\}_{i=1}^N$. In such an ensemble, the single network Y_i is represented using an infinite mixture of ERGMs as follows

$$P_{w,\theta}(Y_i = y_i \,|\, X_i) = \sum_{j=1}^{\infty} w_j \frac{\exp\{\theta_j^\top S(y_i, X_i)\}}{k(\theta_j)}, \tag{3.2.1}$$

where j is the cluster label, w_j is the mixing proportion, $\theta_j \in \mathbb{R}^d$ is the cluster specified parameter vector, $S(y_i, X_i)$ is the summary statistics of network y_i, and $k(\theta_j) = \sum_{y \in \mathcal{Y}} \exp\left\{\theta_j^\top S(y, X_i)\right\}$ is the normalising constant. Without requiring a fixed number of clusters in advance, the infinite mixture model is able to determine the number of clusters adaptively with the data provided.

The likelihood of the ensemble of networks can be expressed as

$$P_{w,\theta}(\{Y_i = y_i\}_{i=1}^N \,|\, \{X_i\}_{i=1}^N) = \prod_{i=1}^{N} \sum_{j=1}^{\infty} w_j \frac{\exp\{\theta_j^\top S(y_i, X_i)\}}{k(\theta_j)},$$

or

$$P_{\theta}(\{Y_i = y_i\}_{i=1}^N \,|\, \{X_i, Z_i = k_i\}_{i=1}^N) = \prod_{i=1}^{N} \frac{\exp\{\theta_{k_i}^\top S(y_i, X_i)\}}{k(\theta_{k_i})},$$

where $Z = (Z_1, Z_2, \ldots, Z_N)$ is a latent variable to indicate the membership of each network, e.g. $Z_i = k_i$ if y_i belongs to cluster k_i. It is informative to consider an infinite mixture model especially when it is not appropriate to have a limit on the number of groups. However, the inference of this model is challenging because the intractable normalising constant has to be evaluated in the infinite sample space.

To perform Bayesian inference on the proposed infinite mixture of ERGMs, we adopt a Dirichlet process prior $\mathrm{DP}(\beta, \mathrm{H})$ (Ferguson, 1973), which is arguably

the most commonly used Bayesian nonparametric prior. Under the constructive definition, also known as the stick-breaking representation (Sethuraman, 1994), the mixing proportion w is constructed using a stick-breaking procedure with an auxiliary variable v. A sequence of independent and identically distributed auxiliary variables $v_1, v_2, \dots$ is sampled from a prior distribution Beta$(1, \beta)$, and the mixing proportions are set as $w_1 = v_1$, $w_j = v_j \prod_{l=1}^{j-1}(1 - v_l)$ $(j = 2, 3, \dots)$. The membership indicator variable Z_i follows a multinomial distribution with probability $w = (w_1, w_2, \dots)$. Given the membership $Z_i = k_i$, the network Y_i is modelled by an ERGM with parameter θ_{k_i}. The choice of prior distributions for Bayesian ERGMs has not been studied in detail, and studies have generally assumed a multivariate normal distributions on the model parameters (Caimo and Friel, 2011; Bouranis et al., 2018). Here, we also use a multivariate normal distribution $\mathcal{N}(\mu_0, \Sigma_0)$ for the prior of ERGM parameter θ_j.

In summary, we propose a Dirichlet process mixture of exponential random graph models (DPM-ERGM) with the following specification,

$$
\begin{aligned}
& v_j \sim \text{Beta}(1, \beta) \quad (j = 1, 2, \dots), \\
& w_1 = v_1, w_j = v_j \prod_{l=1}^{j-1}(1 - v_l), \\
& Z_i | w \sim \text{Multinomial}(w) \quad (i = 1, 2, \dots, N), \\
& \theta_j | \mu_0, \Sigma_0 \sim \mathcal{N}(\mu_0, \Sigma_0), \\
& y_i | Z_i = k_i, \theta \sim P_{\theta_{k_i}}(Y_i = y_i \,|\, X_i), \\
& P_{\theta_{k_i}}(Y_i = y_i \,|\, X_i) = \frac{\exp\{\theta_{k_i}^\top S(y_i, X_i)\}}{k(\theta_{k_i})}.
\end{aligned}
\tag{3.2.2}
$$

3.3 Posterior Computation

The statistical inference for the proposed model is very challenging due to the infinite number of mixture components and the intractable ERGM likelihood. In this section, we first develop a Metropolis-within-slice sampling algorithm to address

the issue of sampling from the infinite sample space of DPM-ERGMs. Afterwards, we provide details of the algorithms based on a full and pseudo likelihood approach separately.

The slice sampling algorithm (Walker, 2007; Kalli et al., 2011) provides a way to sample from the infinite mixture models. Similar to the slice sampling, we first introduce a latent variable u to our proposed model to identify the exact number of components that are required to produce a valid Markov chain with the correct stationary distributions. The joint density of (y, u) is written as

$$P_{w,\theta}(Y = y, u \,|\, X, \xi) = \sum_{j:\xi_j > u} \frac{w_j}{\xi_j} P_{\theta_j}(Y = y|X).$$

The inference can be performed by sampling from the finite set $\{j : \xi_j > u\}$, which simplifies the problem dramatically. ξ is a deterministic decreasing sequence used to address the update of u. See Kalli et al. (2011) for details and choices of ξ.

Furthermore, with the indicator variable Z, the joint density can be expressed as

$$P_{w,\theta}(Y = y, u, Z = k \,|\, X, \xi) = \frac{w_k}{\xi_k} \mathbf{1}(u < \xi_k) P_{\theta_k}(Y = y|X).$$

Hence, the likelihood for the ensemble $\{Y_i\}_{i=1}^N$ with latent variable u and sequence ξ is

$$l_{w,\theta}(\{Y_i = y_i, Z_i = k_i, u_i\}_{i=1}^N \,|\, \{X_i\}_{i=1}^N, \xi) = \prod_{i=1}^N \frac{w_{k_i}}{\xi_{k_i}} \mathbf{1}(u_i < \xi_{k_i}) P_{\theta_{k_i}}(Y_i = y_i|X_i). \tag{3.3.1}$$

With the prior distribution specified in (3.2.2), the full conditional distributions of all variables (u, w, θ, Z) are available. The Metropolis-within-slice sampling scheme is performed by sampling (u, w, θ, Z) from their full conditional distributions in turn. In particular, as the direct sampling from ERGMs is not possible, Metropolis-Hastings algorithm is used to assist the sampling of θ.

3.3.1 Full Likelihood Based Algorithm

In order to overcome the intractability issue and perform accurate estimation to the original model, we propose to employ the intermediate importance sampling technique in the Metropolis-within-slice sampling scheme, and name this method as IMS algorithm. The sampling procedure of the full likelihood based IMS algorithm is as follows:

Step 1. Sample u_i from a uniform distribution,

$$u_i \sim U(0, \xi_{k_i}) \quad (i = 1, 2, \ldots, N), \tag{3.3.2}$$

where k_i is the current allocation of network y_i.

Step 2. Sample v_j from a beta posterior distribution,

$$v_j \sim \text{Beta}(1 + a_j, \beta + b_j) \quad (j = 1, 2, \ldots, K^*). \tag{3.3.3}$$

Here, $a_j = \sum_{i=1}^{N} \mathbf{1}(k_i = j)$ denotes the number of networks in group j and $b_j = \sum_{i=1}^{N} \mathbf{1}(k_i > j)$ corresponds to the number of networks in the groups whose label are bigger than j. K^* denotes the current number of clusters.

Update w_j with

$$w_1 = v_1, w_j = v_j \prod_{l=1}^{j-1} (1 - v_l) \quad (j = 2, \ldots, K^*). \tag{3.3.4}$$

Step 3. Sample Z_i with the following two steps:

(1) Introduce $k(\theta_c)$ to construct a computable normalising constant ratio and estimate the normalising constant ratio $k(\theta_c)/k(\theta_j)$ $(j = 1, \ldots, K^*)$ using an intermediate importance sampling estimator γ_j.

(2) Calculate the conditional probability with the intermediate importance

sampling estimator replacement,

$$P(Z_i = k_i | \cdots) \propto \mathbf{1}(\xi_{k_i} > u_i) \frac{w_{k_i}}{\xi_{k_i}} \cdot \exp\{\theta_{k_i}^\top S(y_i, X_i)\} \cdot \gamma_{k_i}, \quad (i = 1, 2, \ldots, N). \tag{3.3.5}$$

Here, θ_{k_i} is the parameter of group k_i.

Step 4. Sample θ_j $(j = 1, 2, \ldots, K^*)$ using the MMCMH algorithm with the following substeps:

(1) Draw θ_j' from a proposal distribution $h(\cdot|\theta_j)$.

(2) Simulate m_2 networks from each intermediate distribution with parameter θ_r^{im} $(r = 0, 1, \ldots, m_1)$ individually and store the network statistics using $S(z_r^s)$ $(s = 1, 2, \ldots, m_2)$, where θ_r^{im} $(r = 1, 2, \ldots, m_1)$ are m_1 intermediate values between $\theta_0^{im} = \theta_j$ and $\theta_{m_1+1}^{im} = \theta_j'$.

(3) Estimate the normalising constant ratio $k(\theta_j')/k(\theta_j)$ with an intermediate importance sampling estimator γ.

(4) Accept θ_j' with probability

$$\alpha = \min\left(1,\ \frac{\pi(\theta_j')h(\theta_j|\theta_j')}{\pi(\theta_j)h(\theta_j'|\theta_j)} \frac{\exp\{(\theta_j' - \theta_j)^\top \sum_{z_i=j} S(y_i, X_i)\}}{\gamma^{\sum_i \mathbf{1}(z_i=j)}}\right). \tag{3.3.6}$$

If there are no networks allocated to group j, update θ_j using prior $\pi(\theta_j)$.

Next, we show the construction of formula (3.3.5) in sampling Z (Step 3), and explain how the MMCMH algorithm is developed to sample θ (Step 4).

Details of Step 3

The full conditional distribution of Z_i is

$$P(Z_i = k_i \,|\, \cdots) \propto \mathbf{1}(\xi_{k_i} > u_i) \frac{w_{k_i}}{\xi_{k_i}} \frac{\exp\{\theta_{k_i}^\top S(y_i, X_i)\}}{k(\theta_{k_i})}. \tag{3.3.7}$$

The ratio on the right hand side depends on an intractable normalising constant $k(\theta_{k_i})$, which makes the direct sampling infeasible.

Gelman and Meng (1998) provides a way to estimate the normalising constant ratio using the importance sampling technique,

$$\frac{k(\theta_a)}{k(\theta_b)} \approx \frac{1}{m_2}\sum_{s=1}^{m_2}\exp\{(\theta_a-\theta_b)^\top S(z^s)\}, \tag{3.3.8}$$

with z^s $(s = 1, 2\ldots, m_2)$ denoting a sequence of m_2 independent auxiliary networks sampled from the ERGM with parameter θ_b. However, the importance sampling estimate are incorrect if the compared parameters θ_a and θ_b are not close enough (Neal, 2005). This obstacle can be overcome by introducing intermediate distributions between θ_a and θ_b. Specifically, we interpolate m_1 intermediate values, θ_r^{im} $(r = 1, 2, \ldots, m_1)$, so that θ_r^{im} and θ_{r+1}^{im} are close enough, and factorise the normalising constant ratio using intermediate values,

$$\frac{k(\theta_a)}{k(\theta_b)} = \prod_{r=0}^{m_1}\frac{k(\theta_{r+1}^{im})}{k(\theta_r^{im})} = \frac{k(\theta_1^{im})}{k(\theta_0^{im})}\frac{k(\theta_2^{im})}{k(\theta_1^{im})}\cdots\frac{k(\theta_{m_1+1}^{im})}{k(\theta_{m_1}^{im})}, \tag{3.3.9}$$

where $\theta_0^{im} = \theta_b$ and $\theta_{m_1+1}^{im} = \theta_a$. Then, each factor $k(\theta_{r+1}^{im})/k(\theta_r^{im})$ is estimated using the importance sampling estimator, and $k(\theta_a)/k(\theta_b)$ is approximated by

$$\gamma = \prod_{r=0}^{m_1}\frac{1}{m_2}\sum_{s=1}^{m_2}\exp\{(\theta_{r+1}^{im}-\theta_r^{im})^\top S(z_r^s)\}. \tag{3.3.10}$$

where z_r^s $(s = 1, 2, \ldots, m_2)$ is a sequence of m_2 independent networks sampled from the ERGM with parameter θ_r^{im}.

If we can construct a normalising constant ratio in the posterior membership probability, we will be able to borrow the strength of intermediate importance sampling to allocate the network samples. To do so, we multiply a constant $k(\theta_c)$ to each term of the posterior probability vector and obtain

$$P(Z_i = k_i \,|\, \cdots) \propto \mathbf{1}(\xi_{k_i} > u_i)\frac{w_{k_i}}{\xi_{k_i}}\cdot\exp\{\theta_{k_i}^\top S(y_i, X_i)\}\frac{k(\theta_c)}{k(\theta_{k_i})},$$

where the constructed normalising constant ratios $k(\theta_c)/k(\theta_{k_i})$ $(k_i = 1, 2, \ldots, K^*)$ can be approximated using the intermediate importance sampling estimation as

shown in (3.3.10). Thus, the posterior probability ratios do not change and the sampling can be performed.

The choice of θ_c is important to the accuracy of the intermediate importance sampling estimation. The estimation are incorrect if the parameters to be compared, θ_c and θ_j, are not close enough. As each group has a unique θ_j, it is impossible to find one θ_c close to all θ_j at the same time. Simple importance sampling is not applicable here and multiple intermediate values must be used to ensure the quality of estimation. Also, some θ_{k_i} can be quite difficult to sample from, especially when it is representing an empty group.

Details of Step 4

The posterior distribution of group parameter θ_j is proportional to the product of prior $\pi(\theta_j)$ and the joint likelihood of the networks in group j, which is

$$f(\theta_j|\cdots) \propto \pi(\theta_j) \prod_{Z_i=j} \frac{\exp\{\theta_j^\top S(y_i, X_i)\}}{k(\theta_j)}. \tag{3.3.11}$$

Sampling from such a posterior distribution is challenging as it depends on the product of multiple intractable likelihood functions.

The use of MCMC algorithm to sample from this posterior distribution of θ_j involves the calculation of $k(\theta_j')/k(\theta_j)$. As the product of the multiple normalising constant ratios has to be calculated, it is necessary to have a more accurate estimation for each $k(\theta_j')/k(\theta_j)$. To achieve this, we use the intermediate importance sampling estimator γ as in (3.3.10) to substitute $k(\theta_j')/k(\theta_j)$.

Therefore, to sample from (3.3.11) using MMCMH algorithm, we propose θ_j' from $h(\cdot|\theta_j)$, and accept θ_j' with probability

$$\frac{\pi(\theta_j')h(\theta_j|\theta_j')}{\pi(\theta_j)h(\theta_j'|\theta_j)} \frac{\exp\{(\theta_j' - \theta_j)^\top \sum_{z_i=j} S(y_i, X_i)\}}{\gamma^{\sum_i \mathbf{1}(z_i=j)}}. \tag{3.3.12}$$

With the approximation to the normalising constant ratio available, the acceptance ratio is calculable and thus the posterior sampling is feasible. Compared

with importance sampling, the use of intermediate values increases the quality of estimation by introducing intermediate distributions. Similar techniques like annealed importance sampling and linked importance sampling (Neal, 2005) can be used as well.

3.3.2 Pseudo Likelihood Based Algorithm

In addition to the full likelihood approach in 3.3.1, we also propose a fast estimation method based on the pseudo likelihood (Strauss and Ikeda, 1990), which is an approximation to the full likelihood. To be specific, the algorithm is developed by employing a pseudo likelihood approximation in the Metropolis-within-slice sampling algorithm. We name this pseudo likelihood based algorithm as the PMS algorithm. In the PMS algorithm, (u, w) are sampled in the same way as in the IMS algorithm, and (θ, Z) are updated with pseudo likelihood replacement.

The pseudo likelihood method approximates the full likelihood using the product of conditional probabilities of all edges in a network,

$$PL_\theta\left(Y = y \,|\, X\right) = \prod_{r \neq s} P\left(y_{rs} = 1 \,|\, y_{-rs}, X\right)^{y_{rs}} \left\{1 - P\left(y_{rs} = 1 \,|\, y_{-rs}, X\right)\right\}^{1-y_{rs}},$$

where $y_{-rs} = \{y_{kl}, (k,l) \neq (r,s)\}$ denotes all the dyads of the graph excluding y_{rs}. The Change statistic is defined to measure the change of summary statistics when the edge y_{rs} is changed from disconnected to connected, $\Delta S_{rs} = S(y_{rs} = 1, y_{-rs}, X) - S(y_{rs} = 0, y_{-rs}, X)$. The edge variable y_{rs} follows a Bernoulli distribution with probability expressed using change statistics,

$$P\left(y_{rs} = 1 | y_{-rs}, X\right) = \frac{\exp(\theta^\top \Delta S_{rs})}{1 + \exp(\theta^\top \Delta S_{rs})}.$$

If we replace the full likelihood with use pseudo likelihood, then the acceptance

ratio for sampling θ_j using the Metropolis-Hastings algorithm is

$$\frac{\pi(\theta'_j)h(\theta_j|\theta'_j)}{\pi(\theta_j)h(\theta'_j|\theta_j)} \cdot \frac{\prod_{z_i=j} PL_{\theta'_j}(Y_i = y_i|X)}{\prod_{z_i=j} PL_{\theta_j}(Y_i = y_i|X)}, \tag{3.3.13}$$

and the posterior probability of cluster membership Z_i is proportional to

$$\mathbf{1}(\xi_j > u_i)\frac{w_j}{\xi_j} \cdot PL_{\theta_j}(Y_i = y_i). \tag{3.3.14}$$

Thus, the sampling of θ, Z is possible with the pseudo likelihood replacement.

The PMS algorithm is faster than the IMS algorithm, but it is less accurate. The major issue is that it may underestimate the endogenous network formation process, since pseudo likelihood only uses local information within a whole graph (van Duijn et al., 2009). Moreover, when the model is near-degenerate, posterior samples from pseudo likelihood method may fall into the degenerate region (Caimo and Friel, 2011).

3.4 Empirical Results

In this section, we first present some examples on the use of the intermediate importance sampling. Then we show the performance of proposed methods using simulated networks. Finally, we apply DPM-ERGMs to two real network ensembles and discuss the results of the application. The network samples are generated using the `R` package `ergm` (Hunter et al., 2008).

3.4.1 Intermediate Importance Sampling

Here, we use simulation studies to show how the number of intermediate distributions m_1 and the number of auxiliary networks m_2 affect the normalising constant ratio approximation with varying distances between the compared parameters.

We first show that how the estimation changes with different values of m_1 and m_2 when the compared parameters are distant. To do so, we sample two

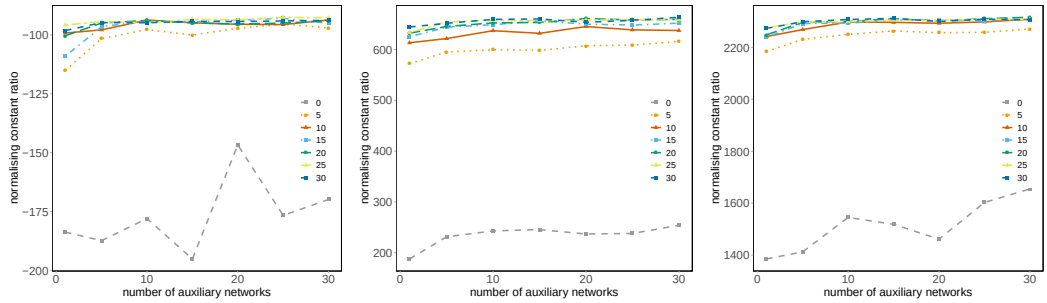

Figure 3.4.1: Estimation of normalising constant ratios when the compared parameters are far from each other. The three plots correspond to three repetitions of the parameters.

parameters θ_1, θ_2 independently from the prior $\mathcal{N}(\mu_0, \Sigma_0)$ and estimate $k(\theta_2)/k(\theta_1)$ with different values of m_1, m_2. The results of three repetitions are shown in the three plots of Figure 3.4.1 separately. In each plot, lines with different colours correspond to different numbers of intermediate values m_1 and x-axis represents different numbers of auxiliary variables m_2. As is shown, the line of $m_1 = 0$ is far from the other lines, meaning that the estimation is incorrect and intermediate distributions have to be used to get a good estimation.

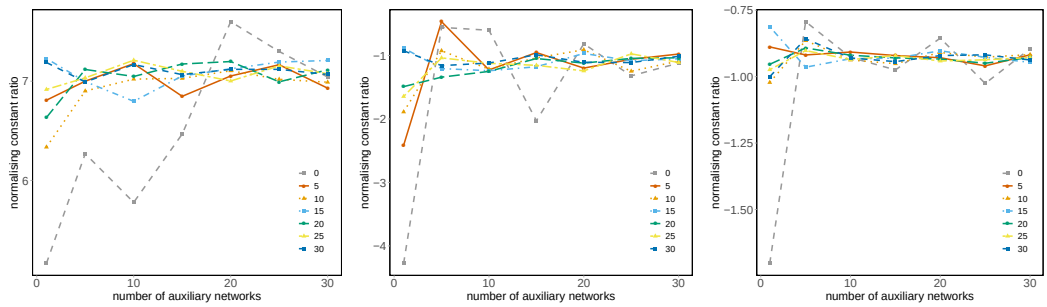

Figure 3.4.2: Estimation of normalising constant ratios when the compared parameters are close to each other. The three plots correspond to three repetitions of the parameters.

In the second simulation, we show how m_1, m_2 affect the intermediate importance sampling estimation when the compared parameters are close. Here, we generate a sample θ_1 from the prior $\mathcal{N}(\mu_0, \Sigma_0)$, and propose θ_2 from a normal

distribution $\mathcal{N}(\theta_1, \Sigma_p)$. Then we estimate normalising constant ratio $k(\theta_2)/k(\theta_1)$ with different m_1, m_2, and show the estimation in Figure 3.4.2. As we can see, all lines merge together with increasing m_1 and m_2, indicating that the intermediate importance sampling estimation is consistent. The simple importance sampling estimation (the line with $m_1 = 0$) has big variations, and intermediate importance sampling estimators (lines with $m_1 > 0$) are more stable.

We recommend $m_1 = 2, m_2 = 10$ for MMCMH algorithm and $m_1 = 5, m_2 = 10$ for the posterior membership sampling as initial values, and similar techniques can be applied to choose m_1, m_2 in the specific dataset.

3.4.2 Simulated Networks

Simulation 1

In this simulation study, we focus on comparing the clustering accuracy of different methods. Firstly, we choose three most commonly used network sufficient statistics for the ERGM distribution,

- $S^1(y_i) = \sum_{r \neq s} y_{rs,i}$, the total number of edges in the network.
- $S^2(y_i) = e^{\phi} \sum_{k=1}^{n-2} \{1 - (1 - e^{-\phi})^k\} EP_k(y_i)$, $\phi = 0.25$, geometrically weighted edgewise shared partner, GWESP, a representation for transitivity. $EP_k(y_i)$ is the number of connected pairs that have k common neighbours.
- $S^3(y_i) = \sum_{r \neq s} y_{rs,i} \mathbf{1}(X_r = X_s)$, the total number of connections between individuals with the same covariate.

X is a binary covariate with half of nodes taking value 0 and the other half taking 1. Then we simulate networks from mixtures of ERGM distributions under four different scenarios. In the first scenario, we consider an ensemble with $N = 30$ networks and $K = 3$ balanced groups. The number of nodes in each network is $n = 40$. In the second scenario, we keep the same ensemble size but increase the network size to $n = 100$. In the third scenario, we consider a larger ensemble with $N = 80$ networks and $K = 4$ groups. The number of networks in each group is

25, 25, 25 and 5. The network size of scenario 3 is $n = 40$. Similarly, the scenario 4 has $N = 80$ networks and $K = 4$ groups with a larger network size $n = 100$. The parameter values θ for each group under different scenarios are specified as follows,

$$\theta^{(40,30)} = \begin{pmatrix} -0.85 & -0.10 & -0.10 \\ -3.45 & 0.75 & 2 \\ -5.10 & 2.5 & 0.5 \end{pmatrix}, \ \theta^{(100,30)} = \begin{pmatrix} -2.03 & -0.10 & -0.10 \\ -4.15 & 0.75 & 2 \\ -5.85 & 2.5 & 0.5 \end{pmatrix},$$

$$\theta^{(40,80)} = \begin{pmatrix} -0.85 & -0.10 & -0.10 \\ -3.45 & 0.75 & 2 \\ -5.10 & 2.5 & 0.5 \\ -2.00 & 0.20 & 1.0 \end{pmatrix}, \ \theta^{(100,80)} = \begin{pmatrix} -2.03 & -0.10 & -0.10 \\ -4.15 & 0.75 & 2 \\ -5.85 & 2.5 & 0.5 \\ -3.00 & 0.20 & 1.0 \end{pmatrix}.$$

Next, we apply the proposed infinite mixture models to the synthetic ensembles. The prior of variable v is a beta distribution Beta$(1, 0.1)$. The prior of ERGM parameters θ is selected to be a multivariate normal distribution $\mathcal{N}(\mu_0, \Sigma_0)$ with $\mu_0 = (-3, 0, 0)$, $\Sigma_0 = 4^2 I_p$, where I_p is a p dimension diagonal matrix, with p denoting the number of sufficient statistics. The proposal distribution in the Metropolis-Hastings algorithm is $\mathcal{N}(0, \Sigma_p)$, $\Sigma_p = 0.05^2 I_p$. For sequence $\xi_1, \xi_2, \ldots,$ we use an exponential decreasing sequence, $\xi_i = e^{-i}$. K_i, the number of components that satisfies $\{j : \xi_j > u_i\}$, is also the smallest integer that satisfies $\{e^{-K_i} > u_i\}$, thus $K_i = \lfloor -log(u_i) \rfloor$. We start with all networks in one group with initial value $\theta_0 = (-2, 0)$ and choose $m_1 = 2, m_2 = 10$ in the MMCMH step and $m_1 = 5, m_2 = 10$ for the sampling of membership variables.

For each scenario, we repeat the same process for 50 times. After running the simulation for 100,000 iterations with 60,000 burn-in and 50 thinning, we report the accuracy of the estimated number of clusters from four different methods in Table 3.4.1. Furthermore, we also evaluate the accuracy of the cluster memberships using Rand index (RI) (Rand, 1971) and adjusted Rand index (ARI). The

Table 3.4.1: Estimation accuracy of K across 50 replicates for four experimental conditions with different estimation methods.

		Accuracy of $\hat{K}$		Average $\hat{K}$	
(n, N)	K	IMS	PMS	IMS	PMS
(40, 30)	3	1	0.56	3	3.40
(100, 30)	3	1	0.54	3	3.52
(40, 80)	4	1	0.26	4	5.26
(100, 80)	4	1	0.46	4	4.60

RI takes values between 0 and 1, with 0 indicating that the two data clusterings do not agree on any pair of points and 1 stands for perfect match. The ARI is the adjusted-for-chance version of the RI. Random labellings have an ARI close to 0 and 1 indicates that the data clusterings are exactly the same. We compare the clustering results of each iteration with the true cluster membership and calculate the average ARI and RI across 50 replicates. The average ARI and average RI are shown in Table 3.4.2.

Table 3.4.2: Average ARI and RI across 50 replicates for four experimental conditions with different estimation methods.

		Average ARI		Average RI	
(n, N)	K	IMS	PMS	IMS	PMS
(40, 30)	3	1	0.926	1	0.968
(100, 30)	3	1	0.944	1	0.977
(40, 80)	4	0.992	0.841	0.997	0.939
(100, 80)	4	1	0.928	1	0.972

From Table 3.4.1, we can see that the IMS method performs very well with respect to the estimation accuracy for the number of clusters K, with an average accuracy of almost 100%. The estimation accuracy of the PMS method is lower as it tends to give a high estimation for K. From Table 3.4.2, we can see that both methods perform well in regard to the accuracy of the cluster memberships. The full likelihood method IMS performs better than the pseudo likelihood method

PMS. After further investigation, we find that although the PMS method provides a higher estimation for K, it only divides the original group into more than one groups. As it does not mix networks from different groups, ARI values from PMS are still satisfactory despite the high estimates on $\hat{K}$.

Table 3.4.3: Average computation time (in seconds) per iteration across 50 replicates.

		Average Time	
(n, N)	K	IMS	PMS
(40, 30)	3	2.4230	0.0244
(100, 30)	3	3.0188	0.0459
(40, 80)	4	2.7854	0.0574
(100, 80)	4	3.4580	0.1196

We also display the average computation time of the proposed methods in Table 3.4.3. The computation is run using a single core (Intel Core i5-11500 @ 2.70GHz). The full likelihood based method IMS takes longer time. IMS aims to get the exact estimation to the model and thus requires estimating the intractable normalising constants at each iteration of the updating procedure. This significantly increases the computation time. The pseudo likelihood method PMS is faster as the calculation of the pseudo likelihood is much simpler than the normalising constant.

Simulation 2

We next concentrate on assessing the quality of ERGM parameter estimation. An ensemble of $N = 40$ undirected networks are generated from a mixture model with $K = 2$ groups. Two statistics are used to describe the networks, the number of edges $S^1(y) = \sum_{i<j} y_{ij}$ to reflect on the network density and the number of triangles $S^2(y) = \sum_{i<j<k} y_{ij}y_{jk}y_{ik}$ to represent the transitivity. The mixing proportion is $w_{true} = (0.5, 0.5)$. The network size is $n = 30$. The ERGM parameters

for group 1 are selected as $\theta^1_{true} = (-3, 0.9)$, which has low density and high transitivity parameter, meaning that some edges are generated because of endogenous formation process. The second group parameter is $\theta^2_{true} = (-1, 0)$, representing Bernoulli networks which have independent edges.

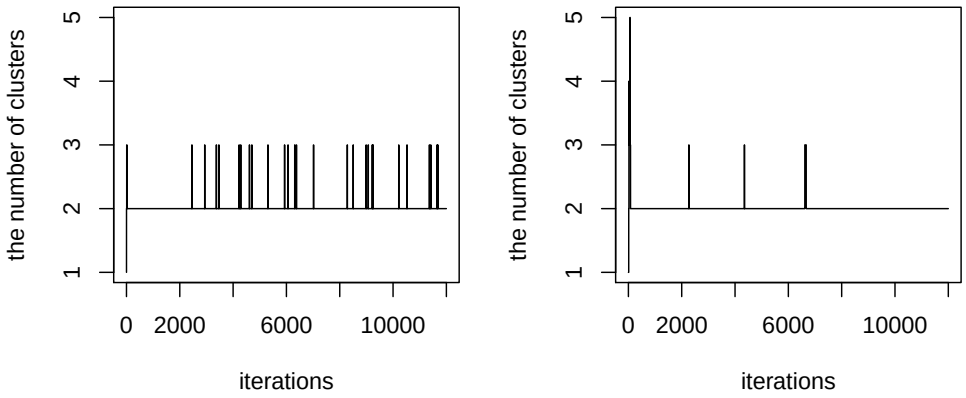

Figure 3.4.3: Clustering results of IMS algorithm (left), and PMS algorithm (right).

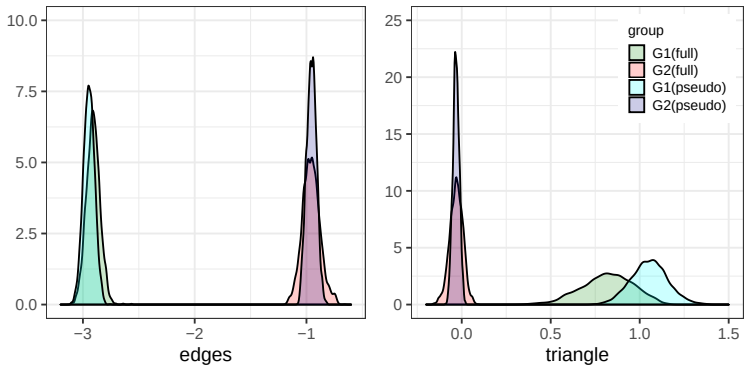

Figure 3.4.4: Density plots of ERGM parameters after 2,000 burn-in.

We apply both the IMS and PMS algorithm to the synthetic ensemble. The simulation is run for 12,000 iterations with the first 2,000 iterations as burn-in. The clustering results are shown in Figure 3.4.3. Both IMS and PMS algorithms are able to detect the true group memberships of all networks correctly. The acceptance ratio is 0.60 for group 1, 0.27 for group 2 using IMS algorithm. The acceptance ratio is 0.62 for group 1, 0.18 for group 2 using PMS algorithm. The posterior density plots are displayed in Figure 3.4.4. As we can see, the triangle

estimator of group 1 from IMS is smaller than PMS. This confirms the finding of van Duijn et al. (2009) that pseudo likelihood method tends to underestimate the endogenous network formation process. For the Bernoulli networks in group 2, the pseudo likelihood method underestimates the parameter variance and provides a narrower interval. This is consistent with the finding of Bouranis et al. (2017).

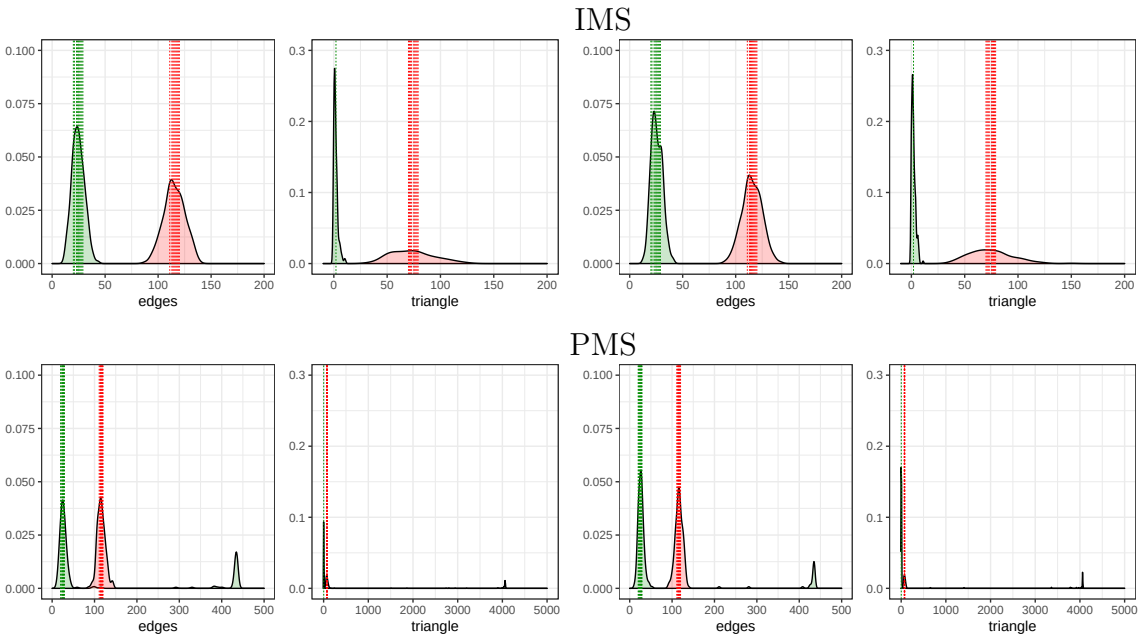

Figure 3.4.5: Density plots of simulated network statistics generated from full likelihood estimation and pseudo likelihood estimation. The left two plots are simulated from 200 different posterior samples, while the right two are sampled from the posterior mean. The vertical lines stand for the value of structure statistics of the simulated networks. Different colour represents different groups.

In order to further assess the quality of estimation, we simulate networks based on the estimation. Specifically, we firstly simulate 200 networks, each from one of the 200 different posterior samples obtained after 2,000 burn-in and 50 thinning, then we simulate another 200 networks from the posterior mean. The simulated network statistics from full likelihood is shown in the first row of Figure 3.4.5, with statistics from the posterior samples on the left side and statistics from the posterior mean on the right side. The observed network statistics is covered well by the simulated network statistics, indicating that the estimator is a good fit to the data. However, in the second row of Figure 3.4.5, there are significant amount

of full graphs (graphs with 435 edges) simulated from the pseudo estimation, because posterior samples from PMS method have degenerate parameter values.

In this simulation, we applied both the IMS and PMS algorithms to the synthetic network ensemble. The IMS algorithm provided accurate estimation to the model. PMS algorithm clustered all the network samples correctly, but the estimated model for group 1 failed to generate networks resembling the observed graphs.

3.4.3 Krackhardt's Advice Networks

We next apply the proposed DPM-ERGMs to an advice network ensemble. Krackhardt (1987) studied a sequence of 21 networks about 21 employees in a high-tech machine manufacturing firm. The networks are constructed based on the data collected from a survey on the query "Who does A go to for advice and help with work?" Everyone is asked not only the advice relationship of themselves but also other people. Therefore, a collection of 21 perception networks $y_i (i = 1, 2, \ldots, 21)$ is built where every network represents an individual's perspective about the advice relationships among the 21 individuals. $y_{rs,i} = 1$ indicates that individual i thinks that individual r asks s for help. The covariate information, the job level of each individual is represented by a vector X. The original paper focuses on exploring the differences of perception networks through node centrality scores to measure the importance of the nodes. Here, we are interested in learning the differences and similarities of the perception networks using the mixture of ERGMs. In this way, the generating mechanism of the perception networks can be analysed. This helps us to better understand the perception network relationships. For the structure statistics, we choose the following:

- $S^1(y_i) = \sum_{r \neq s} y_{rs,i}$, the total number of edges in the network. This reflects on the communication strength.

- $S^2(y_i) = \sum_{r \neq s} y_{rs,i} \mathbf{1}(X_r = X_s)$, the total number of connections between

individuals in the same level. The positive coefficient indicates that people tend to ask for help from people of the same level, while the negative coefficient means that more help is sought from others in a different level.

- $S^3(y_i) = e^{\phi} \sum_{k=1}^{n-2} \{1 - (1 - e^{-\phi})^k\} DP_k(y_i)$, $\phi = 0.25$, geometrically weighted dyad-wise shared partner (gwdsp), representing local clustering property, where $DP_k(y_i)$ denotes the number of dyads that have k shared partners in the network y_i.

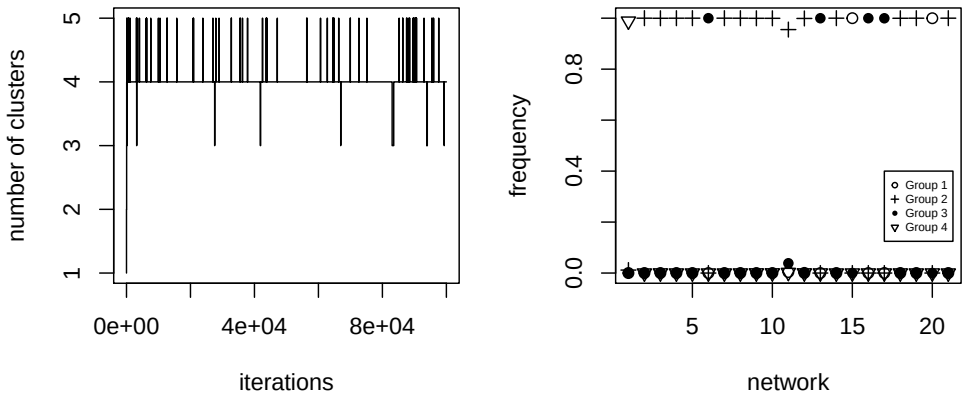

Figure 3.4.6: Clustering results of advice network ensemble using IMS sampling algorithm. Left: the number of clusters at every iteration. Right: the frequency of allocating to each group after 50,000 burn-in.

We apply the IMS algorithm to the advice network ensemble. The hyperparameters are specified as follows. A multivariate Gaussian distribution with mean $\mu_0 = (-3, 0, 0)$ and covariance $\Sigma_0 = 4^2 I_3$ is chosen as the prior distribution for ERGM parameters. The proposal variance in the MMCMH algorithm is set as $\Sigma_q = 0.05^2 I_3$. A beta prior Beta$(1, 0.1)$ is used for the mixing proportion. $\theta_0 = (-2, 0, 0)$ is the initial value for ERGM parameter. In the intermediate importance sampling procedure, we use $m_1 = 2$ intermediate distributions and $m_2 = 10$ auxiliary networks for MMCMH algorithm and $m_1 = 5, m_2 = 10$ in the allocation step.

The number of clusters at each iteration and the allocating frequency of each network from the IMS algorithm are shown in Figure 3.4.6. We can see that

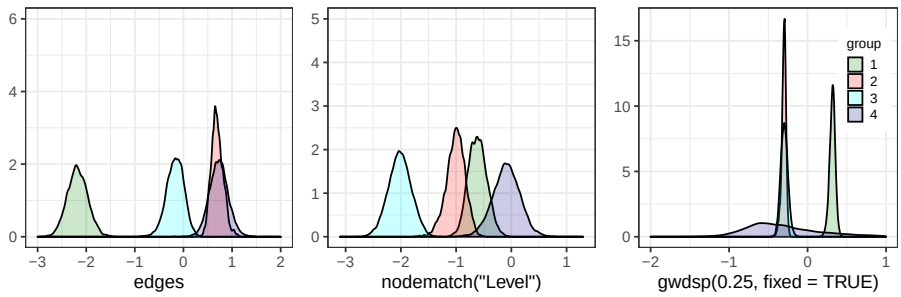

Figure 3.4.7: Density plots of group parameters after 50,000 burn for advice network ensemble.

4 groups are clustered with networks 15, 20 in the first group, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 14, 18, 19, 21 in the second group, 6, 13, 16, 17 in the third group, and network 1 in the fourth group. The acceptance probability in the MMCMH algorithm for 4 groups are 0.43, 0.16, 0.49, 0.38 respectively. To learn about the characteristics of each group, we display the posterior density plots from IMS algorithm in Figure 3.4.7. Group 1 has the smallest coefficient for edges but the biggest for gwdsp. This means that networks 15 and 20 have strong local clustering property, which is consistent with the fact that networks 15 and 20 have hub structures where fewer nodes have most of the connections. The advice relationships they nominate are centred around themselves. Group 2 has a big coefficient for edges and negative coefficient for level effect, indicating that networks are dense in this group and there are more advice between employees of different levels than of same levels. Group 3 has the smallest negative level effect, meaning that the advice relationships they observed are most across employees of different levels. Network 1 individually forms group 4. The level effect of network 1 is around 0, suggesting that individual level does not play a big role in network 1.

Our results are supported by the findings of Krackhardt (1987). Next, we compare our results with the centrality calculated in Krackhardt (1987). Betweenness centrality reflects on the influence of a node has over the flow of information. Group 1 consists of networks 15 and 20, which have unique performances on betweenness centrality. The betweenness centrality of nodes 15, 20 is 81.15 and

65.35, which are much bigger than the rest of nodes. Both of them mentioned a lot of advice relationships they are involved in. This is consistent with our finding of local clustering phenomenon implied by high gwdsp coefficient. The networks in group 3 are distinct from the rest of individuals in terms of low indegree and betweenness centrality. The indegree of individuals 6, 13, 16, 17 is all 0, indicating that they are not asked for advice by anybody. Also, the betweenness centrality of them is 0, 0.2, 0.11, 0.28, smaller than the rest of nodes in the locally aggregated networks. Moreover, employee 1 has high indegree centrality 18, but low betweenness centrality 2.81. It is asked advice often, but rarely asks advice from other people. Of all the 18 edges individual 1 claimed, only 1 relationship is confirmed by others. The speciality of individual 1 explains why the network 1 formed a group of its own.

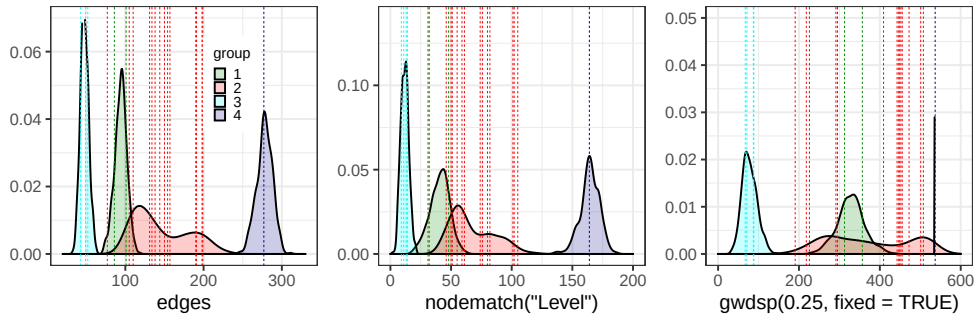

Figure 3.4.8: Density plots of network statistics based on networks simulated from posterior mean. The vertical lines stand for the value of structure statistics of observed networks.

Posterior assessments can be done by comparing the observed network statistics with simulated network statistics sampled from ERGM with estimation as parameters. Specifically, we generate 500 networks using the posterior mean as parameters and draw the density plots of the simulated network statistics in Figure 3.4.8. As we can see, the simulated network statistics are close to the observed network statistics, suggesting that IMS algorithm fits the data well. Note that network 1 located on the right end of the plot is far from other networks regarding the number of total edges and the number of edges within the same level. This is another reason that we think network 1 is better to be in a separate group.

Next, we apply PMS algorithm to the advice ensemble. After 100,000 iterations, 6 stable groups are detected, as shown at Figure 3.4.9. The networks in groups 1, 3, 4 from PMS algorithm are the same as from IMS algorithm. The group 2 from IMS algorithm is divided further into 3 groups, where networks 2, 4, 5, 8, 9, 10, 14, 19, 21 form the new second group, 3, 7, 12, 18 make the new fifth group, and 11 is in the sixth group. The acceptance probability of the MMCMH algorithm for 6 groups are 0.36, 0.23, 0.40, 0.36, 0.29, 0.50 respectively.

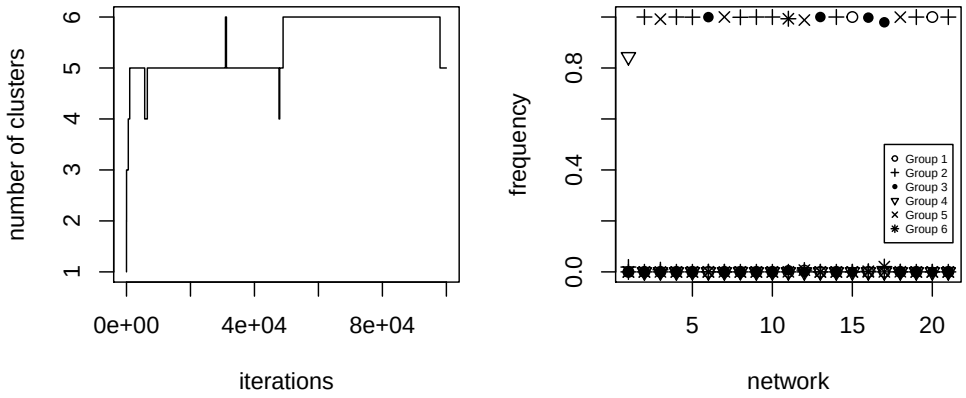

Figure 3.4.9: Clustering results of advice network ensemble using PMS sampling algorithm. Left: the number of clusters at every iteration. Right: the frequency of allocating to each group after 50,000 burn-in.

Furthermore, we calculate the distance between observed network statistics and simulated network statistics as follows,

$$\sum_{i:Z_i=k} \left\{ S(y_i) - \frac{\sum_{l=1}^{500} S(z_l^k)}{500} \right\}^2 \quad (k = 1, 2, \ldots, K),$$

where $S(y_i)$ represents the summary statistics of observed network y_i, and $S(z_l^k)$ stands for the summary statistics of simulated networks from ERGM with group parameter θ_k. The results of are shown in Table 3.4.4. Comparing both results, the estimation for groups 1, 3, 4 from IMS is more accurate, especially that the IMS estimation of group 4 is much better than the PMS estimation. To get more details, we show the density plots of the simulated network statistics on Figure 3.4.10. Simulated network statistics from IMS are centred around the observed

Table 3.4.4: The distance between observed network statistics and simulated network statistics.

Method	Group 1	Group 2	Group 3	Group 4
IMS	1310	191507	411	6
PMS	1358	134267, 10162, 32	720	291048

statistics on the top row, while simulated statistics from PMS are distant from the observed statistics on the second row. This is because the model for group 4 is near-degenerate. For a near-degenerate model, the underlying parameter values are close to a degenerate region, which increases the difficulty for estimation. This can happen quite often when we fit a ERGM with complicated statistics to real datasets. The pseudo likelihood method does not work for the near-degenerate model (Caimo and Friel, 2011). In this case, we can only use full likelihood method. For the 14 networks in group 2, the total distance is smaller for PMS method. This is understandable because the IMS method fits all these 14 networks with one model, while the PMS method fits these networks with 3 models.

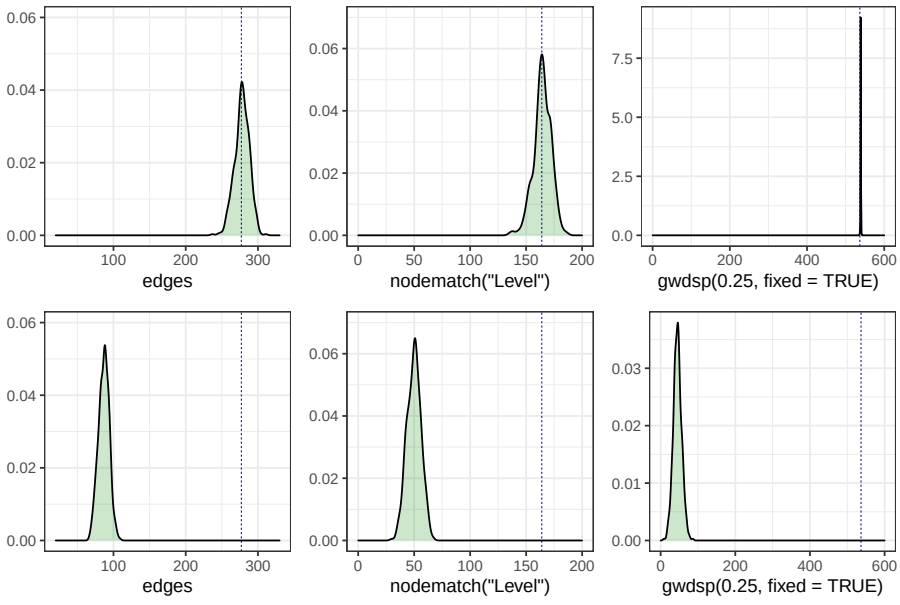

Figure 3.4.10: Simulated network statistics from IMS estimation for network 1 (or group 4) is in the first row. Simulated network statistics from PMS estimation for network 1 (or group 4) is in the second row.

In this simulation, we apply the IMS algorithm to the advice ensemble and found 4 meaningful clusters. Although pseudo likelihood based methods manage to divide the ensemble into reasonable clusters, they fail to represent the features of networks because they are not suitable for estimating the near-degenerate model in this example.

3.4.4 International Trade Networks

We also apply the proposed DPM-ERGMs to a world trade network ensemble. The ensemble of trade networks is observed on 60 countries ($n = 60$) over the period 2001-2016 ($N = 16$), denoted as $y_i (i = 1, 2, \ldots, 16)$. The networks are built based on the annual import data between every two countries from the UN Comtrade website[1]. A directed edge exists from node r to s, $y_{rs,i} = 1$, if the import amount from country r to s is more than 3 billion dollars at year i. The geographic distance between countries, represented by a matrix X, is an important factor in analysing trade relationship. Here, we treat distance as an edge covariate and explore its influence on the trade ensemble. The distance between countries is calculated using the coordinate of the capital city, downloaded from CEPII database[2].

In this application, we choose four statistics to explore the ensembles of trade networks from different aspects:

- $S^1(y_i) = \sum_{r \neq s} y_{rs,i}$, the total number of edges in network y_i. The density of trade networks can reflect the universality of global trade relationship.
- $S^2(y_i) = \sum_{r \neq s} y_{rs,i} y_{sr,i}$, the total number of mutual edges. A mutual edge in the trade networks stands for bilateral trade. It is helping in understanding trade types.
- $S^3(y_i) = e^{\phi} \sum_{k=1}^{n-2} \{1 - (1 - e^{-\phi})^k\} EP_k(y_i)$, $\phi = 0.25$, geometrically weighted

[1] https://comtrade.un.org/
[2] http://www.cepii.fr

edgewise shared partner (gwesp), representing transitivity. $EP_k(y_i)$ is the number of connected pairs that have k common neighbours.

- $S^4(y_i) = \sum_{r \neq s} y_{rs,i} X_{rs}$, the effect of the distance covariate. This helps to explore how distance affects the trade network structure.

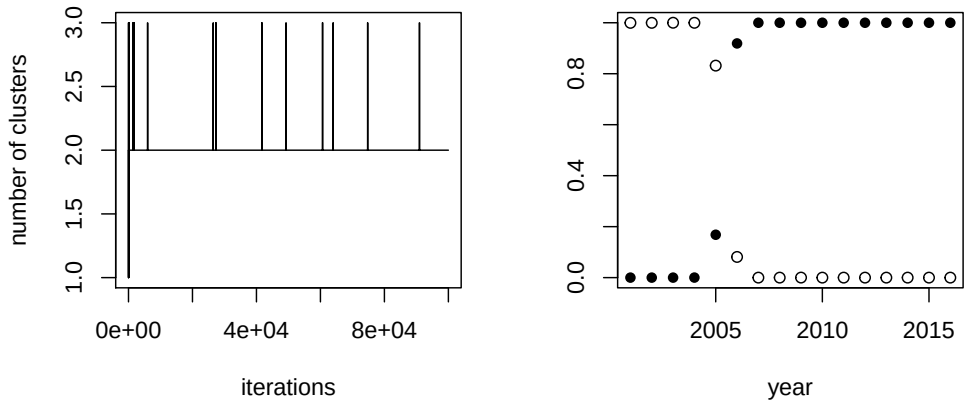

Figure 3.4.11: Clustering results of trade ensemble using IMS sampling algorithm. Left: the number of clusters at every iteration. Right: the frequency of allocating to each group after 50,000 burn-in.

We run 100,000 iterations using IMS sampling algorithm with the first 50,000 iterations as burn-in. The hyperparameters and initial values are set as follows: $\theta_0 = (-2, 0, 0, 0)$ for the ERGM parameter initial, $\mu_0 = (-3, 0, 0, 0)$, $\Sigma_0 = 4^2 I_4$ for the ERGM parameter prior, a diagonal matrix Σ_q with diagonal entries $(0.05^2, 0.02^2, 0.02^2, 0.02^2)$ for the variance of the proposal distribution in MMCMH, Beta$(1, 0.1)$ for the stick-breaking prior. In intermediate importance sampling, we choose $m_1 = 2, m_2 = 10$ for MMCMH and $m_1 = 5, m_2 = 10$ for posterior membership sampling. As shown in Figure 3.4.11, the ensemble of trade networks is clustered into 2 groups. Group 1 corresponds to networks of earlier years, from 2001 to 2005, and group 2 is formed by networks of later years, between 2006 and 2016. The acceptance ratio is 0.45, 0.30 for two groups separately. The network membership is closely related to the time, which is reasonable as trade networks are collected over time.

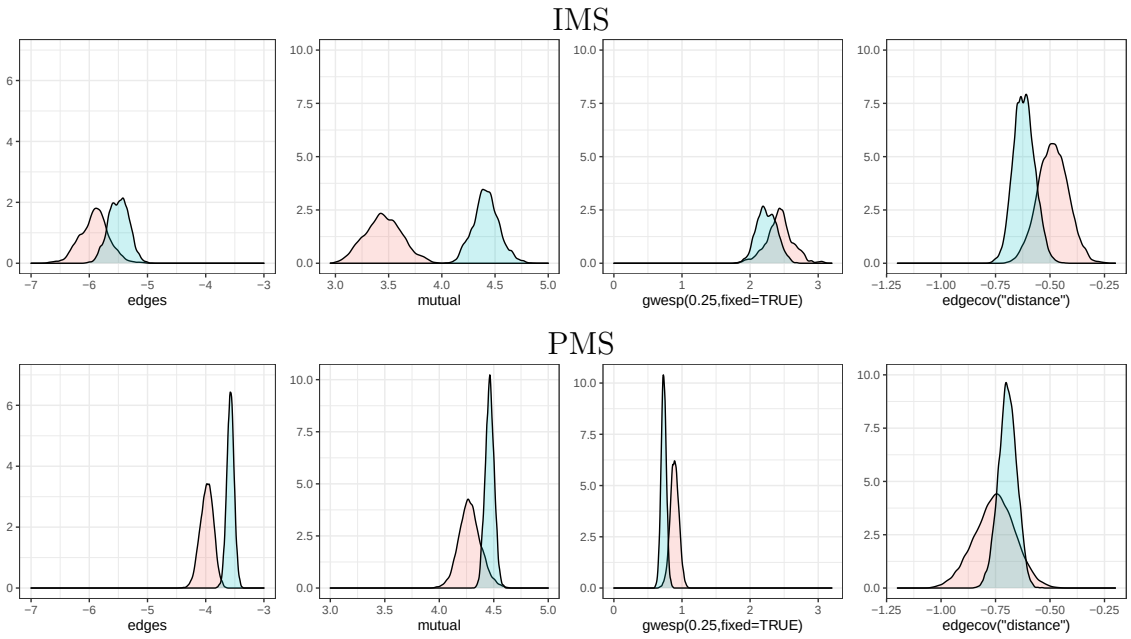

Figure 3.4.12: Density plots of estimation for groups 1 (red), 2 (blue) in trade ensemble.

The characteristics of each group can be further described using a group-specific ERGM and the comparisons between groups can be performed by comparing the parameters of each ERGM. The density plots for the posterior samples are shown in the first row of Figure 3.4.12. As we can see, group 2 has bigger density parameter than group 1, meaning that the trade relationships are denser. It also has bigger mutuality, which indicates that bilateral trade is more common. More countries prefer to form a mutual trade relationship with their trading partners. The smaller transitivity coefficient of group 2 suggests that the international trade is becoming more universal, although that the local clustering phenomenon still exists, implied by the positive transitivity parameter.

Next, we run the PMS sampling algorithm for 100,000 iterations. The clustering result is displayed in Figure 3.4.13, which is similar to the IMS algorithm. Networks from 2001 to 2004 are in the group 1 and networks between 2005 and 2016 are in the group 2. The acceptance ratio of each group is 0.59, 0.32. The density plots for each group are shown in the second row of Figure 3.4.12. Although the clustering result is very similar, the density plots for the ERGM parameter

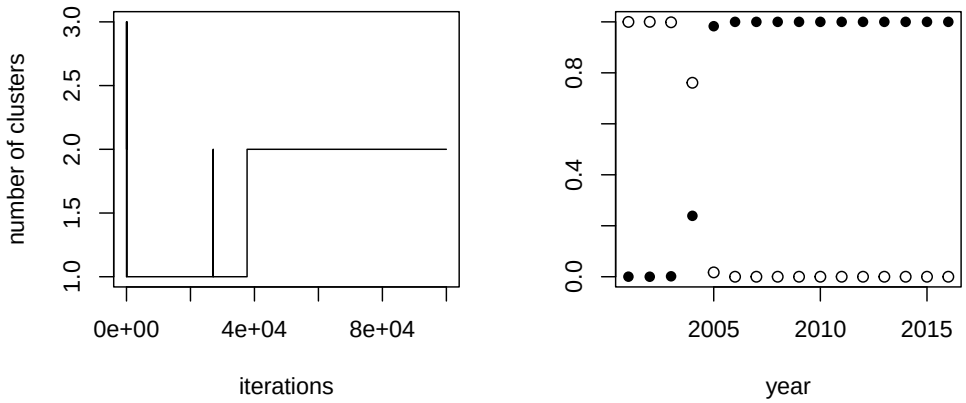

Figure 3.4.13: Clustering results of trade ensemble using PMS sampling algorithm. Left: the number of clusters at every iteration. Right: the frequency of allocating to each group after 50,000 burn-in.

estimation are quite different. Comparing with the IMS algorithm, PMS provides a narrow and sharp estimation, because pseudo likelihood method underestimates the variance of estimation. Moreover, the coefficient for gwesp term from PMS method is much smaller compared with the IMS method. This is because the pseudo likelihood method can not capture the dependency structures within a network.

In order to assess the model results, we generate 1,000 networks from the estimated model, and plot the simulated network statistics in Figure 3.4.14. The black dots stand for networks of group 1 and red dots represent networks of group 2. In the first row of the figure, the density plots of simulated network statistics are close to the observed samples, indicating that IMS algorithm provides good estimation to the data. However, in the second row, the posterior mode of the first group is far away from the 4 samples in the group, implying that the model is not a good fit to the data. As we mentioned before, networks in group 1 have strong transitivity, which can not be captured by PMS algorithm.

Furthermore, we calculate the distance between observed network statistics and simulated network statistics. Results are shown in Table 3.4.5. IMS method fits the networks in group 1 better than PMS method as the distance 89344 is much smaller than 145566.

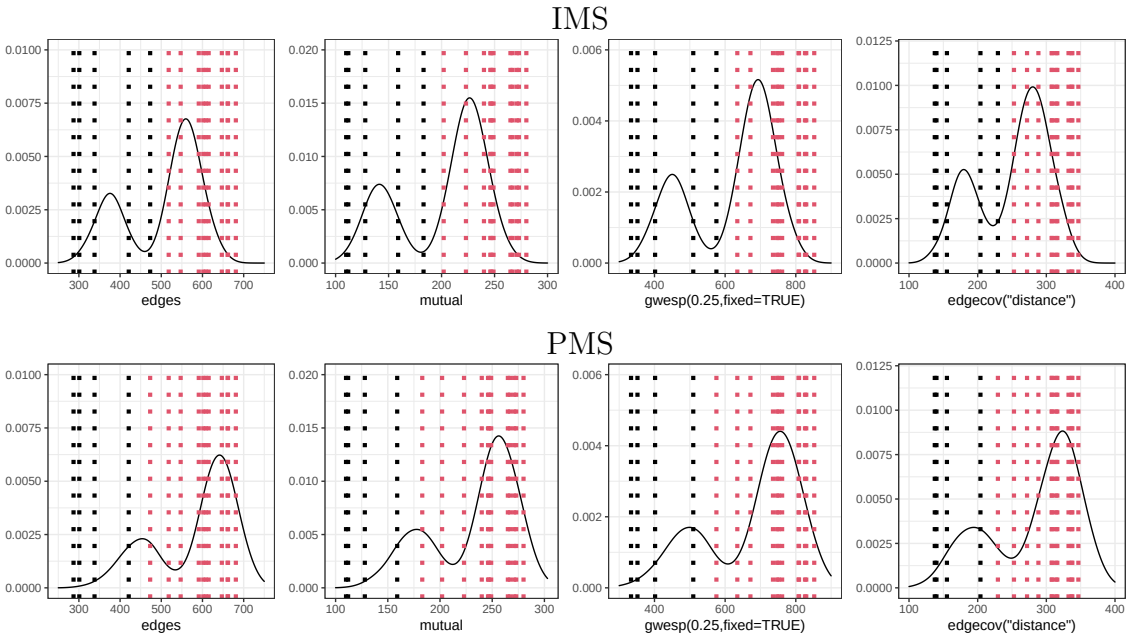

Figure 3.4.14: Density plots of network statistics, edges, mutual, GWESP and edgecov, based on networks simulated from estimation. The black vertical lines represent networks of group 1 and the red lines stand for networks of group 2.

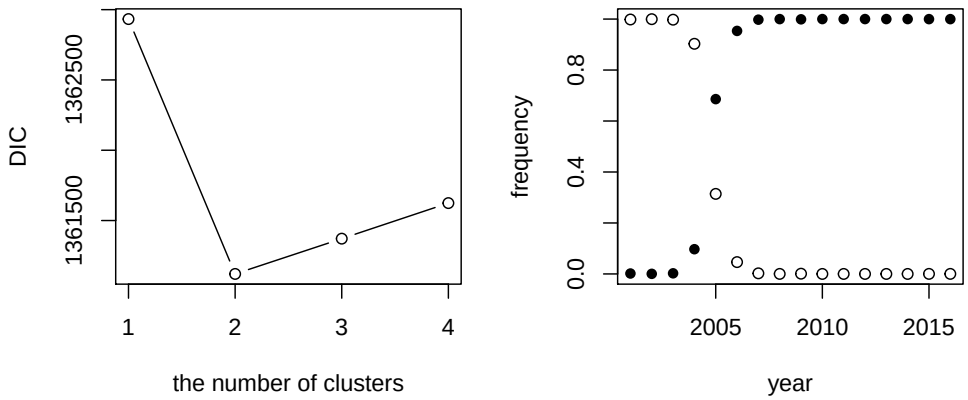

Figure 3.4.15: Clustering results of trade ensemble using PMG sampling algorithm. Left: DIC values for different number of clusters. Right: the frequency of allocating to each group after 50,000 burn-in when there are 2 groups.

Table 3.4.5: The distance between observed network statistics and simulated network statistics.

Method	Group 1	Group 2
IMS	80344	188624
PMS	145566	175461

For comparison, we also applied the pseudo likelihood Metropolis-within-Gibbs (PMG) sampling algorithm, developed by Yin et al. (2022) for a finite mixture of ERGMs. The prior for the group parameter is the same as it for the infinite method. Without knowing the number of clusters in advance, we fit the model with the number of clusters $K = 1, 2, 3, 4$ in sequence and calculate deviance information criteria (DIC) for each model accordingly. The DIC value with different number of clusters is shown in Figure 3.4.15. The best model is the one with the smallest DIC value, meaning that the number of clusters is chosen to be 2 here. Networks between 2001 and 2004 are allocated to the first group and networks between 2005 and 2016 are in the second group. The individual network membership and the posterior density of each group from the PMG algorithm are the same as from our proposed PMS algorithm.

In this simulation, IMS algorithm clusters the trade ensemble into 2 groups and fits each group with a different ERGM. PMS algorithm provides similar clustering result to IMS, but the networks in group 1 are fit poorly because the pseudo likelihood method failed to capture the transitivity of trade networks. The results of PMG algorithm are comparable with PMS algorithm, which supported our method.

Overall, the simulation results are not sensitive to the choice of priors and the number of iterations (burn-in or tuning). The multivariate normal distribution for ERGM model parameters is not an informative prior. As for the hyperparameter setting, we use a flat normal distribution with mean $\mu_0 = (-3, \mathbf{0})$ and covariance $\Sigma_0 = 4^2 I_p$ for the ERGM parameter. A negative number is suggested for the first parameter of μ_0 as nonnegative density parameter gives dense networks which contradict the fact that most real-world networks are sparse. The variance of the

prior can be flexible. For the variance of the proposal distribution in the MMCMH algorithm, we recommend using $\Sigma_0 = 0.05^2 I_p$ as a starting value and tune it based on the acceptance rate.

3.5 Discussion

In this chapter, we have proposed to model the ensembles of networks using a Dirichlet process mixture of ERGMs. Through such a framework, the subpopulations consisting of similar networks can be detected and compared automatically without requiring a fixed number of clusters in advance. On the other hand, multiple networks with similar characteristics are described by the same ERGM, namely, the cluster-specific ERGM. The cluster-specific ERGM is better than a single network ERGM, because it gathers the information from all networks in the same cluster. Moreover, we have also developed a novel IMS sampling algorithm for the full Bayesian inference of the DPM-ERGMs.

The full Bayesian inference of ERGMs is known to be time consuming as generating networks from desired ERGMs requires a long run of Markov chains using MCMC technique. We have provided a PMS sampling algorithm as a fast approximation method which can be used for pre-analysis of the dataset. However, as we mentioned before, PMS estimation may fail when the model is near-degenerate. For a more accurate and robust estimation, the IMS sampling algorithm is recommended.

Chapter 4

Fused Lasso Exponential Random Graph Models

This chapter focuses on the temporal variations in ensembles of networks. Specifically, we propose an extension to the exponential random graph model when the networks are dynamic instead of static. Our extension adds a fused lasso penalty to the objective function, which leads to the name fused lasso exponential random graph model (FL-ERGM). Here, the fused lasso is used to characterise the dynamic evolution of network topology. As a result, the parameter estimate is a piecewise linear function. The piecewise linear structure can capture the similar connectivity patterns between networks observed at consecutive points, and also detect the abrupt changing patterns between networks. To estimate the parameters in the FL-ERGM, we use the alternating direction method of multipliers (ADMM) to achieve a faster speed.

4.1 Introduction

Ensembles of networks that are observed over time are also called dynamic ensembles or dynamic networks. In such settings, where a time series of network observations is available, several formalisms have been proposed to capture the

changes of local and global structural patterns over time. We refer to this property as temporal variations. Stochastic actor-based models (Snijders et al., 2010) are one of the well-known models for network dynamics, which characterise network evolution using an unobserved stochastic process of tie changes. There are also several statistical models that utilise ERGMs to model dynamic networks over discrete time steps. For example, Hanneke et al. (2010) proposed a temporal ERGM to model the differences of network features between two consecutive networks through exponential family distributions. Krivitsky and Handcock (2014) extended the temporal ERGM to the separable temporal ERGM, which consider both the formation and dissolution processes of networks by separately parametrising prevalence and duration of fluctuations. Lee et al. (2020) developed a model based clustering framework based on temporal ERGMs where the block structure of networks are detected simultaneously. However, the temporal ERGM and its extensions ignore the heterogeneity of the differences between networks, and cannot fully capture the time varying patterns of the network structure.

Lee et al. (2020) proposed a varying coefficient ERGM where the parameter functions are assumed to be a smooth function of time. But in many real cases, the dynamic ensemble has a diverse changing pattern. For example, the networks might change suddenly at one time point and remain steady for some period of time. The smooth function in the varying coefficient ERGM can not capture the sudden change of structure or uncover the period that the network structure remains invariant.

Inferring the temporal variations of dynamic ensembles is challenging because it is difficult to simultaneously estimate both the structure within each network and the changes of the structure over time. Each network can exhibit a wide range of dependency structures and the dynamic ensembles can reveal many different types of changes. The range of possibilities includes a sudden shift of the entire network structure, a single node rewiring all of its connections, or even just one or two edges changing in the whole network. Therefore, any method must be flexible enough to discover many types of evolutionary patterns, while also being powerful

enough to learn this temporal structure over a long time period.

To account for the diverse changing patterns of network structures, we employ the idea of fused lasso regression by imposing an L_1 penalty (Tibshirani et al., 2005) on the differences between consecutive model parameters. Specifically, we propose a fused lasso exponential random graph model (FL-ERGM) to describe the dynamic network evolution process. Using this model, the dependent relationships within a network are characterised by ERGMs, and the evolution of networks is described by modelling the parameters associated with different features using a piecewise linear function. The piecewise linear function is a flexible function that can describe many types of relationships. Thus, the FL-ERGM can capture a variety of changing patterns. Moreover, the fused lasso penalty can encourage similar estimates from consecutive networks under some conditions. Since the evolution of networks is time dependent, it is reasonable to believe that the networks observed at consecutive time points share similar connectivity patterns to some extent.

Estimating FL-ERGMs is challenging due to the intractability of ERGM likelihood and the correlation between networks. First, as we mentioned before, the likelihood of ERGM is intractable. Second, the use of the fused lasso penalty in the dynamic networks also increases the difficulty of estimation as the fused lasso penalty is not a separable and smooth function. Also, the penalty we use is different from the fused lasso for linear regression, since we only encourage sparsity on the differences of parameters. Third, when the observed dynamic ensemble has a large number of networks and nodes, the data size is big, requiring strong computational power.

To address the above issues in an effective way, we first employ a pseudo likelihood approximation to the likelihood function for its computational tractability. Then we derive a fast alternating direction method of multipliers (ADMM). The ADMM algorithm solves convex optimisation problems by breaking them into smaller subproblems, each of which are then easier to handle (Boyd et al., 2010).

This allows us to take advantage of known properties and solution methods to derive closed-form ADMM updates for each subproblem. Furthermore, the updates of the ADMM algorithm require the calculation of the inverse of a matrix at each iteration step. Since the matrix inverse calculation is time consuming, we use the lower-upper (LU) matrix decomposition technique to simplify it. The use of LU decomposition speeds up our solver by several orders of magnitude over a naive ADMM implementation.

The rest of this chapter is organised as follows. We formulate the FL-ERGM in Section 4.2. In Section 4.3, we use the ADMM algorithm to solve the resulting optimisation problem. In Section 4.4, we numerically evaluate the performance of our proposed method to the counterpart. We conclude this chapter in Section 4.5.

4.2 Fused Lasso ERGMs

4.2.1 Model Formulation

In dynamic ensembles, networks are observed over time. We look at the dynamic networks collected at discrete time points in this chapter. In addition to describing the effects of local connectivity patterns on the shaping of global network structure, we are also interested in the temporal variations of the effects across networks. Neighbouring networks refer to networks observed at consecutive time points. The fused lasso penalty can encourage neighbouring networks to have similar estimates and describe the associated parameters using a piecewise linear function. Here, we propose a fused lasso exponential random graph model (FL-ERGM) by adding a fused lasso constraint to the ERGM.

Specifically, for a dynamic ensemble $\{Y_t\}_{t=1}^T$ with covariate information $\{X_t\}_{t=1}^T$,

the FL-ERGM is

$$\begin{aligned} \max \quad & \prod_{t=1}^{T} \frac{\exp(\theta_t^\top S(y_t, X_t))}{k(\theta_t)} \\ \text{subject to} \quad & \sum_{t=2}^{T} \|\theta_t - \theta_{t-1}\|_1 \leq m, \end{aligned} \tag{4.2.1}$$

where $\theta_t \in \mathbb{R}^d$ is the parameter vector at time t, $S(y_t, X_t)$ is the summary statistics for network y_t with covariate X^t, and $k(\theta_t) = \sum_{y \in \mathcal{Y}} \exp\left\{\theta_t^\top S(y, X_t)\right\}$ is the normalising constant. m controls the similarity across different network parameters.

Remark 1. Parameters of neighbouring networks should often be close to each other as the evolving process of networks is time dependent, so a fused lasso penalty (Tibshirani et al., 2005) is added to the marginal likelihood of ERGMs. Note that the original fused lasso penalty encourages sparsity of each parameter and the differences between parameters. Here, we only consider differences between parameters. Moreover, with a fused lasso penalty, information from neighbouring networks is used and more reliable estimators are obtained.

Remark 2. The parameter m controls the similarity between θ_t and θ_{t-1} for each $2 \leq t \leq T$. In other words, it determines how much information to use from neighbouring networks. Larger m means weaker constraints, and each parameter θ_t includes more information on the individual network y_t. Smaller m means stronger constraints, and the parameters tend to be similar.

4.2.2 Pseudo Likelihood Approximation

To facilitate effective inference on the proposed FL-ERGM, we further express model (4.2.1) in terms of pseudo likelihood functions.

In the pseudo likelihood function, the probability distribution of each edge is described using the so-called change statistics. The change statistic for edge $y_{ij,t}$ is defined as $\Delta S_{ij,t} = S(y_{ij,t} = 1, y_{-ij,t}, X_t) - S(y_{ij,t} = 0, y_{-ij,t}, X_t)$. This quantity measures the change of global network structures when the status of one edge is

altered from disconnected to connected. Here, $y_{ij,t}$ represents the edge between nodes i and j at time t, and $y_{-ij,t} = \{y_{kl,t}, (k,l) \neq (i,j)\}$ denotes the remaining network excluding edge $y_{ij,t}$.

Under the ERGM framework, the conditional probability distribution of edge $y_{ij,t}$ is

$$P\left(y_{ij,t} = 1 | y_{-ij,t}, X_t\right) = \frac{\exp\left(\theta_t^\top \Delta S_{ij,t}\right)}{1 + \exp(\theta_t^\top \Delta S_{ij,t})}.$$

The pseudo likelihood of network y_t is the joint conditional probability of all edges in the network. For directed networks, the pseudo likelihood is

$$\prod_{i \neq j} P\left(y_{ij,t} | y_{-ij,t}, X_t\right) = \prod_{i \neq j} P\left(y_{ij,t} = 1 | y_{-ij,t}, X_t\right)^{y_{ij,t}} \left[1 - P\left(y_{ij,t} = 1 | y_{-ij,t}, X_t\right)\right]^{1 - y_{ij,t}}.$$

For undirected networks, the pseudo likelihood is

$$\prod_{i < j} P\left(y_{ij,t} | y_{-ij,t}, X_t\right) = \prod_{i < j} P\left(y_{ij,t} = 1 | y_{-ij,t}, X_t\right)^{y_{ij,t}} \left[1 - P\left(y_{ij,t} = 1 | y_{-ij,t}, X_t\right)\right]^{1 - y_{ij,t}}.$$

In undirected networks, $y_{ij} = y_{ji}$, one only need to consider the cases of $i < j$. To keep the notation simple, we focus on the directed networks in the rest of the chapter.

The log pseudo likelihood of y_t is

$$\begin{aligned}
&\sum_{i \neq j} y_{ij,t} \log\left[P\left(y_{ij,t} = 1 | y_{-ij,t}, X_t\right)\right] + (1 - y_{ij,t}) \log\left[1 - P\left(y_{ij,t} = 1 | y_{-ij,t}, X_t\right)\right] \\
&= \sum_{i \neq j} \left\{ y_{ij,t} \theta_t^\top \Delta S_{ij,t} - \log\left[1 + \exp\left(\theta_t^\top \Delta S_{ij,t}\right)\right] \right\}.
\end{aligned}$$

Therefore, the FL-ERGM with pseudo likelihood approximation is written as

$$\max \sum_{t=1}^{T} \sum_{i \neq j} \left\{ y_{ij,t} \theta_t^\top \Delta S_{ij,t} - \log\left[1 + \exp\left(\theta_t^\top \Delta S_{ij,t}\right)\right] \right\} - \lambda \sum_{t=2}^{T} \|\theta_t - \theta_{t-1}\|_1, \tag{4.2.2}$$

where θ_t is the model parameter, $\Delta S_{ij,t}$ is the change statistic and λ is the tuning

parameter.

4.2.3 Choice of Tuning Parameters

The tuning parameter λ controls the similarity of coefficients at different time points. When $\lambda = 0$, parameters for each network observation are estimated individually. In this case, using FL-ERGMs is the same as estimating each network separately with ERGMs. The bigger the λ is, the more information we take from neighbouring networks. As $\lambda \to \infty$, all estimates at different time points tend to be the same. The fused lasso penalty in our model makes it possible to estimate single network parameters while considering information from other networks in the meantime.

There are two reasons for choosing the tuning parameters. The first is that each observed network is only one sample of the real unknown network. Even when the underlying networks at different time points come from the same distribution, the network observations appear to be different as they are different realisations. In this case, using information from other networks is very necessary. The second reason is that the evolution of networks is a time dependent process. The networks that are closer to each other in time should have more similarities, which should be considered in the model. An optimal λ can tell how much information to combine from other networks.

Cross validation methods can be used to find the optimal tuning parameters. Specifically, the dataset is partitioned into K subsets, $\{D_k, k = 1, 2, \ldots, K\}$, with approximately equal size. The cross validation process is then repeated K times, with each of the K subsets used exactly once as the validation data, and the remaining $K - 1$ subsets used as training data. Parameters are estimated using training data, and quality of estimators is measured by Pearson residuals using validation data. Pearson residuals have the form of

$$PR(\lambda) = \frac{(y - \hat{p}(\lambda))^2}{\hat{p}_\lambda(1 - \hat{p}(\lambda))},$$

which can be calculated by

$$\hat{PR}(\lambda) = \sum_{k=1}^{K} \sum_{D_k} \frac{(y_{ij,t} - \hat{p}_{ij,t}^{-k}(\lambda))^2}{\hat{p}_{ij,t}^{-k}(\lambda)(1 - \hat{p}_{ij,t}^{-k}(\lambda))},$$

where $\hat{p}_{ij,t}^{-k}(\lambda)$ is the probability $P\left(y_{ij,t} = 1 | y_{-ij,t}, X_t\right) = \frac{\exp\left\{\hat{\theta}_t^{-k}(\lambda)^\top \Delta S_{ij,t}\right\}}{1+\exp\left\{\hat{\theta}_t^{-k}(\lambda)^\top \Delta S_{ij,t}\right\}}$ with $\hat{\theta}_t^{-k}(\lambda)$ estimated using data other than D_k with tuning parameter λ. For each value of λ, a Pearson residual can be calculated. The optimal tuning parameter is the one with the smallest Pearson residual.

However, the partition of network data is complicated because edge variables are dependent on each other (Li et al., 2020). Splitting network nodes into groups would destroy some of the network structure. Also, the cross validation method for our model is computationally expensive. Therefore, we choose the tuning parameter using Bayesian information criterion (BIC). Specifically, we first choose a sequence of different values of λ. Then we perform model estimations with different λ, and obtain one BIC value for each λ. The λ with the lowest BIC value is the optimal tuning parameter,

$$\begin{aligned} BIC(\lambda) = &-2 \sum_{t=1}^{T} \sum_{i \neq j} \left\{ y_{ij,t} \hat{\theta}_t(\lambda)^\top \Delta S_{ij,t} - \log\left[1 + \exp\left(\hat{\theta}_t(\lambda)^\top \Delta S_{ij,t}\right)\right] \right\} \\ &+ \mathrm{df}(\lambda) \times \log(N \times T). \end{aligned}$$

$\hat{\theta}_t(\lambda)$ is the estimation with tuning parameter λ. $\mathrm{df}(\lambda)$ is the number of degrees of freedom when the tuning parameter is λ. In our model, we use the number of unique parameters as the estimation for the degrees of freedom.

4.3 Model Estimation

In this section, we employ the ADMM algorithm to estimate FL-ERGMs and utilise a LU decomposition technique to speed up the algorithm. We also show the convergence of the algorithm.

4.3.1 ADMM Algorithm

In order to facilitate model inference and use closed-form updates in the estimation procedure, we use a Taylor expansion to form a quadratic approximation (Friedman et al., 2010) to the first part of model (4.2.2) at current estimates $\tilde{\theta}_t$,

$$-\frac{1}{2}\sum_{t=1}^{\top}\sum_{i\neq j} w_{ijt}(\theta_t^{\top}\Delta S_{ijt} - z_{ijt})^2 + C(\tilde{\theta}_t) \quad \text{(second order Taylor expansion)}, \tag{4.3.1}$$

where

$$\begin{aligned}
\tilde{p}_{ijt} &= P(y_{ijt}=1|\tilde{\theta}_t, y_{-ijt}, X_t) = \frac{\exp(\tilde{\theta}_t{}^{\top}\Delta S_{ijt})}{1+\exp(\tilde{\theta}_t{}^{\top}\Delta S_{ijt})},\\
w_{ijt} &= \tilde{p}_{ijt}(1-\tilde{p}_{ijt}) = \frac{\exp(\tilde{\theta}_t{}^{\top}\Delta S_{ijt})}{(1+\exp(\tilde{\theta}_t{}^{\top}\Delta S_{ijt}))^2},\\
z_{ijt} &= \tilde{\theta}_t{}^{\top}\Delta S_{ijt} + \frac{y_{ijt}-\tilde{p}_{ijt}}{w_{ijt}},\\
C(\tilde{\theta}_t) &= -\frac{(y_{ijt}-\tilde{p}_{ijt})^2}{2w_{ijt}} + f(\tilde{\theta}_t).
\end{aligned} \tag{4.3.2}$$

The last term of (4.3.1), $C(\tilde{\theta}_t)$, can be omitted as it does not contain θ_t. Details of the expansion can be found in the Appendix A.1.

Using the quadratic approximation, the formula (4.2.2) can be written in a penalised weighted least square form,

$$\min \frac{1}{2}\sum_{t=1}^{T}\sum_{i\neq j} w_{ijt}(\theta_t^{\top}\Delta S_{ijt} - z_{ijt})^2 + \lambda\sum_{t=2}^{T}\|\theta_t - \theta_{t-1}\|_1. \tag{4.3.3}$$

Thus, the estimation can be performed using an iteratively reweighted least squares method.

Next, we rewrite (4.3.3) in a matrix form,

$$\min \frac{1}{2}\|A\Theta - b\|_2^2 + \lambda\|F\Theta\|_1. \tag{4.3.4}$$

$A \in \mathbb{R}^{N\times(d\times T)}$ is a block diagonal matrix with A_t as diagonal entries,

$$A = \begin{bmatrix} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_T \end{bmatrix}.$$

$A_t \in \mathbb{R}^{N_t\times d}$, $t = 1, 2, \ldots, T$,

$$A_t = \begin{bmatrix} \sqrt{w_{12,t}}\Delta S^1_{12,t} & \sqrt{w_{12,t}}\Delta S^2_{12,t} & \cdots & \sqrt{w_{12,t}}\Delta S^d_{12,t} \\ \sqrt{w_{13,t}}\Delta S^1_{13,t} & \sqrt{w_{13,t}}\Delta S^2_{13,t} & \cdots & \sqrt{w_{13,t}}\Delta S^d_{13,t} \\ \vdots & \vdots & \ddots & \vdots \\ \sqrt{w_{n_t(n_t-1),t}}\Delta S^1_{n_t(n_t-1),t} & \sqrt{w_{n_t(n_t-1),t}}\Delta S^2_{n_t(n_t-1),t} & \cdots & \sqrt{w_{n_t(n_t-1),t}}\Delta S^d_{n_t(n_t-1),t} \end{bmatrix},$$

where n_t is the number of nodes in network y_t, $N_t = n_t(n_t - 1)$ is the number of potential edges at time t, and $N = \sum_{t=1}^{T} N_t$ is the total number of potential edges. Θ is a $(d \times T)$-dimensional vector

$$\Theta = (\theta_t^\top)^\top = (\theta_{1,t}, \theta_{2,t}, \ldots, \theta_{d,t})^\top,\ t = 1, \ldots, T.$$

and b is a N-dimensional vector

$$b = (\sqrt{w_{12,t}}z_{12,t}, \sqrt{w_{13,t}}z_{13,t}, \ldots, \sqrt{w_{n_t(n_t-1),t}}z_{n_t(n_t-1),t})^\top,\ t = 1, \ldots, T.$$

F, a pairwise difference matrix, indicates the position of penalty. It has $d\times(T-1)$ rows, $d \times T$ columns, with the form

$$F = \begin{bmatrix} -1 & 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & -1 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 1 \end{bmatrix}, \tag{4.3.5}$$

where the first element is -1 and the $(d + 1)$th element is 1 in the first row, the

second element is -1 and the $(d+2)$th element is 1 in the second row, and so on.

To use ADMM, we rewrite model (4.3.4) as,

$$\begin{aligned} \min \quad & \frac{1}{2}\|A\Theta - b\|_2^2 + \lambda\|Q\|_1 \\ \text{subject to} \quad & F\Theta - Q = 0. \end{aligned} \tag{4.3.6}$$

Here, vectors $\Theta \in \mathbb{R}^{d\times T}$ and $Q \in \mathbb{R}^{d\times(T-1)}$ are two primal variables.

The augmented Lagrangian form for (4.3.6) is

$$L_\rho(\Theta, Q, y) = \frac{1}{2}\|A\Theta - b\|_2^2 + \lambda\|Q\|_1 + y^\top(F\Theta - Q) + \frac{\rho}{2}\|F\Theta - Q\|_2^2, \tag{4.3.7}$$

where $y \in \mathbb{R}^{d\times(T-1)}$ is a dual variable and $\rho > 0$ is a penalty parameter.

(4.3.7) is equivalent to a scaled form,

$$\begin{aligned} L_\rho(\Theta, Q, U) &= \frac{1}{2}\|A\Theta - b\|_2^2 + \lambda\|Q\|_1 + \rho U^\top(F\Theta - Q) + \frac{\rho}{2}\|F\Theta - Q\|_2^2 \\ &= \frac{1}{2}\|A\Theta - b\|_2^2 + \lambda\|Q\|_1 + \frac{\rho}{2}\|F\Theta - Q + U\|_2^2 - \frac{\rho}{2}\|U\|_2^2, \end{aligned} \tag{4.3.8}$$

where $U = (1/\rho)y$ is the scaled dual variable.

The ADMM algorithm is performed by updating variables Θ, Q, U in turn until convergence:

(1) Θ updates,

$$\begin{aligned} \Theta^{k+1} &= \arg\min_\Theta \frac{1}{2}\|A\Theta - b\|_2^2 + \frac{\rho}{2}\|F\Theta - Q^k + U^k\|_2^2 \\ &= (A^\top A + \rho F^\top F)^{-1}[A^\top b + \rho F^\top(Q - U)]. \end{aligned} \tag{4.3.9}$$

(2) Q updates,

$$\begin{aligned} Q^{k+1} &= \arg\min_Q \ \lambda\|Q\|_1 + \frac{\rho}{2}\|F\Theta^{k+1} - Q + U^k\|_2^2 \\ &= S_{\lambda/\rho}(F\Theta^{k+1} + U^k). \end{aligned} \tag{4.3.10}$$

(3) U updates,

$$U^{k+1} = U^k + F\Theta^{k+1} - Q^{k+1}. \tag{4.3.11}$$

Here, k is the iteration number and $S_\delta(z)$ denote the soft-thresholding operator with value

$$S_\delta(z) = \begin{cases} z - \delta & \text{if } z > 0 \text{ and } z > \delta \\ z + \delta & \text{if } z < 0 \text{ and } z < -\delta \\ 0 & |z| \leq \delta. \end{cases}$$

The termination criterion is that the primal and dual residuals must be smaller than the thresholds,

$$\|F\Theta^{k+1} - Q^{k+1}\|_2 \leq \varepsilon^{\text{pri}} \text{ and } \| - \rho F^\top (Q^{k+1} - Q^k)\|_2 \leq \varepsilon^{\text{dual}},$$

where

$$\begin{aligned} \varepsilon^{\text{pri}} &= \sqrt{(d-1) \times T}\varepsilon^{\text{abs}} + \varepsilon^{\text{rel}} \max\{\|F\Theta^k\|_2, \| - Q^k\|_2\}, \\ \varepsilon^{\text{dual}} &= \sqrt{d \times T}\varepsilon^{\text{abs}} + \rho\varepsilon^{\text{rel}}\|F^\top U^k\|_2. \end{aligned}$$

A reasonable value for the relative stopping criterion might be $\varepsilon^{\text{rel}} = 10^{-4}$ or 10^{-5}.

4.3.2 LU Decomposition

In the fist step of ADMM algorithm (4.3.9), the calculation of $(A^\top A + \rho F^\top F)^{-1}$ is challenging when the time length T is large. Here, we solve this problem by using the lower-upper (LU) decomposition.

F is also a banded matrix with $(T-1)$ rows and T columns,

$$F = \begin{bmatrix} -I_d & I_d & & & \\ & -I_d & I_d & & \\ & & \ddots & \ddots & \\ & & & -I_d & I_d \end{bmatrix},$$

where I_d denotes the diagonal matrix of order d. Then $F^\top F$ is a block tridiagonal matrix,

$$F^\top F = \begin{bmatrix} I_d & -I_d & & & \\ -I_d & 2I_d & -I_d & & \\ & \ddots & \ddots & \ddots & \\ & & -I_d & 2I_d & -I_d \\ & & & -I_d & I_d \end{bmatrix}.$$

$X = A^\top A + \rho F^\top F$ is also a block tridiagonal matrix,

$$X = \begin{bmatrix} A_1^\top(A_1) + \rho I_d & -\rho I_d & & & & \\ -\rho I_d & A_2^\top(A_2) + 2\rho I_d & -\rho I_d & & & \\ & \ddots & \ddots & \ddots & & \\ & & -\rho I_d & A_{T-1}^\top(A_{T-1}) + 2\rho I_d & -\rho I_d & \\ & & & -\rho I_d & A_T^\top(A_T) + \rho I_d \end{bmatrix}.$$

In order to inverse X, we use the LU factorization to decompose the block tridiagonal matrix X as follows,

$$X = (\Delta + L)\Delta^{-1}(\Delta + L^\top).$$

L is the lower part of X,

$$L = \begin{bmatrix} \mathbf{0} & & & & \\ -\rho I_d & \mathbf{0} & & & \\ & \ddots & \ddots & & \\ & & -\rho I_d & \mathbf{0} & \\ & & & -\rho I_d & \mathbf{0} \end{bmatrix}.$$

Δ is a block diagonal matrix with Δ_i as diagonal blocks,

$$\begin{aligned} &\Delta_1 = X_{11} = A_1^\top(A_1) + \rho I_d, \\ &\Delta_i = X_{ii} - \rho^2(\Delta_{i-1})^{-1} = A_i^\top(A_i) + 2\rho I_d - \rho^2(\Delta_{i-1})^{-1} \ (i = 2, \ldots, T-1), \\ &\Delta_T = A_T^\top(A_T) + \rho I_d - \rho^2(\Delta_{T-1})^{-1}. \end{aligned}$$

As $\Delta = \Delta^\top$, then $\Delta + L = (\Delta + L^\top)^\top$, and

$$X^{-1} = (\Delta + L^\top)^{-1}\Delta(\Delta + L)^{-1} = (\Delta + L^\top)^{-1}\Delta((\Delta + L^\top)^{-1})^\top.$$

Let $D = (\Delta + L^\top)^{-1}$, and we have $X^{-1} = D\Delta D^\top$. If we solve D, we solve X^{-1}. Since $\Delta + L^\top$ is a upper triangular matrix, D can be expressed in the following,

$$\begin{aligned} D = (\Delta + L^\top)^{-1} &= \begin{bmatrix} \Delta_1 & -\rho I_d & & & \\ & \Delta_2 & -\rho I_d & & \\ & & \ddots & & \\ & & & \Delta_{T-1} & -\rho I_d \\ & & & & \Delta_T \end{bmatrix}^{-1} \\ &= \begin{bmatrix} D_{11} = \Delta_1^{-1} & D_{12} = \rho\Delta_1^{-1}D_{22} & D_{13} = \rho\Delta_1^{-1}D_{23} & \cdots & D_{1T} = \rho\Delta_1^{-1}D_{2T} \\ & D_{22} = \Delta_2^{-1} & D_{23} = \rho\Delta_2^{-1}D_{33} & \cdots & D_{2T} = \rho\Delta_2^{-1}D_{3T} \\ & & D_{33} = \Delta_3^{-1} & \cdots & D_{3T} = \rho\Delta_3^{-1}D_{4T} \\ & & & \ddots & \vdots \\ & & & & D_{TT} = \Delta_T^{-1} \end{bmatrix}. \end{aligned}$$

$$(\Delta + L^\top)^{-1}\Delta = \begin{bmatrix} \Delta_1^{-1}\Delta_1 & \rho\Delta_1^{-1}D_{22}\Delta_2 & \rho\Delta_1^{-1}D_{23}\Delta_3 & \cdots & \rho\Delta_1^{-1}D_{2T}\Delta_T \\ & \Delta_2^{-1}\Delta_2 & \rho\Delta_2^{-1}D_{33}\Delta_3 & \cdots & \rho\Delta_2^{-1}D_{3T}\Delta_T \\ & & \Delta_3^{-1}\Delta_3 & \cdots & \rho\Delta_3^{-1}D_{4T}\Delta_T \\ & & & \ddots & \vdots \\ & & & & \Delta_T^{-1}\Delta_T \end{bmatrix},$$

$$X^{-1} = D\Delta D^\top = \begin{bmatrix} I_d & D_{12}\Delta_2 & D_{13}\Delta_3 & \cdots & D_{1T}\Delta_T \\ & I_d & D_{23}\Delta_3 & \cdots & D_{2T}\Delta_T \\ & & I_d & \cdots & D_{3T}\Delta_T \\ & & & \ddots & \vdots \\ & & & & I_d \end{bmatrix} \cdot \begin{bmatrix} D_{11} & & & & \\ D_{12} & D_{22} & & & \\ D_{13} & D_{23} & D_{33} & & \\ \vdots & & \ddots & & \\ D_{1T} & D_{2T} & D_{3T} & \cdots & D_{TT} \end{bmatrix}.$$

Therefore, the inverse of $X = A^\top A + \rho F^\top F$, a square matrix with dimension $dT \times dT$, can be solved by calculating the inverse of T small matrices Δ_i of dimension d.

4.3.3 Theoretical Guarantee

In this section, we show that the solution found by the ADMM algorithm converges to the true value of (4.3.3).

First, we show that the optimal solution of the quadratic approximation (4.3.4) converges to the original objective function (4.3.3). To achieve this, we define

$$l(\theta_1, \ldots, \theta_T) = -\sum_{t=1}^{T}\sum_{i\neq j} y_{ijt}\theta_t^\top \Delta S_{ijt} + \log\left\{1 + \exp\left(\theta_t^\top \Delta S_{ijt}\right)\right\},$$

$$\tilde{l}(\theta_1, \ldots, \theta_T) = \frac{1}{2}\sum_{t=1}^{T}\sum_{i\neq j} w_{ijt}(z_{ijt} + \theta_t^\top \Delta S_{ijt})^2 + C(\tilde{\theta}_t)^2.$$

Lemma 1. *$l(\theta_1, \ldots, \theta_T)$ and $\tilde{l}(\theta_1, \ldots, \theta_T)$ are convex.*

We prove Lemma 1 by showing that the Hessian matrix for $l(\theta_1, \ldots, \theta_T)$ and $\tilde{l}(\theta_1, \ldots, \theta_T)$ are positive definite. The details of the calculation can be found in Appendix A.2.

Lemma 2. *For $t \in \{1, \ldots, T\}$, as $\tilde{\theta}_t \to \theta_t$, then we have $\tilde{l}(\theta_1, \ldots, \theta_T) \to l(\theta_1, \ldots, \theta_T)$.*

The proof of Lemma 2 is obvious due to the calculus theory.

Define

$$(\hat{\theta}_1, \ldots, \hat{\theta}_T) = \arg\min l(\theta_1, \ldots, \theta_T), \tag{4.3.12}$$

$$(\hat{\tilde{\theta}}_1, \ldots, \hat{\tilde{\theta}}_T) = \arg\min \tilde{l}(\theta_1, \ldots, \theta_T). \tag{4.3.13}$$

Then we have the following theorem,

Theorem 1. *$(\hat{\theta}_1, \ldots, \hat{\theta}_T)$ and $(\hat{\tilde{\theta}}_1, \ldots, \hat{\tilde{\theta}}_T)$ defined in equations* (4.3.12) *and* (4.3.13) *exist and are unique. Furthermore, as $\tilde{\theta}_t \to \theta_t$, we have $(\hat{\tilde{\theta}}_1, \ldots, \hat{\tilde{\theta}}_T) \to (\hat{\theta}_1, \ldots, \hat{\theta}_T)$.*

The proof of Theorem 1 is straightforward. Lemma 1 shows that $l(\theta_1, \ldots, \theta_T)$ and $\tilde{l}(\theta_1, \ldots, \theta_T)$ are convex. Thus, $(\hat{\theta}_1, \ldots, \hat{\theta}_T)$ and $(\hat{\tilde{\theta}}_1, \ldots, \hat{\tilde{\theta}}_T)$ exist and are unique. Consequently, from Lemma 2, we have $(\hat{\tilde{\theta}}_1, \ldots, \hat{\tilde{\theta}}_T) \to (\hat{\theta}_1, \ldots, \hat{\theta}_T)$, as $\tilde{\theta}_t \to \theta_t$.

Next, we show that the solution found by the ADMM algorithm converges to the true value of (4.3.4). To be specific, we have the following assumptions and theorem.

Assumption 1. *The (extended-value-real) functions $1/2\|A\Theta - b\|_2^2 : \mathbb{R}^{d\times T} \to \mathbb{R} \cup \{+\infty\}$ and $\lambda\|Q\|_1 : \mathbb{R}^{d\times(T-1)} \to \mathbb{R} \cup \{+\infty\}$ are closed, proper and convex.*

Assumption 2. *The unaugmented Lagrangian $1/2\|A\Theta - b\|_2^2 + \lambda\|Q\|_1 + \rho U^\top(F\Theta - Q)$ has a saddle point, which means there exists a point (Θ^*, Q^*, U^*) (not necessarily unique), for which*

$$\begin{aligned}
&1/2\|A\Theta^* - b\|_2^2 + \lambda\|Q^*\|_1 + \rho U^\top(F\Theta^* - Q^*) \\
&\leq 1/2\|A\Theta^* - b\|_2^2 + \lambda\|Q^*\|_1 + \rho U^{*\top}(F\Theta^* - Q^*) \\
&\leq 1/2\|A\Theta - b\|_2^2 + \lambda\|Q\|_1 + \rho U^{*\top}(F\Theta - Q)
\end{aligned}$$

holds for all (Θ, Q, U).

Assumption 3. *The matrix F has full column rank.*

Theorem 2. *(The convergence of ADMM algorithm) When assumptions 1 and 2 hold, $1/2\|A\Theta^k - b\|_2^2 + \lambda\|Q^k\|_1$ converges to (4.3.6) as $k \to \infty$. If assumption 3 also holds, Θ^k converges to a solution of (4.3.4).*

We prove Theorem 2 in Appendix (A.3).

4.4 Empirical Results

To demonstrate the performance of our method, we first apply it to simulated networks under various specifications, and then show its ability to recover the true value in Section 4.4.1. Furthermore, we apply our method to international trade networks in Section 4.4.2 and political co-voting networks in Section 4.4.3. We also compare our method with the competing methods in the simulation studies.

4.4.1 Simulated Networks

We first introduce the five commonly used network statistics as follows:

- $S^1(y_i) = \sum_{r \neq s} y_{rs,i}$, the total number of edges in network y_i.
- $S^2(y_i) = \sum_{r \neq s} y_{rs,i} y_{sr,i}$, the number of mutual connections in network y_i.
- $S^3(y_i) = e^{\phi} \sum_{k=1}^{n-2} \{1 - (1 - e^{-\phi})^k\} DP_k(y_i)$, $\phi = 0.25$, geometrically weighted dyad-wise shared partners (gwdsp), a representation for local clustering phenomenon. $DP_k(y_i)$ is the number of dyads that have k common neighbours in network y_i.
- $S^4(y_i) = \sum_{r \neq s} y_{rs,i} \mathbf{1}(X_r^1 = X_s^1)$, the number of connections between individuals with the same categorical covariate X^1.
- $S^5(y_i) = \sum_{r \neq s} y_{rs,i}(X_r^2 + X_s^2)$, measuring the effect of the continuous covariate X^2.

Each of the above terms represents different features of networks. In order to study various characteristics of networks, we consider three models with different statistics. In model 1, we use three terms: the total number of edges $S^1(y_i)$, the number of mutual edges $S^2(y_i)$ and the categorical covariate term $S^4(y_i)$. In this way, the topological feature and the covariates are both included. $S^2(y_i)$ is a local feature that involves two nodes, while $S^3(y_i)$ is more advanced since $DP_k(y_i)$ takes account of the common neighbours of all dyads. We replace $S^2(y_i)$ with an higher order interaction measure $S^3(y_i)$ in model 2. In model 3, we include the number of edges $S^1(y_i)$, the categorical covariate term $S^4(y_i)$ and the continuous covariate term $S^5(y_i)$ to focus on the covariate effects.

For each model, we simulate 50 dynamic sequences of directed networks with given parameters. We fix X^1 as a binary covariate with half of the nodes taking value 0 and the other half taking value 1. For the continuous covariate term X^2, we sample it from a standard normal distribution. The number of nodes in each network is 30. After generating networks, we estimate each simulated dynamic network sequence with three different methods, FL-ERGM, VCERGM (Lee et al., 2020) and ERGM. Here, the ERGM is used on the dynamic ensemble by estimating each network separately.

To assess the performance of each method, we calculate the integrated absolute error (IAE) to measure the distance between estimated and true coefficients,

$$\text{IAE} = \sum_{t=1}^{T} |\hat{\theta}_t - \theta_t|. \tag{4.4.1}$$

The mean and standard deviation of IAE are shown in Table 4.4.1. The FL-ERGM provide the best results with the smallest IAE when $T = 10, 20, 30, 50$ in all three different models. Next, we investigate the details of each setting. When $T = 10$, the underlying parameters are constant over time as displayed in Figure 4.4.1. The FL-ERGM estimates are also constant over time, which can be seen from the green flat shaded bands illustrating the 0.025 and 0.975 quantiles of the FL-ERGM estimation, while the other shaded bands are not flat. At $T = 20$, the

Table 4.4.1: Simulation results with 30 nodes and 50 replicates: mean and standard deviation of the integrated absolute errors (IAE) for each method.

	T	FL-ERGM	VCERGM	ERGM
Model 1: $S^1(y_i) + S^2(y_i) + S^4(y_i)$	10	**2.14**(1.19)	3.18(1.16)	7.22(1.40)
	20	**7.36**(0.89)	20.60(1.40)	8.71(1.11)
	30	**5.39**(1.82)	7.62(1.46)	18.30(1.90)
	40	7.44(1.20)	**6.73**(1.01)	17.11(1.41)
	50	**10.53**(2.78)	23.00(2.07)	34.46(2.84)
Model 2: $S^1(y_i) + S^3(y_i) + S^4(y_i)$	10	**1.99**(1.20)	4.01(1.35)	6.03(1.06)
	20	**6.54**(1.10)	16.23(0.70)	9.65(3.39)
	30	**5.92**(2.00)	10.61(2.53)	18.24(1.97)
	40	9.27(1.91)	**6.10**(0.99)	15.59(1.40)
	50	**11.09**(2.40)	22.26(2.76)	29.25(2.56)
Model 3: $S^1(y_i) + S^4(y_i) + S^5(y_i)$	10	**1.20**(0.74)	1.73(0.69)	3.90(0.69)
	20	**4.95**(0.73)	14.85(0.51)	5.66(0.64)
	30	**4.71**(1.08)	5.58(0.79)	11.56(1.33)
	40	7.22(1.44)	**4.67**(0.81)	10.75(1.08)
	50	**9.15**(2.04)	15.22(1.05)	20.43(1.64)

underlying parameters are non-smooth as displayed in Figure 4.4.2. In this case, estimates from the FL-ERGM successfully capture the changing patterns of each coefficient regardless of the high difficulty. We also notice that in model 1, the true parameter of $S^2(y_i)$ is piecewise constant, but the VCERGM estimates show a continuously decreasing trend.

When $T = 30$, the underlying parameters are piecewise constant. The overall performance of the FL-ERGM is the best. Although FL-ERGM estimates are more distant to the true parameters than the VCERGM estimates at some time points in the middle panel of Figure 4.4.3, it still captures the piecewise constant structure of the real parameters successfully, while this is not suggested by other methods. At $T = 40$, the underlying parameters are smooth as displayed in Figure 4.4.4. The VCERGM performs the best. The FL-ERGM is worse than the VCERGM but better than the ERGM. When $T = 50$, as shown in Figure 4.4.5, the underlying parameters have some abrupt changes, the FL-ERGM detects and recovers the changing patterns of the real parameter while the VCERGM suggests

a smooth changing pattern.

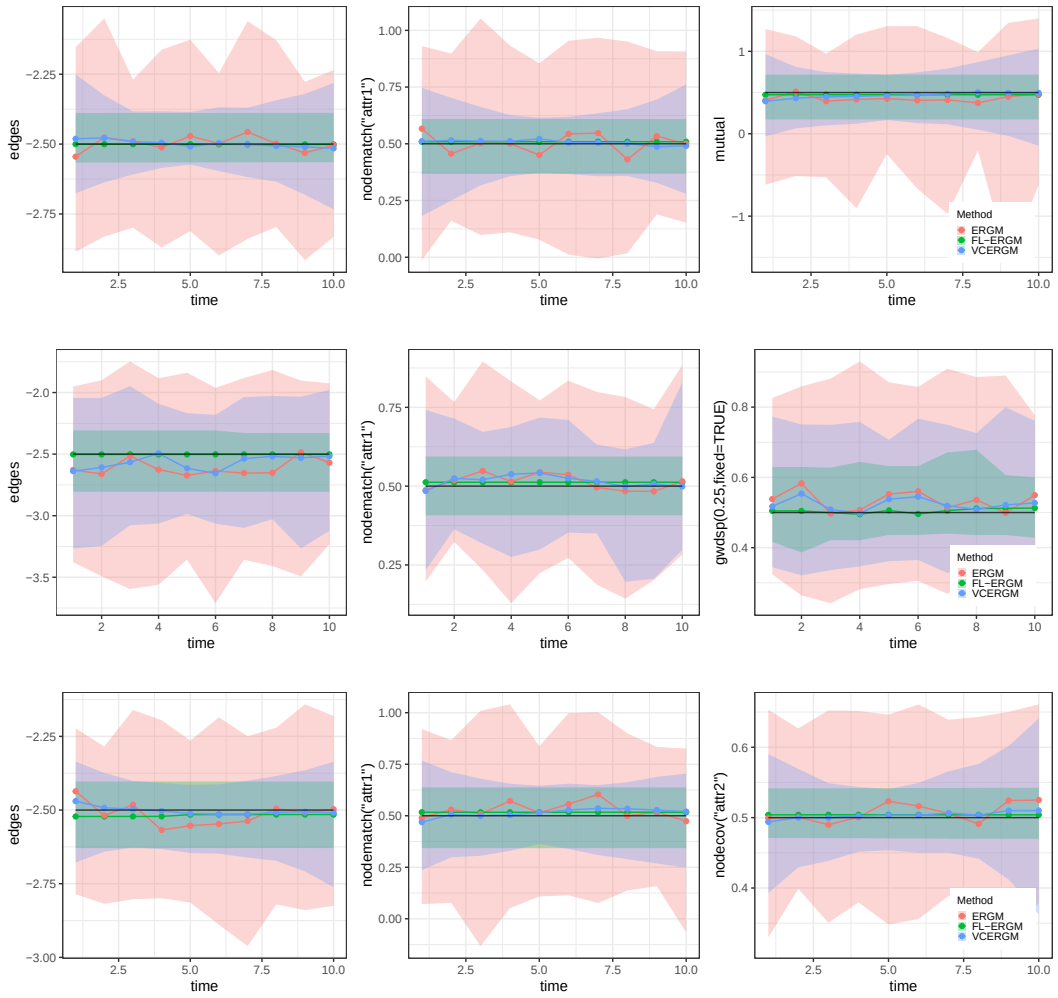

Figure 4.4.1: Parameter estimates for different models with constant underlying parameters. Black lines are the true parameters. Red lines represent the ERGM, blue lines stand for the VCERGM, and green lines show results from the proposed FL-ERGM. For each method, solid lines indicate the average of 50 estimated curves and the shaded bands illustrate the 0.025 and 0.975 quantiles.

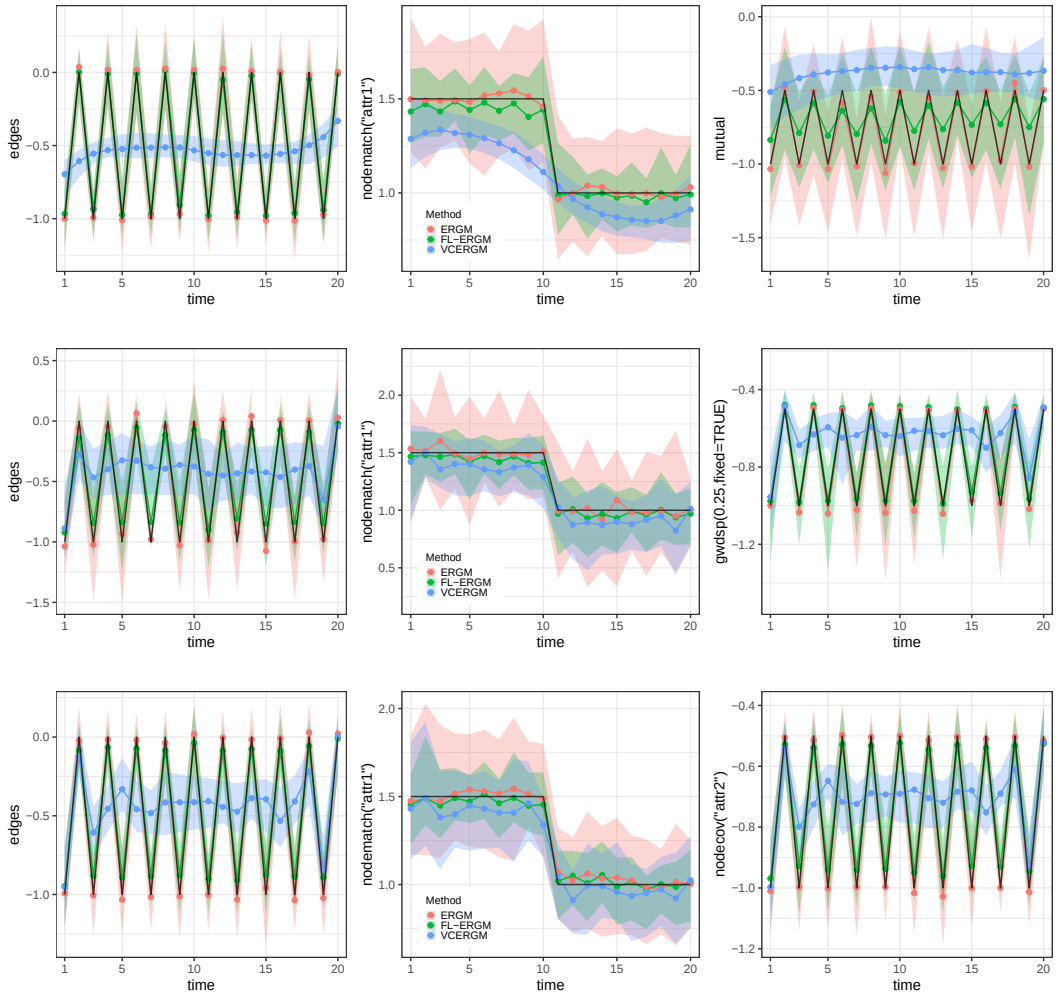

Figure 4.4.2: Parameter estimates for different models with nowhere constant underlying parameters. Black lines are the true parameters. Red lines represent the ERGM, blue lines stand for the VCERGM, and green lines show results from the proposed FL-ERGM. For each method, solid lines indicate the average of 50 estimated curves and the shaded bands illustrate the 0.025 and 0.975 quantiles.

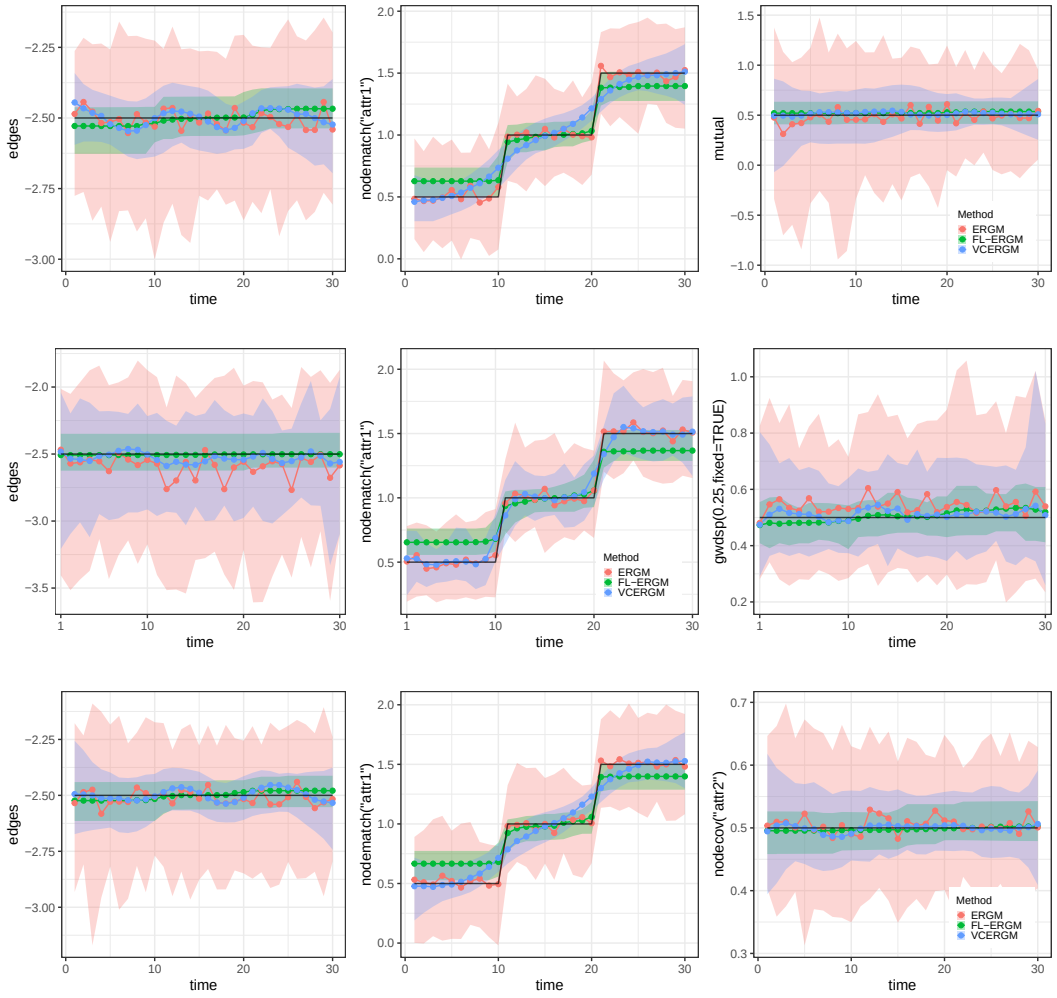

Figure 4.4.3: Parameter estimates for different models with piecewise constant underlying parameters. Black lines are the true parameters. Red lines represent the ERGM, blue lines stand for the VCERGM, and green lines show results from the proposed FL-ERGM. For each method, solid lines indicate the average of 50 estimated curves and the shaded bands illustrate the 0.025 and 0.975 quantiles.

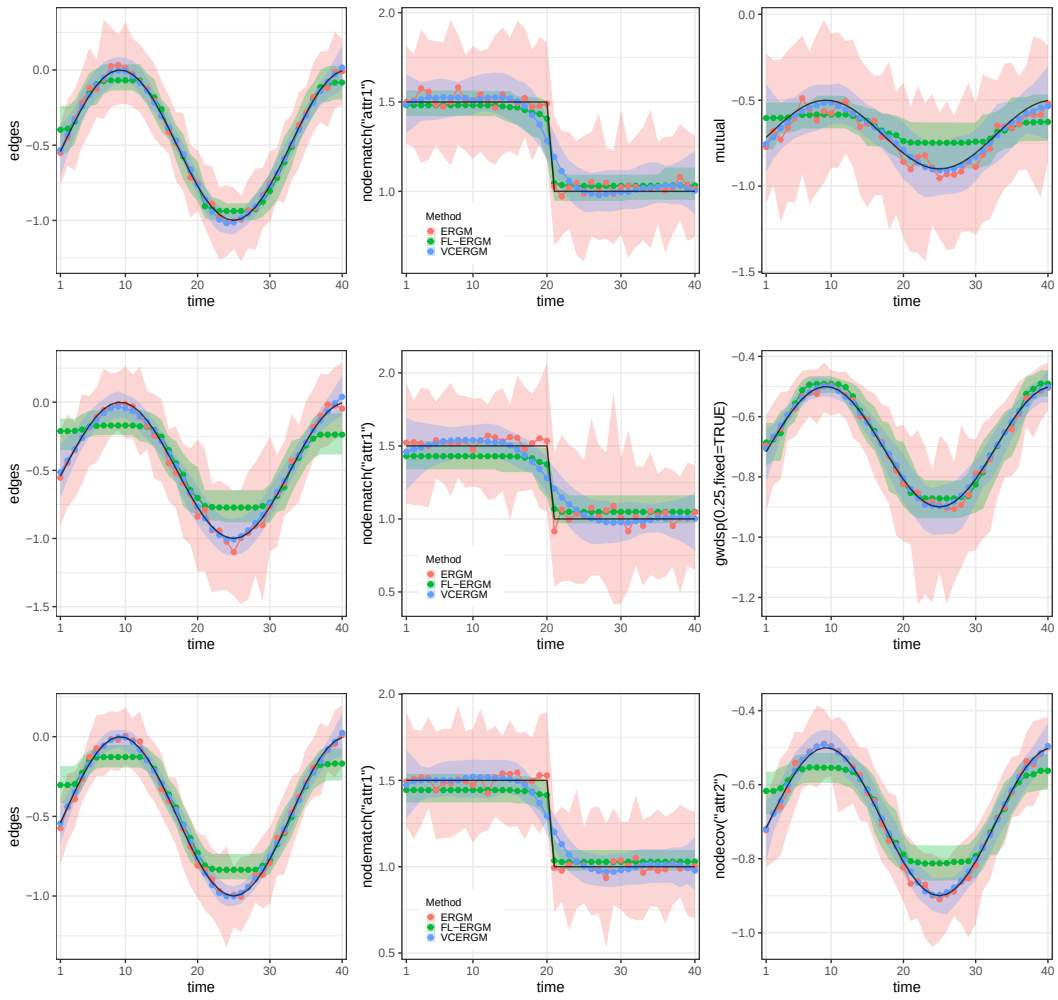

Figure 4.4.4: Parameter estimates for different models with smooth underlying parameters. Black lines are the true parameters. Red lines represent the ERGM, blue lines stand for the VCERGM, and green lines show results from the proposed FL-ERGM. For each method, solid lines indicate the average of 50 estimated curves and the shaded bands illustrate the 0.025 and 0.975 quantiles.

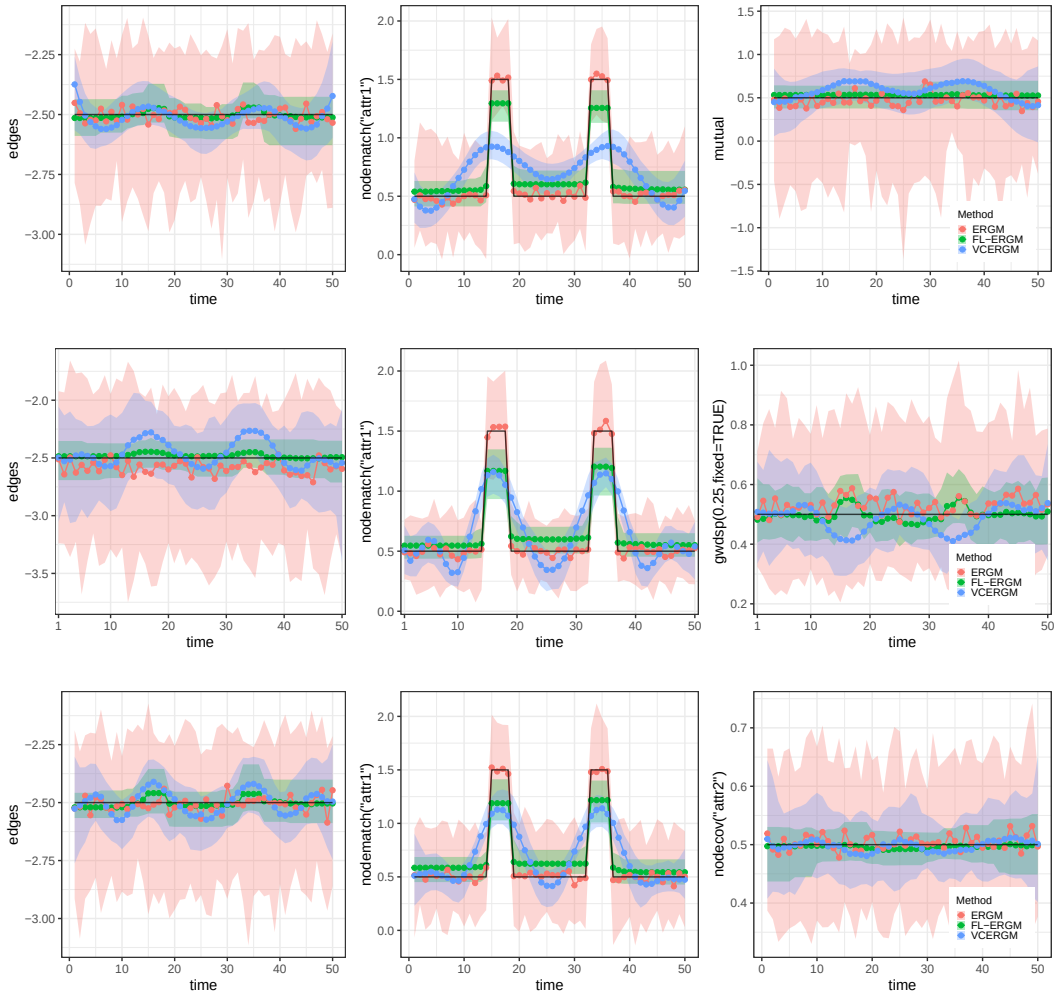

Figure 4.4.5: Parameter estimates for different models with piecewise constant underlying parameters. Black lines are the true parameters. Red lines represent the ERGM, blue lines stand for the VCERGM, and green lines show results from the proposed FL-ERGM. For each method, solid lines indicate the average of 50 estimated curves and the shaded bands illustrate the 0.025 and 0.975 quantiles.

Furthermore, we also run simulations using the settings from Lee et al. (2020). The results of IAE are shown in Table 4.4.2. We notice that our method FL-ERGM performs the best in settings 3 and 4 with the smallest IAE value. In settings 1 and 2, where the underlying parameters are smooth, the VCERGM performs better. We also show the details of estimated parameters together with the true parameters in Figure 4.4.6. Although the FL-ERGM does not perform the best in settings 1 and 2, the estimation from the FL-ERGM can still reflect on the changing trend of underlying parameters. However, in settings 3 and 4, the VCERGM can provide false information about the underlying parameters.

Table 4.4.2: Mean (standard deviation) of IAE for each method under same settings as Lee et al. (2020).

		FL-ERGM	VCERGM	ERGM
Edges	Sinusoidal	8.31(1.47)	**4.78**(1.01)	18.07(9.76
	Quadratic	7.18(2.12)	**2.31**(0.57)	6.34(0.71)
	Erdos-Renyi	**1.89**(1.38)	4.20(1.27)	14.58(2.51)
	Non-smooth	**9.54**(1.13)	33.63(0.47)	18.69(10.64)
Reciprocity	Sinusoidal	10.28(1.96)	**6.67**(1.15)	20.18(9.87)
	Quadratic	8.53(2.61)	**3.04**(0.82)	8.63(0.96)
	Erdos-Renyi	**2.07**(1.66)	4.73(1.40)	16.40(2.58)
	Non-smooth	**12.68**(1.42)	28.63(1.06)	22.24(10.75)

4.4.2 International Trade Networks

The international trade networks describe the trade relationships among 60 countries from 2001 to 2016. We apply the proposed FL-ERGM to international trade networks with the aim of analysing the dynamic evolution of trade relationships. The trade networks are built based on the annual import data between every two countries from the UN Comtrade website[1]. A directed edge exists from node r to s, $y_{rs,i} = 1$, if the import amount from country r to s is more than 3 billion dollars at year i. For each trade network, we choose the following statistics:

[1]https://comtrade.un.org/

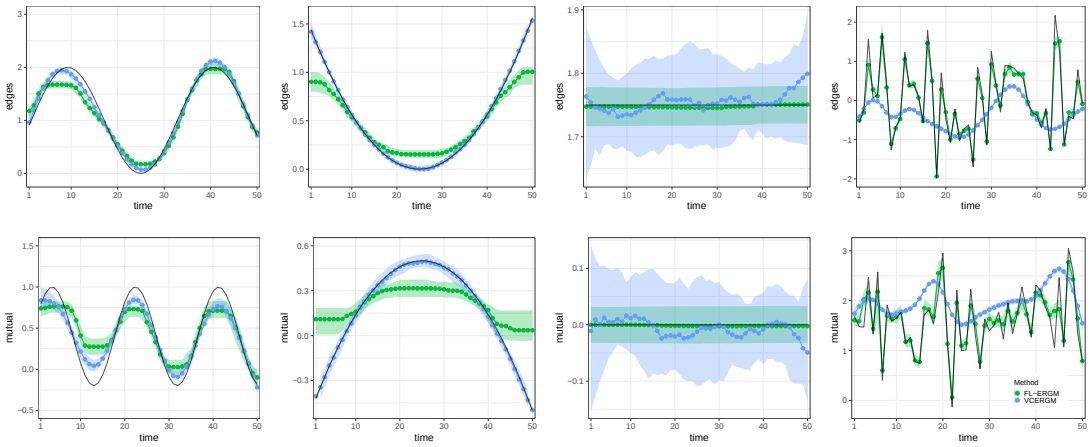

Figure 4.4.6: Parameter estimates for simulated networks with same settings as Lee et al. (2020). Black lines are the true parameters. Blue lines stand for the VCERGM, and green lines show our method the FL-ERGM. For each method, solid lines indicate the average of 100 estimated curves and the shaded bands illustrate the first and third quantiles.

- $S^1(y_i) = \sum_{r \neq s} y_{rs,i}$, the total number of edges in trade network y_i, reflecting on the density of trade relationship.

- $S^2(y_i) = e^{\phi} \sum_{k=1}^{n-2} \{1 - (1 - e^{-\phi})^k\} DP_k(y_i)$, $\phi = 0.25$, geometrically weighted dyad-wise shared partners (GWDSP), a representation for local clustering phenomenon, measuring the extent to which the trade relationship localizes. $DP_k(y_i)$ is the number of dyads that have k common neighbours.

- $S^3(y_i) = \sum_{r \neq s} y_{rs,i} X_{rs}$, measuring the effect of distance for the trade relationship. X_{rs} stands for the geographic distance between countries r and s, downloaded from CEPII database[2].

Next, we fit the FL-ERGM to the dynamic trade networks. BIC is shown in Figure 4.4.7 and parameter estimates of the FL-ERGM are presented in Figure 4.4.8. The parameter estimates in Figure 4.4.8 reveal that the trade evolution has two phases, the increasing period before 2008 and the steady phase after 2008. This matches the fact of global financial crisis in 2008. Before 2008, the coefficient of edges keeps increasing, meaning that the trade relationship was expanding

[2]http://www.cepii.fr

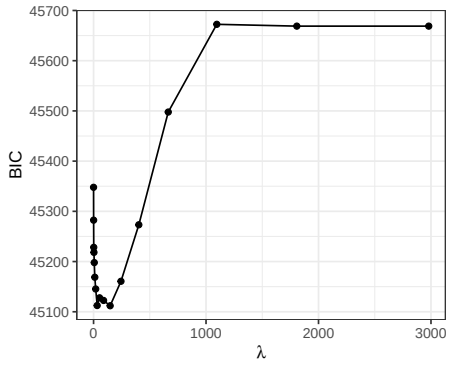

Figure 4.4.7: BIC of the FL-ERGM for dynamic trade networks.

continuously. At the same time, the coefficient of gwdsp is decreasing, which implies that the local clustering phenomenon is slowly disappearing, and the trade relationships are becoming more universal. The distance has a negative and steady effect on trading during this period. After 2008, the coefficients of edges and distance effects are relatively stable over time. We also notice that the gwdsp coefficient has a slight increase from 2012, but we could still claim that the second phase is steady, as the value of gwdsp coefficient is relatively small. Moreover, comparing two phases, the second one has bigger edge coefficients but smaller distance effects, suggesting that the trade is more frequent and the reason might be the increased convenience of global transportation.

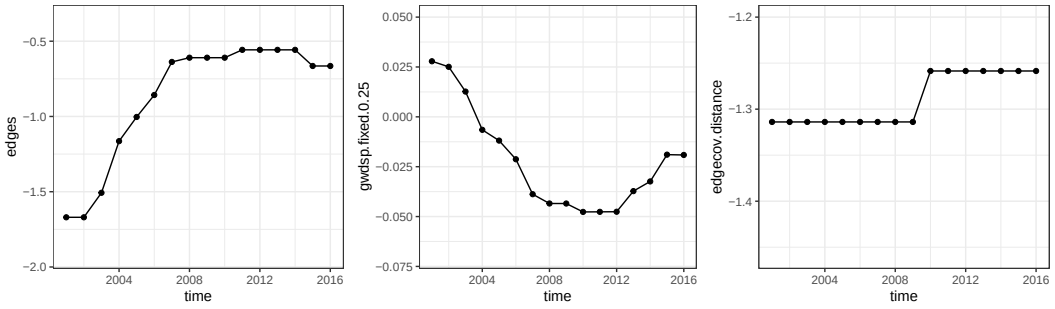

Figure 4.4.8: Parameter estimates of the FL-ERGM for dynamic trade networks.

We also apply the VCERGM and ERGM to the dynamic trade networks. The model results are shown in Figure 4.4.9. We can see that the VCERGM and ERGM provide similar estimates on the coefficients of edges and gwdsp,

regardless of the big spikes from the ERGM. However, the VCERGM suggests a continuously increasing trend on distance effect instead of the two steady phase result from the FL-ERGM. The changing patterns suggested by the VCERGM might not be reliable, because in Figure 4.4.3, VCERGM estimates are increasing continuously when the underlying parameter structure is piecewise constant.

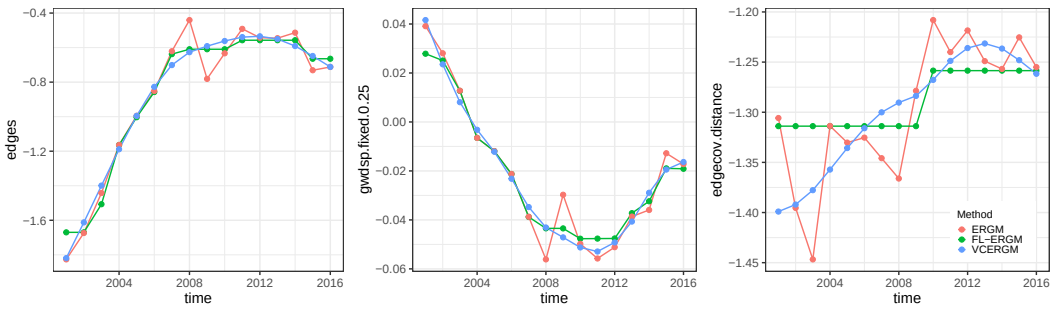

Figure 4.4.9: Parameter estimates of different methods for dynamic trade networks. Red lines represent the results from the ERGM, blue lines stand for the VCERGM, and green lines show results from the proposed FL-ERGM.

We also assess the accuracy of the estimated model through simulations. Specifically, at each time point, we simulate 100 networks from the ERGM distribution with parameter estimate $\hat{\theta}_t$. We present the summary statistics of the simulated networks using box plots in Figure 4.4.10. As is shown, the observed network statistics are close to the simulated network statistics, indicating that the estimated FL-ERGM fits the dynamic trade networks well.

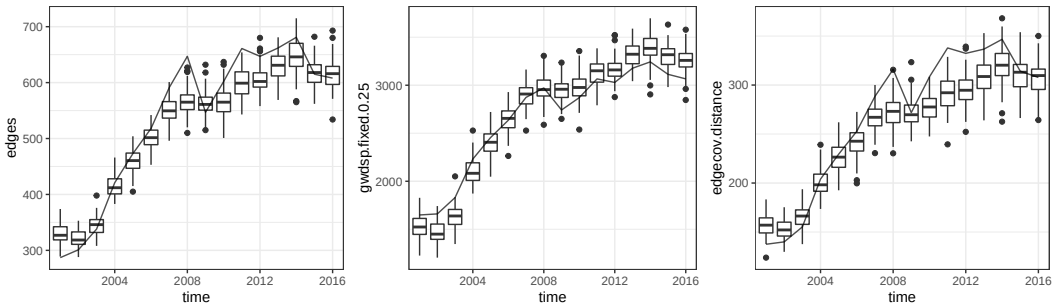

Figure 4.4.10: Model assessments of the FL-ERGM on dynamic trade networks. The solid line indicates the observed network statistics and the box plots represent the statistics of networks simulated from the estimated model.

4.4.3 Political Co-voting Networks

We also apply the FL-ERGM to the political co-voting networks to study the co-voting patterns of senators. The co-voting networks consist of a dynamic sequence of networks from Congress 40 (1867-1869) to Congress 113 (2013-2015). In each network, nodes represent senators, and edges indicate that the corresponding senators vote concurringly (both yay or both nay) on at least 80% of the bills. This dataset has been studied by Lee et al. (2020) using the VCERGM method. In this chapter, we apply the FL-ERGM to the co-voting networks with same statistics listed as follows:

- $S^1(y_i) = e^{\phi} \sum_{k=1}^{n-2} \{1 - (1 - e^{-\phi})^k\} EP_k(y_i)$, $\phi = 1$, geometrically weighted edgewise shared partners (gwesp), representing transitivity. It measures the extent to which the decisions of two senators are the same as their shared co-voters. $EP_k(y_i)$ is the number of connected pairs that have k common neighbours.

- $S^2(y_i) = e^{\phi} \sum_{k=1}^{n-2} \{1 - (1 - e^{-\phi})^k\} DP_k(y_i)$, $\phi = 1$, geometrically weighted dyad-wise shared partners (gwdsp), representing local clustering phenomenon. It measures the extent to which the senators agree on each other. $DP_k(y_i)$ is the number of dyads that have k common neighbours.

- $S^3(y_i) = \sum_{r \neq s} y_{rs,i} \mathbf{1}(X_r = X_s = D)$, the number of connections within Democratic senators.

- $S^4(y_i) = \sum_{r \neq s} y_{rs,i} \mathbf{1}(X_r \neq X_s)$, the number of connections between Democratic and Republican senators.

- $S^5(y_i) = \sum_{r \neq s} y_{rs,i} \mathbf{1}(X_r = X_s = R)$, the number of connections within Republican senators.

Next, we investigate the results of the FL-ERGM. BIC is shown in Figure 4.4.11 and parameter estimates of the FL-ERGM are presented in Figure 4.4.12. The coefficient estimates from the FL-ERGM are functions over time. θ_1 and

θ_2, representatives of the local clustering property, change irregularly with time. $\theta_3, \theta_4, \theta_5$, the effects of affiliation, exhibit piecewise linear changing patterns. The big positive gwesp coefficient and small negative gwdsp coefficient suggest a high transitivity of co-voting networks, meaning that there exist groups of senators that agree with each other on many bills. The negative coefficients of affiliation terms indicate the amount of decrease on the log odds of two senators being connected. In other words, senators being in different parties are most likely disconnected since θ_4 has the biggest absolute value of the three coefficients most time. The effects of Democrats and Republicans are close, and we also notice that the comparison of θ_3 and θ_5 is closely related to the majority party in Senate. In order to show this relationship more clearly, we indicate the majority party of the Senate through the background colour of the top panel, and the party of the US president through the background colour of the bottom panel. Between Congress 47 and 72, Republican is the majority party in Congress for 22 times (out of 27 times) and the coefficient of Republicans, θ_5, is bigger than the coefficient of Democrats, θ_3, for 24 times. Between Congress 78 and 102, the coefficient of Democrats, θ_3, is bigger than the coefficient of Republicans, θ_5, for 23 times (out of 25 times), and Democrat is also the majority party in Congress for 20 times.

Our model found that the sequence of co-voting networks can be divided into 9 periods. We have been able to identify each of these periods with events in American history. Networks at each period have different characteristics represented by distinct coefficients of each period. The first period starts from Congress 40 (1867-1869) to 45 (1877-1879), when the US went through the Reconstruction period after the Civil War. The Congress is gradually dominated by Republican, indicated by the increasing coefficient of θ_5(R-R), while θ_3(D-D) and θ_4(D-R) remain constant during this time. We also notice that there is an abrupt change from Congress 45 to 46, suggested by the sharp increase on θ_1(gwesp) and decrease on θ_2(gwdsp), meaning that new connections are built because of transitivity. The second phase, form Congress 46 (1879-1881) to 63 (1913-1915), started with the Gilded Age. During this period, US finished the Second Industrial Revolution

and became a world economic and military power. The Congress was dominated by Republicans and more agreement is reached inside the Republican party than the Democratic party. There is also an abrupt change on θ_3 and θ_5, at around Congress 50 (1887-1889) and 51 (1889-1891), which matches the fact of the change in the affiliation of the president after a long time.

The third period, from Congress 64 (1915-1917) to 66 (1919-1921), corresponds to time of the World War I. $\theta_3, \theta_4, \theta_5$ all decrease compared with the last period, meaning that the effect of affiliation increases during war time. Also, θ_1 reaches the minimum and θ_2 reaches its maximum at Congress 65. The 4th period, Congress 67 (1921-1923) to 71 (1929-1931), matches the Roaring Twenties when US had the great economic prosperity after the World War I. θ_3 and θ_5 decrease but θ_4 increases compared with the last period. The 5th period is between Congress 72 (1931-1933) and 81 (1949-1951), when US experienced the Great Depression and World War II. We notice that the coefficient of Democrats increases dramatically and becomes bigger than the coefficient of Republicans. The position of Democrats is gained during the Great Depression and kept during the Word War II.

The 6th period, from Congress 82 (1951-1953) to 99 (1985-1987), is related to the Cold War. The coefficient of Democrats is bigger than the coefficient of Republicans most of the time. We also notice that θ_3 has a dip at Congress 83 (1953-1955), which matches the time that the famous president Eisenhower was elected as the first Republican president since 1932. The 7th period, from Congress 100 (1987-1989) to 105 (1998-1999) is the Post-Cold War era. $\theta_3, \theta_4, \theta_5$ all increase compared with the last period, indicating that the affiliation plays a less important role in the decision of senators. θ_4 increases sharply at Congress 106, meaning that there is a lot of agreement between Democrats and Republicans. The 8th period, from Congress 106 (1999-2001) to 109 (2005-2007), lasts for 8 years. $\theta_3, \theta_4, \theta_5$ all remain stable during this time when Bush was the president. The change from phase 8 to 9 is marked by the financial crisis and at the same time, Democrats became the Senate majority again from Congress 110. During the 9th phase, from Congress 110 (2007-2009) to 115 (2013-2015), when Obama

was the president, the coefficient of Democrats increases but the coefficient of Republicans decreases.

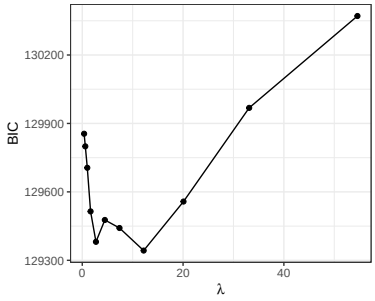

Figure 4.4.11: BIC of the FL-ERGM for dynamic co-voting networks.

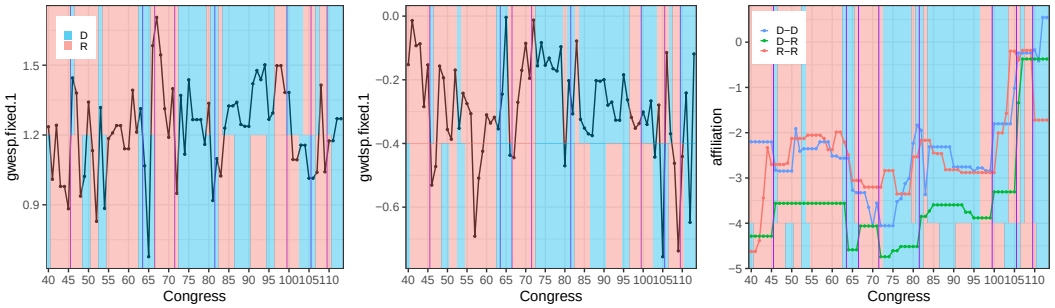

Figure 4.4.12: Parameter estimates of the FL-ERGM for co-voting networks. In all three plots, the background colour of the top panel indicates the majority party of the Senate, and the background colour of the bottom panel represents the party of the US president. In the third plot, blue lines stand for the effects of Democrats-Democrats, green lines stand for the effects of Democrats-Republicans, and red lines show the effects of Republicans-Republicans.

The model assessment is shown in Figure 4.4.13, suggesting that the FL-ERGM fits the networks well. The results of the VCERGM and ERGM are also presented in Figure 4.4.14. Similar to what we found in the simulation studies, ERGM estimates are spiky and VCERGM estimates are smooth. In comparing the Fl-ERGM with competing methods, we see that the coefficient estimates from the FL-ERGM can account for the characteristics of each network as well as show the abrupt change points on the network structure.

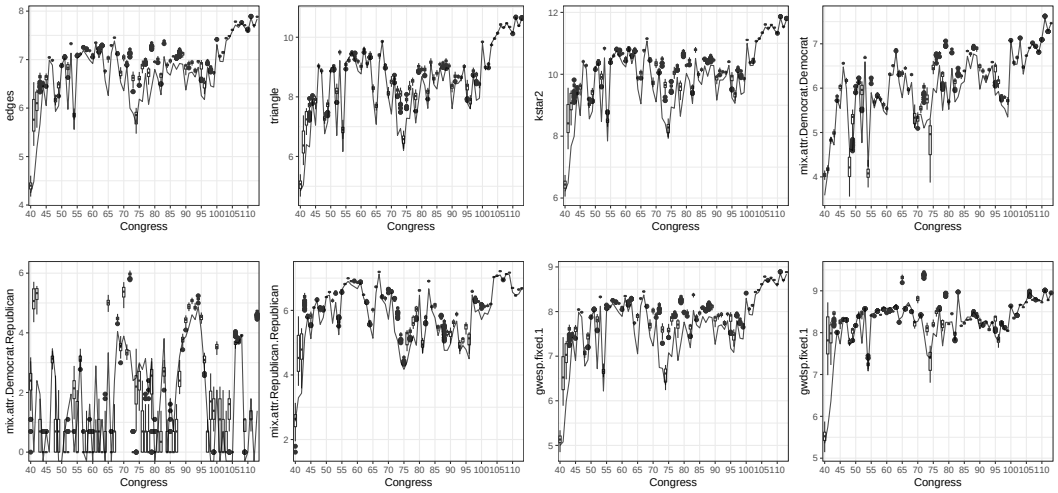

Figure 4.4.13: Model assessments of the FL-ERGM on dynamic co-voting networks. The solid lines indicate the observed network statistics and the box plots represent the statistics of networks simulated from the estimated model.

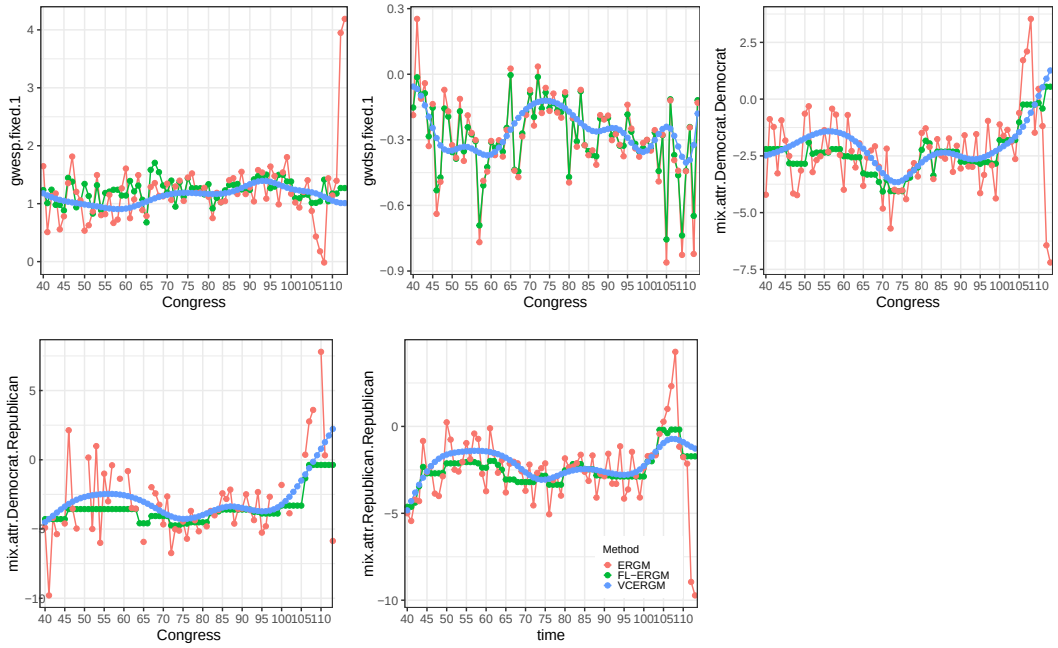

Figure 4.4.14: Parameter estimates of different methods for co-voting networks. Red lines represent the estimates from the ERGM, blue lines stand for the estimates from the VCERGM, and green lines show results from the proposed FL-ERGM.

4.5 Discussion

In this chapter, we have proposed a model for dynamic network analysis which describes various kinds of connectivity patterns within networks and model the changing patterns of network structures using a piecewise linear function. The model is an extension to ERGMs in dynamic fields, and maintains the flexibility of ERGMs. It also provides an interpretable dynamic network model that enables the inference of temporal heterogeneity in dynamic networks. Moreover, the use of fused lasso penalty also enables our model to make full use of information from neighbouring networks to obtain more accurate estimates. In addition, we have shown that the ADMM algorithm is an effective approach for statistical inference of complex dynamic networks.

Chapter 5

Bayesian Hierarchical Block Exponential Random Graph Models

In this chapter, we focus on the individual variations exhibited in ensembles of networks. Specifically, we propose a Bayesian hierarchical block exponential random graph model (BHB-ERGM) for ensembles of networks. The proposed BHB-ERGM represents the whole ensemble using a multilevel framework and describes each network in terms of block, transitive and other local structures using an individual-specific model. The number of blocks and block memberships can be determined automatically and changed adaptively with the data. A Metropolis-within-Gibbs sampling algorithm for the posterior inference of the proposed model is developed and tested through the simulated and real datasets.

5.1 Introduction

Individual variations refer to the differences in structures between individual networks. A multilevel framework provides a natural method to capture the individual variations exhibited in an ensemble of networks. The multilevel framework has been combined with ERGMs to model the individual variations of the complex

connectivity patterns across different networks. For example, in order to model an ensemble of 102 student networks, Lubbers and Snijders (2007) applied ERGMs on each network separately, and then employed a meta-analysis approach based on the ERGM parameter estimation. Wang et al. (2013) extended ERGMs for multilevel networks by introducing a variety of new specifications to represent a range of multilevel structures. Slaughter and Koehly (2016) built a multilevel ERGM for a single group of networks, and Lehmann and White (2021) extended this multilevel model to compare networks from different groups. Yin et al. (2022) proposed to model the ensemble of networks using a finite mixture of ERGMs when the group structure of the networks is unknown. However, the block structure has not been considered in the multilevel ERGMs.

Block structures occur frequently in many real networks, where the nodes in a network can be divided into different groups with similar characteristics. The ensembles of networks observed in nature are also known to exhibit block structures. There are a significant number of papers on the detection of unknown block structure based on stochastic block models and their extensions, as reviewed in Lee and Wilkinson (2019). Some of them focus on inferring block structures in the context of ensembles of networks. For example, Paul and Chen (2020) developed a random effects stochastic block model for joint community detection. Durante et al. (2017) extended latent space models using a Bayesian nonparametric approach. MacDonald et al. (2022) used frequentist latent space models to infer the shared structure of ensembles of networks.

Incorporating block structures into ERGM framework would allow us to capture the topological features within each network. Similar idea has been previously studied in the single network case. Schweinberger and Handcock (2015) is the first to characterise local dependence in exponential random graph models by taking account of the neighbourhood structure of nodes. Later, Babkin et al. (2020) developed a large-scale estimation method of the local dependence model, assuming that the number of blocks is known. Schweinberger and Stewart (2020) established concentration and consistency results for ERGMs with local dependence in

the case where the blocks are known and the block sizes are similar. In the context of ensembles of networks, unknown structures occur at both individual and population network levels. It is more challenging to simultaneously capture the various structural patterns within networks and their variations across individual networks.

To model ensembles of networks with block structures, we propose a Bayesian hierarchical block exponential random graph model (BHB-ERGM). With such a model, each network is described by individual level parameters and the whole ensemble is characterised using population level parameters. Specifically, on the individual level, each network is divided into several subnetworks based on its block structure, and each subnetwork is represented by a different ERGM. On the population level, the information from each network is collected using a Bayesian hierarchical model. Overall, the BHB-ERGM allows us to learn the similarities and differences among multiple network samples with respect to homophily, transitivity and other structural properties simultaneously.

The estimation of BHB-ERGMs is challenging. Firstly, individual and population parameters both have to be estimated to capture the multilevel structures of networks. Secondly, recovering unknown block structures of networks without preliminary information is a tricky problem. There is a rich literature on the inference of block structures using stochastic block models based on the conditional independence assumptions. But in our proposed BHB-ERGMs, edges within the same subnetwork are dependent on each other, making the problem more challenging. Thirdly, the likelihood of the ERGM is incomputable due to the existence of intractable normalising constants. This increases not only the difficulty of ERGM parameter estimation but also the difficulty of block membership identification.

In order to perform the posterior inference of the proposed BHB-ERGMs, we develop a Metropolis-within-Gibbs sampling algorithm. The Gibbs sampling algorithm is used to update variables in turn from their conditional posterior distributions. Within each step of the Gibbs sampling algorithm, the Metropolis-Hastings algorithm is applied to sample from the posterior ERGM distributions.

The proposed algorithm allows the information to be shared across different networks. Furthermore, to identify the block structure of networks, we employ a stick-breaking prior (Ishwaran and James, 2001) for the block membership variables. This allows the number of blocks and the block membership to be learned automatically from the data. As for the ERGM likelihood evaluation, we adopt the pseudo likelihood approximation method to achieve a faster computation speed. Other approximation approaches such as adjusted pseudo likelihood (Bouranis et al., 2017), Monte Carlo estimation (Neal, 2005) can also be plugged in our algorithm.

The rest of the chapter is organized as follows. In Section 5.2, we describe the formulation of BHB-ERGMs. Section 5.3 provides the sampling methodology. Section 5.4 presents the simulation studies. We summarise this chapter in Section 5.5.

5.2 Model Formulation

In this section, we propose BHB-ERGMs for ensembles of networks in three steps. We first develop a block ERGM for individual networks. Then we introduce a Bayesian hierarchical framework to combine information from different individuals. At last, we combine the block ERGM and the Bayesian hierarchical framework.

5.2.1 Block ERGMs

It is known that network data are dependent since they represent the interactions among different entities. As a consequence, each edge may depend on $n^2 - n - 1$ other possible edges in a directed network with n nodes (Babkin et al., 2020). ERGMs capture various types of connectivity patterns exhibited in the graph through exponential family distributions. The complexity of the connectivity patterns increases with the number of nodes in the network, limiting the use of ERGMs in big networks. On the other hand, block structures occur frequently

in many real networks. Given the block structure, the network is divided into a number of subnetworks that are independent of each other. In other words, using the idea of block structures can help us to constrain the dependence within each subnetwork. Here, we develop block ERGMs by combining the idea of block structures and ERGMs.

Next, we introduce some notation. Suppose that we observe an ensemble of N networks, directed or undirected, on a common set of n nodes with no self-loops. The networks are represented by N adjacency matrices $Y = (Y^{(1)}, \ldots, Y^{(N)})$, each with dimension $n \times n$. The covariate information for the ensemble is stored using a sequence of N matrices $X = (X^{(1)}, \ldots, X^{(N)})$. The nodes can be divided into K groups, with the membership variable denoted by $Z = (Z_1, \ldots, Z_n)$. Given the node membership, the edges in a single network $Y^{(i)}$ can be divided into within-group connections $Y^{(i)}_{w,j} = [Y^{(i)}_{rs}]_{Z_r = Z_s = j}$, $j = 1, \ldots, K$, and between-group connections $Y^{(i)}_b = [Y^{(i)}_{rs}]_{Z_r \neq Z_s}$. The subsets of nodes together with the within-group connections form the subnetworks of interest.

We model each subnetwork $Y^{(i)}_{w,j}$, formed by nodes from cluster j, using a separate ERGM with a specific parameter $\theta^{(i)}_{w,j} \in \mathbb{R}^{d_1}$,

$$P(Y^{(i)}_{w,j} = y^{(i)}_{w,j} | \theta^{(i)}_{w,j}, Z) = \frac{\exp\{\theta^{(i)\top}_{w,j} S(y^{(i)}_{w,j}, X^{(i)}_j)\}}{k(\theta^{(i)}_{w,j})}. \tag{5.2.1}$$

$\theta^{(i)}_{w,j}$ is the model parameter for subnetwork $y^{(i)}_{w,j}$. $S(y^{(i)}_{w,j}, X^{(i)}_j)$ is the summary statistics of the subnetwork. $k(\theta^{(i)}_{w,j}) = \sum_{y' \in \mathcal{Y}_{w,j}} \exp\{\theta^{(i)\top}_{w,j} S(y', X^{(i)}_j)\}$ is the normalising constant.

The between-group connections $Y^{(i)}_b$ are modelled via the between-group parameter $\theta^{(i)}_b \in \mathbb{R}^{d_2}$,

$$P(Y^{(i)}_b = y^{(i)}_b | \theta^{(i)}_b, Z) = \prod_{z_r \neq z_s} P(Y^{(i)}_{rs} = y^{(i)}_{rs} | \theta^{(i)}_b), \tag{5.2.2}$$

where $y^{(i)}_b$ is the set of between-group edges, formed by edges from different groups

in network $y^{(i)}$. As the between-group connections are usually much simpler compared to the within-group connections, we assume that between-group connections are independent given the block structure. To be specific, we use a Bernoulli distribution to model each between-group edge variable.

Therefore, given the latent variable Z, the likelihood of a single network $Y^{(i)}$ can be factorized using within-group and between-group connections, written as

$$P(Y^{(i)} = y^{(i)}|\theta^{(i)}, Z) = \prod_{z_r \neq z_s} P(Y_{rs}^{(i)} = y_{rs}^{(i)}|\theta_b^{(i)}) \prod_{j=1}^{K} \frac{\exp\{\theta_{w,j}^{(i)\top} S(y_{w,j}^{(i)}, X_j^{(i)})\}}{k(\theta_{w,j}^{(i)})}. \tag{5.2.3}$$

In this way, the complicated connectivity patterns within a single network $Y^{(i)}$ is captured by K within-group parameters $\theta_{w,j}^{(i)}$ and one between-group parameter $\theta_b^{(i)}$.

5.2.2 Bayesian Hierarchical Frameworks

For ensembles of networks that we are interested in, unknown structures appear at both individual and population network levels. To infer the multilevel structures of ensembles of networks, we further build the block ERGMs into a Bayesian hierarchical framework.

We assume that the individual within-group parameters $\theta_{w,j}^{(i)}$ are drawn from a multivariate normal distribution with mean vector $\mu_{w,j}$ and variance-covariance matrix $\Sigma_{w,j}$,

$$\theta_{w,j}^{(i)} \sim \mathcal{N}(\mu_{w,j}, \Sigma_{w,j}). \tag{5.2.4}$$

Here, $\mu_{w,j}$ represents the characteristics of subnetwork j at a population level. We also assume that the individual between-group parameters $\theta_b^{(i)}$ are drawn from a multivariate normal distribution with mean vector μ_b and variance-covariance

matrix Σ_b,

$$\theta_b^{(i)} \sim \mathcal{N}(\mu_b, \Sigma_b), \tag{5.2.5}$$

where μ_b describes the between-group connections of all networks.

Then we specify conjugate hyperpriors on the population parameters as follows,

$$\mu_{w,j} \sim \mathcal{N}(\mu_w^0, \Sigma_w^0),$$
$$\mu_b \sim \mathcal{N}(\mu_b^0, \Sigma_b^0).$$

For simplicity, we only place hyperpriors on the mean parameters $\{\mu_{w,j}, \mu_b\}$.

5.2.3 Bayesian Hierarchical Block ERGMs

In most cases, the node membership Z is unknown, and needs to be inferred from the observed data. In Chapter 3, we described how to use the stick-breaking process to infer the group structure of networks. Here, we borrow the strength from the stick-breaking process to infer the node membership indicator Z without preliminary information on the number of blocks.

Suppose that node r belongs to group $j = 1, \dots,$ with probability w_j, expressed as $P(Z_r = j) = w_j$. Using a stick-breaking representation, the weight variable w is constructed with an auxiliary variable v. v is sampled from a beta distribution $\text{Beta}(1, \beta)$ and $w_1 = v_1, w_j = v_j \prod_{l=1}^{j-1}(1 - v_l)$ (for $j = 2, 3, \dots$). Therefore, we

propose a Bayesian hierarchical block exponential random graph model (BHB-ERGM) with the following specification,

$$
\begin{aligned}
&v_j \sim \text{Beta}(1, \beta) \quad (j = 1, 2, \dots), \\
&w_1 = v_1, w_j = v_j \prod_{l=1}^{j-1} (1 - v_l), \\
&Z_r | w \sim \text{Multinomial}(w) \quad (r = 1, \dots, n), \\
&\mu_{w,j} \sim \mathcal{N}(\mu_w^0, \Sigma_w^0), \\
&\mu_b \sim \mathcal{N}(\mu_b^0, \Sigma_b^0), \\
&\theta_{w,j}^{(i)} \sim \mathcal{N}(\mu_{w,j}, \Sigma_{w,j}) \quad (i = 1, \dots, N), \\
&\theta_b^{(i)} \sim \mathcal{N}(\mu_b, \Sigma_b), \\
&P(Y^{(i)} = y^{(i)} | \theta^{(i)}, Z) = \prod_{z_r \neq z_s} P(Y_{rs}^{(i)} = y_{rs}^{(i)} | \theta_b^{(i)}) \prod_{j=1}^{\infty} \frac{\exp\{\theta_{w,j}^{(i)\top} S(y_{w,j}^{(i)}, X_j^{(i)})\}}{k(\theta_{w,j}^{(i)})}.
\end{aligned}
$$

Here, $\{\theta_{w,j}^{(i)}, \theta_b^{(i)}\}$ represent the characteristics of each subnetwork, and $\{\mu_{w,j}, \mu_b\}$ describe the characteristics of all subnetworks at a population level. Z indicates the shared block structure.

5.3 Posterior Computation

The statistical inference of the proposed model is challenging due to the following three reasons: the infinite number of clusters, the intractability of the ERGM likelihood, and the within-subnetwork dependence. In this section, we develop a Metropolis-within-Gibbs sampling algorithm to sample from the posterior distributions of BHB-ERGMs.

To address the infinity issue, we approximate the stick-breaking prior using a truncated stick-breaking prior based on the maximum number of clusters K_{max}, along the lines of Ishwaran and James (2001). In their work, they offered some general advice concerning the choice of K_{max}: (1) trying out multiple values of K_{max} and comparing the goodness of fit of the model; (2) exploiting available

information; (3) setting $K_{max} = n$, the number of samples to be allocated.

Here, we focus on using the goodness of fit method to find K_{max}. Given K_{max}, the membership probability w is constructed in the following way,

$$
\begin{aligned}
&v_j \sim \text{Beta}(1, \beta) \quad (j = 1, 2, \ldots, K_{max} - 1), \\
&v_{K_{max}} = 1, \\
&w_1 = v_1, w_j = v_j \prod_{l=1}^{j-1} (1 - v_l) \quad (j = 2, \ldots, K_{max}).
\end{aligned}
$$

With the approximated stick-breaking prior, the posterior distribution can be expressed as

$$
\begin{aligned}
&p(w, Z, \mu_{w,j}, \mu_b, \theta_{w,j}^{(i)}, \theta_b^{(i)} | y^{(i)}, \ldots, y^{(N)}) \\
&\propto p(v) p(Z|w) p(\mu_{w,j}) p(\mu_b) \prod_{i=1}^{N} p(\theta_{w,j}^{(i)} | \mu_{w,j}) p(\theta_b^{(i)} | \mu_b) P(Y^{(i)} = y^{(i)} | \theta^{(i)}, Z) \\
&= p(v) p(Z|w) p(\mu_{w,j}) p(\mu_b) \prod_{i=1}^{N} p(\theta_{w,j}^{(i)} | \mu_{w,j}) p(\theta_b^{(i)} | \mu_b) \prod_{z_r \neq z_s} P(Y_{rs}^{(i)} = y_{rs}^{(i)} | \theta_b^{(i)}) \\
&\times \prod_{j=1}^{K_{max}} \frac{\exp\{\theta_{w,j}^{(i)\top} S(y_{w,j}^{(i)}, X_j^{(i)})\}}{k(\theta_{w,j}^{(i)})}.
\end{aligned}
$$

This allows us to derive the full conditional distribution of each variable from the above formula. The remaining obstacle is the intractable normalising constant $k(\theta_{w,j}^{(i)})$.

Next, we propose a Metropolis-within-Gibbs sampling algorithm to sample variables $(w, Z, \mu_{w,j}, \mu_b, \theta_{w,j}^{(i)}, \theta_b^{(i)})$ in turn, and describe our solution to the intractability problem at Steps 3 and 4.

Step 1. Sample v_j from the beta distribution,

$$
\begin{aligned}
&v_j \sim \text{Beta}(1 + a_j, \beta + b_j) \quad (j = 1, 2, \ldots, K_{max}), \\
&v_{K_{max}} = 1.
\end{aligned}
$$

Here, $a_j = \sum_{r=1}^{n} \mathbf{1}(Z_r = j)$ denotes the number of nodes in group j and $b_j = \sum_{r=1}^{n} \mathbf{1}(Z_r > j)$ corresponds to the number of nodes in the groups that have larger label than j.

Update w_j with

$$w_1 = v_1, w_j = v_j \prod_{l=1}^{j-1}(1 - v_l) \quad (j = 2, \ldots, K_{max}). \tag{5.3.1}$$

Step 2. Sample $\mu = \{\mu_{w,j}, \mu_b\}$ from the multivariate normal distribution,

$$\begin{aligned} \mu_{w,j} &\sim \mathcal{N}(\mu_{w,j}^{pst}, \Sigma_{w,j}^{pst}) \quad (j = 1, 2, \ldots, K_{max}), \\ \mu_b &\sim \mathcal{N}(\mu_b^{pst}, \Sigma_b^{pst}), \end{aligned}$$

with

$$\begin{aligned} \mu_{w,j}^{pst} &= \Sigma_w^0 \left(\Sigma_w^0 + \frac{1}{N}\Sigma_{w,j}\right)^{-1} \left(\frac{1}{N}\sum_{i=1}^{N} \theta_{w,j}^{(i)}\right) + \frac{1}{N}\Sigma_{w,j} \left(\Sigma_w^0 + \frac{1}{N}\Sigma_{w,j}\right)^{-1} \mu_w^0, \\ \Sigma_{w,j}^{pst} &= \Sigma_w^0 \left(\Sigma_w^0 + \frac{1}{N}\Sigma_{w,j}\right)^{-1} \frac{1}{N}\Sigma_{w,j}, \\ \mu_b^{pst} &= \Sigma_b^0 \left(\Sigma_b^0 + \frac{1}{N}\Sigma_b\right)^{-1} \left(\frac{1}{N}\sum_{i=1}^{N} \theta_b^{(i)}\right) + \frac{1}{N}\Sigma_b \left(\Sigma_b^0 + \frac{1}{N}\Sigma_b\right)^{-1} \mu_b^0, \\ \Sigma_b^{pst} &= \Sigma_b^0 \left(\Sigma_b^0 + \frac{1}{N}\Sigma_b\right)^{-1} \frac{1}{N}\Sigma_b. \end{aligned}$$

The conjugacy property of normal distributions facilitates the sampling of $\{\mu_{w,j}, \mu_b\}$.

Step 3. Sample $\theta = \{\theta_{w,j}^{(i)}, \theta_b^{(i)}\}$ using the Metropolis-Hastings algorithm.

The posterior distribution of $\theta_{w,j}^{(i)}$ is proportional to the product of the prior

distribution and the likelihood of subnetwork $y_{w,j}^{(i)}$,

$$\begin{aligned} p(\theta_{w,j}^{(i)}|\cdots) &\propto p(\theta_{w,j}^{(i)}|\mu_{w,j})P(Y^{(i)} = y^{(i)}|\theta^{(i)}, Z) \\ &\propto p(\theta_{w,j}^{(i)}|\mu_{w,j})P(Y_{w,j}^{(i)} = y_{w,j}^{(i)}|\theta_{w,j}^{(i)}) \\ &= p(\theta_{w,j}^{(i)}|\mu_{w,j})\frac{\exp\{\theta_{w,j}^{(i)\top} S(y_{w,j}^{(i)}, X_j^{(i)})\}}{k(\theta_{w,j}^{(i)})}. \end{aligned}$$

The direct application of Metropolis-Hastings algorithm to sample from this posterior distribution is impossible as the likelihood includes an intractable normalising constant. As we have discussed in Chapter 3, there are mainly two approaches to address this issue. The first approach is to use a Monte Carlo method to estimate the normalising constant ratio in the acceptance probability of the Metropolis-Hastings algorithm. The second approach is to use a pseudo likelihood approximation to replace the likelihood of the subnetwork. Here, we adopt the second method to achieve a faster estimation speed. If there are no nodes allocated to group j, then $\theta_{w,j}^{(i)} \sim \mathcal{N}(\mu_{w,j}, \Sigma_{w,j})$.

The posterior distribution of $\theta_b^{(i)}$ is proportional to the product of the prior distribution and the likelihood of between-group connections,

$$\begin{aligned} p(\theta_b^{(i)}|\cdots) &\propto p(\theta_b^{(i)}|\mu_b)P(Y^{(i)} = y^{(i)}|\theta^{(i)}, Z) \\ &\propto p(\theta_b^{(i)}|\mu_b)P(Y_b^{(i)} = y_b^{(i)}|\theta_b^{(i)}) \\ &= p(\theta_b^{(i)}|\mu_b) \prod_{z_r \neq z_s} P(Y_{rs}^{(i)} = y_{rs}^{(i)}|\theta_b^{(i)}), \end{aligned}$$

and edge $y_{rs}^{(i)}$ is modelled using Bernoulli distribution with parameter $\theta_b^{(i)}$.

Step 4. Sample Z based on

$$\begin{aligned} P(Z_r = k|\cdots) \propto& P(Z_r = k|w)\prod_{i=1}^{N} P(Y^{(i)} = y^{(i)}|\theta^{(i)}, Z_r = k, Z_{-r}) \qquad (5.3.2) \\ =& P(Z_r = k|w)\prod_{i=1}^{N}\prod_{z_r \neq z_s} P(Y_{rs}^{(i)} = y_{rs}^{(i)}|\theta_b^{(i)}) \prod_{j=1}^{K_{max}} \frac{\exp\{\theta_{w,j}^{(i)\top} S(y_{w,j}^{(i)}, X_j^{(i)})\}}{k(\theta_{w,j}^{(i)})}. \end{aligned}$$

In stochastic block models, edges are conditionally independent given the block structure of a network. In other words, the membership of node r does not affect the probability of other edges that do not involve node r, which simplifies the computation dramatically. However, in our framework, edges from the same subnetwork depend on each other. If the membership of node r changes from k to k', the likelihood of all edges in subnetworks k and k' may be all affected. This requires us to calculate the likelihood of network $y^{(i)}$ with different values of Z_r, which is computational demanding. To address this issue, we adopt the pseudo likelihood approximation for the likelihood of each subnetwork and borrow the idea from stochastic block models to simplify between-group connections.

5.4 Empirical Results

To illustrate our proposed BHB-ERGMs, we first apply it to a set of simulated networks to show its ability to recover the ground truth. Then we compare it with HB-ERGMs (Slaughter and Koehly, 2016), the ERGMs under a hierarchical setting. Furthermore, we apply the BHB-ERGMs to an ensemble of trade networks.

5.4.1 Simulated Networks

We generate an ensemble of $N = 20$ undirected networks. Each network has $n = 90$ nodes with $K = 3$ equally sized subnetworks. For each subnetwork, we choose the number of edges $S^1(y^{(i)}) = \sum_{r<s} y_{rs}^{(i)}$ and geometrically weighted edgewise shared partner (gwesp), $S^2(y^{(i)}) = e^{\phi} \sum_{k=1}^{n-2} \{1 - (1 - e^{-\phi})^k\} EP_k(y^{(i)})$, $\phi = 0$, as the subnetwork statistics. $EP_k(y_i)$ is the number of connected pairs that have k common neighbours. With $\phi = 0$, we have $S^2(y^{(i)}) = \sum_{k=1}^{n-2} EP_k(y^{(i)})$. For the between-subnetwork connections, we consider the number of edges only. There are three steps in generating simulated networks. Firstly, we specify the

within-subnetwork parameters as follows,

$$\mu_w = \begin{pmatrix} -2 & -2 & -2 \\ 0 & 0.25 & 0.5 \end{pmatrix}, \ \Sigma_w = \begin{pmatrix} 0.05^2 & 0 \\ 0 & 0.05^2 \end{pmatrix}.$$

The parameters for the between-subnetwork connections are $\mu_b = -4$, $\Sigma_b = 0.05^2$. Secondly, we simulate the individual parameters from the population parameters, $\theta_{w,j}^{(i)} \sim \mathcal{N}(\mu_{w,j}, \Sigma_{w,j})$ $(j = 1, \ldots, K)$, $\theta_b^{(i)} \sim \mathcal{N}(\mu_b, \Sigma_b)$. Thirdly, we generate simulated networks $y^{(i)}$ $(i = 1, \ldots, N)$ with $\theta^{(i)}$ as parameters.

For the simulated network ensemble $y^{(i)}$, we fit it with the Metropolis-within-Gibbs sampling algorithm. The maximum number of clusters is set as $K_{max} = 10$. The hyperparameters in the prior distribution are set as $\mu_w^0 = (-2, 0)$, $\Sigma_w^0 = I_2$, $\mu_b^0 = -3$ and $\Sigma_b^0 = 1$. The proposal distribution is $\mathcal{N}(\mathbf{0}, 0.1^2 I_2)$ for within-subnetwork parameters and $\mathcal{N}(0, 0.2^2)$ for between-subnetwork connections. I_2 denotes a two-dimensional diagonal matrix. The simulation is run for 10,000 iterations with the first 2,000 as burn-in. We present the number of occupied clusters at Figure 5.4.1. The nodes are correctly allocated to the true group. The We also check that MCMC chains for all individual parameters $\{\theta_{w,j}^{(i)}, \theta_b^{(i)}\}$ mix well. The acceptance ratios are between 0.4 and 0.5. We also present the posterior density plots of population parameters $\mu_{w,j}$ in Figure 5.4.2 and μ_b in Figure 5.4.3. The posterior density plots cover the true parameter values well.

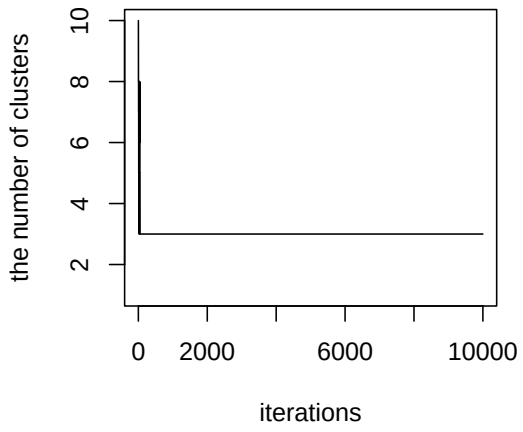

Figure 5.4.1: Trace plot for the number of occupied clusters.

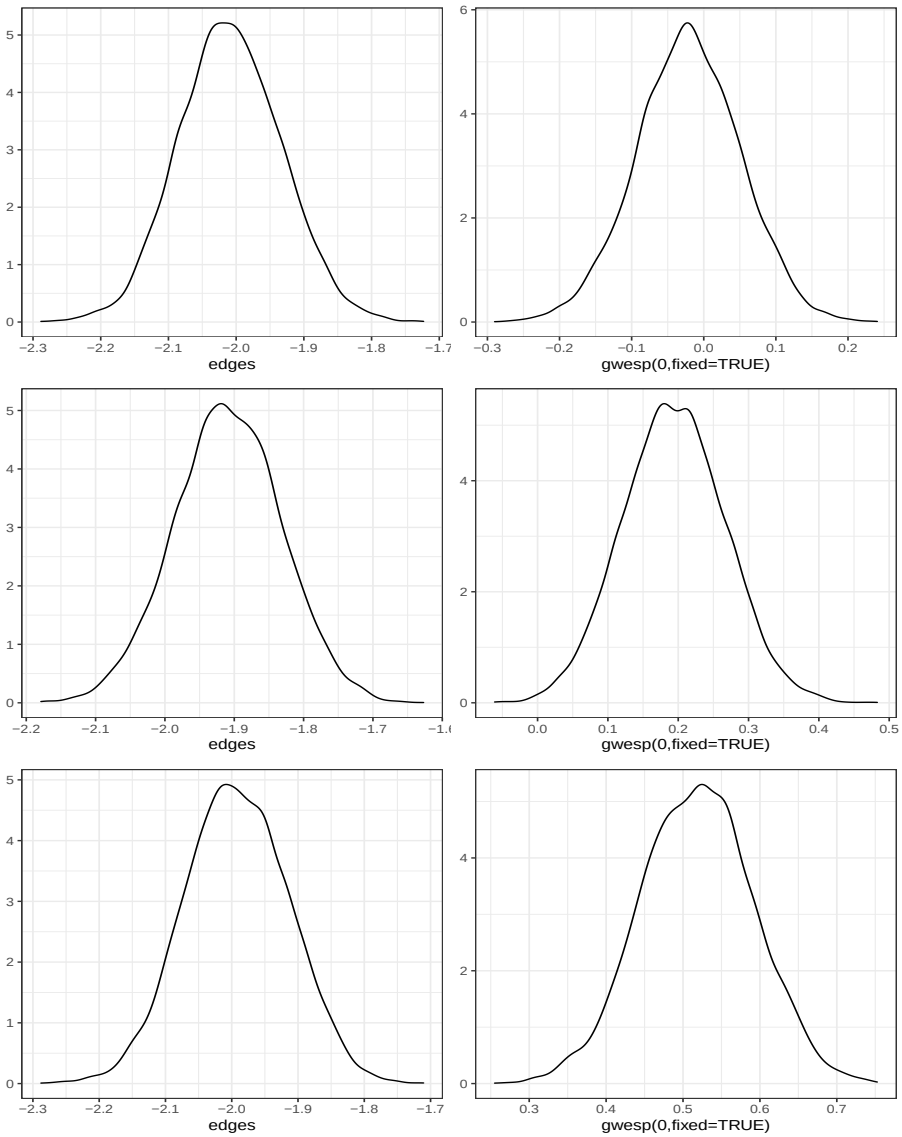

Figure 5.4.2: Posterior density plots of the within-subnetwork parameters using BHB-ERGMs. Each row corresponds to the characteristics of a population of subnetworks formed a specific set of nodes.

We repeat the same process 50 times and calculate the mean and standard deviation of all posterior means, shown in Table 5.4.1. As we can see, the estimates are close to the true value.

Furthermore, we also apply ERGMs to the same simulated networks at the hierarchical Bayesian setting, following the method of Slaughter and Koehly (2016). We refer to this method as HB-ERGM. Compared to our proposed BHB-ERGM,

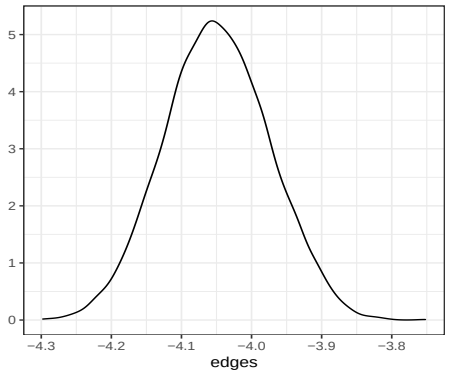

Figure 5.4.3: The posterior density plot of the between-subnetwork parameter using BHB-ERGMs at the population level.

Table 5.4.1: Mean (standard deviation) of posterior means of 50 repetitions.

	subnetwork 1	subnetwork 2	subnetwork 3	between-subnetwork
edges	-2.00 (0.06)	-2.02 (0.06)	-2.02 (0.07)	-4.01(0.03)
gwesp	-0.03 (0.05)	0.25 (0.04)	0.50 (0.05)	0

the HB-ERGM does not consider block structures of networks. The posterior density plots of the population parameters μ are shown in Figure 5.4.4.

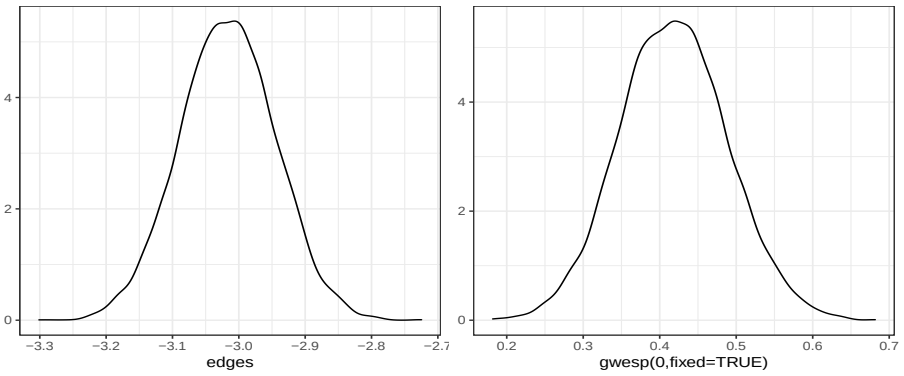

Figure 5.4.4: Posterior density plots of the population parameters using HB-ERGMs.

To assess the performance of both methods, we calculate the mean squared error (MSE) and the mean absolute error (MAE) to measure the distance between predicted network summary statistics and observed network summary statistics

as follows,

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{S}(z^{(i)}) - S(y^{(i)})\right)^2,$$
$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{S}(z^{(i)}) - S(y^{(i)})|.$$

Here, $S(y^{(i)})$ stands for the statistics of the networks we are modelling, $y^{(i)}$ $(i = 1, \ldots, N)$. $\hat{S}(z^{(i)})$ is the average summary statistics of the predicted networks, which are generated based on individual estimation $\hat{\theta}^{(i)}$. We generate predicted networks using both estimation of BHB-ERGMs and HB-ERGMs and present the results in Table 5.4.2. It is shown that BHB-ERGMs perform better than HB-ERGMs suggested by the smaller values of MSE and MAE.

Table 5.4.2: Mean (standard deviation) of MSE and MAE of 50 repetitions.

		BHB-ERGMs	HB-ERGMs
edges	MSE	**26.82** (9.35)	372.57 (128.27)
	MAE	**4.28** (0.84)	16.31 (2.73)
gwesp	MSE	**48.00** (13.15)	227.03 (107.91)
	MAE	**5.58** (0.86)	11.47 (2.46)

5.4.2 International Trade Networks

We also apply the proposed BHB-ERGMs to an ensemble of trade networks. The ensemble of trade networks is observed on 60 countries ($n = 60$) over the period 2001-2016 ($N = 16$), denoted as $y^{(i)}(i = 1, 2, \ldots, 16)$. The networks are built based on the annual import data between every two countries from the UN Comtrade website[1]. A directed edge exists from node r to s, $y_{rs}^{(i)} = 1$, if the import amount from country r to s is more than 3 billion dollars at year i.

In this application, we choose two statistics to model the trade networks:

[1]https://comtrade.un.org/

- $S^1(y_i) = \sum_{r \neq s} y_{rs,i}$, the total number of edges in network y_i. The density of trade networks can reflect the universality of global trade relationship.

- $S^2(y_i) = e^{\phi} \sum_{k=1}^{n-2} \{1 - (1 - e^{-\phi})^k\} DP_k(y_i)$, $\phi = 0.25$, geometrically weighted dyadic shared partner (gwdsp), representing transitivity. $DP_k(y_i)$ is the number of connected pairs that share k common neighbours.

Next, we fit the model using the Metropolis-within-Gibbs sampling algorithm. The maximum number of clusters is set as $K_{max} = 10$. We use the same initial points as in the simulation studies. The hyperparameter in the population mean prior distribution is set as $\mu_w^0 = (-2, 0)$, $\Sigma_w^0 = 0.5 I_2$, $\mu_b^0 = -3$ and $\Sigma_b^0 = 1$. The proposal distribution in the Metropolis-Hastings algorithm is $\mathcal{N}(\mathbf{0}, 0.1^2 I_2)$ for within-subnetwork parameters and $\mathcal{N}(0, 0.2^2)$ for between-subnetwork connections. I_2 denotes a two-dimensional diagonal matrix. We run four independent chains for 20,000 iterations and present the number of occupied clusters at Figure 5.4.5. We also check the convergence of each parameter.

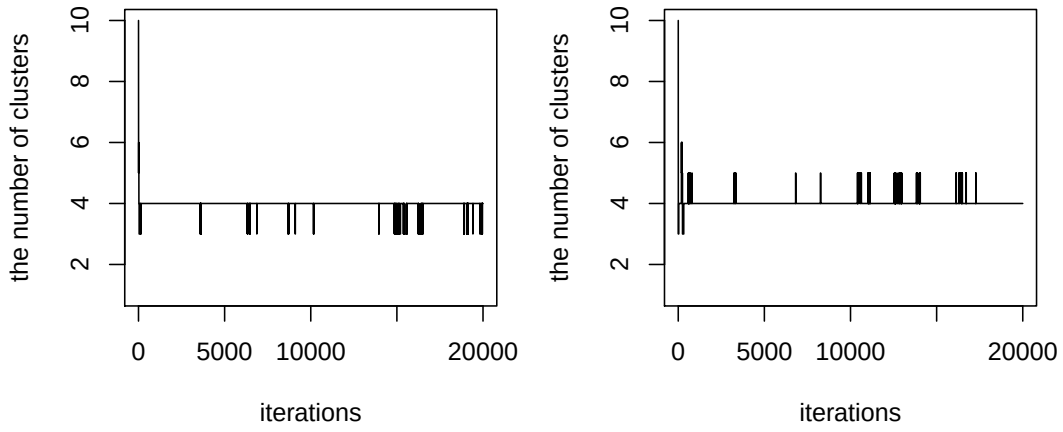

Figure 5.4.5: Trace plot for the number of occupied clusters of two different chains.

Four stable groups are obtained from the model results. The members of each subnetwork are listed as follows:

- Group 1: United States, China, Germany, United Kingdom, Japan, France, Korea, Italy, Canada, Netherlands, Belgium, India, Spain, Singapore, Switzerland, Turkey, Thailand, Australia, Russia, Malaysia, Austria, Sweden, Brazil,

Indonesia, Denmark, Ireland, Norway, Finland,

- Group 2: Philippines, South Africa, Romania, Portugal, Israel, Chile, Greece, Colombia, New Zealand, Peru, Slovenia, Bulgaria, Belarus, Lithuania, Kazakhstan, Croatia, Sri Lanka, Tunisia, Luxembourg, Dominica Republic, Guatemala, Ethiopia, Ecuador, Estonia, Costa Rica, Uruguay, Tanzania,
- Group 3: Czech Republic, Slovakia, Hungary, Poland,
- Group 4: Mexico.

We get four subnetworks, each of which has its unique characteristics. We present the summary statistics of each subnetwork in Table 5.4.3. Subnetwork 1 is densely connected, with a positive gwdsp parameter (as shown in Figure 5.4.6) indicating high transitivity. Most of the 28 countries in subnetwork 1 are developed countries. Only a few connections are observed in subnetwork 2, and the gwdsp statistics are always 0. Subnetwork 3 is formed by four countries that are geographically located next to each other. They are all members of the Visegrad (V4) group. These four countries do not have any connections at years 2001 and 2002, but are fully connected at years 2011, 2013, 2014, 2015 and 2016, as shown in Figure 5.4.7. Mexico forms its own group most of the time. Although it is a big manufacturing country, its trade relationship is less wide ranging compared to countries in subnetwork 1 as it mostly trades with a couple of specific countries. For example, Mexico has a large amount of trade with the United States. The posterior density plot of between-subnetwork connections is shown in Figure 5.4.8.

Table 5.4.3: Mean (standard deviation) of summary statistics of the subnetworks.

	subnetwork 1	subnetwork 2	subnetwork 3	subnetwork 4
size	28	27	4	1
edges	371.81 (67.48)	0.31 (0.79)	7.44(4.53)	0 (0)
gwdsp	954.08 (28.26)	0 (0)	8.20 (6.19)	0 (0)

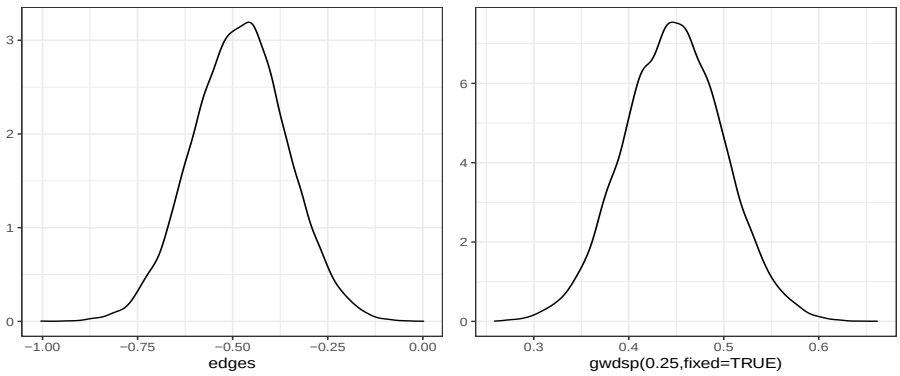

Figure 5.4.6: Posterior density plots of model parameters for subnetwork 1.

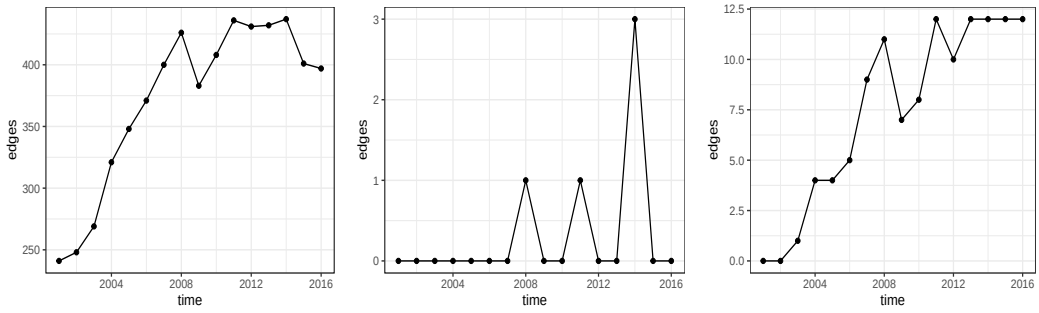

Figure 5.4.7: The number of edges of subnetworks 1 (left), 2(middle) and 4(right) at different years.

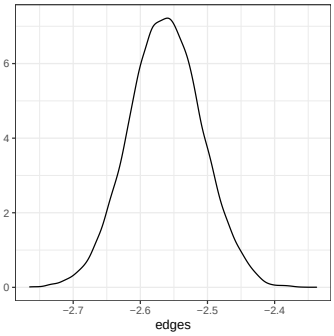

Figure 5.4.8: The posterior density plot of the parameter for between-subnetwork connections.

5.5 Discussion

The analysis of connectivity patterns is crucial in understanding the process underlying the formation of networks. The main contribution of this work is that we have developed a multilevel framework to simultaneously model the various connectivity patterns and their variations within and across blocks in the ensemble of networks. The inference of such a framework is challenging due to the within-subnetwork dependency, the intractability of the ERGM likelihood and the problem associated with MCMC for hierarchical models. To address these issues, we have developed a novel MCMC algorithm. The proposed algorithm can adapt easily to different datasets. Moreover, we have applied our method to an ensemble of trade networks. Our model detects the common block structure of the trade ensemble, captures the dependency within each individual subnetworks and also presents the characteristics of subnetworks at a population level.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this thesis, three strategies on the modelling of ensembles of networks have been presented. After an introduction and literature review, different characteristics of the ensemble have been individually described in Chapters 3, 4, 5.

In Chapter 3, we have presented a novel methodology for modelling the generative process of ensembles of networks. This model is the first approach to apply a Bayesian nonparametric method on ERGMs, to the best of our knowledge. It provides a highly flexible and interpretable framework for characterising the complex generative process of ensembles. With such a model, the subpopulations consisting of similar networks can be detected and compared automatically without requiring a fixed number of clusters in advance. We have also developed a novel IMS sampling algorithm for the full Bayesian inference of the model and a PMS sampling algorithm for the fast estimation of the model.

In Chapter 4, we have proposed a new method to model the temporal variations of dynamic ensembles in terms of the effects of various local connectivity patterns on the shaping of global network structure over time. By incorporating a fused lasso penalty in the model, the time correlation between different networks has been modelled realistically using a piecewise linear function. We have also discussed how to use a powerful optimisation method, the ADMM algorithm, to

estimate a sequence of dynamic networks simultaneously. In particular, we have employed a LU matrix decomposition technique to facilitate the inference on the ensemble with a large number of network observations. We have also shown that our proposed model can present the changing trend of different effects as well as detect abrupt changes of network structures.

In Chapter 5, we have proposed a Bayesian hierarchical framework to model the ensembles of networks with block structures using ERGMs. The proposed model does not only detect the common block structures of all network samples, but it also describes the individual and population characteristics of subnetworks. One obstacle of the estimation is how to distinguish block structures and various structural patterns from ERGMs in the generative process of networks. We have also developed a Metropolis-within-Gibbs sampling algorithm to address these issues while keeping the flexibility on the number of blocks.

6.2 Future Work

The flexibility of both ERGMs and hierarchical models offers us many promising opportunities to further develop our framework. First, we plan to develop a unified framework to infer the multilevel group structures of ensembles using ERGMs, in other words, to develop a framework that can cluster networks and nodes simultaneously. Second, we plan to include network level covariates in the current Bayesian hierarchical framework to study how covariates affect the differences between networks. For example, this can provide insights on how personal activities such as exercise influence the connectivity patterns of different brain regions. Third, in addition to the common block structure of ensembles, we will also consider the network block structure of each network. This would help us to account for more heterogeneity on the generative processes of ensembles. Fourth, for ensembles with a time order, it is often of interest to detect change points in the network. It would be interesting to further analyse how to utilize dynamic network models like the FL-ERGM to identify those points where the network

undergoes significant local or global structural changes.

More accurate and faster approximations of the ERGM likelihood are also potential directions for future research. Current available Monte Carlo approximations are accurate but time consuming as generating networks from desired ERGMs requires a long run of Markov chains. Pseudo likelihood approximations are fast but should be treated with caution because they can lead to an unreasonable inference. More recently, faster Bayesian inference methods, such as variation inference (Tan and Friel, 2020) and approximate Bayesian inference (Yin and Butts, 2020), have been applied to ERGMs to pursue a better estimation. All the methods mentioned above are developed for the case of a single network. It is worthwhile to explore the use of fast Bayesian inference methods on the ensembles of networks. Moreover, if we could make use of the features of the ensemble, such as the similarities between network samples, to calculate the likelihood of the ensemble without calculating the likelihood of each network sample, we could also save computing time.

Although Bayesian nonparametric analysis based on stochastic block models and latent space models has attracted much attention in the last decade, the Bayesian nonparametric analysis of ERGMs has not been given the desired attention. This is largely due to the intractable normalising constant in the ERGM likelihood. However, as we mentioned before, Bayesian nonparametric methods have both the advantages of Bayesian analysis and nonparametric models, the model complexity of which can change with the data. On the other hand, ERGMs are very flexible in the way that various dependency structures can be incorporated into the model. They have the advantages that other network models do not have. Therefore, it is interesting to extend ERGMs using other stochastic processes such as Gaussian processes or beta processes, in addition to Dirichlet processes. The Bayesian inference of such frameworks would also be worthwhile to explore.

Appendix A

Appendix

A.1 Taylor Expansion

The function to expand is

$$f(\theta_t) = y_{ijt}\theta_t^\top \Delta S_{ijt} - \log\left\{1 + \exp\left(\theta_t^\top \Delta S_{ijt}\right)\right\},$$

with first and second derivative as follows,

$$\frac{\partial f(\theta_t)}{\partial \theta_t} = y_{ijt}\Delta S_{ijt} - \frac{\exp(\theta_t^\top \Delta S_{ijt})}{1 + \exp(\theta_t^\top \Delta S_{ijt})}\Delta S_{ijt},$$
$$\frac{\partial^2 f(\theta_t)}{\partial \theta_t^2} = -\frac{\exp(\theta_t^\top \Delta S_{ijt})}{(1 + \exp(\theta_t^\top \Delta S_{ijt}))^2}\Delta S_{ijt}\Delta S_{ijt}^\top.$$

The second Taylor expansion at $\tilde{\theta}_t$ is

$$
\begin{aligned}
& f(\tilde{\theta}_t) + (\theta_t - \tilde{\theta}_t)^\top f'(\tilde{\theta}_t) + \frac{1}{2}(\theta_t - \tilde{\theta}_t)^\top f''(\tilde{\theta}_t)(\theta_t - \tilde{\theta}_t) \\
=& f(\tilde{\theta}_t) + (\theta_t - \tilde{\theta}_t)^\top \Delta S_{ijt}(y_{ijt} - \frac{\exp(\tilde{\theta}_t{}^\top \Delta S_{ijt})}{1 + \exp(\tilde{\theta}_t{}^\top \Delta S_{ijt})}) \\
& - \frac{\exp(\tilde{\theta}_t{}^\top \Delta S_{ijt})}{2(1 + \exp(\tilde{\theta}_t{}^\top \Delta S_{ijt}))^2}(\theta_t - \tilde{\theta}_t)^\top \Delta S_{ijt} \Delta S_{ijt}^\top (\theta_t - \tilde{\theta}_t) \\
=& -\frac{1}{2} w_{ijt}[(\theta_t - \tilde{\theta}_t)^\top \Delta S_{ijt}]^2 + (\theta_t - \tilde{\theta}_t)^\top \Delta S_{ijt}(y_{ijt} - \tilde{p}_{ijt}) + f(\tilde{\theta}_t) \\
=& -\frac{1}{2} w_{ijt} \left\{ [(\theta_t - \tilde{\theta}_t)^\top \Delta S_{ijt}]^2 - \frac{2(\theta_t - \tilde{\theta}_t)^\top \Delta S_{ijt}(y_{ijt} - \tilde{p}_{ijt})}{w_{ijt}} \right\} + f(\tilde{\theta}_t) \\
=& -\frac{1}{2} w_{ijt} \left\{ [(\theta_t - \tilde{\theta}_t)^\top \Delta S_{ijt}]^2 - \frac{2(\theta_t - \tilde{\theta}_t)^\top \Delta S_{ijt}(y_{ijt} - \tilde{p}_{ijt})}{w_{ijt}} + (\frac{y_{ijt} - \tilde{p}_{ijt}}{w_{ijt}})^2 \right\} \\
& - \frac{(y_{ijt} - \tilde{p}_{ijt})^2}{2w_{ijt}} + f(\tilde{\theta}_t) \\
=& -\frac{1}{2} w_{ijt} \left\{ (\theta_t - \tilde{\theta}_t)^\top \Delta S_{ijt} - \frac{y_{ijt} - \tilde{p}_{ijt}}{w_{ijt}} \right\}^2 - \frac{(y_{ijt} - \tilde{p}_{ijt})^2}{2w_{ijt}} + f(\tilde{\theta}_t) \\
=& -\frac{1}{2} w_{ijt} \left\{ \theta_t^\top \Delta S_{ijt} - \tilde{\theta}_t{}^\top \Delta S_{ijt} - \frac{y_{ijt} - \tilde{p}_{ijt}}{w_{ijt}} \right\}^2 - \frac{(y_{ijt} - \tilde{p}_{ijt})^2}{2w_{ijt}} + f(\tilde{\theta}_t) \\
=& -\frac{1}{2} w_{ijt} \left\{ \theta_t^\top \Delta S_{ijt} - z_{ijt} \right\}^2 + C(\tilde{\theta}_t),
\end{aligned}
$$

where

$$
\begin{aligned}
\tilde{p}_{ijt} &= \frac{\exp(\tilde{\theta}_t{}^\top \Delta S_{ijt})}{1 + \exp(\tilde{\theta}_t{}^\top \Delta S_{ijt})}, \\
w_{ijt} &= \tilde{p}_{ijt}(1 - \tilde{p}_{ijt}) = \frac{\exp(\tilde{\theta}_t{}^\top \Delta S_{ijt})}{(1 + \exp(\tilde{\theta}_t{}^\top \Delta S_{ijt}))^2}, \\
z_{ijt} &= \tilde{\theta}_t{}^\top \Delta S_{ijt} + \frac{y_{ijt} - \tilde{p}_{ijt}}{w_{ijt}}, \\
C(\tilde{\theta}_t) &= -\frac{(y_{ijt} - \tilde{p}_{ijt})^2}{2w_{ijt}} + f(\tilde{\theta}_t).
\end{aligned}
$$

A.2 Proof of Lemma 1

To prove Lemma 1, we only need to show that the Hessian matrix for $l(\theta_1, \ldots, \theta_T)$ and $\tilde{l}(\theta_1, \ldots, \theta_T)$ are positive definite.

For any $t \in \{1, \ldots, T\}$, we have

$$\frac{\partial^2 l(\theta_1, \ldots, \theta_T)}{\partial \theta_t^2} = \sum_{i \neq j} \frac{\Delta S_{ijt} \Delta S_{ijt}^\top \exp(\theta_t^\top \Delta S_{ijt})}{(1 + \exp(\theta_t^\top \Delta S_{ijt}))^2}.$$
$$\frac{\partial^2 \tilde{l}(\theta_1, \ldots, \theta_T)}{\partial \theta_t^2} = \sum_{i \neq j} w_{ijt} \Delta S_{ijt} \Delta S_{ijt}^\top.$$

In addition, for $t \neq s, t, t \in \{1, \ldots, T\}$, we have

$$\frac{\partial^2 l(\theta_1, \ldots, \theta_T)}{\partial \theta_t \partial \theta_s} = 0,$$
$$\frac{\partial^2 \tilde{l}(\theta_1, \ldots, \theta_T)}{\partial \theta_t \partial \theta_s} = 0.$$

Thus the Hessian matrix for $l(\theta_1, \ldots, \theta_T)$ is

$$\begin{bmatrix} \frac{\partial^2 l}{\partial \theta_1^2} & \frac{\partial^2 l}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 l}{\partial \theta_1 \partial \theta_T} \\ \frac{\partial^2 l}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 l}{\partial \theta_2^2} & \cdots & \frac{\partial^2 l}{\partial \theta_2 \partial \theta_T} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 l}{\partial \theta_T \partial \theta_1} & \frac{\partial^2 l}{\partial \theta_T \partial \theta_2} & \cdots & \frac{\partial^2 l}{\partial \theta_T^2} \end{bmatrix}$$
$$= \sum_{i \neq j} \begin{bmatrix} \frac{\Delta S_{ij1} \Delta S_{ij1}^\top \exp(\theta_1^\top \Delta S_{ij1})}{(1+\exp(\theta_1^\top \Delta S_{ij1}))^2} & 0 & \cdots & 0 \\ 0 & \frac{\Delta S_{ij2} \Delta S_{ij2}^\top \exp(\theta_2^\top \Delta S_{ij2})}{(1+\exp(\theta_2^\top \Delta S_{ij2}))^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\Delta S_{ijT} \Delta S_{ijT}^\top \exp(\theta_T^\top \Delta S_{ijT})}{(1+\exp(\theta_T^\top \Delta S_{ijT}))^2} \end{bmatrix},$$

and the Hessian matrix of $\tilde{l}(\theta_1, \ldots, \theta_T)$ is

$$\begin{bmatrix} \frac{\partial^2 \tilde{l}}{\partial \theta_1^2} & \frac{\partial^2 \tilde{l}}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 \tilde{l}}{\partial \theta_1 \partial \theta_T} \\ \frac{\partial^2 \tilde{l}}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 \tilde{l}}{\partial \theta_2^2} & \cdots & \frac{\partial^2 \tilde{l}}{\partial \theta_2 \partial \theta_T} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \tilde{l}}{\partial \theta_T \partial \theta_1} & \frac{\partial^2 \tilde{l}}{\partial \theta_T \partial \theta_2} & \cdots & \frac{\partial^2 \tilde{l}}{\partial \theta_T^2} \end{bmatrix}$$

$$= \sum_{i \neq j} \begin{bmatrix} w_{ij1} \Delta S_{ij1} \Delta S_{ij1}^\top & 0 & \cdots & 0 \\ 0 & w_{ij2} \Delta S_{ij2} \Delta S_{ij2}^\top & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_{ijT} \Delta S_{ijT} \Delta S_{ijT}^\top \end{bmatrix}.$$

It is obvious to see that the Hessian matrix for $l(\theta_1, \ldots, \theta_T)$ and $\tilde{l}(\theta_1, \ldots, \theta_T)$ are positive definite, so $l(\theta_1, \ldots, \theta_T)$ and $\tilde{l}(\theta_1, \ldots, \theta_T)$ are convex.

A.3 Proof of Theorem 2

The main idea of the proof is to use three inequalities.

We define a Lyapunov function for the algorithm (which is a nonnegative quantity that decreases at each iteration) as below,

$$V^k = \rho \|U^k - U^*\|_2^2 + \rho \| - (Q^k - Q^*)\|_2^2.$$

The first inequality is

$$V^{k+1} \leq V^k - \rho \|F\Theta^{k+1} - Q^{k+1}\|_2^2 - \rho \| - (Q^{k+1} - Q^k)\|_2^2, \tag{A.3.1}$$

which states that V^k decreases at each iteration. In addition, the amount of decrease depends on the norm of the prime residual $F\Theta^{k+1} - Q^{k+1}$ and on the change of Q over one iteration. Since $V^k \leq V^0$, thus U^k and $-Q^k$ are bounded.

From inequality (A.3.1), we have

$$\rho \sum_{k=0}^{\infty} (\|F\Theta^{k+1} - Q^{k+1}\|_2^2 + \| - (Q^{k+1} - Q^k)\|_2^2) \leq V^0, \tag{A.3.2}$$

which implies that $F\Theta^{k+1} - Q^{k+1} \to 0$ and $-(Q^{k+1} - Q^k) \to 0$ as $k \to \infty$. Thus the primal residual $F\Theta^{k+1} - Q^{k+1}$ and dual residual $-\rho F^\top (Q^{k+1} - Q^k)$ converge to zero.

The second inequality is

$$\frac{1}{2}\|A^*\Theta^* - b^*\|_2^2 + \lambda\|Q^*\|_1 - \frac{1}{2}\|A^{k+1}\Theta^{k+1} - b^{k+1}\|_2^2 - \lambda\|Q^{k+1}\|_1 \leq \rho U^{*\top}(F\Theta^{k+1} - Q^{k+1}). \tag{A.3.3}$$

The third inequality we are going to prove is

$$\begin{aligned} &\frac{1}{2}\|A^{k+1}\Theta^{k+1} - b^{k+1}\|_2^2 + \lambda\|Q^{k+1}\|_1 - \frac{1}{2}\|A^*\Theta^* - b^*\|_2^2 - \lambda\|Q^*\|_1 \qquad \text{(A.3.4)} \\ &\leq -(\rho U^{k+1})^\top (F\Theta^{k+1} - Q^{k+1}) - \rho\{(Q^{k+1} - Q^k)\}^\top \{(F\Theta^{k+1} - Q^{k+1}) + (Q^{k+1} - Q^*)\}. \end{aligned}$$

Using the results of inequality (A.3.2), we see that the righthand side of (A.3.3) also converges to 0 when $k \to \infty$, since the prime residual $F\Theta^{k+1} - Q^{k+1}$ converges to 0. The righthand side in (A.3.4) converges to 0 when $k \to \infty$, since $-(Q^{k+1} - Q^*)$ is bounded and $F\Theta^{k+1} - Q^{k+1}$ and $-(Q^{k+1} - Q^k)$ converge to 0. Thus we have $\lim_{k\to\infty}\{(1/2)\|A^{k+1}\Theta^{k+1} - b^{k+1}\|_2^2 + \lambda\|Q^{k+1}\|_1\} = (1/2)\|A^*\Theta^* - b^*\|_2^2 + \lambda\|Q^*\|_1$, which is the objective convergence. If assumption 3 holds, Θ^k converges to a solution of (4.3.4).

Next, we prove the three inequalities. We first prove inequality (A.3.3). Since (Θ^*, Q^*, U^*) is a saddle point for $(1/2)\|A\Theta - b\|_2^2 + \lambda\|Q\|_1 + \rho U^\top (F\Theta - Q)$, we

have

$$
\begin{aligned}
&(1/2)\|A^*\Theta^* - b^*\|_2^2 + \lambda\|Q^*\|_1 + \rho U^{*\top}(F\Theta^* - Q^*) \\
&\leq (1/2)\|A^{k+1}\Theta^{k+1} - b^{k+1}\|_2^2 + \lambda\|Q^{k+1}\|_1 + \rho U^{*\top}(F\Theta^{k+1} - Q^{k+1}).
\end{aligned}
$$

As $F\Theta^* - Q^* = 0$, the lefthand side of the above inequality equals to $(1/2)\|A^*\Theta^* - b^*\|_2^2 + \lambda\|Q^*\|_1$. Subtracting both parts with $(1/2)\|A^{k+1}\Theta^{k+1} - b^{k+1}\|_2^2 + \lambda\|Q^{k+1}\|_1$, then we have proved inequality (A.3.3).

Second, we prove inequality (A.3.4). Θ^{k+1} minimizes the augmented Lagrangian $(1/2)\|A\Theta - b\|_2^2 + \lambda\|Q^k\|_1 + \rho U^{k\top}(F\Theta - Q^k) + (\rho/2)\|F\Theta - Q^k\|_2^2$. Because $(1/2)\|A\Theta - b\|_2^2$ is closed, proper, and convex, it is subdifferentiable, and so is the augmented Lagrangian $(1/2)\|A\Theta - b\|_2^2 + \lambda\|Q^k\|_1 + \rho U^{k\top}(F\Theta - Q^k) + (\rho/2)\|F\Theta - Q^k\|_2^2$. The (necessary and sufficient) optimality condition is

$$
0 \in \partial\{(1/2)\|A^{k+1}\Theta^{k+1} - b^{k+1}\|_2^2\} + \rho F^\top U^k + \rho F^\top(F\Theta^{k+1} - Q^k). \tag{A.3.5}
$$

Since $U^{k+1} = U^k + F\Theta^{k+1} - Q^{k+1}$, we plug in $U^k = U^{k+1} - (F\Theta^{k+1} - Q^{k+1})$ and rearrange (A.3.5):

$$
0 \in \partial\{(1/2)\|A^{k+1}\Theta^{k+1} - b^{k+1}\|_2^2\} + F^\top\{\rho U^{k+1} + \rho(Q^{k+1} - Q^k)\}. \tag{A.3.6}
$$

(A.3.6) shows that Θ^{k+1} minimizes

$$
\{(1/2)\|A\Theta - b\|_2^2\} + \{\rho U^{k+1} + \rho(Q^{k+1} - Q^k)\}^\top F\Theta.
$$

Thus we have

$$
\begin{aligned}
&(1/2)\|A^{k+1}\Theta^{k+1} - b^{k+1}\|_2^2 + \{\rho U^{k+1} + \rho(Q^{k+1} - Q^k)\}^\top F\Theta^{k+1} \\
&\leq (1/2)\|A^*\Theta^* - b^*\|_2^2 + \{\rho U^{k+1} + \rho(Q^{k+1} - Q^k)\}^\top F\Theta^*.
\end{aligned} \tag{A.3.7}
$$

A similar argument implies that Q^{k+1} minimizes $\lambda\|Q\|_1 - \rho(U^{k+1})^\top Q$, and we

have

$$\lambda\|Q^{k+1}\|_1 - \rho(U^{k+1})^\top Q^{k+1} \leq \lambda\|Q^*\|_1 - \rho(U^{k+1})^\top Q^*. \tag{A.3.8}$$

Adding inequalities (A.3.7) and (A.3.8), we obtain

$$\begin{aligned}
&\frac{1}{2}\|A^{k+1}\Theta^{k+1} - b^{k+1}\|_2^2 - \frac{1}{2}\|A^*\Theta^* - b^*\|_2^2 + \lambda\|Q^{k+1}\|_1 - \lambda\|Q^*\|_1 \\
&\leq \{\rho U^{k+1} + \rho(Q^{k+1} - Q^k)\}^\top F\Theta^* - \{\rho U^{k+1} + \rho(Q^{k+1} - Q^k)\}^\top F\Theta^{k+1} \\
&- (\rho U^{k+1})^\top Q^* + (\rho U^{k+1})^\top Q^{k+1}.
\end{aligned}$$

Then we plug in $F\Theta^* - Q^* = 0$ to the righthand side of the above inequality, and we obtain the righthand side of the (A.3.4)

$$\begin{aligned}
&\{\rho U^{k+1} + \rho(Q^{k+1} - Q^k)\}^\top F(\Theta^* - \Theta^{k+1}) + (\rho U^{k+1})^\top (Q^{k+1} - Q^*) \\
=&\{\rho U^{k+1}\}^\top F(\Theta^* - \Theta^{k+1}) + \{\rho(Q^{k+1} - Q^k)\}^\top F(\Theta^* - \Theta^{k+1}) + (\rho U^{k+1})^\top (Q^{k+1} - Q^*) \\
=&\{\rho U^{k+1}\}^\top (Q^* - F\Theta^{k+1}) + (\rho U^{k+1})^\top (Q^{k+1} - Q^*) + \{\rho(Q^{k+1} - Q^k)\}^\top (Q^* - F\Theta^{k+1}) \\
=&(\rho U^{k+1})^\top (Q^{k+1} - F\Theta^{k+1}) + \rho\{(Q^{k+1} - Q^k)\}^\top \{Q^* - F\Theta^{k+1})\} \\
=&-(\rho U^{k+1})^\top (F\Theta^{k+1} - Q^{k+1}) - \rho\{(Q^{k+1} - Q^k)\}^\top \{(F\Theta^{k+1} - Q^{k+1}) + (Q^{k+1} - Q^*).\}
\end{aligned}$$

Thus (A.3.4) is proved.

The third inequality to prove is (A.3.1). To achieve this, we first add (A.3.3) and (A.3.4), and multiply the sum by 2,

$$\begin{aligned}
&2(\rho U^{k+1} - \rho U^*)^\top (F\Theta^{k+1} - Q^{k+1}) + 2\rho(Q^{k+1} - Q^k)^\top (F\Theta^{k+1} - Q^{k+1}) \\
&+ 2\rho(Q^{k+1} - Q^k)^\top (Q^{k+1} - Q^*) \leq 0.
\end{aligned} \tag{A.3.9}$$

Then inequality (A.3.1) can be obtained from (A.3.9) after some manipulation and rewriting.

Substituting $U^{k+1} = U^k + (F\Theta^{k+1} - Q^{k+1})$ in the first term of inequality (A.3.9)

gives

$$2\rho(U^k - U^*)^\top(F\Theta^{k+1} - Q^{k+1}) + \rho\|F\Theta^{k+1} - Q^{k+1}\|_2^2 + \rho\|F\Theta^{k+1} - Q^{k+1}\|_2^2. \tag{A.3.10}$$

Replacing $F\Theta^{k+1} - Q^{k+1} = (U^{k+1} - U^k)$ in the first two terms of (A.3.10) gives

$$2\rho(U^k - U^*)^\top(U^{k+1} - U^k) + \rho\|U^{k+1} - U^k\|_2^2 + \rho\|F\Theta^{k+1} - Q^{k+1}\|_2^2. \tag{A.3.11}$$

Since $U^{k+1} - U^k = (U^{k+1} - U^*) - (U^k - U^*)$, (A.3.11) can be written as

$$\rho\left(\|U^{k+1} - U^*\|_2^2 - \|U^k - U^*\|_2^2\right) + \rho\|F\Theta^{k+1} - Q^{k+1}\|_2^2. \tag{A.3.12}$$

Substituting (A.3.12) in the first term of (A.3.9) we obtain:

$$\begin{aligned}&\rho\left(\|U^{k+1} - U^*\|_2^2 - \|U^k - U^*\|_2^2\right) + \rho\|F\Theta^{k+1} - Q^{k+1}\|_2^2 \\ &+ 2\rho(Q^{k+1} - Q^k)^\top(F\Theta^{k+1} - Q^{k+1}) + 2\rho(Q^{k+1} - Q^k)^\top(Q^{k+1} - Q^*) \leq 0.\end{aligned} \tag{A.3.13}$$

Replacing $Q^{k+1} - Q^* = (Q^{k+1} - Q^k) + (Q^k - Q^*)$ in the last term of (A.3.13), we obtain

$$\begin{aligned}&\rho\left(\|U^{k+1} - U^*\|_2^2 - \|U^k - U^*\|_2^2\right) \\ &+ \rho\|F\Theta^{k+1} - Q^{k+1} + (Q^{k+1} - Q^k)\|_2^2 + \rho\|Q^{k+1} - Q^k\|_2^2 + 2\rho(Q^{k+1} - Q^k)^\top(Q^k - Q^*) \leq 0.\end{aligned} \tag{A.3.14}$$

Substituting $Q^{k+1} - Q^k = (Q^{k+1} - Q^*) - (Q^k - Q^*)$ in the last two terms of (A.3.14), we obtain

$$\begin{aligned}&\rho\left(\|U^{k+1} - U^*\|_2^2 - \|U^k - U^*\|_2^2\right) + \rho\|F\Theta^{k+1} - Q^{k+1} + (Q^{k+1} - Q^k)\|_2^2 \\ &+ \rho\left(\|Q^{k+1} - Q^*\|_2^2 - \|Q^k - Q^*\|_2^2\right) \leq 0.\end{aligned} \tag{A.3.15}$$

Then we can rewrite (A.3.15) as

$$V^k - V^{k+1} \geq \rho \|F\Theta^{k+1} - Q^{k+1} + (Q^{k+1} - Q^k)\|_2^2. \tag{A.3.16}$$

(A.3.16) is equivalent to

$$V^{k+1} \leq V^k - \rho \|F\Theta^{k+1} - Q^{k+1}\|_2^2 - \rho \|(Q^{k+1} - Q^k)\|_2^2 - 2\rho (F\Theta^{k+1} - Q^{k+1})(Q^{k+1} - Q^k). \tag{A.3.17}$$

Next, we show that $2\rho(F\Theta^{k+1} - Q^{k+1})(Q^{k+1} - Q^k)$ is nonnegative. Note that Q^{k+1} minimizes $\lambda\|Q\|_1 - (\rho U^{k+1})^\top Q$ and Q^k minimizes $\lambda\|Q\|_1 - (\rho U^k)^\top Q$. Thus we have

$$\lambda\|Q^{k+1}\|_1 - (\rho U^{k+1})^\top Q^{k+1} \leq \lambda\|Q^k\|_1 - (\rho U^{k+1})^\top Q^k, \tag{A.3.18}$$

and

$$\lambda\|Q^k\|_1 - (\rho U^k)^\top Q^k \leq \lambda\|Q^{k+1}\|_1 - (\rho U^k)^\top Q^{k+1}, \tag{A.3.19}$$

From (A.3.18) and (A.3.19), we have

$$(\rho U^{k+1})^\top Q^k - (\rho U^{k+1})^\top Q^{k+1} \leq \lambda\|Q^k\|_1 - \lambda\|Q^{k+1}\|_1 \leq (\rho U^k)^\top Q^k - (\rho U^k)^\top Q^{k+1}. \tag{A.3.20}$$

From (A.3.20) we will have

$$\rho(U^{k+1} - U^k)^\top (Q^{k+1} - Q^k) \geq 0.$$

Since $U^{k+1} - U^k = F\Theta^{k+1} - Q^{k+1}$, thus we have proved that $2\rho(F\Theta^{k+1} - Q^{k+1})(Q^{k+1} - Q^k)$ is nonnegative.

Thus, (A.3.17) leads to

$$V^{k+1} \leq V^k - \rho\|F\Theta^{k+1} - Q^{k+1}\|_2^2 - \rho\| - (Q^{k+1} - Q^k)\|_2^2. \tag{A.3.21}$$

We have proved inequality (A.3.1).

Bibliography

Airoldi, E. M., D. M. Blei, S. E. Fienberg, and E. P. Xing (2008). Mixed membership stochastic blockmodels. *Journal of Machine Learning Research 9*, 1981–2014.

Babkin, S., J. R. Stewart, X. Long, and M. Schweinberger (2020). Large-scale estimation of random graph models with local dependence. *Computational Statistics & Data Analysis 152*, 107029.

Betancourt, B., A. Rodríguez, and N. Boyd (2017). Bayesian fused lasso regression for dynamic binary networks. *Journal of Computational and Graphical Statistics 26*(4), 840–850.

Bouranis, L., N. Friel, and F. Maire (2017). Efficient bayesian inference for exponential random graph models by correcting the pseudo-posterior distribution. *Social Networks 50*, 98–108.

Bouranis, L., N. Friel, and F. Maire (2018). Bayesian model selection for exponential random graph models via adjusted pseudolikelihoods. *Journal of Computational and Graphical Statistics 27*(3), 516–528.

Boyd, S., N. Parikh, E. Chu, B. Peleato, and J. Eckstein (2010). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning 3*(1), 1–122.

Breiger, R. L., S. A. Boorman, and P. Arabie (1975). An algorithm for clustering

relational data with applications to social network analysis and comparison with multidimensional scaling. *Journal of Mathematical Psychology 12*(3), 328–383.

Caimo, A. and N. Friel (2011). Bayesian inference for exponential random graph models. *Social Networks 33*(1), 41–55.

Durante, D. and D. B. Dunson (2018). Bayesian inference and testing of group differences in brain networks. *Bayesian Analysis 13*(1), 29–58.

Durante, D., D. B. Dunson, and J. T. Vogelstein (2017). Nonparametric bayes modeling of populations of networks. *Journal of the American Statistical Association 112*(520), 1516–1530.

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics 1*, 209–230.

Frank, O. and D. Strauss (1986). Markov graphs. *Journal of the American Statistical Association 2*, 832–842.

Friedman, J., T. Hastie, H. Höfling, and R. Tibshirani (2007). Pathwise coordinate optimization. *The annals of applied statistics 1*(2), 302–332.

Friedman, J., T. Hastie, and R. Tibshirani (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software 33*(01), 1–22.

Gelman, A. and X.-L. Meng (1998). Simulating normalizing constants: from importance sampling to bridge sampling to path sampling. *Statistical Science 13*(2), 163–185.

Geyer, C. and E. Thompson (1992). Constrained monte carlo maximum likelihood for depedent data. *Journal of the Royal Statistical Society Series B 54*, 657–699.

Hallac, D., Y. Park, S. Boyd, and J. Leskovec (2017). Network inference via the time-varying graphical lasso. In *Proceedings of the 23rd ACM SIGKDD*

International Conference on Knowledge Discovery and Data Mining, pp. 205–213.

Hanneke, S., W. Fu, and E. P. Xing (2010). Discrete temporal models of social networks. *Electronic Journal of Statistics 4*(3), 585–605.

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

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

Hunter, D. R. and M. S. Handcock (2006). Inference in curved exponential family models for networks. *Journal of Computational and Graphical Statistics 15*(3), 565–583.

Hunter, D. R., M. S. Handcock, C. T. Butts, S. M. Goodreau, and M. Morris (2008). ergm: A package to fit, simulate and diagnose exponential-family models for networks. *Journal of Statistical Software 24*(3), 1–29.

Ishwaran, H. and L. F. James (2001). Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association 96*(453), 161–173.

Kalli, M., J. E. Griffin, and S. G. Walker (2011). Slice sampling mixture models. *Statistics and Computing 21*, 93–105.

Koskinen, J. (2008). The linked importance sampler auxiliary variable metropolis hastings algorithm for distributions with intractable normalising constants. *MelNet Social Networks Laboratory Technical Report 08-01. Department of Psychology, School of Behavioural Science, University of Melbourne, Australia*.

Krackhardt, D. (1987). Cognitive social structures. *Social Networks 9*(2), 109–134.

Krivitsky, P. N. and M. S. Handcock (2014). A separable model for dynamic networks. *Journal of the Royal Statistical Society Series B 76*(1), 29–46.

Lee, C. and D. J. Wilkinson (2019). A review of stochastic block models and extensions for graph clustering. *Applied Network Science 4*(1), 1–50.

Lee, J., G. Li, and J. D. Wilson (2020). Varying-coefficient models for dynamic networks. *Computational Statistics & Data Analysis 152*, 107052.

Lee, K. H., L. Xue, and D. R. Hunter (2020). Model-based clustering of time-evolving networks through temporal exponential-family random graph models. *Journal of Multivariate Analysis 175*, 104540.

Lehmann, B. and S. White (2021). Bayesian exponential random graph models for populations of networks. *arXiv:2104.05110*.

Li, T., E. Levina, and J. Zhu (2020). Network cross-validation by edge sampling. *Biometrika 107*(2), 257–276.

Liang, F. and I.-H. Jin (2013). A Monte Carlo Metropolis-Hastings algorithm for sampling from distributions with intractable normalizing constants. *Neural Computation 25*(8), 2199–2234.

Lubbers, M. J. and T. A. Snijders (2007). A comparison of various approaches to the exponential random graph model: A reanalysis of 102 student networks in school classes. *Social Networks 29*(4), 489–507.

MacDonald, P. W., E. Levina, and J. Zhu (2022). Latent space models for multiplex networks with shared structure. *Biometrika*, 1–24.

Møller, J., A. N. Pettitt, R. Reeves, and K. K. Berthelsen (2006). An efficient markov chain monte carlo method for distributions with intractable normalising constants. *Biometrika 93*(2), 451–458.

Morris, M., M. S. Handcock, and D. R. Hunter (2008). Specification of exponential-family random graph models: terms and computational aspects. *Journal of statistical software 24*(4), 1548.

Murray, I., Z. Ghahramani, and D. J. C. MacKay (2006). Mcmc for doubly-intractable distributions. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, pp. 359–366.

Neal, R. M. (2005). Estimating ratios of normalizing constants using linked importance sampling. *Technical Report No. 0511, Department of Statistics, University of Toronto*.

Nicola, G. D., B. Sischka, and G. Kauermann (2022). Mixture models and networks: The stochastic blockmodel. *Statistical Modelling 22*(1-2), 67–94.

Paul, S. and Y. Chen (2020). A random effects stochastic block model for joint community detection in multiple networks with applications to neuroimaging. *Annals of Applied Statistics 14*(2), 993–1029.

Propp, J. G. and D. B. Wilson (1996). Exact sampling with coupled markov chains and applications to statistical mechanics. *Random Structures & Algorithms 9*(1-2), 223–252.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association 66*(336), 846–850.

Ren, S. (2019). Dynamic network analysis via exponential random graph models. *Master Thesis submited for Northwest University*.

Robbins, H. and S. Monro (1951, 09). A stochastic approximation method. *Annals of Mathematical Statistics 22*(3), 400–407.

Schweinberger, M. and M. S. Handcock (2015). Local dependence in random graph models: characterization, properties and statistical inference. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 77*(3), 647–676.

Schweinberger, M., P. N. Krivitsky, C. T. Butts, and J. R. Stewart (2020). Exponential-family models of random graphs: Inference in finite, super and infinite population scenarios. *Statistical Science 35*(4), 627–662.

Schweinberger, M. and J. Stewart (2020). Concentration and consistency results for canonical and curved exponential-family models of random graphs. *The Annals of Statistics 48*(1), 374–396.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica 4*(2), 639–650.

Sewell, D. K. and Y. Chen (2015). Latent space models for dynamic networks. *Journal of the American Statistical Association 110*(512), 1646–1657.

Signorelli, M. and E. C. Wit (2020). Model-based clustering for populations of networks. *Statistical Modelling 20*(1), 9–29.

Simpson, S. L., R. G. Lyday, S. Hayasaka, A. P. Marsh, and P. J. Laurienti (2013). A permutation testing framework to compare groups of brain networks. *Frontiers in computational neuroscience 7*(171), 1–13.

Slaughter, A. J. and L. M. Koehly (2016). Multilevel models for social networks: Hierarchical bayesian approaches to exponential random graph modeling. *Social Networks 44*, 334–345.

Snijders, T. A., G. G. V. de Bunt, and C. E. Steglich (2010). Introduction to stochastic actor-based models for network dynamics. *Social Networks 32*(1), 44–60.

Snijders, T. A. B. (2002). Markov chain monte carlo estimation of exponential random graph models. *Journal of Social Structure 3*(2), 1–40.

Strauss, D. and M. Ikeda (1990). Pseudolikelihood estimation for social networks. *Journal of the American Statistical Association 85*(409), 204–212.

Sweet, T. M., A. Flynt, and D. Choi (2019). Clustering ensembles of social networks. *Network Science 7*(2), 141–159.

Tan, L. S. and N. Friel (2020). Bayesian variational inference for exponential random graph models. *Journal of Computational and Graphical Statistics 29*(4), 910–928.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological) 58*(1), 267–288.

Tibshirani, R., M. Sauders, S. Rosset, J. Zhu, and K. Knight (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society Series B 67*(1), 91–108.

Tibshirani, R. J. and J. Taylor (2011). The solution path of the generalized lasso. *The Annals of Statistics 39*(3), 1335 – 1371.

van Duijn, M. A., K. J. Gile, and M. S. Handcock (2009). A framework for the comparison of maximum pseudo-likelihood and maximum likelihood estimation of exponential family random graph models. *Social Networks 31*(1), 52–62.

Walker, S. G. (2007). Sampling the Dirichlet mixture model with slices. *Communications in Statistics. Simulation and Computation 36*, 45–54.

Wang, P., G. Robins, P. Pattison, and E. Lazega (2013). Exponential random graph models for multilevel networks. *Social Networks 35*(1), 96–115.

Yin, F. and C. T. Butts (2020). Kernel-based approximate bayesian inference for exponential family random graph models. *arXiv:2004.08064*.

Yin, F., W. Shen, and C. T. Butts (2022). Finite mixtures of ergms for modeling ensembles of networks. *Bayesian Analysis*, 1–39.