# Coevolution Regression and Composite Likelihood Estimation for Social Networks

Yanjun He

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2018

Reading Committee:

Peter D. Hoff, Chair

Tyler H. McCormick

Hariharan Narayanan

Program Authorized to Offer Degree:
Department of Statistics

University of Washington

**Abstract**

Coevolution Regression and Composite Likelihood Estimation for Social Networks

Yanjun He

Chair of the Supervisory Committee:
Professor Peter D. Hoff
Department of Statistical Science, Duke University
Department of Statistics, University of Washington

We study how social networks and nodal attributes influence each other over time. A multiplicative coevolution regression (MCR) model is proposed for longitudinal network and nodal attribute data. The coevolution model is based on the following three principles: autocorrelation, homophily and contagion. For the Gaussian MCR model, the maximum likelihood estimates can be obtained using ordinary least squares. We also extend the Gaussian MCR so that it can include latent factors or model ordinal data. A Bayesian method using Markov Chain Monte Carlo (MCMC) is used to estimate the parameters and latent factors.

We then focus on developing a scalable method to estimate the parameters in models of very large binary network datasets. Maximum likelihood estimates are generally impossible to obtain because the full likelihood involves an intractable high dimensional integral. Also, full-likelihood Bayesian estimation is impractical for very large datasets as the MCMC algorithm is very slow. We propose a triadic composite likelihood estimation method for exchangeable latent Gaussian network models, and extend it to $q$-node composite likelihood estimation for other exchangeable and non-exchangeable models. The maximum composite likelihood estimates are obtained by optimizing the composite likelihood using a stochastic gradient-based algorithm, where the gradients are approximated using Monte Carlo samples. For networks of moderate size, we show via simulations that composite likelihood estimation

provides estimates as accurate as those provided by fully Bayesian estimation using MCMC. For very large datasets, fully Bayesian estimation is impractical, but composite likelihood estimation is feasible as its computational cost is essentially constant as a function of the network size.

**Keyword**: longitudinal social networks, multiplicative coevolution regression model, composite likelihood estimation, fully Bayesian estimation, gradient-based method

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ACKNOWLEDGMENTS

The author wishes to express her sincere appreciation to those who guided and funded her as she developed her research skills. In particular, Daquan Jiang, Hongman Wang, Adrain E. Raftery, Samuel J. Clark, Tyler H. McCormick, Hariharan Narayanan and her advisor Peter D. Hoff.

# DEDICATION

This dissertation is dedicated to my dear parents, Changhua He and Minfang Zhao.

# Chapter 1
# INTRODUCTION

## *1.1 Overview*

Network data on a set of $m$ nodes can be represented as an $m \times m$ matrix (sociomatrix) $\mathbf{Y}$ with $y_{ij}$ referring to the relation from node $i$ to node $j$, $i \neq j$. This framework can be applied to many areas, including the friendship patterns on Facebook, conflicts between countries and protein-protein interactions. Often a network evolves over time, in which case the dataset is no longer an $m \times m$ matrix, but an $m \times m \times n$ array, with the third dimension representing time points and each time slice is a sociomatrix for the same set of nodes. Besides the networks, individuals' characteristics may evolve over time as well. Therefore, network time series are often accompanied by some time series of nodal attributes. Longitudinal studies of network relations are often interested in two particular network phenomena: homophily and contagion. Homophily describes the fact that how similar two individuals are may affect their connection with each other. Contagion describes the impact on the nodal attributes from those the node connects with.

We propose a multiplicative coevolution regression model for network and nodal attribute data in Chapter 2 to model the these phenomena together. The coevolution model is based on the following three principles. First, network relations and nodal attributes may evolve smoothly over time (autocorrelation). Secondly, people may change relations based on how similar their characteristics are (homophily). Finally, people may change their characteristics based on the characteristics of those they have relations with (contagion). The coevolution model is designed to separately estimate these different effects of how networks and individuals' characteristics coevolve overtime and how they influence each other. It can be applied to various types of network and nodal attributes data, including continuous and ordinal net-

work data or nodal characteristics. For continuous data, we can apply a Gaussian model and ordinary least square (OLS) estimates can be easily obtained. For ordinal networks, including binary ones, we use a probit link and obtain Bayes estimates of model parameters and random effects using Markov Chain Monte Carlo (MCMC).

Recent network studies gather data on very large populations, resulting in networks with thousands or millions of nodes. For example, some social media networks have millions or billions of nodes. How to estimate the parameters in a network model for such datasets is very challenging. Bayes estimates obtained via MCMC are generally impractical because the algorithm require operations on matrices of dimension related to the number of nodes. For probit network models, maximum likelihood estimates (MLEs) are in general infeasible to obtain for large networks, because the full likelihood is proportional to an intractable integral. One approximate estimation method is variational inference. Airoldi et al. [2008] used variational inference with a mean field approximation to estimate the parameters in mixed membership stochastic blockmodels for large networks including a physical interaction network among 871 proteins in yeast. The estimation relies on the choice of the variational distribution and a reasonable one may not be practical or easily obtained.

In this paper, we propose a scalable method to estimate the parameters in random effects and latent factor models for binary network data. Our approach is to optimize a composite likelihood, which is the sum of marginal likelihoods of subgraphs. For small subgraphs, we can easily approximate the gradients of the logarithm of their marginal distributions using MCMC samples, and then use them in a gradient-based optimization algorithm. This method can be applied to various network models for binary networks, including exchangeable models such as the social relations model (SRM) and the additive and multiplicative effect (AME) model, as well as nonexchangeable models such as models with nodal or dyadic covariates.

Chapter 3 focuses on latent Gaussian exchangeable models for binary networks. One example is the social relations model (SRM), for which we can take advantage of a summation trick in triadic composite likelihood estimation, which reduces the parameter estimation problem to small pieces. For each triadic component, the marginal distribution is an integral

over a vector of size 6, and it is feasible to get an accurate approximation of its gradient with respect to the parameters. A scalable gradient based algorithm using MCMC approximated gradients is thus proposed to optimize the triadic composite likelihood. We will show by simulation that triadic composite likelihood estimation can provide estimates that are as accurate as the fully Bayesian estimates obtained using MCMC.

In Chapter 4, we extend the method to a group of more general exchangeable models for binary networks, in particular the AME model, for which a multiplicative term of latent factors is introduced. As we introduce more parameters and the latent factors, triadic composite likelihood estimation does not include enough information for accurate parameter estimation. We thus introduce a $q$-node composite likelihood, which allows specification of different choices of $q$. The challenge is that for $q > 3$, it is impractical to enumerate all isomorphic subgraphs and thus the composite likelihood can not be written as a sum of a small number of marginal likelihood of subgraphs. In chapter 5, we extend the approach to nonexchangeable models, such as those with nodal or dyadic covariates. In this case, no subgraphs are isomorphic and therefore, composite likelihood can not be reduced to a small number of marginal likelihood of subgraphs as well, no matter what value $q$ takes.

For these two models, we use a stochastic gradient based algorithm to optimize the $q$-node composite likelihood and obtain the maximum composite likelihood estimates (MCLEs). Instead of calculating the gradient for each isomorphic subgraphs, we sample one subgraph at each iteration of the algorithm and generate MCMC samples based on the subgraph to approximate the gradient. In these two chapters, we carry out simulation studies to discuss how to choose the value of $q$, and compare the performance of the $q$-node composite likelihood estimation and the fully Bayesian estimation using MCMC. For moderate sized networks, both methods provide accurate estimates. We show that as the network size increases, the $q$-node composite likelihood estimation can still provide reasonable estimates with a constant time consumption, while the fully Bayesian estimation fails due to exploding computational cost.

## *1.2 Notation*

Unless mentioned specifically, thoughout the thesis we use lowercase letters to represent scalars, for example, $y_{ij}$ for the connection between node $i$ and $j$, and $\mu$ for a univariate mean. Lowercase letters in bold fonts are used to represent vectors, for example, $\mathbf{y}$ for the vectorization of the sociomatrix and $\boldsymbol{\theta}$ for the parameter vector. We use uppercase letters in bold fonts to represent matrices, for example $\mathbf{Y}$ for a sociomatrix and $\boldsymbol{\Sigma}$ for a covariance matrix.

# Chapter 2

# MULTIPLICATIVE COEVOLUTION REGRESSION MODELS FOR LONGITUDINAL NETWORKS AND NODAL ATTRIBUTES

## *2.1 Introduction*

Modern studies of social networks often involve longitudinal measurements over time. Such data can be represented as a sequence of sociomatrices $\mathbf{Y}_0, \ldots, \mathbf{Y}_n$, where each $\mathbf{Y}_t$ is a square $m \times m$ matrix with entry $y_{ij,t}$ representing the value of a relationship between nodes $i$ and $j$ at time $t$ (the diagonal entries are typically undefined). Several methods for the analysis of such data have been developed: Important early work in this area has involved stochastic actor-oriented models [Snijders, 2005, Snijders et al., 2010]. This approach is based on an economic model of rational choice, whereby individuals make unilateral changes to their networks in order to maximize personal utility functions. Other methods for dynamic network analysis have evolved out of earlier methods for static network data. For example, methods based on temporal exponential random graph models (TERGM) have been developed based on the popular static exponential random graph modeling framework (ERGM) [Hunter et al., 2008, Krivitsky and Handcock, 2014]. An alternative approach to static network modeling is one where network patterns are represented with node-specific latent variables [Nowicki and Snijders, 2001, Hoff et al., 2002]. Dynamic versions of these models have been developed in Sarkar and Moore [2005], Xing et al. [2010], Ward et al. [2013], Durante and Dunson [2014], Sewell and Chen [2015], among others.

Longitudinal network data will often be accompanied by longitudinal node-level attributes $\mathbf{X}_0, \ldots, \mathbf{X}_n$, where each $\mathbf{X}_t$ is an $m \times p$ matrix whose $i$th row is a vector $\mathbf{x}_{i,t}$ of characteristics of node $i$ at time $t$. In such cases, it is often of interest to infer how the network and

nodal attributes might influence each other over time. To this end, statistical methodology and software have been developed that extends the actor-oriented approach described above (Snijders et al. [2007], http://www.stats.ox.ac.uk/~snijders/siena/). While this work has been groundbreaking, the applicability of an actor-oriented model may be limited to certain types of networks and individual-level characteristics. As described by the primary developers of this approach [Snijders et al., 2007], such a model may not be appropriate in situations where network and behavioral data depend on unobserved latent variables. Such a situation may be present in the study of social networks and obesity: An individual's body mass index may be related to their social network, but this relationship is likely mediated by other variables such as socioeconomic status, diet, exercise, participation in sports and other variables that may potentially be unobserved. Furthermore, parameter estimation for such actor oriented models is computationally intensive, involving an iterative optimization scheme that requires simulation of hypothetical networks at each iteration.

As an alternative to this actor-oriented approach, in this article we develop a class of coevolution models for network and nodal attribute data that are based on simple and scalable linear regression and latent factor models. Like regression modeling, the framework we present is flexible and extendable, and can be modified to accommodate continuous and ordinal measurements for both the nodal and network data. The framework is built upon a simple autoregressive model that describes the association of both the network $\mathbf{Y}_t$ and the nodal attributes $\mathbf{X}_t$ at time $t$ with the values $\{\mathbf{Y}_{t-1}, \mathbf{X}_{t-1}\}$ from the previous time point. The associations are modeled in terms of products of the network and nodal outcomes, and so we refer to such models as multiplicative coevolution regression (MCR) models.

As we discuss in the next section, the parameters of MCR models can quantify three important data features: First, that both the network and nodal attributes may vary smoothly from time point to time point; second, the relations between individuals may be influenced by the similarity of their attributes; and third, individuals may change their attributes based upon the attributes of those with whom they relate. We refer to these three features as autocorrelation, homophily, and contagion, respectively. While the basic MCR model may

simply be represented as a type of regression model, in Section 2.2 we discuss extensions of this model to accommodate network and nodal data that may be binary or ordinal, as well as extensions for data where certain types of network patterns may be well-represented with latent nodal factors. In Section 2.3 we discuss estimation and inference, including maximum likelihood estimates for fully observed continuous data, and Bayesian inference for a variety of model extensions. In Section 2.4 we present two case studies. The first involves monthly interactions between 50 countries over a 10 year period. The second analyzes the coevolution of friendship ties and an ordinal measure of delinquency for 26 high-school students. A discussion follows in Section 2.5.

### *2.2 Multiplicative Coevolution Regression*

A coevolution model for dynamic network and nodal attribute data should be able to quantify *autocorrelation*, *homophily* and *contagion*. Autocorrelation quantifies the tendency for relations and attributes to vary gradually over time. Homophily refers to the possibility that changes to the relations between nodes may be partly determined by how similar their attributes are. Contagion describes how nodes may change their attributes based on the attributes of those with whom they have relations. For the case of undirected relational data, we propose the following simple multiplicative regression model for describing these three phenomena:

$$\begin{aligned} y_{ij,t+1} &= \mu_{ij} + \alpha y_{ij,t} + \mathbf{x}_{i,t}^T \mathbf{H} \mathbf{x}_{j,t} + \epsilon_{ij,t+1}, \\ \mathbf{x}_{i,t+1} &= \boldsymbol{\theta}_i + \mathbf{A}\mathbf{x}_{i,t} + \mathbf{C}\mathbf{X}_t^T \mathbf{y}_{i\cdot,t} + \mathbf{e}_{i,t+1}, \end{aligned} \tag{2.1}$$

where $\mathbf{y}_{i\cdot,t}$ is the $i$th row $\mathbf{Y}_t$ (with $y_{ii,t} = 0$), the $\epsilon_{ij,t}$'s are i.i.d. $N(0, \sigma^2)$ and the $\mathbf{e}_{i,t}$'s are i.i.d. $N(\mathbf{0}, \boldsymbol{\Sigma})$. Alternatively, the intercept terms $\mu_{ij}$ and $\boldsymbol{\theta}_i$ can be replaced with regression terms involving exogenous predictors and possibly depending on time.

The parameters $\{\alpha, \mathbf{A}\}$, $\mathbf{H}$ and $\mathbf{C}$ respectively represent the phenomena of autocorrelation, homophily and contagion described above. To see this, note that if $\mathbf{H}$ and $\mathbf{C}$ were zero, then the model reduces to two first order autoregressive models, with $\alpha$ and $\mathbf{A}$ being

the autoregression parameters. Regarding homophily, the matrix $\mathbf{H} \in \mathbb{R}^{p \times p}$ represents the influence of the similarity between the characteristics of two nodes on their relations. As a simple example, consider the case where $\mathbf{H} = h\mathbf{I}$ with $h > 0$, and so $\mathbf{x}_{i,t}^T \mathbf{H} \mathbf{x}_{j,t} = h\mathbf{x}_{i,t}^T \mathbf{x}_{j,t}$. In this case we have positive homophily, in that the more similar $i$ and $j$ are in terms of their attributes at time $t$, the larger the expected relation between them at the next time point. Finally, the matrix $\mathbf{C}$ describes contagion, the effect of nodal attributes at time $t$ on those of a given node $i$ at time $t+1$, weighted by the relations of node $i$. For example, assume for the moment that $y_{ij,t} \in \{0, 1\}$. In this case, $\mathbf{X}_t^T \mathbf{y}_{i\cdot,t}$ is proportional to the average of the characteristic values of those to whom node $i$ is linked.

Model (2.1) describes the simplest situation we consider in this article, in which the network and nodal attributes are assumed to be Gaussian and fully observed. We refer to this model as a multiplicative coevolution regression (MCR) model. The model is multiplicative in $\mathbf{Y}_t$ and $\mathbf{X}_t$ via the homophily and contagion effects. However, as will be discussed in Section 2.3, it is linear in the parameters and so can be viewed as a multivariate linear regression model.

The assumption of additive effects and normally distributed outcomes is not appropriate for many network datasets. In particular, many network relations are binary or ordinal, and are possibly asymmetric in that $y_{ij,t}$ is not necessarily equal to $y_{ji,t}$. Furthermore, it is often likely to be the case that some variables that drive network formation are unobserved, and not part of the the dataset. In this case, we may want to augment the model to accommodate latent, unobserved nodal characteristics. We consider extensions of the model in (2.1) to accommodate each of these situations in the following paragraphs.

**Ordinal data:** The relational variable $y_{ij,t}$ in many network datasets is binary, indicating whether or not two nodes have some sort of tie between them, such as friendship or social interaction. In other cases this variable is ordinal, such as when $y_{ij,t}$ is recorded as being negative, neutral or positive, or when $y_{ij,t}$ measures the number or intensity of social interactions between two individuals. While the assumptions of Gaussian noise and additive effects

of the MCR model will not generally be appropriate for such data, the model can be used to formulate a probit regression model for general ordinal network relations. This is done by expressing the relations $y_{ij,t}$ as a non-decreasing function of latent relations $z_{ij,t}$ that that do follow the MCR model. Specifically, we assume that $y_{ij,t} = f(z_{ij,t})$ for some non-decreasing function $f$, and that the process $\{(\mathbf{Z}_t, \mathbf{X}_t) : t = 0, \ldots, n\}$ follows the Gaussian MCR model. The only adjustment to the model is that the error variance $\sigma^2$ may be assumed to be 1, as otherwise this scale parameter is not separately identifiable from $f$. Furthermore, if the nodal characteristic process $\{\mathbf{X}_t, t = 1, \ldots, n\}$ is not well represented with a normal model then an ordinal probit model can be used here as well. In this case, we model $x_{i,k,t} = g_k(w_{i,k,t})$ where $g_1, \ldots, g_p$ are nondecreasing functions and $w_{i,k,t}$ is a latent Gaussian process that determines $x_{i,k,t}$. Letting $\mathbf{W}_t$ be the $n \times p$ matrix with elements $w_{i,k,t}$, the model is completed by assuming $\{(\mathbf{Z}_t, \mathbf{W}_t) : t = 0, \ldots, n\}$ follows the MCR model. An example data analysis in which the both the relational and attribute variables are ordinal is presented in Section 2.4.

**Directed relations:** Many network datasets include directed relations where $y_{ij,t}$ is not necessarily equal to $y_{ji,t}$. The natural extension of the multiplicative coevolution model in equation (2.1) to accommodate directed relations is as follows:

$$\begin{aligned} y_{ij,t+1} &= \mu_{ij} + \alpha_1 y_{ij,t} + \alpha_2 y_{ji,t} + \mathbf{x}_{i,t}^T \mathbf{H} \mathbf{x}_{j,t} + \epsilon_{ij,t+1}, \\ \mathbf{x}_{i,t+1} &= \boldsymbol{\theta}_i + \mathbf{A}\mathbf{x}_{i,t} + \mathbf{C}_1 \mathbf{X}_t^T \mathbf{y}_{i\cdot,t} + \mathbf{C}_2 \mathbf{X}_t^T \mathbf{y}_{\cdot i,t} + \mathbf{e}_{i,t+1}. \end{aligned} \tag{2.2}$$

The modifications to the model for the network process are that the homophily parameter $\mathbf{H}$ is not necessarily symmetric, and that $y_{ij,t+1}$ may be influenced by $y_{ji,t}$ via the reciprocity parameter $\alpha_2$. The model for the attribute process now includes two different contagion parameters $\mathbf{C}_1$ and $\mathbf{C}_2$. The former represents the relationship-weighted effect of the nodal characteristics of those to which one sends ties, while the latter represents the effect of those from which one receives ties. An example data analysis using a probit version of this directed MCR model appears in Section 2.4.

**Latent nodal attributes:** When nodal attribute data are either not available or only weakly associated with the network process, it may be useful to add latent nodal attributes to the model. In the case of static network modeling, inclusion of latent nodal attributes can provide identification of clusters of nodes, improved model fit and better out-of-sample predictions of unmeasured relations. The basic framework is to model the relation $y_{ij}$ between nodes $i$ and $j$ as depending on the similarity of latent, unobserved characteristics $\mathbf{x}_i$ and $\mathbf{x}_j$. For example, the latent class model of Nowicki and Snijders [2001] is equivalent to letting $\mathbf{x}_i$ represent a vector indicating membership of node $i$ to one of several latent classes. The latent distance model of Hoff et al. [2002] assumes $y_{ij}$ depends on the Euclidean distance between the latent location vectors $\mathbf{x}_i$ and $\mathbf{x}_j$. Hoff [2008] shows how both of these approaches are generalized by a multiplicative approach, in which $y_{ij}$ is modeled as a function of the inner product $\mathbf{x}_i^T\mathbf{H}\mathbf{x}_j$. This suggests that, in the absence of nodal characteristics strongly associated with the network process, we allow $\mathbf{x}_{i,t}$ in the MCR model (2.1) to represent latent, unobserved nodal attributes. In this case, both the parameters of the MCR model in (2.1) and the latent attribute process $\{\mathbf{X}_t : t = 0, \ldots, n\}$ can be estimated from the data. However, the parameters in the MCR model are not fully identifiable when the nodal attributes are latent. For example, the model is invariant to orthogonal rotations of the $\mathbf{X}_t$'s, that is, replacement of each $\mathbf{X}_t$ by $\mathbf{X}_t\mathbf{R}$, where $\mathbf{R}$ is a $p \times p$ orthogonal matrix so that $\mathbf{R}\mathbf{R}^T = \mathbf{I}$. For this reason we simplify the latent MCR model by parameterizing the homophily parameter $\mathbf{H}$ as being diagonal, and setting $\mathbf{\Sigma}$ equal to the $p \times p$ identity matrix. Even so, the model remains invariant to simultaneous permutations of the columns of the $\mathbf{X}_t$'s. This issue is discussed further in the data analysis example in Section 2.4.

This latent MCR model is similar to several other models developed for the analysis of longitudinal network data that lack nodal attributes. For example, Ward et al. [2013], Durante and Dunson [2014] and Sewell and Chen [2015] each utilize models where the network $\mathbf{Y}_t$ at each time point is modeled as a function of nodal latent variables $\mathbf{X}_t$, which in turn follows a stochastic process. These are hidden Markov models for the observed network process, and can be graphically depicted by the dependence graph in the first panel of

Figure 2.1. Such models essentially only include a homophily parameter, modeling a relation between two nodes as a function of their time-varying latent attributes. In contrast, our latent MCR model (depicted in the second panel of the figure) permits a richer description of the evolution of the network by inclusion of an autocorrelation term for the network, and a contagion parameter that allows for the possibility that nodes may change their nodal attributes depending on their past relations.

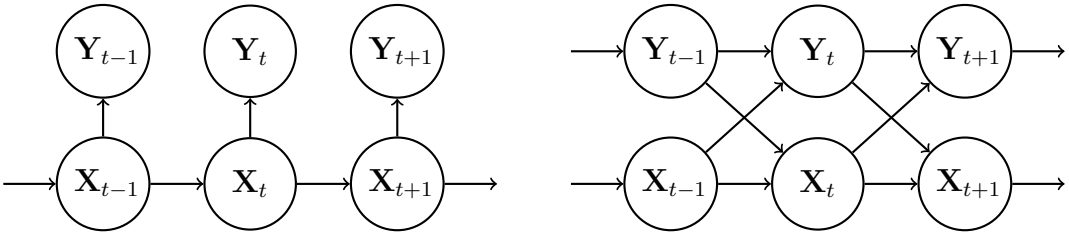


Figure 2.1: Dependence graphs for longitudinal network models. Hidden Markov model (left) and latent MCR model (right).

## *2.3 Estimation and Inference*

One feature of the MCR model is its simplicity: It can be expressed as a pair of linear regression models. As we show in the next subsection, this permits very easy parameter estimation in the case of a normal model for the observed network and attributes. The linear regression framework also serves as a building block for data analysis in more complicated situations, such as the case of ordinal relational and attribute variables and latent attribute models. As we show in Section 2.3.2, Bayesian inference in such situations can be obtained using relatively straightforward Gibbs sampling algorithms.

### *2.3.1 MLEs for normal models*

To see how the network evolution model in Equation 2.1 can be expressed as a linear regression model, first parameterize $\mu_{ij}$ as $\mu_{ij} = \boldsymbol{\gamma}^T \mathbf{s}_{ij}$, where $\mathbf{s}_{ij}$ is a vector of observed exogenous

covariates and $\boldsymbol{\gamma}$ is a vector of unknown parameters. If there are no exogenous covariates then we can take $\boldsymbol{\gamma}$ to simply be a vector consisting of the values of $\mu_{ij}$ and $\mathbf{s}_{ij}$ to be the appropriate binary vector with a single entry equal to one and the remaining entries equal to zero. Then, note that the term $\mathbf{x}_{i,t}^T \mathbf{H} \mathbf{x}_{j,t}$ can be written as $\mathbf{h}^T \mathbf{x}_{ij,t}$, where $\mathbf{h} = \text{vech}(\mathbf{H})$ is the "half vectorization" of the matrix $\mathbf{H}$ obtained by concatenating the lower-triangular elements of $\mathbf{H}$ (including the diagonal), and $\mathbf{x}_{ij,t} = \text{vech}(\mathbf{x}_{i,t}\mathbf{x}_{j,t}^T + \mathbf{x}_{j,t}\mathbf{x}_{i,t}^T - \text{diag}(\mathbf{x}_{i,t}\mathbf{x}_{j,t}^T))$. For example, if each $\mathbf{x}_{i,t}$ is two-dimensional, then $\mathbf{x}_{ij,t} = (x_{i,1,t}x_{j,1,t},\, x_{i,1,t}x_{j,2,t} + x_{i,2,t}x_{j,1,t},\, x_{i,2,t}x_{j,2,t})$. We can therefore write the network component of model (2.1) as

$$y_{ij,t} = \boldsymbol{\beta}^T \mathbf{w}_{ij,t} + \epsilon_{ij,t},$$

where $\boldsymbol{\beta} = (\boldsymbol{\gamma}, \alpha, \mathbf{h})$ and $\mathbf{w}_{ij,t} = (\mathbf{s}_{ij}, y_{ij,t-1}, \mathbf{x}_{ij,t})$. The residual sum of squares can be expressed as

$$\sum_t \sum_{i<j} (y_{ij,t} - \boldsymbol{\beta}^T \mathbf{w}_{ij,t})^2 = \left( \sum_t \sum_{i<j} y_{ij,t}^2 \right) - 2\boldsymbol{\beta}^T \mathbf{l} + \boldsymbol{\beta}^T \mathbf{Q} \boldsymbol{\beta},$$

where

$$\begin{aligned} \mathbf{l} &= \sum_{t=1}^{n} \sum_{i<j} \mathbf{w}_{ij,t}^T y_{ij,t} \\ \mathbf{Q} &= \sum_{t=1}^{n} \sum_{i<j} \mathbf{w}_{ij,t} \mathbf{w}_{ij,t}^T. \end{aligned} \tag{2.3}$$

The maximum likelihood estimate of $\boldsymbol{\beta}$ is therefore given by $\hat{\boldsymbol{\beta}} = \mathbf{Q}^{-1}\mathbf{l}$.

The attribute evolution model is also a linear regression model. Parameterizing $\boldsymbol{\theta}_i$ as $\boldsymbol{\Gamma}\mathbf{s}_i$ for an exogenous covariate vector $\mathbf{s}_i$ and parameter matrix $\boldsymbol{\Gamma}$, we have $\mathbf{x}_{i,t+1} = \mathbf{B}\mathbf{w}_{i,t+1} + \mathbf{e}_{i,t+1}$, where $\mathbf{B}$ is the column-wise concatenation of $\boldsymbol{\Gamma}$, $\mathbf{A}$ and $\mathbf{C}$, and $\mathbf{w}_{i,t+1}$ is the vector obtained by concatenating the vectors $\mathbf{s}_i$, $\mathbf{x}_{i,t}$ and $\mathbf{X}_t^T \mathbf{y}_{i\cdot,t}$. The attribute evolution model can be written in matrix form as

$$\begin{aligned} \mathbf{X}_{t+1} &= \mathbf{S}\boldsymbol{\Gamma}^T + \mathbf{X}_t \mathbf{A}^T + \mathbf{Y}_t \mathbf{X}_t \mathbf{C}^T + \mathbf{E}_{t+1} \\ &= \mathbf{W}_{t+1} \mathbf{B}^T + \mathbf{E}_{t+1}, \end{aligned}$$

where the $i$th row of $\mathbf{W}_{t+1}$ is the vector $\mathbf{w}_{i,t+1}$ defined above. A standard result from multivariate regression is that the MLE of $\mathbf{B}$ is given by $\hat{\mathbf{B}} = \mathbf{L}\mathbf{Q}^{-1}$, where

$$\mathbf{L} = \sum_{t=1}^{n}\sum_{i=1}^{m} \mathbf{x}_{i,t}\mathbf{w}_{i,t}^T = \sum_{t=1}^{n} \mathbf{X}_t^T\mathbf{W}_t \tag{2.4}$$
$$\mathbf{Q} = \sum_{t=1}^{n}\sum_{i=1}^{m} \mathbf{w}_{i,t}\mathbf{w}_{i,t}^T = \sum_{t=1}^{n} \mathbf{W}_t^T\mathbf{W}_t.$$

Estimation for directed relations proceeds with a few modifications. For estimation of the network process, $\boldsymbol{\beta} = (\boldsymbol{\gamma}, \alpha_1, \alpha_2, \mathbf{h})$ where $\mathbf{h} = \text{vec}(\mathbf{H})$, and $\mathbf{w}_{ij,t} = (\mathbf{s}_{ij}, y_{ij,t-1}, y_{ji,t-1}, \mathbf{x}_{j,t} \otimes \mathbf{x}_{i,t})$, where "$\otimes$" is the Kronecker product. Additionally, the summation in (2.3) is replaced by a summation over all ordered pairs $\{(i,j) : i \neq j\}$. For estimation of the attribute process, the matrix $\mathbf{B}$ is the concatenation of $\boldsymbol{\Gamma}$, $\mathbf{A}$, $\mathbf{C}_1$ and $\mathbf{C}_2$, and $\mathbf{w}_{i,t+1}$ is the concatenation of the vectors $\mathbf{s}_i$, $\mathbf{x}_{i,t}$, $\mathbf{X}_t^T\mathbf{y}_{i\cdot,t}$ and $\mathbf{X}_t^T\mathbf{y}_{\cdot i,t}$.

### *2.3.2 Bayesian estimation for model extensions*

In cases where nodal attributes are not observed or the network and attribute processes are not plausibly Gaussian, the MCR model will have to be extended as described in Section 2.2. For these cases we propose a Bayesian approach to inference, as a posterior approximation scheme based on Gibbs sampling is modular and can be easily modified to accommodate different features of the data. We first discuss Bayesian inference for the basic MCR model described in Equation 2.1, and then discuss two modifications, permitting the modeling of unobserved latent attributes and the modeling of ordinal network and attribute data.

Let $\boldsymbol{\beta} = (\gamma, \alpha, \mathbf{h})$ and $\mathbf{B} = [\boldsymbol{\Gamma}\, \mathbf{A}\, \mathbf{C}]$ be the regression parameters in the network and attribute processes respectively. Using semiconjugate prior distributions for the unknown parameters $\boldsymbol{\beta}$, $\mathbf{B}$, $\sigma^2$ and $\boldsymbol{\Sigma}$, their joint posterior distribution can be approximated with a Gibbs sampler that iteratively simulates values of these parameters from their full conditional distributions. Specifically, if the prior distributions are $\boldsymbol{\beta} \sim N(\mathbf{0}, \mathbf{V}_\beta)$, $\mathbf{b} = \text{vec}(\mathbf{B}) \sim N(\mathbf{0}, \mathbf{V}_\text{b})$, $1/\sigma^2 \sim \text{gamma}(\nu_0/2, \nu_0\sigma_0^2/2)$, and $\boldsymbol{\Sigma}^{-1} \sim$Wishart$(\mathbf{S}_0^{-1}, \eta_0)$, then the Gibbs sampler proceeds by iterating the following steps:

1. Simulate $\boldsymbol{\beta}$ from its multivariate normal full conditional distribution with mean $(\mathbf{V}_{\beta}^{-1} + \mathbf{Q})^{-1}\mathbf{l}$ and variance $(\mathbf{V}_{\beta}^{-1} + \mathbf{Q})^{-1}$, where $\mathbf{Q}$ and $\mathbf{l}$ are as in (2.3).

2. Simulate $\mathbf{b}$ from its multivariate normal full conditional distribution with mean $(\mathbf{V}_{b}^{-1} + \mathbf{Q} \otimes \boldsymbol{\Sigma}^{-1})^{-1}\text{vec}(\mathbf{L})$ and variance $(\mathbf{V}_{b}^{-1} + \mathbf{Q} \otimes \boldsymbol{\Sigma}^{-1})^{-1}$, where $\mathbf{Q}$ and $\mathbf{L}$ are as in (2.4).

3. Simulate $1/\sigma^2 \sim \text{gamma}([\nu_0 + nm(m-1)/2]/2, [\nu_0\sigma_0^2 + RSS)]/2)$, where

$$RSS = \sum_{t=1}^{n}\sum_{i<j}(y_{ij,t} - [\mu_{ij} + \alpha y_{ij,t-1} + \mathbf{x}_{i,t-1}^T\mathbf{H}\mathbf{x}_{j,t-1}])^2.$$

4. Simulate $\boldsymbol{\Sigma}^{-1} \sim \text{Wishart}([\mathbf{S}_0 + \mathbf{RSS}]^{-1}, \eta_0 + mn)$, where

$$\mathbf{RSS} = \sum_{t=1}^{n}(\mathbf{X}_t - [\Theta + \mathbf{X}_{t-1}\mathbf{A}^T + \mathbf{Y}_{t-1}\mathbf{X}_{t-1}\mathbf{C}^T])^T(\mathbf{X}_t - [\Theta + \mathbf{X}_{t-1}\mathbf{A}^T + \mathbf{Y}_{t-1}\mathbf{X}_{t-1}\mathbf{C}^T]).$$

Iteration of this algorithm generates a Markov chain with a stationary distribution equal to the posterior distribution of $(\boldsymbol{\beta}, \mathbf{b}, \sigma^2, \boldsymbol{\Sigma})$. The empirical distribution of the simulated parameter values can be used to obtain approximate posterior means, quantiles and confidence intervals. Furthermore, the Gibbs sampling algorithm can be modified or extended to provide inference for related models and data structures. We consider two such modifications below.

**Latent attribute models:** The Gibbs sampling algorithm may be easily modified to accommodate the case that the $\mathbf{x}_{i,t}$'s are estimated latent attributes rather than observed attributes. Recall from the discussion in Section 2.2 that in this case we fix $\boldsymbol{\Sigma} = \mathbf{I}$ for reasons of identifiability. As such, we replace Step 4 in the Gibbs sampler described above with the following step that iteratively simulates values of the $\mathbf{x}_{i,t}$'s from their full conditional distributions:

4. Iteratively over nodes $i = 1, \ldots, m$ and time points $t = 0, \ldots, n$, simulate $\mathbf{x}_{i,t}$ from its multivariate normal full conditional distribution. For a time point $t$ such that

$0 < t < n$, this full conditional distribution has mean $\mathbf{Q}^{-1}\mathbf{l}$ and variance $\mathbf{Q}^{-1}$, where $\mathbf{l} = \sum_{k=1}^{3} \mathbf{W}_k^T \mathbf{z}_k$ and $\mathbf{Q} = \sum_{k=1}^{3} \mathbf{W}_k^T \mathbf{W}_k$ are given as follows:

$$\mathbf{W}_1 = \mathbf{I} \qquad \mathbf{z}_1 = \boldsymbol{\theta}_i + \mathbf{A}\mathbf{x}_{i,t-1} + \mathbf{C}\mathbf{X}_{t-1}^T \mathbf{y}_{i\cdot,t-1}$$

$$\mathbf{W}_2 = \tilde{\mathbf{X}}_t \mathbf{H}/\sigma \qquad \mathbf{z}_2 = (\tilde{\mathbf{y}}_{i\cdot,t+1} - \tilde{\boldsymbol{\mu}}_{i\cdot} - \alpha \tilde{\mathbf{y}}_{i\cdot,t})/\sigma$$

$$\mathbf{W}_3 = \mathbf{e}_i \otimes \mathbf{A} + \mathbf{y}_{i\cdot t} \otimes \mathbf{C} \qquad \mathbf{z}_3 = \mathrm{vec}(\mathbf{X}_{t+1} - \Theta - \tilde{\mathbf{I}}^T \tilde{\mathbf{X}}_t \mathbf{A}^T - \tilde{\mathbf{Y}}_t^T \tilde{\mathbf{X}}_t \mathbf{C}^T),$$

where $\mathbf{e}_i$ is a vector of zeros except for a one in the $i$th entry, and the tildes in the formulas for $\mathbf{W}_2, \mathbf{W}_3$ and $\mathbf{z}_2, \mathbf{z}_3$ indicate the removal of the $i$th row of a matrix or the $i$th element of a vector. The three terms in the sums for $\mathbf{l}$ and $\mathbf{Q}$ represent information about $\mathbf{x}_{i,t}$ from the past, from the future network, and from the future attributes, respectively. The values of $\mathbf{x}_{i,0}$ and $\mathbf{x}_{i,n}$ are updated similarly, except in the former case we have $\mathbf{z}_1 = \mathbf{0}$, and in the latter case we have $\mathbf{l} = \mathbf{W}_1^T \mathbf{z}_1$ and $\mathbf{Q} = \mathbf{W}_1^T \mathbf{W}_1$. As discussed in Section 2.2, we also restrict $\mathbf{H}$ to be a diagonal matrix when the attributes are latent. As a result, the calculation of $\mathbf{l}$ in Step 1 of the Gibbs sampler is as in (2.3) except that it is computed with $\mathbf{x}_{ij,t} = (\mathbf{x}_{j,t} \circ \mathbf{x}_{j,t})$, where "$\circ$" denotes element-wise multiplication. This is because $\mathbf{x}_{i,t}^T \mathbf{H} \mathbf{x}_{j,t} = (\mathbf{x}_{j,t} \circ \mathbf{x}_{j,t})^T \mathbf{h}$ in this case where $\mathbf{H}$ is diagonal. A numerical illustration of this Gibbs sampler as applied to longitudinal international relations data is provided in Section 2.4.1.

**Probit models for ordinal outcomes:** Ordinal network and attribute data may be accommodated by modeling the observed network and attribute processes as non-decreasing functions of latent processes that do follow the Gaussian MCR model in Equation 2.1. Specifically, let $y_{ij,t}$ be the observed ordinal-valued relation between nodes $i$ and $j$ at time $t$, and let $x_{i,k,t}$ be the value of the $k$th ordinal-valued attribute of node $i$ at time $t$. We then model the network and attribute process by assuming $y_{ij,t} = f(z_{ij,t})$ and $x_{i,k,t} = g_k(w_{i,k,t})$, where $f$ and $g_1, \ldots, g_p$ are unknown non-decreasing step functions, and the $z_{ij,t}$'s and $w_{i,k,t}$'s follow the Gaussian MCR model. A Gibbs sampler for this probit MCR model may be obtained by adding to Steps 1-4 above a few additional steps to simulate values of the $z_{ij,t}$'s, the $w_{i,k,t}$'s from their full conditional distributions, as well as the values defining $f$ and $g_1, \ldots, g_p$. Such

steps are standard in the literature on Bayesian modeling of ordinal data: Assuming normal prior distributions for the locations of the jumps in $f$, $g_1, \ldots, g_p$, the full conditional distributions of all of these quantities are constrained normal distributions, which may be simulated from using the inverse-CDF method. For information on such procedures in general, see Albert and Chib [1993]. Details of the Gibbs sampler for the MCR model in particular can be found in Appendix A.1. An example data analysis using the probit MCR model appears in Section 2.4.2.

## *2.4 Example Data Analyses*

In this section we illustrate the use of the MCR model with two example data analyses. The first example applies the model to a time series of international relations between 50 countries over a ten year period using a latent Gaussian MCR model. The second example studies the coevolution of the friendships and delinquency behaviors of 26 high-school students. In this latter example the network is binary and the nodal attribute is ordinal, and so an ordinal MCR model is employed.

### *2.4.1 International Relations*

The ICEWS project (http://www.lockheedmartin.com/us/products/W-ICEWS/iData.html) gathers data on international events occurring between countries. For this article, we analyze a monthly summary of the undirected dyadic relations between the 50 most active countries in the ICEWS database during a 112 month period from 2006 to 2015. Events between countries are assigned event codes, and each event has an associated intensity score ranging from -10 for extreme negative relations to +10 for extreme positive relations [Boschee et al., 2016]. For this analysis, we computed the monthly sum of these intensity scores for each pair of countries, and then applied a normal quantile-quantile transformation to all values. This resulted in a time series of 112 $50 \times 50$ sociomatrices $\mathbf{Y}_0, \ldots, \mathbf{Y}_{111}$, where $y_{ij,t}$ is the (transformed) intensity score sum between countries $i$ and $j$ for month $t$.

We fit the latent MCR model described in Section 2.2 with $p = 2$ latent attributes for each

| quantile | $a_{1,1}$ | $a_{1,2}$ | $a_{2,1}$ | $a_{2,2}$ | $c_{1,1}$ | $c_{1,2}$ | $c_{2,1}$ | $c_{2,2}$ |
|---|---|---|---|---|---|---|---|---|
| 2.5% | 0.148 | 0.044 | -0.004 | 0.339 | 0.005 | -0.002 | 0.001 | 0.015 |
| 50% | 0.193 | 0.094 | 0.047 | 0.388 | 0.006 | 0.000 | 0.004 | 0.018 |
| 97.5% | 0.241 | 0.144 | 0.098 | 0.438 | 0.008 | 0.002 | 0.006 | 0.020 |

Table 2.1: Posterior quantiles of parameters in the attribute evolution process for the ICEWS data.

country at each time point. With all regression coefficients being a priori i.i.d. $N(0, 100)$, and $\nu_0 = \sigma_0^2 = 1$, the Gibbs sampler described in Section 2.3.2 was run for 27,500 iterations. The first 2,500 iterations of the algorithm were dropped to allow for burn-in, and every 10th iteration thereafter was saved, yielding 2,500 simulated values for each parameter with which to approximate the posterior distribution. The average effective sample size across parameters in the MCR model was 789.

A 95% posterior credible interval for $\alpha$ is (0.134,0.145), indicating strong evidence for positive autocorrelation, and the diagonal values of $\mathbf{H}$ were positive for every iteration of the Gibbs sampler, indicating positive homophily. To get a sense of the magnitude of these coefficients, we computed the relative sum of squares contributions of the four terms of the network coevolution model, averaged across time points. These contributions were 28.2, 2.3, 16.2 and 53.2 percent, respectively, for the $\mu_{i,j}$'s, the autoregressive term, the homophily term and the error variance, respectively.

Posterior medians and 95% credible intervals for the $\mathbf{A}$ and $\mathbf{C}$ parameters of the attribute evolution model are given in Table 2.1. The most significant terms in these two matrices are the diagonal terms, indicating that the two latent attribute processes both show positive autocorrelation and positive contagion, but not strong interdependence with each other. The magnitude of the autocorrelation and contagion effects can be assessed by computing the sum of squares of these terms relative to the $\boldsymbol{\theta}_i$'s and the error term, averaged across time

points. These contributions were 60.0, 9.3, 4.6 and 26.1 percent, respectively, for the $\boldsymbol{\theta}_i$'s, the autoregressive term, the contagion term and the error variance, respectively.

Figure 2.2 plots the times series of the estimated latent attributes for a few selected countries. The top panel plots the first attribute (corresponding to the larger of the two homophily effects) for the United States, the United Kingdom and Iran. The plot indicates that this factor contributes positively to the relationship between the United States and the United Kingdom throughout the time period (as the estimated attributes have the same sign), whereas the contribution to the relationships of these countries with Iran is neutral until early 2013, when Hassan Rouhani was elected President over several hardline candidates and indicated a desire to negotiate a nuclear accord. The second panel of the figure plots a time series of the second latent attribute for Ukraine, Germany and Russia. In this plot we see that the time series for Russia and Ukraine are similar until the very beginning of 2014, when the protests against the Russian-backed government of President Yanukovych began.

Finally, we performed a small out-of-sample forecasting study to assess the benefit of the proposed model over the type of hidden Markov model considered in Ward et al. [2013], Durante and Dunson [2014], and displayed graphically in the left-hand side of Figure 2.1. Such models lack the network autocorrelation term $\alpha$ and the contagion term $\mathbf{C}$. To assess the predictive benefit of these effects we considered four models - with and without $\alpha$ and with and without $\mathbf{C}$. We obtained five one-month-ahead forecasts for each model, using data up to and including months 87, 92, 97, 102, 107 to predict the value of the network at time 88, 93, 98, 103 and 108 respectively. In terms of prediction error sum of squares, the full MCR model with network autocorrelation and contagion effects performed the best for each month forecasted. However, the submodel without contagion effects only performed 1.5% worse, on average over the five months forecasted. However, the submodel lacking both network autocorrelation and contagion performed on average 6.8% worse, suggesting that for these data, network autocorrelation effects are more important than contagion effects for forecasting the network.

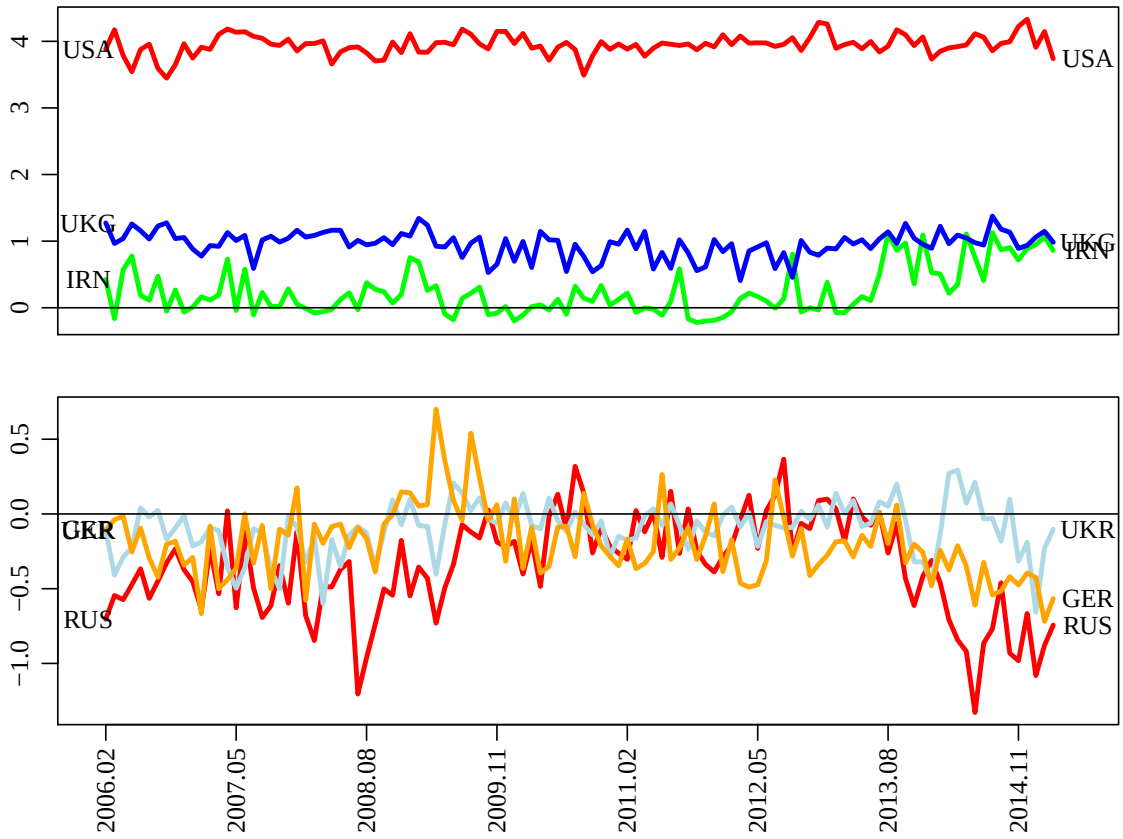


Figure 2.2: Time series of selected country-specific latent attributes. The top plot gives the estimated values of the first factor for the United States (USA), Iran (IRN) and the United Kingdom (UKG). The lower plot gives values of the second factor for Ukraine (UKR), Germany (GER) and Russia (RUS).

### *2.4.2 Friendship and Delinquency*

Knecht et al. [2007] gathered gathered data on a small directed friendship network of 25 Dutch secondary school students, along with nodal attributes including sex and a five-level ordinal measure of delinquency. Both delinquency and the friendship network were measured at four time points during a year-long period.

We model the coevolution of friendship and delinquency over the study period with an ordinal MCR model. Specifically, we model the binary friendship indicator $y_{ij,t}$ as $y_{ij,t} = f(z_{ij,t})$, and the delinquency category $x_{i,t}$ as $x_{i,t} = g(w_{i,t})$, where $f$ and $g$ are non-decreasing functions and $z_{ij,t}$ and $w_{i,t}$ follow a Gaussian MCR model:

$$\begin{aligned} z_{ij,t+1} &= \boldsymbol{\beta}^T \mathbf{s}_{i,j} + \alpha_1 z_{ij,t} + \alpha_2 z_{ji,t} + h w_{i,t} w_{j,t} + \epsilon_{ij,t+1} \\ w_{i,t+1} &= b s_i + a w_{i,t} + c_1 \mathbf{w}_t^T \mathbf{z}_{i\cdot,t} + c_2 \mathbf{w}_t^T \mathbf{z}_{\cdot i,t} + e_{i,t+1} \\ \{\epsilon_{i,j,t}\}, \{e_{i,t}\} &\sim \text{ i.i.d. } N(0,1), \end{aligned}$$

where $\alpha_1$, $\alpha_2$ and $a$ describe network autocorrelation, network reciprocity, and delinquency autocorrelation respectively, $h$ is a homophily parameter and $c_1$ and $c_2$ are contagion parameters. Additionally, $s_i$ is the binary indicator that student $i$ is female, and $\mathbf{s}_{i,j} = (s_i, s_j, 1(s_i = s_j))$ is a vector describing the gender characteristics of the directed dyad $(i, j)$. The unknown parameters $\boldsymbol{\beta}$ and $b$ describe the effects of gender on temporal changes to the network and the nodal attributes, respectively. We also note that the latent variables $z_{i,j,1}$ and $w_{i,1}$ at the first time point were modeled as $z_{i,j,1} \sim N(\boldsymbol{\gamma}^T \mathbf{s}_{i,j}, \sigma^2)$ and $w_{i,1} \sim N(g s_i, \tau^2)$, respectively. The parameters $\boldsymbol{\beta}_1$ and $b_1$ describe the effects of gender on the initial state of the network and delinquency.

The parameters in this model were estimated using the Gibbs sampler for ordinal data described in Section 2.3. We ran the MCMC algorithm for 40,000 iterations, and dropped the first 20,000 iterations to allow for burn-in of the Markov chain. The lowest effective sample size among the regression parameters was 643, and the median effective sample size was around 3000. Posterior medians and 95% posterior credible intervals are given in Table

| | $\beta_1$ | $\beta_2$ | $\beta_3$ | $\alpha_1$ | $\alpha_2$ | $h$ | $b$ | $a$ | $c_1$ | $c_2$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 2.5 % | -0.412 | -0.205 | 0.081 | 0.438 | 0.286 | 0.023 | -0.219 | 0.319 | -0.061 | -0.033 |
| 50 % | -0.278 | -0.072 | 0.240 | 0.530 | 0.374 | 0.084 | 0.269 | 0.583 | -0.001 | 0.028 |
| 97.5 % | -0.145 | 0.064 | 0.395 | 0.621 | 0.463 | 0.200 | 0.778 | 0.845 | 0.057 | 0.090 |

| | $\gamma_1$ | $\gamma_2$ | $\gamma_3$ | $g$ |
|---|---|---|---|---|
| 2.5% | -0.399 | -0.277 | 0.662 | -1.966 |
| 50% | -0.177 | -0.060 | 0.879 | -0.864 |
| 97.5% | 0.027 | 0.140 | 1.104 | -0.088 |

Table 2.2: Posterior quantiles of MCR model parameters for the friendship and delinquency data.

2.2.

The results indicate evidence of positive autocorrelation for both the network and attribute processes (represented by $\alpha_1$, $\alpha_2$ and $a$), positive homophily with respect to the delinquency attribute ($h$), but not not evidence of contagion ($c_1$ and $c_2$). This lack of evidence for contagion is in accord with the results of Snijders et al. [2010], who used a stochastic actor-based utility model to analyze these data. Additionally, the posterior distributions of $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ indicate evidence of homophily with respect to sex, and that males had a higher rate of increase in friendship nominations over time. The posterior distributions of $b$ and $g$ indicated a lower rate of delinquency among females at the beginning of the study ($g$) but not a further effect of sex on the delinquency process ($b$).

## *2.5 Discussion*

In this chapter we proposed a multiplicative coevolution regression (MCR) model for dynamic network and nodal attribute data, which is able to quantify patterns of autoregression, homophily and contagion in social network data. In the Gaussian case, this model can be

regarded as a linear regression and we can use OLS to obtain the MLEs of the parameters.

One advantage of MCR models is that they can be easily extended and modified to accommodate different data types and network patterns. For example, latent attributes can be included to explain the variation in network relations that is believed to have not been explained by the observed covariates. In this case, we can also consider the the evolution of the latent factors that is influenced by the network and the observed nodal attributes. Also using a latent variable representation, an MCR model for ordinal network and nodal attributes can be constructed. For this model, a Bayesian method using MCMC can be implemented to get the estimates for the parameters and the latent variables. The Bayesian estimation algorithm is helpful in the imputation of missing values and forecasting future networks or nodal attributes. By adding extra steps at each MCMC iteration to update those values, we can obtain the imputations or forecasts using the samples from their posterior distributions.

The work of Snijders et al. [2010] and Hanneke et al. [2010] provide methods for modeling evolution of network based on network statistics including density, stability, reciprocity and transitivity. Therefore, those models are flexible and the interpretation becomes straightforward. Compared to their models, we do not explicitly put those statistics in the model, but such effects can be estimated by including network statistics in the regression model. For example, to estimate reciprocity, we include $y_{ji,t-1}$ as a predictor for $y_{ij,t}$.

One thing to be mentioned that some terms in the model are possibly confounded with each other. For example, if the network presents strong reciprocity, then $\mathbf{C}_1\mathbf{X}_t^T\mathbf{y}_{i\cdot,t}$ and $\mathbf{C}_2\mathbf{X}_t^T\mathbf{y}_{\cdot i,t}$ will be highly correlated. The estimation procedure can hardly identify these two terms, especially for probit MCR models. Though it has little influence on prediction, it is still necessary to come up with a method of model selection for the coevolution model in the future.

# Chapter 3

# COMPOSITE LIKELIHOOD ESTIMATION OF BINARY EXCHANGEABLE NETWORK MODELS

## *3.1 Overview*

Random effects and latent factor models are widely used for network data. The typical way to estimate the parameters of those models is to use Bayesian computational methods such as MCMC. However, this method is not scalable and will not be computationally efficient for large network. Our goal is to develop a scalable way to estimate the parameters in network models.

For example, for binary network data, we can use a probit social relations model (SRM) to describe the relationships in the network. The probit SRM is as follows. Assuming that the observed relationship between node $i$ and $j$ is $y_{ij} \in \{0, 1\}$, the relationship $y_{ij}$ between node $i$ and node $j$ can be modeled in terms of a hidden relationship $z_{ij}$, which we model with a probit link. $z_{ij}$ is decomposed into the effect from the row effect $a_i$, column effect $b_j$ and some error term $e_{ij}$.

$$y_{ij} = 1(z_{ij} > 0),\ i, j \in \{1, \cdots, m\},\ i \neq j$$
$$z_{ij} = \mu + a_i + b_j + e_{ij}.$$

The row effects $\{a_i\}_{i=1}^m$ are associated with the out-degrees of the network, which describe the tendency of node $i$ to extend connections to other nodes, i.e., sociability. The column effects $\{b_j\}_{j=1}^m$ are associated with the in-degrees and represent how popular node $j$ is in terms of receiving connections from others. Social relations models assume a bivariate Gaussian distribution for $(a_i, b_i)$. Since $y_{ij}$ is determined only by the sign of $z_{ij}$, the scale of $z_{ij}$ is not identifiable, so we set the variance of the error term to 1. The correlation $\rho_e$ is called dyadic correlation, which is related to the reciprocity of the network, i.e., the tendency of sending

a link back to where one receives a connection from. Formally, the model can be written as follows:

$$
\begin{aligned}
y_{ij} &= 1(z_{ij} > 0),\ i, j \in \{1, \cdots, m\},\ i \neq j \\
z_{ij} &= \mu + a_i + b_j + e_{ij}, \\
\begin{pmatrix} a_1 \\ b_1 \end{pmatrix}, \cdots, \begin{pmatrix} a_m \\ b_m \end{pmatrix} &\overset{i.i.d.}{\sim} \mathrm{N}_2 \left( \mathbf{0}, \begin{pmatrix} \sigma_a^2 & \rho_{ab}\sigma_a\sigma_b \\ \rho_{ab}\sigma_a\sigma_b & \sigma_b^2 \end{pmatrix} \right), \\
\left\{ \begin{pmatrix} e_{ij} \\ e_{ji} \end{pmatrix} \right\}_{i \neq j} &\overset{i.i.d.}{\sim} \mathrm{N}_2 \left( \mathbf{0}, \begin{pmatrix} 1 & \rho_e \\ \rho_e & 1 \end{pmatrix} \right).
\end{aligned}
\tag{3.1}
$$

To obtain the maximum likelihood estimates (MLE) for Model (3.1), we need to optimize the full likelihood $\ell(\boldsymbol{\theta} : \mathbf{y}) \propto p(\mathbf{y}|\boldsymbol{\theta}) = \int_{S(\mathbf{y})} p(\mathbf{z}|\boldsymbol{\theta}) d\mathbf{z}$, with $\boldsymbol{\theta} = (\mu, \sigma_a^2, \sigma_b^2, \rho_{ab}, \rho_e)$ representing the parameter vector. This full likelihood is intractable because it involves an integral that has no closed form expression. Therefore, it is infeasible to optimize it directly to get the MLE.

Literature is available for generalized linear models or generalized linear mixed models, e.g., [Schall, 1991, Piegorsch and Casella, 1996, Pinheiro and Chao, 2006]. If $\mathbf{z}$ is a function of random effects, then Laplace's method can be a useful tool to approximate the integral. It is based on the mode and curvature of the random effects in terms of the parameters. Well established software is also available for this purpose including the R package lme4 [Bates et al., 2014], INLA [Rue et al., 2009]. However, this method can perform poorly or even fail for cases such as SRM, where we assume dyadic correlations. This approach will work poorly for estimates of dyadic correlations, since each dyad-level random effect is estimated with only two points, i.e,. $y_{ij}$ and $y_{ji}$.

In this chapter, we propose a scalable method to estimate the parameters in the random effects and latent factor models for network data. Our approach is to optimize a composite likelihood, which is the sum of he marginal likelihoods of the subsets of the data. Bivariate (pairwise) likelihoods are popular choices [Renard et al., 2004, Varin et al., 2011]. In those literature, the authors often choose the dimension of the subsets to be 2 in various contexts,

such as serially correlated models, and show that composite likelihood methods can provide reasonable inferences. The applications include Gibbs random field [Friel, 2012, Stoehr, 2015], autoregressive models [Varin and Czado, 2010] and spatial analyses [Heagerty and Lele, 1998]. In terms of social network analysis, the working paper from Chen et al. [2014] focuses on efficient estimation of social intercorrelations in large-scale networks using the spatial model and Bartolucci et al. [2015] proposes a method for dynamic networks that is based on a hidden Markov model. Asuncion et al. [2010] works on the composite likelihood methods for exponential random graph models (ERGM). They apply a machine learning algorithm called contrastive divergence (CD) to optimize the composite likelihood by iterating repeatedly between obtaining samples from the current model, used to calculate a gradient estimate, and optimizing the model parameters given that gradient. In this chapter, we focus on the application of composite likelihood estimation on exchangeable network models. We make use of the triad structure of subnetworks so that the method is scalable and the computational cost does not grow with the network size.

The organization of this chapter is as follows. In Section 3.2, we will introduce the method of composite likelihood estimation and how we can apply it to exchangeable models for network data. In particular, the SRM will be used as an example to illustrate the usage of triadic composite likelihood estimation. Section 3.3 covers the topic of calculating the gradients of the composite likelihood which allows us to do gradient-based optimization. We will introduce a sampling method to approximate the gradients. In Section 3.4, the algorithm for optimizing composite likelihood is introduced and we compare it to two other methods, the Laplace approximation and Bayesian estimation using MCMC, with a simulation study on the SRM for binary network data. In Section 3.5, we will summarize the methods talked about in the previous sections and discuss how we can extend the composite likelihood estimation method to broader scenarios.

### *3.2 Composite Likelihood Estimation*

The topic of this chapter is estimation of the parameters in random effects and latent factor models for binary network data. We represent such data with an $m \times m$ matrix $\mathbf{Y}$ with undefined diagonal elements. The $ij$-th element $y_{ij}$ of $\mathbf{Y}$ refers to the (directed) relationship from individual $i$ to $j$ for $i \neq j$. Several methods, including latent factor models [Hoff, 2009a] and exponential random graph models (ERGM) [Hunter et al., 2008] provide ways for modeling binary networks. Suppose there is a probability model for $\mathbf{Y}$ which is associated with a parameter vector $\boldsymbol{\theta}$. Our goal is to estimate the parameter $\boldsymbol{\theta}$. One estimator is the maximum likelihood estimate (MLE) which is the value of $\boldsymbol{\theta}$ that maximizes the likelihood. However it is often hard to obtain because the log-likelihood usually contains a multiple integration for which there is no closed form expression.

**Example 3.2.1.** *Probit models for binary network data.*

Assume that the observed binary outcome $y_{ij}$ is the indicator that some continuous variable $z_{ij}$ exceeds some threshold, say zero. We call $\mathbf{Z}$ the continuous representative of $\mathbf{Y}$. This provides us with a variety of choices for modeling $\mathbf{Y}$ in terms of $\mathbf{Z}$, such as random effects and latent factor models. This generic model for $\mathbf{Y}$ can be expressed as

$$\begin{aligned} y_{ij} &= 1(z_{ij} > 0),\ i, j \in \{1, \cdots, m\},\ i \neq j \\ z_{ij} &= \mu_{ij} + e_{ij}, \end{aligned} \tag{3.2}$$

where $1(z > 0)$ refers to the indicator function which takes value 1 when $z > 0$ and 0 otherwise. The mean term $\mu_{ij}$ can be replaced by a function of nodal or dyadic covariates, random effects or latent factors. Denoting $\mathbf{E} = \{e_{ij}\}$ as the error matrix and $\mathbf{e}$ being its vectorization, we use a Gaussian model for the error term,

$$\mathbf{e} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma}(\boldsymbol{\phi})),$$

with $\boldsymbol{\phi}$ being the covariance parameters. Assuming the mean term is associated with parameter $\boldsymbol{\mu}$, our goal is to obtain the MLE for parameter $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\phi})$. Denote two $m(m-1)$-dimensional vectors $\mathbf{y}$ and $\mathbf{z}$ as the vectorization (without diagonals) of $\mathbf{Y}$ and $\mathbf{Z}$ respectively.

Then the log-likelihood for Model (3.2) is

$$\begin{aligned}\ell(\boldsymbol{\theta} : \mathbf{y}) &= \log\left[p(\mathbf{y}|\boldsymbol{\theta})\right] \\ &= \log\left[\int_{S(\mathbf{y})} p(\mathbf{z}|\boldsymbol{\theta})d\mathbf{z}\right],\end{aligned}$$

where $S(\mathbf{y}) = \{\mathbf{z} : p(\mathbf{z}|\mathbf{y}) \neq 0\}$ represents the support of $\mathbf{z}$ that is determined by $\mathbf{y}$. Since $\mathbf{y}$ is a binary vector, $S(\mathbf{y})$ will be a $m(m-1)$-dimensional hyperrectangular $H_{y_1} \times H_{y_2} \times \cdots \times H_{y_{m(m-1)}}$, with $H_0 = (-\infty, 0]$ and $H_1 = (0, \infty)$. The MLE $\hat{\boldsymbol{\theta}}$ is a root of the score function $\mathbf{s}(\hat{\boldsymbol{\theta}}) \equiv \nabla_{\boldsymbol{\theta}}\ell(\hat{\boldsymbol{\theta}} : \mathbf{y}) = 0$. The solution can be obtained with gradient methods, while computing the gradient is hard. The likelihood involves a multiple integration with respect to $m(m-1)$ variables. So does the gradient of the log-likelihood:

$$\begin{aligned}\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta} : \mathbf{y}) &= \frac{\nabla_{\boldsymbol{\theta}} p(\mathbf{y}|\boldsymbol{\theta})}{p(\mathbf{y}|\boldsymbol{\theta})} \\ &= \frac{\int_{S(\mathbf{y})} \nabla_{\boldsymbol{\theta}} p(\mathbf{z}|\boldsymbol{\theta})d\mathbf{z}}{\int_{S(\mathbf{y})} p(\mathbf{z}|\boldsymbol{\theta})d\mathbf{z}}.\end{aligned}$$

In most cases, the integral is not tractable, especially when $m$ is large. □

When $m$ is small, however, we will show in the next section that an accurate Monte Carlo approximation to the gradient is available. This suggests that we use marginal distributions of sub-networks. Let $k \in \{1, \cdots, \binom{m}{q}\}$ index $q$-tuples ($q \ll m$) of nodes, and let $\mathbf{Y}_k$ be the subgraph of $\mathbf{Y}$ corresponding to $q$-tuple $k$. More precisely, $\mathbf{Y}_k$ is defined as $\{y_{ij} : (i,j) \in \mathscr{K} \times \mathscr{K}\}$, with $\mathscr{K} \subset \{1, \cdots, m\}$. Similarly, let $\mathbf{y}_k$ be the vectorization of $\mathbf{Y}_k$. Considering all the sub-networks composed of $q$ nodes and multiplying their marginal distributions together, we obtain a pseudo-likelihood

$$\tilde{L}(\boldsymbol{\theta} : \mathbf{y}) = \prod_{k=1}^{K} p(\mathbf{y}_k|\boldsymbol{\theta}),$$

where $K = \binom{m}{q}$ is the total number of sub-networks that consist of $q$ nodes. Denoting $\ell_k(\boldsymbol{\theta} : \mathbf{y}) = \log\left[p(\mathbf{y}_k|\boldsymbol{\theta})\right]$ as the logarithm of the marginal likelihood, the pseudo-log-likelihood

is

$$\begin{aligned}\tilde{\ell}(\boldsymbol{\theta} : \mathbf{y}) &= \sum_{k=1}^{K} \log\left[p(\mathbf{y}_k|\boldsymbol{\theta})\right] \\ &= \sum_{k=1}^{K} \ell_k(\boldsymbol{\theta} : \mathbf{y}_k).\end{aligned}$$

The maximum pseudo-likelihood estimate (MPLE) satisfies $\tilde{\mathbf{s}}(\hat{\boldsymbol{\theta}}) \equiv \nabla_{\boldsymbol{\theta}}\tilde{\ell}(\hat{\boldsymbol{\theta}} : \mathbf{y}) = 0$. Just like the score function of the full likelihood, $\tilde{\mathbf{s}}(\boldsymbol{\theta})$ defines an estimating equation whose expectation is zero at the true value of $\boldsymbol{\theta}$. The score function is defined by taking the derivative of $\tilde{\ell}(\boldsymbol{\theta} : \mathbf{y})$ with respect to $\boldsymbol{\theta}$ as

$$\begin{aligned}\tilde{\mathbf{s}}(\boldsymbol{\theta}) &= \sum_{k=1}^{K} \nabla \log[p(\mathbf{y}_k|\boldsymbol{\theta})] \\ &= \sum_{k=1}^{K} \frac{\nabla p(\mathbf{y}_s|\boldsymbol{\theta})}{p(\mathbf{y}_k|\boldsymbol{\theta})}.\end{aligned} \tag{3.3}$$

The expectation of the score function is

$$\begin{aligned}\mathrm{E}[\tilde{\mathbf{s}}(\boldsymbol{\theta})] &= \sum_{k=1}^{K} \mathrm{E}\left[\frac{\nabla_{\boldsymbol{\theta}} p(\mathbf{y}_k|\boldsymbol{\theta})}{p(\mathbf{y}_k|\boldsymbol{\theta})}\right] \\ &= \sum_{k=1}^{K} \int_{S(\mathbf{y}_k)} \frac{\nabla_{\boldsymbol{\theta}} p(\mathbf{y}_k|\boldsymbol{\theta})}{p(\mathbf{y}_k|\boldsymbol{\theta})} p(\mathbf{y}_k|\boldsymbol{\theta}) d\mathbf{y}_k \\ &= \sum_{k=1}^{K} \int_{S(\mathbf{y}_k)} \nabla_{\boldsymbol{\theta}} p(\mathbf{y}_k|\boldsymbol{\theta}) d\mathbf{y}_k \\ &= \sum_{k=1}^{K} \nabla_{\boldsymbol{\theta}} \int_{S(\mathbf{y}_k)} p(\mathbf{y}_k|\boldsymbol{\theta}) d\mathbf{y}_k \\ &= \sum_{k=1}^{K} \nabla_{\boldsymbol{\theta}} \mathbf{1} \\ &= \mathbf{0}.\end{aligned}$$

In many cases, $\boldsymbol{\theta}$ is the maximizer of $\mathrm{E}[\tilde{\ell}(\boldsymbol{\theta} : \mathbf{y})]$. This suggests that, if $\hat{\boldsymbol{\theta}}$ maximizes the pseudo-likelihood $\tilde{\ell}(\boldsymbol{\theta} : \mathbf{y})$, then $\tilde{\boldsymbol{\theta}}$ will be close to $\boldsymbol{\theta}$. According to Cox and Reid [2004], this

is necessary for getting consistent and asymptotically normal estimates. In this chapter, we will show how to make use of subsets with smaller dimension $q \ll m$ to get an estimate of $\boldsymbol{\theta}$ in a computationally feasible way.

Estimation of $\boldsymbol{\theta}$ using the pseudo-likelihood above turns out to be a special case of what is called composite likelihood estimation. The idea was introduced in Besag [1975]. Besag proposed a pseudo-likelihood method for analyzing spatially correlated data, i.e. irregularly distributed data points with spatial stochastic interactions. Lindsay [1988] introduced the name "composite likelihood" in consideration of the way the likelihood is constructed. In general, the composite likelihood estimate is the maximizer of $\tilde{\ell}(\boldsymbol{\theta} : \mathbf{y}) = \sum_{k=1}^{K} \ell_k(\boldsymbol{\theta} : \mathbf{y}_k)$, where $\ell_k(\boldsymbol{\theta} : \mathbf{y}) = \log[p(\mathbf{y}_k|\boldsymbol{\theta})]$ or $\ell_k(\boldsymbol{\theta} : \mathbf{y}) = \log[p(\mathbf{y}_k|\mathbf{y}_{-k}, \boldsymbol{\theta})]$, $k = 1, \cdots, K$ are the log-likelihoods associated with the subsets of data. Assume $\mathbf{y}$ is an $m$-dimensional vector of data and $\mathbf{y}_1, \cdots, \mathbf{y}_K$ are the $q$-dimensional subvectors of $\mathbf{y}$, $q \ll m$. For example, we have $K = \binom{m}{q}$. Depending on the application, the log-likelihood $\ell_k(\boldsymbol{\theta} : \mathbf{y}_k)$ can represent the marginal distribution of $\mathbf{y}_k$, i.e., $\ell_k(\boldsymbol{\theta} : \mathbf{y}_k) = \log[p(\mathbf{y}_k|\boldsymbol{\theta})]$, or it can be associated with the conditional distribution of $\mathbf{y}_k$ on other variables $\mathbf{y}_{-k}$, such as the neighbors of $s$ or the complementary vector. In this chapter, we focus on the marginal composite likelihood with $\ell_k(\boldsymbol{\theta} : \mathbf{y}_k) = \log[p(\mathbf{y}_k|\boldsymbol{\theta})]$, the marginal distribution of $\mathbf{y}_k$. The composite log-likelihood we propose is defined as $\tilde{\ell}(\boldsymbol{\theta} : \mathbf{y}) = \sum_{k=1}^{K} \ell_k(\boldsymbol{\theta} : \mathbf{y}_k)$. This makes it practical to get accurate estimation of its gradient with respect to $\boldsymbol{\theta}$, and thus it becomes feasible to optimize the composite likelihood using a gradient method and get an estimate for $\boldsymbol{\theta}$.

### *3.2.1 Composite likelihood for exchangeable models*

The focus of this chapter is exchangeable network models. Methods for non-exchangeable models will be discussed in Chapter 5. Exchangeability means that the distribution of the graph is invariant to permutations of the node labels. We will show that the composite log-likelihood for exchangeable models of binary networks can be simplified to a weighted summation of a small number of likelihood functions. Optimizing such a composite likelihood has a computational cost that does not grow with the network size.

In other applications [Renard et al., 2004, Varin et al., 2011], bivariate composite likelihoods have been used to estimate model parameters. For network data, we consider subgraphs of dyads, triads or $q$ nodes. A dyad is defined as a group of two nodes while a triad is a group of three nodes.

Let $(i, j)$ be a dyad and $\mathbf{y}_{ij} = (y_{ij}, y_{ji})$ be the relationships in the dyad. The composite log-likelihood based on dyads is called dyadic composite likelihood. It is defined as

$$\tilde{\ell}^{(2)}(\boldsymbol{\theta} : \mathbf{y}) = \sum_{ij} \log[p(\mathbf{y}_{ij}|\boldsymbol{\theta})]. \tag{3.4}$$

For binary network data, there are only 4 possible outcomes for dyads, which are $(0, 0)$, $(0, 1)$, $(1, 0)$ and $(1, 1)$. If the model for $\mathbf{Y}$ is exchangeable, we have $p(\mathbf{y}_{ij}|\boldsymbol{\theta}) = p(\mathbf{y}_{i'j'}|\boldsymbol{\theta})$, for example, $\Pr(\mathbf{y}_{ij} = (0, 1)|\boldsymbol{\theta}) = \Pr(\mathbf{y}_{i'j'} = (0, 1)|\boldsymbol{\theta})$ and also $\Pr(\mathbf{y}_{ij} = (0, 1)|\boldsymbol{\theta}) = \Pr(\mathbf{y}_{ji} = (1, 0)|\boldsymbol{\theta}) = \Pr(\mathbf{y}_{ij} = (1, 0)|\boldsymbol{\theta})$. This results in a simpler form of dyadic composite likelihood as

$$\begin{aligned}\tilde{\ell}^{(2)}(\boldsymbol{\theta} : \mathbf{y}) =& n_{00}\log[\Pr(\mathbf{y}_{12} = (0, 0)|\boldsymbol{\theta})] + n_{(01)}\log[\Pr(\mathbf{y}_{12} = (0, 1)|\boldsymbol{\theta})] \\ &+ n_{11}\log[\Pr(\mathbf{y}_{12} = (1, 1)|\boldsymbol{\theta})],\end{aligned} \tag{3.5}$$

where $n_{ab}$ stands for the number of subgraphs with outcome $y_{ij} = (a, b)$, $i < j$ and $n_{(01)} = n_{01} + n_{10}$. The total number of subgraphs is $n_{00} + n_{01} + n_{10} + n_{11} = \binom{m}{2}$.

When the model is complicated, dyadic composite likelihood may not be informative enough for estimating all the parameters. In this case, we consider higher order composite likelihood, for example, triadic composite likelihood. Let $(i, j, k)$ be a triad and $\mathbf{y}_{ijk} = (y_{ij}, y_{ji}, y_{jk}, y_{kj}, y_{ki}, y_{ik})$ represent the relationships between the nodes in the triad. Similar to the dyadic composite likelihood, a triadic composite likelihood is defined as

$$\tilde{\ell}^{(3)}(\boldsymbol{\theta} : \mathbf{y}) = \sum_{ijk} \log[p(\mathbf{y}_{ijk}|\boldsymbol{\theta})].$$

For binary network data, there are a total of $2^6 = 64$ possible outcomes for $\mathbf{y}_{ijk}$. We introduce a function $t(\cdot)$ that maps a binary vector to an integer from 0 to 63. Formally,

$$t(\mathbf{y}_{ijk}) = 2^5 y_{ij} + 2^4 y_{ji} + 2^3 y_{jk} + 2^2 y_{kj} + 2 y_{ki} + y_{ik}.$$

If the model is exchangeable, then it is straightforward to show that $\Pr(t(\mathbf{y}_{ijk}) = t|\boldsymbol{\theta}) = \Pr(t(\mathbf{y}_{i'j'k'}) = t|\boldsymbol{\theta})$ for any $t = 0, \cdots, 63$, $(i, j, k)$ and $(i', j', k')$. The definition of triadic composite log-likelihood is thus equivalent to

$$\tilde{\ell}^{(3)}(\boldsymbol{\theta} : \mathbf{y}) = \sum_{t=0}^{63} n_t \log[\Pr(t(\mathbf{y}_{ijk}) = t|\boldsymbol{\theta})], \tag{3.6}$$

where $n_t$ refers to the number of outcomes that corresponds to the integer $t$ under the mapping $t(\cdot)$ and $\sum_{t=0}^{63} n_t = \binom{m}{3}$. However, exchangeability implies that this summation can be simplified further. $\mathbf{y}_{ijk} = (0, 0, 0, 1, 0, 1)$ has the same probability as $\mathbf{y}_{ijk} = (0, 0, 1, 0, 1, 0)$ by relabelling nodes $(i, j, k)$ as $(i, k, j)$. This is because the graph of $(0, 0, 0, 1, 0, 1)$ is isomorphic with that of $(0, 0, 1, 0, 1, 0)$. Under exchangeability, two isomorphic graphs correspond to the same probability. This implies that the summation (3.6) over 64 outcomes can be reduce to the summation over 16 non-isomorphic graphs. Figure 3.1 shows the representative of the 16 groups. Given one binary outcome of $\mathbf{y}_{ijk}$, we can permute the labels and make it identical to one of the 16 graphs.

Remember that each outcome of $\mathbf{y}_{ijk}$ can be mapped to an integer between 0 and 63. So we can also categorize the 64 integers into 16 groups. Table 3.1 summarizes the categories of the integer representatives $t(\mathbf{y}_{ijk})$. Defining a function $c(\cdot)$ that map the integer representative $t(\mathbf{y}_{ijk})$ to the corresponding category in Table 3.1, we can further simplify the triadic composite log-likelihood in Equation (3.7) to

$$\tilde{\ell}^{(3)}(\boldsymbol{\theta} : \mathbf{y}) = \sum_{c=1}^{16} n_c \log[\Pr(c \circ t(\mathbf{y}_{ijk}) = c|\boldsymbol{\theta})], \tag{3.7}$$

where $n_c$ now stands for the number of triads with outcomes corresponds to integer of category $c$, and $\sum_{c=1}^{16} n_c = \binom{m}{3}$. Calculating $\{n_c = 1, \cdots, 16\}$ can be done quickly using the method introduced in Batagelj and Mrvar [1998], especially when the network is sparse.

Both dyadic and triadic composite likelihood depend on the marginal probability of only 3 or 16 subgraphs of small dimensions respectively. Gradients of the likelihoods associated with such small subgraphs may be computed numerically. At each step of the gradient method, we only need to approximate the gradient of 3 or 16 marginal likelihoods, each

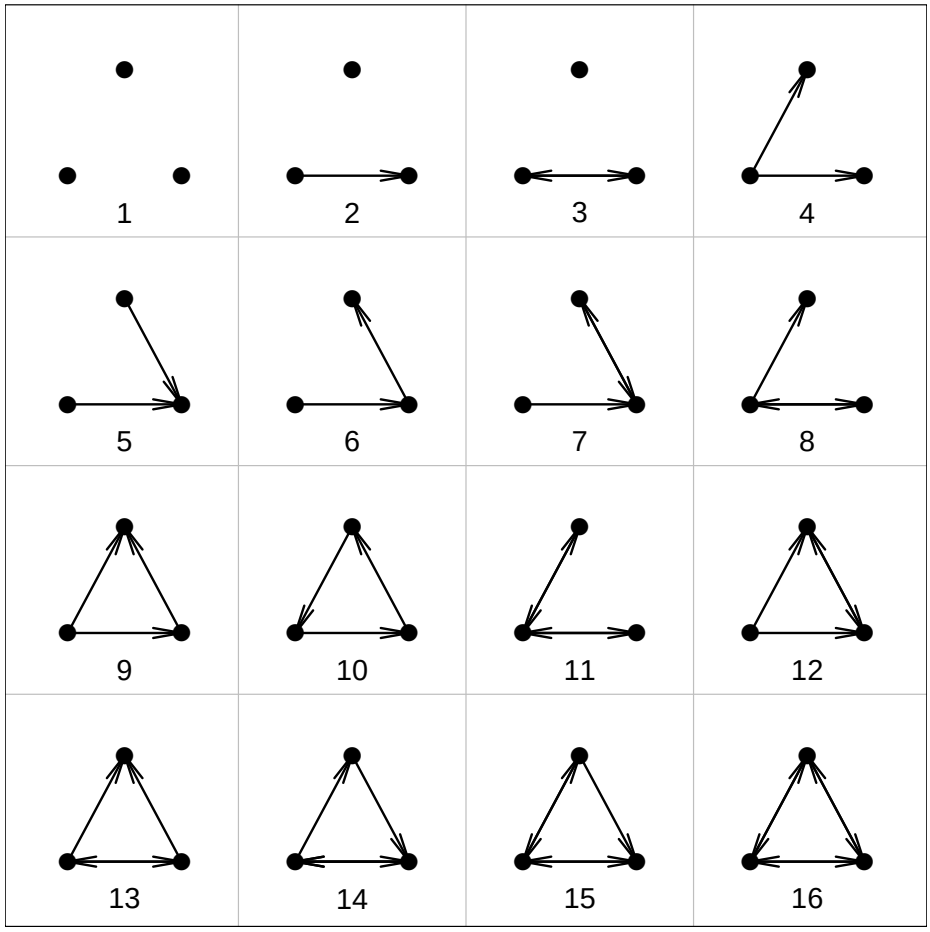


Figure 3.1: Representatives of the 16 outcome groups that have different probabilities under an exchangeable model. The number represents the corresponding category in Table 3.1.

based on only 2 or 3 nodes. Therefore we can use a scalable way to optimize the composite likelihood and get an estimate for $\boldsymbol{\theta}$. Similarly, we can define tetradic (group of 4 nodes) or $q$-node composite likelihood if needed. Taking advantage of exchangeability, we can also write the composite log-likelihood as a weighted sum of a finite number of marginal distributions. When $q > 3$, the number of different categories is very large. As we will discuss in Chapter 4, instead of calculating the number of isomorphic subgraphs, we will process the subgraphs using a stochastic method when optimizing higher ordered composite likelihood.

| Category | $d(\mathbf{y}_{ijk})$ | Category | $d(\mathbf{y}_{ijk})$ |
|---|---|---|---|
| 1 | 0 | 9 | 22, 25, 26, 37, 38, 41 |
| 2 | 1, 2, 4, 8, 16, 32 | 10 | 21, 42 |
| 3 | 3, 12, 48 | 11 | 15, 51, 60 |
| 4 | 6, 24, 33 | 12 | 27, 45, 54 |
| 5 | 9, 18, 36 | 13 | 30, 39, 57 |
| 6 | 5, 10, 17, 20, 34, 40 | 14 | 23, 29, 43, 46, 53, 58 |
| 7 | 11, 13, 19, 44, 50, 52 | 15 | 31, 47, 55, 59, 61, 62 |
| 8 | 7, 14, 28, 35, 49, 56 | 16 | 63 |

Table 3.1: Integer representatives of triad census

### *3.2.2 Choices of q*

In many literature on composite likelihood, bivariate (pairwise) likelihood is a popular choice due to its simplicity. For network data, an analogy would be considering dyads (pairs of nodes) and their marginal distributions. For exchangeable network models, this results in a composite likelihood of three different marginal distributions as shown in Equation (3.5), and each of them is associated with a bivariate random vector. Optimizing this composite likelihood is much easier than optimizing a composite likelihood based on larger subsets of the date. This is because the number of non-isomorphic graphs grows as the size of the subsets increases, and also a larger subset results in more difficulty in calculating the gradient of the marginal likelihoods. However, we might be concerned that the marginal likelihood of dyads might not contain enough information to estimate models parameters, for example, the dyadic correlation, e.g, $\text{corr}(e_{ij}, e_{ji}) \neq 0$ in Model (3.2).

**Definition 3.2.1.** *A class of estimating function* $\mathscr{L} = \{\ell(\boldsymbol{\theta} : \mathbf{y}),\ \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$ *is identifiable if for any* $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \boldsymbol{\Theta}$, $\ell(\boldsymbol{\theta}_2 : \mathbf{y}) = \ell(\boldsymbol{\theta}_1 : \mathbf{y})$ *implies* $\boldsymbol{\theta}_1 = \boldsymbol{\theta}_2$.

Identifiability is important because without it, different parameter values are not distin-

guishable from the data. When choosing the size of subsets for the composite likelihood, we use identifiability as a guideline. In this section, we explore these issues in the context of estimating the parameters in a probit social relations model introduced in Section 3.1. Recall that the model is given by

$$
\begin{aligned}
y_{ij} &= 1(z_{ij} > 0),\ i, j \in \{1, \cdots, m\},\ i \neq j \\
z_{ij} &= \mu + a_i + b_j + e_{ij}, \\
\begin{pmatrix} a_1 \\ b_1 \end{pmatrix}, \cdots, \begin{pmatrix} a_m \\ b_m \end{pmatrix} &\overset{i.i.d.}{\sim} \mathrm{N}_2 \left( \mathbf{0}, \begin{pmatrix} \sigma_a^2 & \rho_{ab}\sigma_a\sigma_b \\ \rho_{ab}\sigma_a\sigma_b & \sigma_b^2 \end{pmatrix} \right), \\
\left\{ \begin{pmatrix} e_{ij} \\ e_{ji} \end{pmatrix} \right\}_{i \neq j} &\overset{i.i.d.}{\sim} \mathrm{N}_2 \left( \mathbf{0}, \begin{pmatrix} 1 & \rho_e \\ \rho_e & 1 \end{pmatrix} \right).
\end{aligned}
$$

In this model, the continuous representative $z_{ij}$ is modeled as equal to an overall mean $\mu$ plus the additive row and column effects $a_i$ and $b_j$ plus the error term $e_{ij}$. The row and column effects of a node $i$ are potentially correlated with correlation $\rho_{ab}$. A dyadic correlation between $e_{ij}$ and $e_{ji}$ is also allowed. The parameter $\mu$ influences the density of the network $\mathbf{Y}$, and a higher $\mu$ results in a denser network. The random effect $a_i$ describes how node $i$ sends connections to others, i.e., sociability, while $b_i$ describes the level of connections node $i$ would receive, i.e., popularity. Higher $a_i$ or higher $b_i$ result in higher out- or in-degrees for node $i$. We are interested in estimating $\boldsymbol{\theta} = (\mu, \sigma_a^2, \sigma_b^2, \rho_{ab}, \rho_e)$ using composite likelihood estimation. In terms of choosing the size of node subsets $q$, we have the following lemma.

**Lemma 3.2.1.** *For binary SRM, triads are the minimal components that make composite likelihood estimation identifiable.*

*Proof.* First, using dyads as composite likelihood components does not make the parameters identifiable. For dyadic composite likelihood, the covariance matrix of the continuous representative $\mathbf{z}_{ij}$ is equal to

$$
\begin{pmatrix} \sigma_a^2 + \sigma_b^2 + 1 & 2\sigma_a\sigma_b\rho_{ab} + \rho_e \\ 2\sigma_a\sigma_b\rho_{ab} + \rho_e & \sigma_a^2 + \sigma_b^2 + 1 \end{pmatrix}.
$$

Based on this covariance matrix, we can see that dyadic pairs provide information to estimate $\sigma_a^2+\sigma_b^2$ and $2\sigma_a\sigma_b\rho_{ab}+\rho_e$, but not each parameter separately. This means that there exists at least two different parameter vectors $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^*$ that define the identical distributions $p(\mathbf{z}_{ij}|\boldsymbol{\theta})$ and $p(\mathbf{z}_{ij}|\boldsymbol{\theta}^*)$. Since $p(\mathbf{y}_{ij}|\boldsymbol{\theta})$ is the integral of $p(\mathbf{z}_{ij}|\boldsymbol{\theta})$ over a given quadrant, $p(\mathbf{y}_{ij}|\boldsymbol{\theta}) = p(\mathbf{y}_{ij}|\boldsymbol{\theta}^*)$, and thus $\tilde{\ell}^{(2)}(\boldsymbol{\theta}:\mathbf{y}) = \tilde{\ell}^{(2)}(\boldsymbol{\theta}^*:\mathbf{y})$ for all $\mathbf{y}$.

Second, we show that triadic composite likelihood for binary SRM is identifiable. For triad $(i,j,k)$, the marginal distribution of $\mathbf{y}_{ijk}$ is given by

$$\begin{aligned} \mathbf{y}_{ijk} &= 1(\mathbf{z}_{ijk}), \\ \mathbf{z}_{ijk} &\sim \mathrm{N}(\mu\mathbf{1}, \boldsymbol{\Sigma}(\boldsymbol{\theta})), \end{aligned} \tag{3.8}$$

where the covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ is equal to

$$\begin{pmatrix} \sigma_a^2+\sigma_b^2+1 & 2\sigma_a\sigma_b\rho_{ab}+\rho_e & \sigma_a\sigma_b\rho_{ab} & \sigma_b^2 & \sigma_a\sigma_b\rho_{ab} & \sigma_a^2 \\ 2\sigma_a\sigma_b\rho_{ab}+\rho_e & \sigma_a^2+\sigma_b^2+1 & \sigma_a^2 & \sigma_a\sigma_b\rho_{ab} & \sigma_b^2 & \sigma_a\sigma_b\rho_{ab} \\ \sigma_a\sigma_b\rho_{ab} & \sigma_a^2 & \sigma_a^2+\sigma_b^2+1 & 2\sigma_a\sigma_b\rho_{ab}+\rho_e & \sigma_a\sigma_b\rho_{ab} & \sigma_b^2 \\ \sigma_b^2 & \sigma_a\sigma_b\rho_{ab} & 2\sigma_a\sigma_b\rho_{ab}+\rho_e & \sigma_a^2+\sigma_b^2+1 & \sigma_a^2 & \sigma_a\sigma_b\rho_{ab} \\ \sigma_a\sigma_b\rho_{ab} & \sigma_b^2 & \sigma_a\sigma_b\rho_{ab} & \sigma_a^2 & \sigma_a^2+\sigma_b^2+1 & 2\sigma_a\sigma_b\rho_{ab}+\rho_e \\ \sigma_a^2 & \sigma_a\sigma_b\rho_{ab} & \sigma_b^2 & \sigma_a\sigma_b\rho_{ab} & 2\sigma_a\sigma_b\rho_{ab}+\rho_e & \sigma_a^2+\sigma_b^2+1 \end{pmatrix}.$$

Assume there are two parameter vectors $\boldsymbol{\theta} = (\mu, \sigma_a^2, \sigma_b^2, \rho_{ab}, \rho_e)$ and $\boldsymbol{\theta}^* = (\mu^*, \sigma_a^{*2}, \sigma_b^{*2}, \rho_{ab}^*, \rho_e^*)$ that result in the same composite likelihood, i.e., $\tilde{\ell}^{(3)}(\boldsymbol{\theta}:\mathbf{y}) = \tilde{\ell}^{(3)}(\boldsymbol{\theta}^*:\mathbf{y})$ for any $\mathbf{y}$, then we have $\sum_{t=0}^{63} n_t \log[\Pr(t(\mathbf{y}_{ijk})=t|\boldsymbol{\theta})] = \sum_{t=0}^{63} n_t \log[\Pr(t(\mathbf{y}_{ijk})=t|\boldsymbol{\theta}^*)]$, for any $n_t$'s. This means that $\Pr(t(\mathbf{y}_{ijk})=t|\boldsymbol{\theta}) = \Pr(t(\mathbf{y}_{ijk})=t|\boldsymbol{\theta}^*)$ for any $t$, which indicates that the probability of each quadrant of $p(\mathbf{z}_{ijk}|\boldsymbol{\theta})$ is identical to that of $p(\mathbf{z}_{ijk}|\boldsymbol{\theta}^*)$. To make this condition hold, we must have the distribution of $\mathbf{z}_{ijk}|\boldsymbol{\theta}$ and $\mathbf{z}_{ijk}|\boldsymbol{\theta}^*$ be identical, i.e., $\mu = \mu^*$ and $\boldsymbol{\Sigma}(\boldsymbol{\theta}) = \boldsymbol{\Sigma}(\boldsymbol{\theta}^*)$. The latter implies that each element in the covariance matrices is identical, i.e.,

$$\begin{cases} \sigma_a^2 = \sigma_a^{*2}, \\ \sigma_b^2 = \sigma_b^{*2}, \\ \sigma_a\sigma_b\rho_{ab} = \sigma_a^*\sigma_b^*\rho_{ab}^*, \\ 2\sigma_a\sigma_b\rho_{ab}+\rho_e = 2\sigma_a^*\sigma_b^*\rho_{ab}^*+\rho_e^*, \end{cases} \Rightarrow \begin{cases} \sigma_a^2 = \sigma_a^{*2}, \\ \sigma_b^2 = \sigma_b^{*2}, \\ \rho_{ab} = \rho_{ab}^*, \\ \rho_e = \rho_e^*. \end{cases}$$

This shows that the two vectors are identical. Therefore, using triads as composite likelihood components makes the parameters identifiable. □

Based on Lemma 3.2.1, we will estimate $\boldsymbol{\theta}$ by optimizing the triadic composite log-likelihood for binary SRM:

$$\tilde{\ell}^{(3)}(\boldsymbol{\theta} : \mathbf{y}) = \sum_{c=1}^{16} n_c \log \int_{S(\mathbf{y}_c)} (2\pi)^{-3} |\boldsymbol{\Sigma}(\boldsymbol{\theta})|^{-1/2} \exp\{-\frac{1}{2}(\mathbf{z} - \mu\mathbf{1})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta})(\mathbf{z} - \mu\mathbf{1})\} d\mathbf{z}. \quad (3.9)$$

### *3.3 Calculation of Gradients for Composite Likelihood*

One advantage of composite likelihood estimation is that it breaks the full likelihood, an integral of an $m(m-1)$-dimensional vector, to the summation of integrals of $q(q-1)$-dimensional vectors, with $q \ll m$. The gradients for the composite likelihood are thus composed of integrals of small dimensional vectors. These gradients can be approximated numerically, and then used to optimize the composite likelihood and obtain an estimate for the parameters in an exchangeable model for binary network data.

Applying the gradient methods for optimizing $\tilde{\ell}(\boldsymbol{\theta} : \mathbf{y})$ requires calculation of the gradients, which are the derivatives of the log likelihood with respect to the parameters:

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} \tilde{\ell}(\boldsymbol{\theta} : \mathbf{y}) &= \sum_{k=1}^{K} \nabla_{\boldsymbol{\theta}} \ell_k(\boldsymbol{\theta} : \mathbf{y}_k) \\
&= \sum_{k=1}^{K} \nabla_{\boldsymbol{\theta}} \log \left[ \int_{S(\mathbf{y}_k)} p(\mathbf{z}|\boldsymbol{\theta}) d\mathbf{z} \right] \\
&= \sum_{k=1}^{K} \frac{\nabla_{\boldsymbol{\theta}} \int_{S(\mathbf{y}_k)} p(\mathbf{z}|\boldsymbol{\theta}) d\mathbf{z}}{\int_{S(\mathbf{y}_k)} p(\mathbf{z}|\boldsymbol{\theta}) d\mathbf{z}}, \\
&= \sum_{k=1}^{K} \frac{\int_{S(\mathbf{y}_k)} \nabla_{\boldsymbol{\theta}} p(\mathbf{z}|\boldsymbol{\theta}) d\mathbf{z}}{\int_{S(\mathbf{y}_k)} p(\mathbf{z}|\boldsymbol{\theta}) d\mathbf{z}}.
\end{aligned}$$

Note that the derivative of the distribution of $\mathbf{z}$ can be decomposed into a product of $p(\mathbf{z}|\boldsymbol{\theta})$

and $\nabla_{\boldsymbol{\theta}} p(\mathbf{z}|\boldsymbol{\theta})$, and therefore

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} \tilde{\ell}(\boldsymbol{\theta} : \mathbf{y}) &= \sum_{k=1}^{K} \frac{\int_{S(\mathbf{y}_k)} \frac{\nabla_{\boldsymbol{\theta}} p(\mathbf{z}|\boldsymbol{\theta})}{p(\mathbf{z}|\boldsymbol{\theta})} p(\mathbf{z}|\boldsymbol{\theta}) d\mathbf{z}}{\int_{S(\mathbf{y}_k)} p(\mathbf{z}|\boldsymbol{\theta}) d\mathbf{z}} \\ &= \sum_{k=1}^{K} E\left[\frac{\nabla_{\boldsymbol{\theta}} p(\mathbf{z}|\boldsymbol{\theta})}{p(\mathbf{z}|\boldsymbol{\theta})} | \mathbf{y}_k\right] \\ &= \sum_{k=1}^{K} E\left[\nabla_{\boldsymbol{\theta}} \log\left(p(\mathbf{z}|\boldsymbol{\theta})\right) | \mathbf{y}_k\right],\end{aligned} \tag{3.10}$$

where the expectation is with respect to the conditional distribution of $\mathbf{z}$ given $\mathbf{y}_s$. Unless the model $p(\mathbf{z}|\boldsymbol{\theta})$ assumes that the elements of $\mathbf{z}$ are independent with each other, there will generally be no closed form expression for this expectation. One strategy would be to get a certain number of samples from the distribution of $\mathbf{z}$ given $\mathbf{y}$ and $\boldsymbol{\theta}$, and approximate this expectation by averaging over the values of $\nabla_{\boldsymbol{\theta}} \log[p(\mathbf{z}|\boldsymbol{\theta})]$ with the samples. However, directly sampling from the distribution is usually not feasible, so we consider Monte Carlo or Markov Chain Monte Carlo samples.

**Example 3.3.1.** *Triadic composite likelihood for binary SRM.*

Equation (3.9) gives the form of triadic composite log-likelihood for binary SRM. Denoting the parameters related to the covariance $\boldsymbol{\Sigma}$ as $\boldsymbol{\phi} = (\sigma_a^2, \sigma_b^2, \rho_{ab}, \rho_e)$, we have the gradients of the triadic composite log-likelihood based on Equation (3.10) as

$$\begin{aligned}\frac{\partial}{\partial \mu} \tilde{\ell}^{(3)}(\boldsymbol{\theta} : \mathbf{y}) &= \sum_{c=1}^{16} n_c \mathrm{E}\left[\boldsymbol{\Lambda}(\boldsymbol{\theta})(\mathbf{z} - \mu \mathbf{1}) | \mathbf{y}_c\right] \\ \frac{\partial}{\partial \phi_i} \tilde{\ell}^{(3)}(\boldsymbol{\theta} : \mathbf{y}) &= \frac{1}{2} \sum_{c=1}^{16} n_c \mathrm{tr}\left(\boldsymbol{\Sigma}(\boldsymbol{\theta}) \frac{\partial}{\partial \phi_i} \boldsymbol{\Lambda}(\boldsymbol{\theta})\right) \\ &\quad - \frac{1}{2} \sum_{c=1}^{16} n_c \mathrm{E}\left[(\mathbf{z} - \mu \mathbf{1})^T \frac{\partial}{\partial \phi_i} \boldsymbol{\Lambda}(\boldsymbol{\theta})(\mathbf{z} - \mu \mathbf{1}) | \mathbf{y}_c\right],\end{aligned} \tag{3.11}$$

where $\boldsymbol{\Lambda}(\boldsymbol{\theta}) = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta})$ is the precision matrix and the expectation $\mathrm{E}[\cdot|\mathbf{y}_c]$ is with respect to the 6-dimensional truncated multivariate normal distribution $p(\mathbf{z}|\mathbf{y}_c, \mu, \boldsymbol{\theta})$. In most cases, there are no closed form expressions for the gradients, which involves calculating the expectation of a truncated multivariate normal distribution.

To approximate the expectations, we need to get samples from the distribution of $\mathbf{z}$ given $\mathbf{y}$ and $\boldsymbol{\theta}$. Different sampling methods can be used including rejection sampling, importance sampling, Gibbs sampling and Hamiltonian Monte Carlo (HMC) [Pakman and Paninski, 2013], etc. Rejection sampling is performed by sampling from the distribution of $\mathbf{z}$ and only keeping the points in the restricted intervals determined by the sign of $\mathbf{y}$. It is easy to implement, but for high-dimensional $\mathbf{y}$, it will reject too many points and take a long time to simulate one $\mathbf{z}$ in the desired intervals. Both Gibbs sampling and Hamiltonian Monte Carlo provide Monte Carlo samples. The former works with the conditional distribution of one element at a time, and can be very slow if the dimension is large. The latter makes use of the Hamiltonian equations of motion such that the chain mixes faster and is more efficient than Gibbs sampling. However, its runtime depends on the number and the shape of the constraints, so if the size of $\mathbf{y}$ is large, the number of constraints will be accordingly large, and thus HMC is prohibitively time-consuming. When the network model is simple, importance sampling can be a useful tool to get independent samples. The shortcoming of this method is that it largely depends on the choice of the proposal distributions. When the dimension of $\mathbf{y}$ is large, all of those methods results in small effective sample size per unit time. Table 3.2 shows the effective sample sizes gained per second using those sampling methods (excluding rejection sampling since for large network size, it takes too long to generate one sample) for the SRM.

| SRM | $m = 5$ | $m = 10$ | $m = 25$ |
|---|---|---|---|
| Importance sampling | 11110.66 | 156.44 | 1.43 |
| Gibbs sampling | 1103.61 | 47.88 | 0.23 |
| HMC | 13775.73 | 736.62 | 0.37 |

Table 3.2: Effective sample size per second in SRM

Even with network size $m = 25$, the effect sample sizes per second for all the sampling methods are around 1. This means that for the full likelihood, it would not be feasible to

obtain samples for high-dimensional $\mathbf{z}$ given $\mathbf{y}$ and $\boldsymbol{\theta}$ and get approximations for its gradient. However, composite likelihood estimation does not suffer from this problem because for small dimensions, these sampling methods can provide a large enough effective sample size in a short period of time.

### *3.3.1 Gradient approximation of triadic composite likelihood*

Compared to the other methods of sampling, Gibbs sampling is easy to code and can be applied to general cases. In what follows, we use Gibbs sampling to generate an MC sample with which to approximate each gradient of the composite likelihood components.

For triadic composite likelihood, Equation (3.8) gives the model for each component. Sampling from the distribution of $\mathbf{z}_{ijk}$ given $\mathbf{y}_{ijk}$ means sampling from a truncated multivariate Gaussian distribution. Since direct sampling is not available for this distribution, we sample element by element. Note that the marginal distribution of the elements of $\mathbf{z}_{ijk}$ is not (truncated) Gaussian any more, but the conditional distribution of one element given all the others is. Therefore, we can use Gibbs sampling to iteratively generate samples from the elements of $\mathbf{z}_{ijk}$, which can be used to approximate the joint distribution of $\mathbf{z}_{ijk}$. The following algorithm summarizes the Gibbs sampling steps for getting samples of $\mathbf{z}_{ijk}$ given $\mathbf{y}_{ijk}$.

**Algorithm 3.3.1.** *Gibbs sampling from truncated multivariate Gaussian distribution for* $\mathbf{z}_{ijk}$ *given* $\mathbf{y}_{ijk}$.

1. Choose a starting point $\mathbf{z}_0 = (z_{ij}^{(0)}, z_{ji}^{(0)}, z_{jk}^{(0)}, z_{kj}^{(0)}, z_{ki}^{(0)}, z_{ik}^{(0)})$.

2. For $s = 1, \cdots, S$:

   2.1 The distribution of $z_{ij}^{(s)} | z_{ji}^{(s-1)}, z_{jk}^{(s-1)}, z_{kj}^{(s-1)}, z_{ki}^{(s-1)}, z_{ik}^{(s-1)}, y_{ij} \sim \mathrm{N}(\mu_{ij}, \sigma_{ij}^2)$, which is a univariate truncated Gaussian distribution, so we simulate

$$z_{ij}^{(s)} = \begin{cases} \Phi^{(-1)}(u), & u \sim \mathrm{U}(\Phi(-\frac{\mu_{ij}}{\sigma_{ij}}), 1), \quad \text{if } y_{ij} = 1, \\ \Phi^{(-1)}(u), & u \sim \mathrm{U}(0, \Phi(-\frac{\mu_{ij}}{\sigma_{ij}})), \quad \text{if } y_{ij} = 0. \end{cases}$$

2.2 Sample from $z_{ji}^{(s)}|z_{ij}^{(s)}, z_{jk}^{(s-1)}, z_{kj}^{(s-1)}, z_{ki}^{(s-1)}, z_{ik}^{(s-1)}, y_{ji}$.

2.3 Sample from $z_{jk}^{(s)}|z_{ij}^{(s)}, z_{ji}^{(s)}, z_{kj}^{(s-1)}, z_{ki}^{(s-1)}, z_{ik}^{(s-1)}, y_{jk}$.

2.4 Sample from $z_{kj}^{(s)}|z_{ij}^{(s)}, z_{ji}^{(s)}, z_{jk}^{(s)}, z_{ki}^{(s-1)}, z_{ik}^{(s-1)}, y_{kj}$.

2.5 Sample from $z_{ki}^{(s)}|z_{ij}^{(s)}, z_{ji}^{(s)}, z_{jk}^{(s)}, z_{kj}^{(s)}, z_{ik}^{(s-1)}, y_{ki}$.

2.6 Sample from $z_{ik}^{(s)}|z_{ij}^{(s)}, z_{ji}^{(s)}, z_{jk}^{(s)}, z_{kj}^{(s)}, z_{ki}^{(s-1)}, y_{ik}$.

Once the MC samples $\{\mathbf{z}^{(s)} = (z_{ij}^{(s)}, z_{ji}^{(s)}, z_{jk}^{(s)}, z_{kj}^{(s)}, z_{ki}^{(s)}, z_{ik}^{(s)})\}_{s=1}^{S}$ are generated, we use them to numerically approximate the gradient in Equation (3.11) as follows:

$$\begin{aligned}
\frac{\partial}{\partial \mu}\widehat{\tilde{\ell}^{(3)}(\boldsymbol{\theta} : \mathbf{y})} &= \sum_{c=1}^{16} \frac{n_c}{S} \sum_{s=1}^{S} \boldsymbol{\Lambda}(\boldsymbol{\theta})(\mathbf{z}^{(s)} - \mu \mathbf{1}), \\
\frac{\partial}{\partial \phi_i}\widehat{\tilde{\ell}^{(3)}(\boldsymbol{\theta} : \mathbf{y})} &= \frac{1}{2} \sum_{c=1}^{16} n_c \mathrm{tr}\left(\boldsymbol{\Sigma}(\boldsymbol{\theta}) \frac{\partial}{\partial \phi_i} \boldsymbol{\Lambda}(\boldsymbol{\theta})\right) \\
&\quad - \frac{1}{2} \sum_{c=1}^{16} \frac{n_c}{S} \sum_{s=1}^{S} (\mathbf{z}^{(s)} - \mu \mathbf{1})^T \frac{\partial}{\partial \phi_i} \boldsymbol{\Lambda}(\boldsymbol{\theta})(\mathbf{z}^{(s)} - \mu \mathbf{1}).,
\end{aligned}$$

## *3.4 Simulation Study*

One way to optimize the composite likelihood is to use the gradient method. In the previous section, we explained how to use a Gibbs sampler to approximate the gradient of a composite likelihood component given a parameter vector. Here, we will introduce the application of the gradient method on optimizing the triadic composite likelihood for SRM of binary network data using the approximated gradients.

### *3.4.1 Algorithm*

Since there is no closed form expression for the gradient, it is impossible to use it directly in a gradient ascent algorithm. Therefore, we consider a stochastic gradient ascent algorithm, which uses the stochastic approximation of the gradients. It is slightly different from the stochastic gradient ascent algorithm usually studied in the literature, which obtains stochastic gradient approximations by subsampling the data. Instead, our gradient approximations

are stochastic because they are obtained from an MCMC algorithm, for example, Algorithm 3.3.1. Bottou [1998] justifies our method. Given several regularity assumptions, the proof of the algorithm convergence relies on the unbiasedness of the gradient approximation, no matter the approximation is obtained by subsampling the data or sampling from an MCMC algorithm. Formally, the stochastic gradient ascent algorithm we propose to optimize composite likelihood (3.9) for $\hat{\boldsymbol{\theta}}$ is as follows:

**Algorithm 3.4.1.** *Stochastic gradient ascent algorithm used to obtain triadic composite likelihood estimates of parameters in binary SRM.*

0. Choose a starting point $\boldsymbol{\theta}_0$, an initial learning rate $\alpha_0$, a decay factor $\kappa \in (0.5, 1)$ for the learning rate, the number of MCMC samples $S$ and a tolerance $\epsilon$. Calculate the counts of graphs in each isomorphic group as $(n_1, \cdots, n_{16})$.

1. For $t = 1, 2, \cdots$ until the stopping criterion is met:

   1.1 For each category $c = 1, \cdots, 16$, generate $S$ MCMC samples from the conditional distribution of $\mathbf{z}$ given $\mathbf{y} = \mathbf{y}_c$ using Algorithm 3.3.1.

   1.2 Approximate the gradient at $\boldsymbol{\theta}_{t-1}$ as $\hat{\nabla}_{\boldsymbol{\theta}} \tilde{\ell}^{(3)}(\boldsymbol{\theta}_{t-1} : \mathbf{y}) = \sum_{c=1}^{16} n_c \hat{\nabla} \ell(\boldsymbol{\theta}_{t-1} : \mathbf{y}_c)$.

   1.3 Update the parameter as $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} + \frac{\alpha_0}{t^\kappa} \hat{\nabla}_{\boldsymbol{\theta}} \tilde{\ell}^{(3)}(\boldsymbol{\theta}_{t-1} : \mathbf{y})$. □

Since it is hard to get the exact value or even an approximation to the objective function, i.e., the composite likelihood, we propose using a stopping criterion that checks the change of gradients between two adjacent iterations. Denoting the parameter vector as $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_p)$, with $p = 5$ for SRM, and $g_i(\boldsymbol{\theta}) = \frac{\partial}{\partial \theta_i} \tilde{\ell}^{(3)}(\boldsymbol{\theta} : \mathbf{y})$ as the gradient with respect to the $i$-th parameter, $i = 1, \cdots, p$, we stop the iterations when the maximum gradient change among all the parameters is below the tolerance $\epsilon$, i.e.,

$$\max_{i=1,\cdots,p} \left| \frac{\hat{g}_i(\boldsymbol{\theta}_t) - \hat{g}_i(\boldsymbol{\theta}_{t-1})}{\hat{g}_i(\boldsymbol{\theta}_{t-1})} \right| < \epsilon.$$

One drawback of the stochastic gradient ascent algorithm 3.3.1 is the need to tune the learning rate parameters. It might take multiple attempts to figure out a reasonable combination of the base learning rate $\alpha_0$ and the decay factor $\kappa$ so that the chain converges fast and also has the capability to jump between local modes. As an alternative to tuning these two parameters, we consider using an adaptive algorithm called AdaGrad (short for adaptive gradient method) by replacing the decay of the learning rate $\frac{\alpha_0}{t^\kappa}$ to $\alpha \mathbf{G}_t^{-1/2}$, with $\mathbf{G}_t = \mathbf{G}_{t-1} + \hat{\nabla}_{\boldsymbol{\theta}} \tilde{\ell}^{(3)}(\boldsymbol{\theta}_t : \mathbf{y}) \hat{\nabla}_{\boldsymbol{\theta}} \tilde{\ell}^{(3)}(\boldsymbol{\theta}_t : \mathbf{y})^T$ and $\mathbf{G}_1 = \mathbf{I}$. The advantage of using AdaGrad is that it adjusts the learning rate adaptively as the algorithm proceeds, and we only need to tune the base learning rate $\alpha$.

Note that the parameters in the SRM for binary network data can not take all the values in $\mathbb{R}^p$. Specifically, the variance parameters $\sigma_a^2$ and $\sigma_b^2$ should always be positive and the correlation parameters $\rho_{ab}$ and $\rho_e$ need to stay between $-1$ and 1. However, Algorithm 3.3.1 does not put a restriction on the values of parameters and allows the parameter vector to be any element of $\mathbb{R}^p$. To keep the parameters within their allowed ranges, we consider the following transformation of the parameters:

$$\begin{aligned} \sigma_a^2 &= e^{\gamma_a}, \gamma_a \in \mathbb{R} \\ \sigma_b^2 &= e^{\gamma_b}, \gamma_b \in \mathbb{R} \\ \rho_{ab} &= \frac{e^{\eta_{ab}} - 1}{e^{\eta_{ab}} + 1}, \eta_{ab} \in \mathbb{R} \\ \rho_e &= \frac{e^{\eta_e} - 1}{e^{\eta_e} + 1}, \eta_e \in \mathbb{R}. \end{aligned} \tag{3.12}$$

The mapping between $\boldsymbol{\theta} = (\mu, \sigma_a^2, \sigma_b^2, \rho_{ab}, \rho_e)$ and $\boldsymbol{\delta} = (\mu, \gamma_a, \gamma_b, \eta_{ab}, \eta_e)$ is strictly monotonic, which means that optimizing the composite likelihood with respect to $\boldsymbol{\theta}$ is equivalent to optimizing with respect to $\boldsymbol{\delta}$. Making use of the chain rule, we can easily calculate the gradients based on Equations (3.11). The gradients with respect to $\boldsymbol{\delta}$ are thus

$$
\begin{aligned}
\frac{\partial}{\partial \mu}\tilde{\ell}^{(3)}(\boldsymbol{\delta}:\mathbf{y}) &= \sum_{c=1}^{16} n_c \mathrm{E}\left[\boldsymbol{\Lambda}(\boldsymbol{\theta}(\boldsymbol{\delta}))(\mathbf{z}-\mu\mathbf{1})|\mathbf{y}_c\right], \\
\frac{\partial}{\partial \gamma_a}\tilde{\ell}^{(3)}(\boldsymbol{\delta}:\mathbf{y}) &= \frac{\partial}{\partial \sigma_a^2}\tilde{\ell}^{(3)}(\boldsymbol{\theta}(\boldsymbol{\delta}):\mathbf{y})\frac{\partial \sigma_a^2}{\partial \gamma_a} \\
&= e^{\gamma_a}\left\{\frac{1}{2}\sum_{c=1}^{16} n_c \mathrm{tr}\left(\boldsymbol{\Sigma}(\boldsymbol{\theta}(\boldsymbol{\delta}))\frac{\partial}{\partial \sigma_a^2}\boldsymbol{\Lambda}(\boldsymbol{\theta}(\boldsymbol{\delta}))\right)\right. \\
&\quad \left. -\frac{1}{2}\sum_{c=1}^{16} n_c \mathrm{E}\left[(\mathbf{z}-\mu\mathbf{1})^T\frac{\partial}{\partial \sigma_a^2}\boldsymbol{\Lambda}(\boldsymbol{\theta}(\boldsymbol{\delta}))(\mathbf{z}-\mu\mathbf{1})|\mathbf{y}_c\right]\right\}, \\
\frac{\partial}{\partial \gamma_b}\tilde{\ell}^{(3)}(\boldsymbol{\delta}:\mathbf{y}) &= \frac{\partial}{\partial \sigma_b^2}\tilde{\ell}^{(3)}(\boldsymbol{\theta}(\boldsymbol{\delta}):\mathbf{y})\frac{\partial \sigma_b^2}{\partial \gamma_b} \\
&= e^{\gamma_b}\left\{\frac{1}{2}\sum_{c=1}^{16} n_c \mathrm{tr}\left(\boldsymbol{\Sigma}(\boldsymbol{\theta}(\boldsymbol{\delta}))\frac{\partial}{\partial \sigma_b^2}\boldsymbol{\Lambda}(\boldsymbol{\theta}(\boldsymbol{\delta}))\right)\right. \\
&\quad \left. -\frac{1}{2}\sum_{c=1}^{16} n_c \mathrm{E}\left[(\mathbf{z}-\mu\mathbf{1})^T\frac{\partial}{\partial \sigma_b^2}\boldsymbol{\Lambda}(\boldsymbol{\theta}(\boldsymbol{\delta}))(\mathbf{z}-\mu\mathbf{1})|\mathbf{y}_c\right]\right\}, \qquad (3.13) \\
\frac{\partial}{\partial \eta_{ab}}\tilde{\ell}^{(3)}(\boldsymbol{\delta}:\mathbf{y}) &= \frac{\partial}{\partial \rho_{ab}}\tilde{\ell}^{(3)}(\boldsymbol{\theta}(\boldsymbol{\delta}):\mathbf{y})\frac{\partial \rho_{ab}}{\partial \eta_{ab}} \\
&= \frac{2e^{\eta_{ab}}}{(e^{\eta_{ab}}+1)^2}\left\{\frac{1}{2}\sum_{c=1}^{16} n_c \mathrm{tr}\left(\boldsymbol{\Sigma}(\boldsymbol{\theta}(\boldsymbol{\delta}))\frac{\partial}{\partial \rho_{ab}}\boldsymbol{\Lambda}(\boldsymbol{\theta}(\boldsymbol{\delta}))\right)\right. \\
&\quad \left. -\frac{1}{2}\sum_{c=1}^{16} n_c \mathrm{E}\left[(\mathbf{z}-\mu\mathbf{1})^T\frac{\partial}{\partial \rho_{ab}}\boldsymbol{\Lambda}(\boldsymbol{\theta}(\boldsymbol{\delta}))(\mathbf{z}-\mu\mathbf{1})|\mathbf{y}_c\right]\right\}, \\
\frac{\partial}{\partial \eta_e}\tilde{\ell}^{(3)}(\boldsymbol{\delta}:\mathbf{y}) &= \frac{\partial}{\partial \rho_e}\tilde{\ell}^{(3)}(\boldsymbol{\theta}(\boldsymbol{\delta}):\mathbf{y})\frac{\partial \rho_e}{\partial \eta_e} \\
&= \frac{2e^{\eta_e}}{(e^{\eta_e}+1)^2}\left\{\frac{1}{2}\sum_{c=1}^{16} n_c \mathrm{tr}\left(\boldsymbol{\Sigma}(\boldsymbol{\theta}(\boldsymbol{\delta}))\frac{\partial}{\partial \rho_e}\boldsymbol{\Lambda}(\boldsymbol{\theta}(\boldsymbol{\delta}))\right)\right. \\
&\quad \left. -\frac{1}{2}\sum_{c=1}^{16} n_c \mathrm{E}\left[(\mathbf{z}-\mu\mathbf{1})^T\frac{\partial}{\partial \rho_e}\boldsymbol{\Lambda}(\boldsymbol{\theta}(\boldsymbol{\delta}))(\mathbf{z}-\mu\mathbf{1})|\mathbf{y}_c\right]\right\}.
\end{aligned}
$$

Since the MC samples are used to approximate the gradient for the AdaGrad algorithm, we name this algorithm as MC-AdaGrad. To summarize, the MC-AdaGrad algorithm for optimizing the triadic composite likelihood of binary SRM is as follows.

**Algorithm 3.4.2.** *MC-AdaGrad algorithm used to get triadic composite likelihood estimates*

*of parameters in binary SRM.*

0. Choose a starting point $\boldsymbol{\delta}_0$, a base learning rate $\alpha$, the number of MCMC samples $S$ and a tolerance $\epsilon$. Calculate the counts of graphs in each isomorphic group as $(n_1, \cdots, n_{16})$.

1. While $\max_{i=1,\cdots,p} \left| \frac{\hat{g}_i(\boldsymbol{\delta}_t) - \hat{g}_i(\boldsymbol{\delta}_{t-1})}{\hat{g}_i(\boldsymbol{\delta}_{t-1})} \right| < \epsilon$:

   1.1 For each category $c = 1, \cdots, 16$, generate $S$ MCMC samples from the conditional distribution of $\mathbf{z}$ given $\mathbf{y} = \mathbf{y}_c$ using Algorithm 3.3.1.

   1.2 Approximate the gradient at $\boldsymbol{\delta}_{t-1}$ in Equation(3.13) as $\hat{\nabla}_\delta \tilde{\ell}^{(3)}(\boldsymbol{\delta}_{t-1} : \mathbf{y})$.

   1.3 Update the parameter as $\boldsymbol{\delta}_t = \boldsymbol{\delta}_{t-1} + \alpha \mathbf{G}_t^{-1/2} \tilde{\ell}^{(3)}(\boldsymbol{\delta}_{t-1} : \mathbf{y})$, with $\mathbf{G}_t = \mathbf{G}_{t-1} + \hat{\nabla}_\delta \tilde{\ell}^{(3)}(\boldsymbol{\delta}_t : \mathbf{y}) \hat{\nabla}_\delta \tilde{\ell}^{(3)}(\boldsymbol{\delta}_t : \mathbf{y})^T$ and $\mathbf{G}_1 = \mathbf{I}$.

2. Transform back using Equation (3.12) to get the estimate as $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}(\hat{\boldsymbol{\delta}})$.

### *3.4.2 Simulation scheme*

The goal of this simulation study is to examine the performance of triadic composite likelihood estimation for the binary probit SRM. We implement the composite likelihood estimation algorithm on networks of different sizes, and compare to other methods.

To examine how the procedure scales with the network size, we generate networks with node sizes of $m = 50, 200, 500$ using the same values of parameters $\boldsymbol{\theta} = (\mu, \sigma_a^2, \sigma_b^2, \rho_{ab}, \rho_e) = (-1.50, 0.29, 0.34, 0.28, 0.63)$. For each node size, we generate 20 datasets independently from the SRM given by Equation (3.1), and for each dataset we estimate the parameter $\boldsymbol{\theta}$ using three methods: composite likelihood estimation, Laplace approximation and fully Bayesian estimation using MCMC.

**Method 1 (Composite likelihood estimation):** The MC-AdaGrad algorithm 3.4.2 is used to optimize the objective with the base learning rate $\alpha = 3$, the number of MCMC samples $S = 2000$ after a burnin period of 200 iterations and a tolerance $\epsilon = \sqrt{1e-5}$. In

order to get a good starting value that is close to the optimum, we choose $\boldsymbol{\delta}_0$ as follows. First, we impute the continuous representative $\mathbf{Z}_0$ of the binary network $\mathbf{Y}$ using the $z$-score of the rank. Denoting the number of 1's in $\mathbf{Y}$ is $M_1$ and the number of 0's is $M_0 = m(m-1) - M_1$, we define the $Z$-score as

$$z_{0,ij} = \begin{cases} \Phi^{-1}\left(\frac{M_0+1}{2[m(m+1)+1]}\right), & \text{if } y_{ij} = 0, \\ \Phi^{-1}\left(\frac{M_0+M_1+1}{2[m(m+1)+1]}\right), & \text{if } y_{ij} = 1, \end{cases} \tag{3.14}$$

where $\Phi(\cdot)$ stands for the cumulative distribution function (CDF) of univariate Gaussian distribution. Second, the starting value of $\mu_0$ is calculated using the mean of $\mathbf{Z}_0$, the row effects $a_{0,i}$ takes the row means of $\mathbf{Z}_0 - \mu_0 \mathbf{1}\mathbf{1}^T$ and $b_{0,i}$ is from its column means. The starting value of $\rho_{0,ab}$ is then computed as the correlation between the vectors $\mathbf{a}$ and $\mathbf{b}$. Third, we take the dyadic pairs $(e_{0,ij}, e_{0,ji})$ from the residual matrix $\mathbf{E}_0 = \mathbf{Z}_0 - \mu_0 \mathbf{1}\mathbf{1}^T - \mathbf{a}_0\mathbf{b}_0^T$ and calculate its correlation as $\rho_{0,e}$. Fourth, since we require the variance of the error terms to be 1 for identifiability, we need to scale $\mu_0$ and the standard deviance terms $\sigma_{0,a}$, $\sigma_{0,b}$ with the standard deviance of $\mathbf{e}_0$. Finally, using the inverse transformation of Equations (3.12) to get the starting point of $\boldsymbol{\delta}_0$. Basically, we are obtaining our starting values using moment estimates based on a rough estimate of the matrix $\mathbf{Z}$.

**Method 2 (Laplace approximation):** For generalized linear mixed models (GLMM), Laplace approximation is often used to approximate the integrals of the marginal distributions. In general, a probit GLMM of binary networks looks like the following,

$$\begin{aligned} y_{ij} &= 1(z_{ij} > 0),\ i, j \in \{1, \cdots, m\},\ i \neq j \\ z_{ij} &= w_{s_{ij}} + e_{ij},\ s_{ij} \in \{1, \cdots, r\} \\ \mathbf{w} &\sim \mathrm{N}_r(\mu\mathbf{1}, \boldsymbol{\Sigma}(\boldsymbol{\theta})), \\ \mathbf{e} &\sim \mathrm{N}(0, \mathbf{I}), \end{aligned}$$

with $\mathbf{w}$ representing some $r$-dimensional random effects. We can write the marginal distribution of $\mathbf{y}$ as

$$\begin{aligned} p(\mathbf{y}|\mu, \boldsymbol{\theta}) &= \int_{S(\mathbf{y})} p(\mathbf{z}|\mu, \boldsymbol{\theta}) d\mathbf{z}, \\ &= \int_{\mathbb{R}^r} p(\mathbf{y}|\mathbf{w}) p(\mathbf{w}|\mu, \boldsymbol{\theta}) d\mathbf{w}. \end{aligned}$$

The Laplace approximation makes use of the mode and curvature of $\log\left[p(\mathbf{y}|\mathbf{w})p(\mathbf{w}|\mu, \boldsymbol{\theta})\right]$ and approximates it with a quadratic form, which happens to be a penalized weighted residual sum of squares: $\|\mathbf{A}_1(\mathbf{z}-\mathbf{w})\|^2 + \|\mathbf{A}_2(\mathbf{w}-\mu\mathbf{1})\|^2$, where $\mathbf{A}_1$ and $\mathbf{A}_2$ represent weight matrices [Bates et al., 2014]. For the SRM, the application of this Laplace approximation is not straightforward because the error terms $e_{ij}$ and $e_{ji}$ are not independent. Therefore, we define a random effect $d_{(ij)}$ representing the additive effect of the dyadic pair, with $d_{(ij)} \equiv d_{(ji)}$, $\text{var}(d_{(ij)}) = \rho_e$ and $d_{(ij)} \perp d_{(kl)}$ if $(ij) \neq (kl)$. The SRM (3.1) is then equivalent to the following.

$$\begin{aligned} y_{ij} &= 1(z_{ij} > 0), \ i, j \in \{1, \cdots, m\}, \ i \neq j \\ z_{ij} &= \mu + a_i + b_j + d_{(ij)} + e_{ij}, \\ \begin{pmatrix} a_1 \\ b_1 \end{pmatrix}, \cdots, \begin{pmatrix} a_m \\ b_m \end{pmatrix} &\overset{i.i.d.}{\sim} \text{N}_2\left(\mathbf{0}, \begin{pmatrix} \sigma_a^2 & \rho_{ab}\sigma_a\sigma_b \\ \rho_{ab}\sigma_a\sigma_b & \sigma_b^2 \end{pmatrix}\right), \\ d_{(ij)} &\sim \text{N}(0, \rho_e), \\ e_{ij} &\sim \text{N}(0, 1-\rho_e). \end{aligned} \tag{3.15}$$

The Laplace approximation can now be applied to Model (3.15) considering $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{d}$ as random effects. Several tools have been developed in R to perform such Laplace approximations, including lme4 [Bates et al., 2014] and INLA [Rue et al., 2009]. We apply lme4 on the simulated datasets to get estimates of $\boldsymbol{\theta}$.

**Method 3 (Fully Bayesian estimation using MCMC):** The idea of this method is to work on the full conditionals of the parameters as well as the random effects or latent factors given certain priors. We use MCMC to iteratively sample from the full conditionals.

These samples are used to approximate the posterior distributions, and provide posterior mean estimates. To estimate $\boldsymbol{\theta}$ in the SRM of binary networks, we use Gibbs sampling to iteratively sample from the full conditionals of $\mathbf{z}$, $\mu$, $\mathbf{a}$, $\mathbf{b}$, $(\sigma_a^2, \sigma_b^2, \rho_{ab})$ and $\rho_e$. The posterior mean provides a point estimate for $\boldsymbol{\theta}$ and the variance of the posterior distribution describes the uncertainty of the estimation. The amen package in R uses this Bayesian methods for additive and multiplicative effects model for various types of networks. It can be applied to SRM of binary network directly and we will use it to do Bayesian estimation for $\boldsymbol{\theta}$.

### *3.4.3 Comparison*

For the Laplace approximation, we use a simpler SRM model to illustrate the drawbacks of this method. Setting $\rho_{ab} \equiv 0$ and $(\mu, \sigma_a^2, \sigma_b^2, \rho_e) = (0, 1, 2, 0.5)$, we generate 100 datasets and estimate the parameters using Method 2 (Laplace approximation) on each dataset. Figure 3.2 shows the results of the estimation. Each set of points represents the parameter estimation over 100 datasets. As we can see, the estimation of $\sigma_a^2$ and $\sigma_b^2$ is good as the points jump around the truth across different simulated data. However, the estimation of parameter $\rho_e$ is off as none of the datasets provides estimate close to the truth. The Laplace approximation fails in providing reasonable estimates for the probit SRM. We speculate that this failure is because when estimating the dyadic correlation $\rho_e$, the Laplace approximation relies on estimates of the random effects $\{d_{(ij)}\}$. However, only two data points $y_{ij}$ and $y_{ji}$ provide information about $d_{(ij)}$.

Moreover, there are two additional disadvantages of the Laplace approximation method. First, it depends on the form of a random effects model. As we will discuss in Chapter 4, some random effects models cannot be written as a combination of fixed effect, random effect and an error term with identity covariance matrix. Second, for the SRM as Model (3.15), this equivalent form only works for positive dyadic correlation $\rho_e > 0$ because it is modeled as the variance of a random effect $d_{(ij)}$. It is then impossible to model the case for $\rho_e \leq 0$ with random effects, which means the application of the Laplace approximation is limited.

Method 1 (composite likelihood estimation) and Method 3 (Bayes estimates using MCMC)

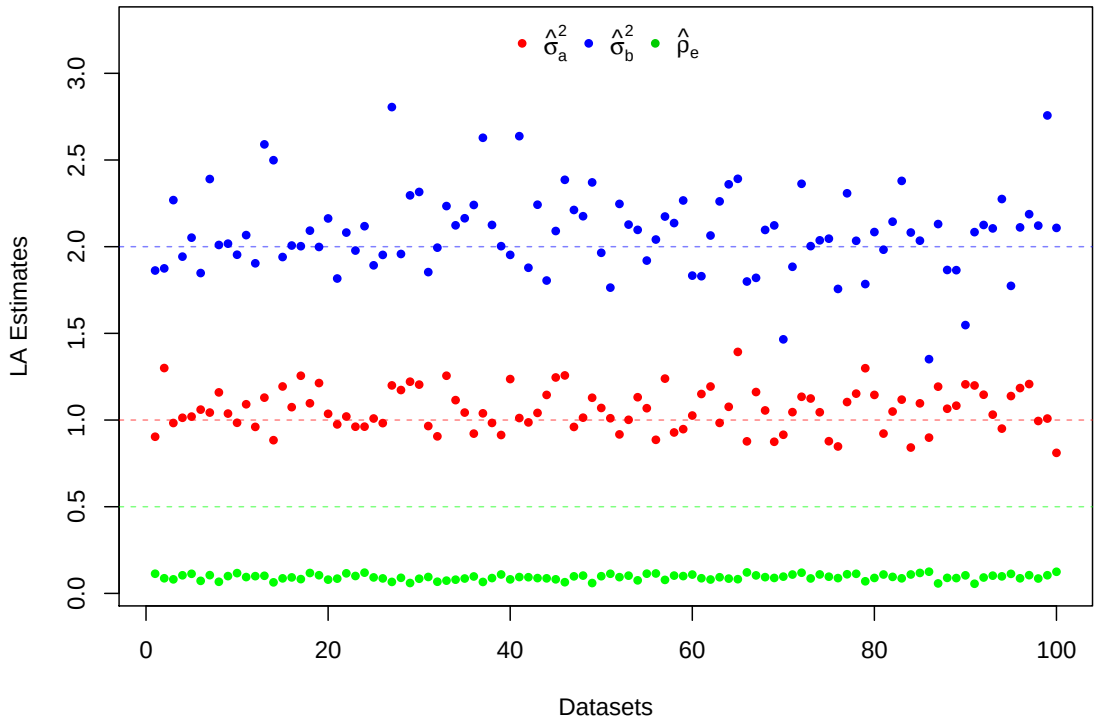


Figure 3.2: Estimation of SRM parameters using Laplace approximation. The points on the plot represent the estimates for each parameter across 100 datasets.

do not suffer from those drawbacks and can be applied to various kind of models of binary networks, beyond the SRM As we know, the Bayesian methods provide consistent estimates for the probit models of binary networks. We then compare the composite likelihood estimates from the 20 datasets generated with the Bayesian method using the amen package in R. If the two results are close to each other, then this suggest that the composite likelihood estimation provides reasonable estimates. Figure 3.3 presents the box plots of the parameter estimates across the 20 datasets under different network dimensions. The orange horizontal lines stand for the true values of the parameters.

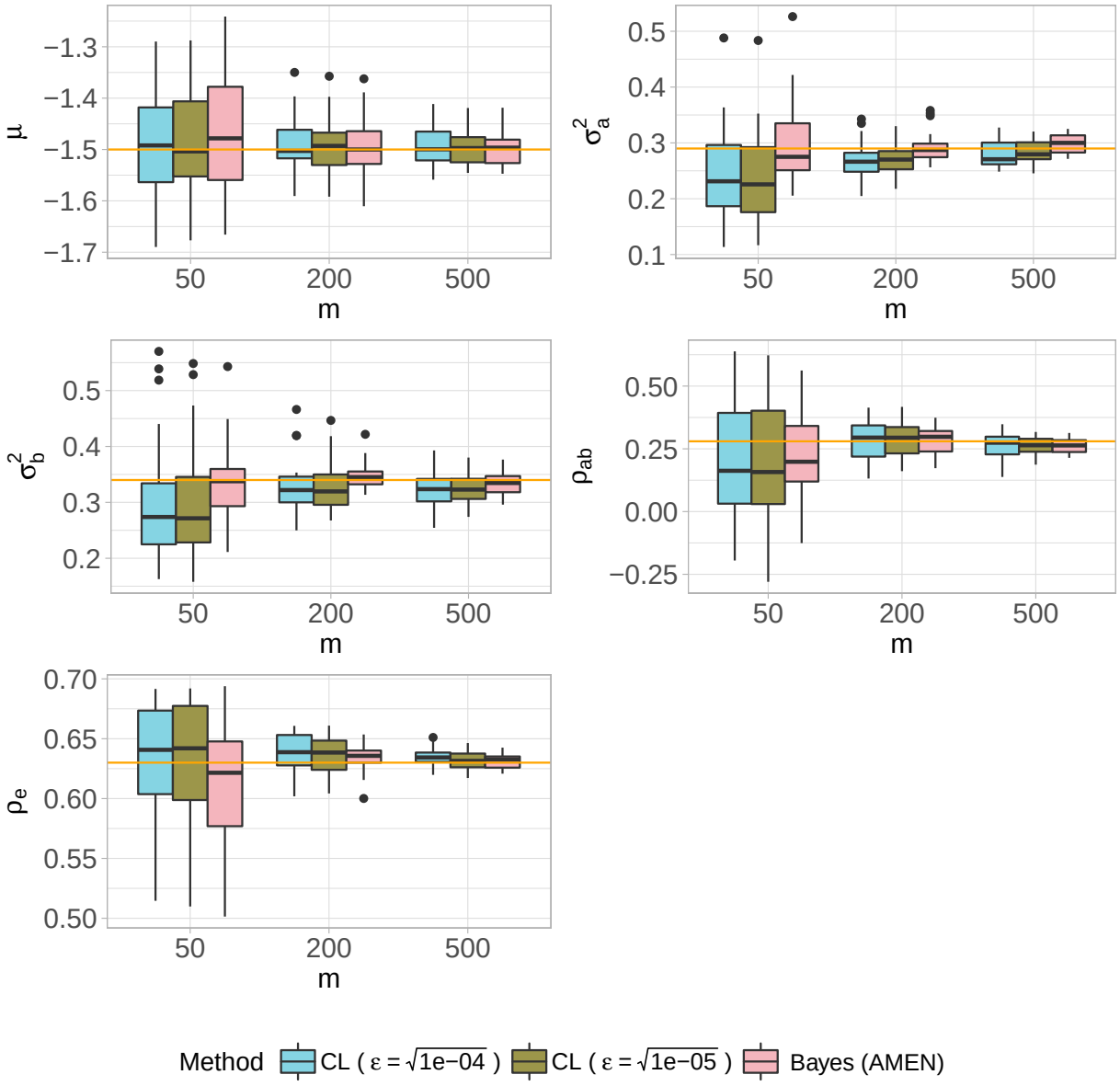


Figure 3.3: Comparison between the composite likelihood estimates and the full Bayesian method using MCMC. The box plots show the distribution of estimates across the 20 datasets. The orange line represents the true values of parameters.

As the network size increases, the variance of the composite likelihood estimates around the true values decreases. This means that the triadic composite likelihood estimates of the SRM for binary network data are potentially consistent. When the network size is small, we can see that the composite likelihood estimates are biased. This is because the composite likelihood estimation only uses triads and for small networks, the number of triads is too small to provide enough information. In this case, we suggest using Bayesian methods such as MCMC to estimate the parameters. However, when $m$ is large, say $m = 500$, using composite likelihood estimation can be a reasonable and scalable alternative to the Bayesian methods since the run time of MCMC increases as $m$ increases, while the complexity of composite likelihood is almost constant. Moreover, when performing MCMC, we need to sample $m \times m$ matrices, which may be impractical when $m$ is large. While for composite likelihood, we only need perform calculations on small matrices (say $6 \times 6$ matrices for triadic composite likelihoods), which has shorter run time and also helps avoid the problem caused by system precisions.

One thing to be noted is that in Figure 3.3, we show the composite likelihood estimates under different tolerances, $\epsilon = \sqrt{1e-4}$ and $\epsilon = \sqrt{1e-5}$. A smaller tolerance means more iterations before convergence in the MC-AdaGrad algorithm. From the plots, we can see that the differences in results between the two tolerances are small. This suggests we use the larger one $\epsilon = \sqrt{1e-4}$, which results in about 10 times fewer iterations, i.e., the run time can be about 10 times shorter.

### *3.5 Discussion*

In this chapter, we introduced a composite likelihood estimation approach to exchangeable network models, with a specific application to the probit SRM. Since the composite likelihood is a summation over multiple likelihood of small subsets, we can use a sampling method to estimate the gradient for the composite likelihood and then use a gradient method to optimize it. In particular, we proposed a triadic composite likelihood that works on all triads of networks. Given exchangeability, the triadic composite likelihood can be written

into a summation of only 16 components. This makes the optimization problem simpler and the algorithm more scalable, i.e., the computation time is fixed with the size of the networks. We compared its performance with two other methods, the Laplace approximation and the Bayes estimates using MCMC, on the probit SRM. The Laplace approximation relies on the form of the latent model being able to be decomposed into the sum of random effects and the current software does not support to specify a structure of the covariance matrix of the error terms in probit models. The Laplace approximation fails to accurately estimate the dyadic correlation in the SRM due to its mechanism in imputing random effects first and then estimating the parameters based on the imputations. The performance of the composite likelihood is closer to that of the Bayes estimates of the full likelihood using MCMC, which provide consistent estimates, while composite likelihood estimation is much more scalable and its run time does not depend on the size of the network. Therefore, when estimating the parameters in exchangeable models of large networks, we suggest using composite likelihood estimation to get reasonable estimates in less time.

However, there are cases when triadic composite likelihood estimation may provide poor performance. For example in the next chapter, we will introduce the additive and multiplicative effects models where the latent models of $z_{ij}$ are not Gaussian. In this case, using only the information from triads is not sufficient. We will use a higher order composite likelihood based on tetrads (sets of four nodes), pentads (sets of five nodes) and so on. This makes composite likelihood estimation capable of providing estimates for a larger variety of models. The challenge is that for higher orders, it is infeasible to enumerate all the isomorphic subgraphs for the exchangeable network models. Similarly, in Chapter 5, we will discuss the estimation problem for non-exchangeable models, e.g., when considering nodal or dyadic covariates in the models. In both cases, we can no longer write the composite likelihood into a summation of a small number of distributions of subgraphs, and therefore, we need to acquire a new stochastic algorithm to optimize the composite likelihood.

# Chapter 4

# COMPOSITE LIKELIHOOD ESTIMATION OF BINARY EXCHANGEABLE LATENT FACTOR NETWORK MODELS

## *4.1 Overview*

In the previous chapter we introduced a scalable method to estimate the parameters in an exchangeable model for binary network data by optimizing a composite likelihood. We used the SRM as an example to show the advantage of using a triadic composite likelihood, which reduced the optimization objective function to a sum of 16 terms, with each term being associated with an integral over a 6-dimensional vector. Because the dimension is low, we are able to use Gibbs sampling to obtain an accurate approximation of the gradients of the 16 terms and thus the gradient of the composite likelihood. These gradients can be used in a stochastic optimization algorithm to obtain the maximum composite likelihood estimate (MCLE).

However, the class of exchangeable network models contains not only the SRM but also a larger variety of other models. For example, the additive and multiplicative effects (AME) model extends the SRM by including a multiplicative term $\mathbf{u}_i^T\mathbf{v}_j$ when modeling the connection between node $i$ and node $j$. The terms $\mathbf{u}_i$ and $\mathbf{v}_i$ are $r$-dimensional latent factors. The former describes some latent characteristics that drive node $i$ to send connections to others, and the latter describes the latent characteristics of node $i$ that attract connections from others. We can formally write the AME model as follows.

$$\begin{aligned} y_{ij} &= 1(z_{ij} > 0),\ i, j \in \{1, \cdots, m\},\ i \neq j \\ z_{ij} &= \mu + a_i + b_j + \mathbf{u}_i^T\mathbf{v}_j + e_{ij}, \end{aligned} \tag{4.1}$$

with

$$\begin{pmatrix} a_1 \\ b_1 \end{pmatrix}, \cdots, \begin{pmatrix} a_m \\ b_m \end{pmatrix} \overset{i.i.d.}{\sim} N_2\left(\mathbf{0}, \begin{pmatrix} \sigma_a^2 & \rho_{ab}\sigma_a\sigma_b \\ \rho_{ab}\sigma_a\sigma_b & \sigma_b^2 \end{pmatrix}\right),$$
$$\begin{pmatrix} \mathbf{u}_i \\ \mathbf{v}_i \end{pmatrix}, \cdots, \begin{pmatrix} \mathbf{u}_m \\ \mathbf{v}_m \end{pmatrix} \overset{i.i.d.}{\sim} N_{2r}\left(\mathbf{0}, \mathbf{I}_2 \otimes \mathrm{diag}(\tau_1^2, \cdots, \tau_r^2)\right),$$
$$\left\{ \begin{pmatrix} e_{ij} \\ e_{ji} \end{pmatrix} \right\}_{i \neq j} \overset{i.i.d.}{\sim} N_2\left(\mathbf{0}, \begin{pmatrix} 1 & \rho_e \\ \rho_e & 1 \end{pmatrix}\right).$$

The parameters to be estimated are $\boldsymbol{\theta} = (\mu, \sigma_a^2, \sigma_b^2, \rho_{ab}, \rho_e, \tau_1^2, \cdots, \tau_r^2)$. The model is invariant under the permutation of the $r$ latent factors, and therefore, we assume $\tau_1^2 \leq \cdots \leq \tau_r^2$. Similar to the SRM, the marginal distribution of Model (4.1) is an intractable integral, so it is impractical to optimize the full likelihood to obtain the maximum likelihood estimate. So we consider using composite likelihood to estimate $\boldsymbol{\theta}$. However, it is not appropriate to directly apply the triadic composite likelihood we defined in Chapter 3 for the following two reasons: First, the marginal distribution of $z_{ij}$, unconditional on the random effects $\mathbf{u}_i$ and $\mathbf{v}_j$, is no longer Gaussian. This makes it difficult to generate samples of $\mathbf{z}$ given $\mathbf{y}$ and approximate the gradients using those samples. Second, since the AME model has a more complicated structure than the SRM and more parameters to estimate, a triadic composite likelihood may not carry enough information to provide accurate parameter estimates. Therefore, a higher order composite likelihood, i.e., a composite likelihood with larger subsets of networks, is required.

In this chapter, we will discuss the solution for the two problems addressed above. In Section 4.2, we will derive the form of the gradients for non-Gaussian random effects models and introduce a new Gibbs sampling algorithm by generating samples from the joint distribution of $(\mathbf{z}, \mathbf{u}, \mathbf{v})$ given $\mathbf{y}$, where $\mathbf{y}$, $\mathbf{z}$, $\mathbf{u}$ and $\mathbf{v}$ are the vectorizations of the network $\mathbf{Y}$, the continuous network representation $\mathbf{Z}$ and the $m \times r$ matrices of the random effects $\mathbf{U}$, $\mathbf{V}$. Then we use those samples to approximate the gradients. For the second problem above, we will show by simulation that triadic composite likelihood estimation is not suffi-

cient to the AME model and thus in Section 4.3, a $q$-node composite likelihood estimation will be discussed by extending the components from triads to groups of $q$ nodes. In this case, enumerating isomorphic subgraphs is infeasible, which makes it impossible to write the composite likelihood as a weighted sum of a small number of marginal distributions of subgraphs. Therefore, a stochastic algorithm to optimize the $q$-node composite likelihood will be proposed. In Section 4.4, a simulation study and a real case study will be carried out to illustrate the performance of composite likelihood estimates and compare them to full Bayes estimates using MCMC.

## *4.2 Gradient Approximation for Non-Gaussian Models*

Recall that in Chapter 3, we used Gibbs sampling to obtain samples of the distribution $\mathbf{z}$ given $\mathbf{y}$, which was a truncated multivariate normal distribution, and then we numerically approximated the gradient of the triadic composite likelihood for the binary SRM using those samples. This replied on the normality of the marginal distribution of $\mathbf{z}$, integrated over all node and dyad level random effects. In the SRM, the $z_{ij}$ could be represented in terms of a sum of normally distributed random effects, and so the model, integrated over these random effects, could be written as

$$\begin{aligned} \mathbf{y} &= 1(\mathbf{z} > 0), \\ \mathbf{z} &\sim N_{m^2}\left(\mu\mathbf{1}, \boldsymbol{\Sigma}(\boldsymbol{\phi})\right). \end{aligned} \tag{4.2}$$

The likelihood is an integral of multivariate normal distribution over truncated intervals, i.e.,

$$\ell\left(\boldsymbol{\theta} : \mathbf{y}\right) = \int_{S(\mathbf{y})} p(\mathbf{z}|\boldsymbol{\theta}) d\mathbf{z},$$

where $\boldsymbol{\theta} = (\mu, \boldsymbol{\phi})$. The gradients of the likelihood are

$$\begin{aligned} \frac{\partial}{\partial \mu}\ell(\boldsymbol{\theta} : \mathbf{y}) &= \mathrm{E}\left[\boldsymbol{\Lambda}(\boldsymbol{\phi})(\mathbf{z} - \mu\mathbf{1})|\mathbf{y}\right], \\ \frac{\partial}{\partial \phi_i}\ell(\boldsymbol{\theta} : \mathbf{y}) &= \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}(\boldsymbol{\phi})\frac{\partial}{\partial \phi_i}\boldsymbol{\Lambda}(\boldsymbol{\phi})\right) - \frac{1}{2}\mathrm{E}\left[(\mathbf{z} - \mu\mathbf{1})^T \frac{\partial}{\partial \phi_i}\boldsymbol{\Lambda}(\boldsymbol{\phi})(\mathbf{z} - \mu\mathbf{1})|\mathbf{y}\right], \end{aligned} \tag{4.3}$$

where $\boldsymbol{\Lambda}(\boldsymbol{\phi}) = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\phi})$ is the precision matrix and the expectation $\mathrm{E}[\cdot|\mathbf{y}]$ is with respect to the truncated multivariate normal distribution $p(\mathbf{z}|\mathbf{y}, \boldsymbol{\theta})$. This explains why we use the

samples from $p(\mathbf{z}|\mathbf{y},\boldsymbol{\theta})$ to approximate the gradients. The normality of $\mathbf{z}$ makes the Gibbs sampling straightforward.

For AME models, although $\mathbf{u}_i$ and $\mathbf{v}_j$ are both Gaussian distributed, their product is not. This results in the fact that $z_{ij}$ is no longer marginally Gaussian, so the AME models cannot be written in the form of Model (4.2). This means that we cannot directly obtain samples of $\mathbf{z}$ from the distribution $p(\mathbf{z}|\mathbf{y})$ and use them to approximate the gradients.

In general, an exchangeable latent factor model for binary network is given as follows:

$$\begin{aligned} y_{ij} &= 1(z_{ij} > 0), \\ z_{ij} &= \mu + f(\mathbf{u}_i, \mathbf{v}_j) + e_{ij}, \\ \begin{pmatrix} \mathbf{u}_i \\ \mathbf{v}_i \end{pmatrix}, \cdots, \begin{pmatrix} \mathbf{u}_m \\ \mathbf{v}_m \end{pmatrix} &\overset{i.i.d.}{\sim} N_{2r}\left(\mathbf{0}, \mathbf{I}_2 \otimes \mathrm{diag}(\tau_1^2, \cdots, \tau_r^2)\right), \\ \mathbf{e} = \mathrm{vec}(\mathbf{E}) &\sim N_{m^2}(\mathbf{0}, \boldsymbol{\Sigma}(\boldsymbol{\phi})), \end{aligned} \tag{4.4}$$

where $\mathbf{E}$ is the matrix representation of the error terms and its vectorization $\mathbf{e}$ is normally distributed. Since the row and column effects $\mathbf{a}$ and $\mathbf{b}$ are linear in the model and normally distributed, we absorb them into the error term $\mathbf{E}$ and its covariance matrix is denoted as $\boldsymbol{\Sigma}(\boldsymbol{\phi})$. The parameters to estimate are $\boldsymbol{\theta} = (\mu, \boldsymbol{\phi})$. The function $f(\cdot)$ is not necessarily linear and thus the marginal distribution of $\mathbf{z}$ is not Gaussian. The full likelihood, which is an integral of $p(\mathbf{z}|\mathbf{y},\boldsymbol{\theta})$ with respect to $\mathbf{z}$ over its support $S(\mathbf{y})$, can be decomposed into the integral of the product $p(\mathbf{z}|\mathbf{u},\mathbf{v},\boldsymbol{\theta})p(\mathbf{u},\mathbf{v}|\boldsymbol{\theta})$, i.e.,

$$\begin{aligned} \ell(\boldsymbol{\theta} : \mathbf{y}) &= \log\left[\int_{S(\mathbf{y})} p(\mathbf{z}|\boldsymbol{\theta}) d\mathbf{z}\right], \\ &= \log\left[\int_{S(\mathbf{y})} \int_{\mathbb{R}^{2mr}} p(\mathbf{z}|\mathbf{u},\mathbf{v},\boldsymbol{\theta}) p(\mathbf{u},\mathbf{v}|\boldsymbol{\theta}) d\mathbf{u} d\mathbf{v} d\mathbf{z}\right]. \end{aligned}$$

The gradients are thus

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} \ell\left(\boldsymbol{\theta}:\mathbf{y}\right) &= \nabla_{\boldsymbol{\theta}} \log\left[\int_{S(\mathbf{y})}\int_{\mathbb{R}^{2mr}} p(\mathbf{z}|\mathbf{u},\mathbf{v},\boldsymbol{\theta})p(\mathbf{u},\mathbf{v}|\boldsymbol{\theta})d\mathbf{u}d\mathbf{v}d\mathbf{z}\right], \\
&= \frac{\int_{S(\mathbf{y})}\int_{\mathbb{R}^{2mr}}\left[\nabla_{\boldsymbol{\theta}} p(\mathbf{z}|\mathbf{u},\mathbf{v},\boldsymbol{\theta})p(\mathbf{u},\mathbf{v}|\boldsymbol{\theta}) + p(\mathbf{z}|\mathbf{u},\mathbf{v},\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}} p(\mathbf{u},\mathbf{v}|\boldsymbol{\theta})\right]d\mathbf{u}d\mathbf{v}d\mathbf{z}}{\int_{S(\mathbf{y})}\int_{\mathbb{R}^{2mr}} p(\mathbf{z}|\mathbf{u},\mathbf{v},\boldsymbol{\theta})p(\mathbf{u},\mathbf{v}|\boldsymbol{\theta})d\mathbf{u}d\mathbf{v}d\mathbf{z}}, \\
&= \frac{\int_{S(\mathbf{y})}\int_{\mathbb{R}^{2mr}} \frac{\nabla_{\boldsymbol{\theta}} p(\mathbf{z}|\mathbf{u},\mathbf{v},\boldsymbol{\theta})}{p(\mathbf{z}|\mathbf{u},\mathbf{v},\boldsymbol{\theta})} p(\mathbf{z}|\mathbf{u},\mathbf{v},\boldsymbol{\theta})p(\mathbf{u},\mathbf{v}|\boldsymbol{\theta})d\mathbf{u}d\mathbf{v}d\mathbf{z}}{\int_{S(\mathbf{y})}\int_{\mathbb{R}^{2mr}} p(\mathbf{z}|\mathbf{u},\mathbf{v},\boldsymbol{\theta})p(\mathbf{u},\mathbf{v}|\boldsymbol{\theta})d\mathbf{u}d\mathbf{v}d\mathbf{z}} \\
&\quad + \frac{\int_{S(\mathbf{y})}\int_{\mathbb{R}^{2mr}} \frac{\nabla_{\boldsymbol{\theta}} p(\mathbf{u},\mathbf{v}|\boldsymbol{\theta})}{p(\mathbf{u},\mathbf{v}|\boldsymbol{\theta})} p(\mathbf{z}|\mathbf{u},\mathbf{v},\boldsymbol{\theta})p(\mathbf{u},\mathbf{v}|\boldsymbol{\theta})d\mathbf{u}d\mathbf{v}d\mathbf{z}}{\int_{S(\mathbf{y})}\int_{\mathbb{R}^{2mr}} p(\mathbf{z}|\mathbf{u},\mathbf{v},\boldsymbol{\theta})p(\mathbf{u},\mathbf{v}|\boldsymbol{\theta})d\mathbf{u}d\mathbf{v}d\mathbf{z}}, \\
&= \mathbf{E}\left[\frac{\nabla_{\boldsymbol{\theta}} p(\mathbf{z}|\mathbf{u},\mathbf{v},\boldsymbol{\theta})}{p(\mathbf{z}|\mathbf{u},\mathbf{v},\boldsymbol{\theta})} + \frac{\nabla_{\boldsymbol{\theta}} p(\mathbf{u},\mathbf{v}|\boldsymbol{\theta})}{p(\mathbf{u},\mathbf{v}|\boldsymbol{\theta})}|\mathbf{y}\right],
\end{aligned}$$

where the expectation term $\mathrm{E}[\cdot|\mathbf{y}]$ is with respect to the joint distribution $p(\mathbf{z},\mathbf{u},\mathbf{v}|\mathbf{y})$. This inspires us to obtain the samples from this joint distribution and use them to approximate the gradients numerically. We illustrate this process using the following example.

**Example 4.2.1.** *Gradient approximation for binary AME models.*

The log-likelihood for the binary AME Model (4.1) is

$$\begin{aligned}
\ell\left(\boldsymbol{\theta}:\mathbf{y}\right) &= \log\left[\int_{S(\mathbf{y})} p(\mathbf{z}|\boldsymbol{\theta})d\mathbf{z}\right], \\
&= \log\left[\int_{S(\mathbf{y})}\int_{\mathbb{R}^{2mr}} p(\mathbf{z}|\mathbf{u},\mathbf{v},\boldsymbol{\theta})p(\mathbf{u},\mathbf{v}|\boldsymbol{\theta})d\mathbf{u}d\mathbf{v}d\mathbf{z}\right],
\end{aligned}$$

with

$$\begin{aligned}
\mathbf{z}|\mathbf{u},\mathbf{v},\boldsymbol{\theta} &\sim N_{m^2}\left(\mu + (\mathbf{V}\otimes\mathbf{U})\mathbf{i}, \boldsymbol{\Sigma}(\boldsymbol{\phi})\right), \\
p(\mathbf{u},\mathbf{v}|\boldsymbol{\theta}) &\propto \left[\prod_{i=1}^{r}(\tau_i^2)^{-m}\right]\exp\left\{-\frac{1}{2}\mathbf{U}\left[\mathrm{diag}(\tau_1^2,\cdots,\tau_r^2)\right]^{-1}\mathbf{V}^T\right\},
\end{aligned}$$

where $\mathbf{i}$ is the vectorization of the identity matrix $\mathbf{I}_r$. Notice that the covariance $\boldsymbol{\Sigma}(\boldsymbol{\phi})$ is associated with $\boldsymbol{\phi}_1 = (\sigma_a^2, \sigma_b^2, \rho_{ab}, \rho_e)$ and the variance in $p(\mathbf{u},\mathbf{v}|\boldsymbol{\theta})$ is only associated with

$\boldsymbol{\phi}_2 = (\tau_1^2, \cdots, \tau_r^2)$. Therefore, the gradients can be calculated as

$$\begin{aligned}
\frac{\partial}{\partial \mu}\ell(\boldsymbol{\theta} : \mathbf{y}) &= \mathrm{E}\left[\mathbf{1}^T \boldsymbol{\Lambda}(\boldsymbol{\phi}_1)(\mathbf{z} - \mu\mathbf{1} - (\mathbf{V} \otimes \mathbf{U})\mathbf{i})|\mathbf{y}\right], \\
\frac{\partial}{\partial \phi_{1,i}}\ell(\boldsymbol{\theta} : \mathbf{y}) &= \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}(\boldsymbol{\phi}_1)\frac{\partial}{\partial \phi_{1,i}}\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\right) \\
&\quad - \frac{1}{2}\mathrm{E}\left[(\mathbf{z} - \mu\mathbf{1} - (\mathbf{V} \otimes \mathbf{U})\mathbf{i})^T \frac{\partial}{\partial \phi_{1,i}}\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)(\mathbf{z} - \mu\mathbf{1} - (\mathbf{V} \otimes \mathbf{U})\mathbf{i})|\mathbf{y}\right], \\
\frac{\partial}{\partial \tau_i^2}\ell(\boldsymbol{\theta} : \mathbf{y}) &= -\frac{m}{\tau_i^2} - \frac{1}{2}\mathrm{E}\left[\mathbf{U}\frac{\partial\left[\mathrm{diag}(\tau_1^2, \cdots, \tau_r^2)\right]^{-1}}{\partial \tau_i^2}\mathbf{V}^T|\mathbf{y}\right].
\end{aligned} \tag{4.5}$$

To approximate the gradients numerically, we need to sample from the joint distribution of $p(\mathbf{z}, \mathbf{u}, \mathbf{v}|\mathbf{y})$. Since the distribution $p(\mathbf{z}|\mathbf{y})$ is not Gaussian for AME models, it is hard to directly sample from the joint distribution or use the Gibbs sampling method that is discussed in Chapter 3. However, the conditional distributions $p(\mathbf{z}|\mathbf{u}, \mathbf{v}, \mathbf{y})$, $p(\mathbf{u}|\mathbf{v}, \mathbf{z})$ and $p(\mathbf{v}|\mathbf{u}, \mathbf{z})$ are all Gaussian, so we can easily iteratively sample from these three conditional distribution using a Gibbs sampling algorithm, and it will provide us with approximate Monte Carlo (MC) samples of the joint distribution $p(\mathbf{z}, \mathbf{u}, \mathbf{v}|\mathbf{y})$. Algorithm 4.2.1 describes the Gibbs sampling steps.

**Algorithm 4.2.1.** *Gibbs sampling from the joint distribution* $p(\mathbf{z}, \mathbf{u}, \mathbf{v}|\mathbf{y})$.

1. Choose a starting point $\mathbf{z}_0$, $\mathbf{u}_0$ and $\mathbf{v}_0$.

2. For $s = 1, \cdots, S$:

   2.1 The distribution of $\mathbf{z}^{(s)}|\mathbf{u}^{(s-1)}, \mathbf{v}^{(s-1)}, \mathbf{y}$ is $\mathrm{N}(\mu\mathbf{1} + (\mathbf{V}^{(s-1)} \otimes \mathbf{U}^{(s-1)})\mathbf{i}, \boldsymbol{\Sigma}(\boldsymbol{\phi}))$ and truncated by the support $S(\mathbf{y})$ of $\mathbf{z}$. We iteratively sample one element at a time using Algorithm 3.3.1 in Chapter 3 except now the conditional mean depends on $\mathbf{u}$ and $\mathbf{v}$. Note that when sampling from the conditional distribution of $z_{ii}$, i.e., the diagonal elements, we obtain the sample from the univariate Gaussian distribution, which is not truncated as $y_{ii}$ is undefined.

2.2 Given $\mathbf{z}^{(s)}$ and $\mathbf{v}^{(s-1)}$, we sample $\mathbf{u}^{(s)}$ from the full conditional of $\mathbf{u}^{(s)}|\mathbf{z}^{(s)}, \mathbf{v}^{(s-1)}$ given by Equations (A.3) in Appendix A.2.

2.3 Given $\mathbf{z}^{(s)}$ and $\mathbf{u}^{(s)}$, we sample $\tilde{\mathbf{v}}^{(s)}$ from the full conditional of $\tilde{\mathbf{v}}^{(s)}|\mathbf{z}^{(s)}, \mathbf{u}^{(s)}$ given by Equations (A.5) in Appendix A.2.

Once we obtain the Gibbs samples from Algorithm 4.2.1 as $\{(\mathbf{z}^{(s)}, \mathbf{u}^{(s)}, \mathbf{v}^{(s)})\}_{s=1}^{S}$, the gradients in Equations (4.5) can be numerically approximated as follows:

$$\begin{aligned}
\widehat{\frac{\partial}{\partial \mu}\ell(\boldsymbol{\theta} : \mathbf{y})} &= \frac{1}{S}\sum_{s=1}^{S}\left[\mathbf{1}^T\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\mathbf{w}^{(s)}\right], \\
\widehat{\frac{\partial}{\partial \phi_{1,i}}\ell(\boldsymbol{\theta} : \mathbf{y})} &= \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}(\boldsymbol{\phi}_1)\frac{\partial}{\partial \phi_{1,i}}\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\right) - \frac{1}{2S}\sum_{s=1}^{S}\left[{\mathbf{w}^{(s)}}^T\frac{\partial}{\partial \phi_{1,i}}\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\mathbf{w}^{(s)}\right], \\
\widehat{\frac{\partial}{\partial \tau_r^2}\ell(\boldsymbol{\theta} : \mathbf{y})} &= -\frac{m}{\tau_r^2} + \frac{1}{2S(\tau_r^2)^2}\sum_{s=1}^{S}\left[{\mathbf{u}^{r(s)}}^T\mathbf{u}^{r(s)} + {\mathbf{v}^{r(s)}}^T\mathbf{v}^{r(s)}\right],
\end{aligned} \tag{4.6}$$

with $\mathbf{w}^{(s)} \equiv \mathbf{z}^{(s)} - \mu\mathbf{1} - (\mathbf{V}^{(s)} \otimes \mathbf{U}^{(s)})\mathbf{i}$. □

This method of approximating gradients applies in general to latent factors models. However, as we discussed in Chapter 3, if the dimension of $\mathbf{Y}$ is high, it is impossible to get enough MC samples of the joint distribution $p(\mathbf{z}, \mathbf{u}, \mathbf{v}|\mathbf{y})$ in a short time. In the next section, we will revisit composite likelihood estimation of exchangeable models for binary network data. In that context, the algorithm introduced in this section can be used on subsets of networks and provides accurate approximation to the gradients of the composite likelihood.

## *4.3 q-node Composite Likelihood Estimation*

We showed in Chapter 3 that triadic composite likelihood estimation works well in estimating parameters in the Gaussian SRM for binary network data. It had the following two main advantages. First, the triadic composite likelihood reduces the integral in the full likelihood into smaller pieces and then makes it possible to obtain an accurate approximation of the gradient using the sampling method. This makes the gradient-based optimization method feasible. Second, after obtaining triad census, we can categorize the triadic subgraphs into

isomorphic groups, which makes the composite likelihood a weighted sum of only 16 terms and thus the computational cost is essentially constant as a function of the network size.

However, if the model becomes more complicated, for example, adding multiplicative terms like the AME model, the triadic composite likelihood may not be sufficient. Note that the variance of $z_{ij}$ in the AME model is $\sigma_a^2 + \sigma_b^2 + \sum_{i=1}^{r} \tau_i^2$. Therefore as the number of parameters increases, it will become hard for the model to distinguish the $\tau^2$'s from the variance term based only on a 6-dimensional vector, which results in more local modes or local modes closer to the truth. It may be hard for the gradient method to distinguish between those local modes. To show this, we use the triadic composite likelihood estimation method to estimate the parameters in the binary AME model. In Figure 4.1, we plot the trajectories of the parameters during the MC-AdaGrad algorithm, which optimizes the triadic composite likelihood of the AME model for a single dataset. The chains may not have converged yet after 3000 iterations, and the trajectory of $\tau^2$ stays far from the truth.

This inspires us to consider a $q$-node composite likelihood. Denote $m_k = (i_{k,1}, \cdots, i_{k,q})$, $k = 1, \cdots, \binom{m}{q}$ as the $q$-dimensional subset of the nodes and $\mathbf{y}_{m_k}$ as the vectorization of the subnetwork of nodes $m_k$. The $q$-node composite likelihood for exchangeable latent factors models is defined as follows:

$$\begin{aligned} \tilde{\ell}^{(q)}(\boldsymbol{\theta} : \mathbf{y}) &= \sum_{k=1}^{\binom{m}{q}} \log\left[p(\mathbf{y}_{m_k}|\boldsymbol{\theta})\right] \\ &= \sum_{k=1}^{\binom{m}{q}} \log\left[\int_{S(\mathbf{y}_{m_k})} \int_{\mathbb{R}^{2qr}} p(\mathbf{z}_{m_k}|\mathbf{u}_{m_k}, \mathbf{v}_{m_k}, \boldsymbol{\theta}) p(\mathbf{u}_{m_k}, \mathbf{v}_{m_k}|\boldsymbol{\theta}) d\mathbf{u}_{m_k} d\mathbf{v}_{m_k} d\mathbf{z}_{m_k}\right]. \end{aligned} \tag{4.7}$$

### *4.3.1 Stochastic MC-AdaGrad algorithm*

One challenge is that when $q > 3$, it is difficult to enumerate the $2^{q(q-1)}$ possible outcomes of $\mathbf{y}_{m_k}$ and categorize them into isomorphic subgraphs. This means that it is impractical to calculate the gradients for each category and add them up to get the gradient approximation of the composite likelihood. Therefore, we use a stochastic gradient descent algorithm.

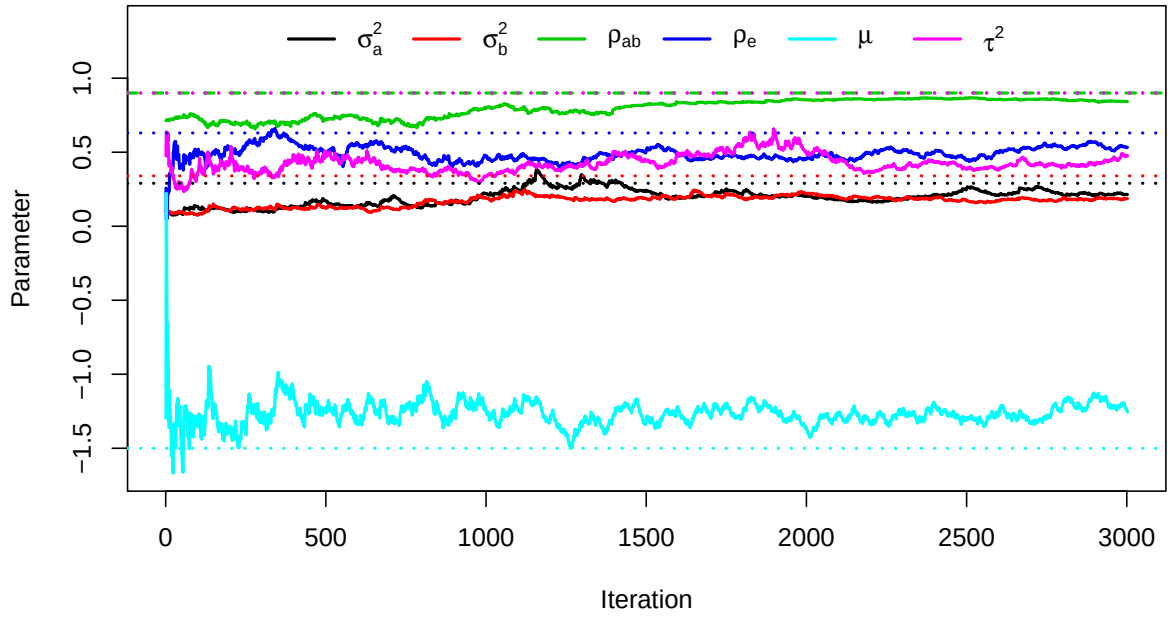


Figure 4.1: Triadic composite likelihood estimation of binary AME models. The dashed line refers to the true values of the parameters.

Instead of calculating gradients for all subgraphs, we sample one subgraph at each iteration and estimate the gradient of the composite likelihood using the approximated gradient of the marginal distribution of that subgraph. Since this estimate is unbiased, using it in a stochastic gradient ascent algorithm will guarantee convergence to a local optimum [Bottou, 1998]. The following stochastic MC-AdaGrad algorithm is what we propose for optimizing a $q$-node composite likelihood of binary exchangeable latent factor models.

**Algorithm 4.3.1.** *Stochastic MC-AdaGrad algorithm used to obtain q-node composite likelihood estimates of parameters in the binary exchangeable latent factor models.*

0. Choose a starting point $\boldsymbol{\theta}_0$, a base learning rate $\alpha$, the number of MCMC samples $S$

and a tolerance $\epsilon$.

1. Denote $\hat{g}_i(\boldsymbol{\theta})$ as the approximation of the gradient with respect to the $i$-th parameter given $\boldsymbol{\theta}$. While
$$\max_{i=1,\cdots,p} \left| \frac{\hat{g}_i(\boldsymbol{\theta}_t) - \hat{g}_i(\boldsymbol{\theta}_{t-1})}{\hat{g}_i(\boldsymbol{\theta}_{t-1})} \right| < \epsilon :$$

   1.1 Sample one $q$-node subset $m_k$ and obtain the corresponding subgraph $\mathbf{y}_{m_k}$ , generate $S$ MCMC samples from the conditional distribution of $\mathbf{z}$ given $\mathbf{y} = \mathbf{y}_{m_k}$ using Algorithm 4.2.1.

   1.2 Approximate the gradient at $\boldsymbol{\theta}_{t-1}$ as $\hat{\nabla}_{\boldsymbol{\theta}} \tilde{\ell}^{(q)}(\boldsymbol{\theta}_{t-1} : \mathbf{y})$.

   1.3 Update the parameter as $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} + \alpha \mathbf{G}_t^{-1/2} \tilde{\ell}^{(q)}(\boldsymbol{\theta}_{t-1} : \mathbf{y})$, with $\mathbf{G}_t = \mathbf{G}_{t-1} + \hat{\nabla}_{\boldsymbol{\theta}} \tilde{\ell}^{(q)}(\boldsymbol{\theta}_t : \mathbf{y}) \hat{\nabla}_{\boldsymbol{\theta}} \tilde{\ell}^{(q)}(\boldsymbol{\theta}_t : \mathbf{y})^T$ and $\mathbf{G}_1 = \mathbf{I}$.

When applying Algorithm 4.3.1 on $q$-node composite likelihood of binary AME models, we need to do the following parameter transformation, which is similar to what is discussed in Chapter 3.

$$\begin{aligned}
\sigma_a^2 &= e^{\gamma_a}, \ \gamma_a \in \mathbb{R} \\
\sigma_b^2 &= e^{\gamma_b}, \ \gamma_b \in \mathbb{R} \\
\rho_{ab} &= \frac{e^{\eta_{ab}} - 1}{e^{\eta_{ab}} + 1}, \ \eta_{ab} \in \mathbb{R} \\
\rho_e &= \frac{e^{\eta_e} - 1}{e^{\eta_e} + 1}, \ \eta_e \in \mathbb{R} \\
\tau_i^2 &= e^{\gamma_i}, \ \gamma_i \in \mathbb{R}, \ i = 1, \cdots, r.
\end{aligned} \tag{4.8}$$

We obtain the gradient with respect to the parameter $\boldsymbol{\delta} = (\mu, \gamma_a, \gamma_b, \eta_{ab}, \eta_e, \gamma_1, \cdots, \gamma_r)$ by using the chain rule. Then we can use Algorithm 4.3.1 to get the maximum composite likelihood estimate (MCLE) $\hat{\boldsymbol{\delta}}$ and apply the transformation of Equations (4.8) to obtain the estimate $\hat{\boldsymbol{\theta}}$.

### *4.3.2 Uncertainty quantification with standard errors*

Approximating confidence intervals and hypothesis tests can be based on standard errors of the composite likelihood estimates. To obtain the standard errors for the composite likelihood estimates, note that composite likelihood maybe regarded as a misspecified likelihood, which is built upon the incorrect assumption of independence of subgraphs. The theory for robust standard errors in misspecified models was developed by White [1980], who introduced the robust "sandwich" variance estimates. For composite likelihood, Varin et al. [2011] showed that the standard errors are the square root of the diagonal elements of the following "sandwich" form:

$$\mathbf{H}^{-1}(\boldsymbol{\theta})\mathbf{J}(\boldsymbol{\theta})\mathbf{H}^{-1}(\boldsymbol{\theta}), \tag{4.9}$$

where $\mathbf{H}(\boldsymbol{\theta})$ is the expectation of the Hessian matrix and the $\mathbf{J}(\boldsymbol{\theta})$ is the covariance matrix of the gradient:

$$\begin{aligned}
\mathbf{H}(\boldsymbol{\theta}) &= \mathrm{E}\left(-\nabla^2_{\boldsymbol{\theta}}\ell^{(q)}(\boldsymbol{\theta}:\mathbf{y})\right) \\
&= -\int \left(\nabla^2_{\boldsymbol{\theta}}\ell^{(q)}(\boldsymbol{\theta}:\mathbf{y})\right) p(\mathbf{y}|\boldsymbol{\theta})d\mathbf{y} \\
\mathbf{J}(\boldsymbol{\theta}) &= \mathrm{Cov}\left(\nabla_{\boldsymbol{\theta}}\ell^{(q)}(\boldsymbol{\theta}:\mathbf{y})\right) \\
&= \mathrm{E}\left(\nabla_{\boldsymbol{\theta}}\ell^{(q)}(\boldsymbol{\theta}:\mathbf{y})\nabla^T_{\boldsymbol{\theta}}\ell^{(q)}(\boldsymbol{\theta}:\mathbf{y})\right) \\
&= \int \left(\nabla_{\boldsymbol{\theta}}\ell^{(q)}(\boldsymbol{\theta}:\mathbf{y})\nabla^T_{\boldsymbol{\theta}}\ell^{(q)}(\boldsymbol{\theta}:\mathbf{y})\right) p(\mathbf{y}|\boldsymbol{\theta})d\mathbf{y}
\end{aligned} \tag{4.10}$$

We estimate these two matrices with the values at the MCLE, i.e., $\mathbf{H}(\hat{\boldsymbol{\theta}})$ and $\mathbf{J}(\hat{\boldsymbol{\theta}})$. However, there is no closed form expression for them, which are needed to calculate the standard error at the optima $\hat{\boldsymbol{\theta}}$. However, Monte Carlo approximations of these quantities can be obtained using a nested approach. First of all, we obtain a batch of $q \times q$ subgraphs from the observed data, and MCMC samples of $(\mathbf{z}, \mathbf{u}, \mathbf{v})$ with respect to each subgraph. Then we use those samples to approximate the gradient as well as the Hessian matrix. Secondly, we repeat the first step multiple times to get (estimated) samples of the gradient and Hessian matrix. Then we approximate the two expectations in Equations (4.10) with the average

of Hessian matrices and the outer product of gradients over the samples. Formally, we implement the following algorithm to obtain the approximation of the standard error for $\hat{\boldsymbol{\delta}}$ and build the confidence interval of $\hat{\boldsymbol{\delta}}$. The standard error of $\hat{\boldsymbol{\delta}}$ can be used to obtain a standard error for $\hat{\boldsymbol{\theta}}$ via the delta method or a confidence interval for $\hat{\boldsymbol{\theta}}$ via a confidence interval for $\hat{\boldsymbol{\delta}}$.

**Algorithm 4.3.2.** *Algorithm of approximation of the standard error of MCLE.*

1. For $c = 1, \cdots, C$:

   1.1 Sample $K$ (batch size) $q \times q$ subgraphs $\mathbf{Y}_{c,k}$, $k = 1, \cdots, K$, from the data.

   1.2 For each subgraph, we implement Algorithm 4.2.1 to obtain the MCMC samples from $p(\mathbf{z}, \mathbf{u}, \mathbf{v}|\mathbf{y}_k)$ and use them to approximate the gradient as $\mathbf{g}_{c,k}(\hat{\boldsymbol{\delta}})$ and the second derivative $\mathbf{H}_{c,k}(\hat{\boldsymbol{\delta}})$.

   1.3 Get one sample of the (approximated) gradient as $\mathbf{g}_c(\hat{\boldsymbol{\delta}}) = \frac{1}{K}\sum_{k=1}^{K}\mathbf{g}_{c,k}(\hat{\boldsymbol{\delta}})$ and the (approximated) second derivative as $\mathbf{H}_c(\hat{\boldsymbol{\delta}}) = \frac{1}{K}\sum_{k=1}^{K}\mathbf{H}_{c,k}(\hat{\boldsymbol{\delta}})$.

2. Obtain the approximation of the matrices $\mathbf{H}(\hat{\boldsymbol{\delta}})$ and $\mathbf{J}(\hat{\boldsymbol{\delta}})$ as

$$\hat{\mathbf{H}}(\hat{\boldsymbol{\delta}}) = \frac{1}{C}\sum_{c=1}^{C}\mathbf{H}_c(\hat{\boldsymbol{\delta}})$$

$$\hat{\mathbf{J}}(\hat{\boldsymbol{\delta}}) = \frac{1}{C}\sum_{c=1}^{C}\mathbf{g}_c(\hat{\boldsymbol{\delta}})\mathbf{g}_c^T(\hat{\boldsymbol{\delta}}) - \left[\frac{1}{C}\sum_{c=1}^{C}\mathbf{g}_c(\hat{\boldsymbol{\delta}})\right]\left[\frac{1}{C}\sum_{c=1}^{C}\mathbf{g}_c(\hat{\boldsymbol{\delta}})\right]^T$$

3. The standard error is approximated using the square root of the diagonal elements of $\mathrm{SE}(\hat{\boldsymbol{\delta}}) = \hat{\mathbf{H}}^{-1}(\hat{\boldsymbol{\delta}})\hat{\mathbf{J}}(\hat{\boldsymbol{\delta}})\hat{\mathbf{H}}^{-1}(\hat{\boldsymbol{\delta}})$. From this we can obtain an approximates $1-\alpha$ confidence interval for $\hat{\delta}_j$ as $\hat{\delta}_j \pm z_{1-\alpha/2}\mathrm{SE}(\hat{\delta}_j)$. After applying the transformation of Equations (4.8), we can obtain the confidence interval of $\hat{\boldsymbol{\theta}}$.

## *4.4 Numerical Studies*

In this section, we perform a simulation study to compare the performance of composite likelihood estimation and fully Bayesion estimation using the amen package. We also implement the composite likelihood estimation algorithm on a real dataset that is large in size for which the fully Bayesian procedure is impractical.

### *4.4.1 Simulation study*

The goal of simulation studies is to answer two main questions. First is how to choose a reasonable $q$ in the $q$-node composite likelihood of the AME model and the second is to compare the performance between composite likelihood estimation and fully Bayesian estimation in terms of accuracy and time consumption. In this section, we present three simulation results. First is on networks of moderate size $m = 500$. We implement the $q$-node composite likelihood estimation algorithm on different choices of $q$ and compare the performance with fully Bayesian estimation using the amen package. By doing this, we show that both methods provide accurate estimates for networks of moderate sizes. The second simulation is carried out in the same way except that the network datasets are generated with a larger size $m = 5000$. We show that in this case composite likelihood estimates are reasonable comparing to the truth while fully Bayesian estimation can be very slow or impractical. Thirdly, we compare the time consumption between the two methods with another simulation on networks of different sizes.

**Choice of $q$:** Recall that Figure 4.1 showed the stochastic gradient descent trajectories when optimizing the triadic composite likelihood for a single dataset. Comparing that to Figure 4.2, which plots the trajectories when optimizing the nonadic (9-node) composite likelihood of the same network dataset, we see that the nonadic composite likelihood estimation algorithm converged after about 2500 iterations under tolerance of $\epsilon = \sqrt{1e-5}$. The final point is closer to the truth than that obtained using triadic composite likelihood estimation.

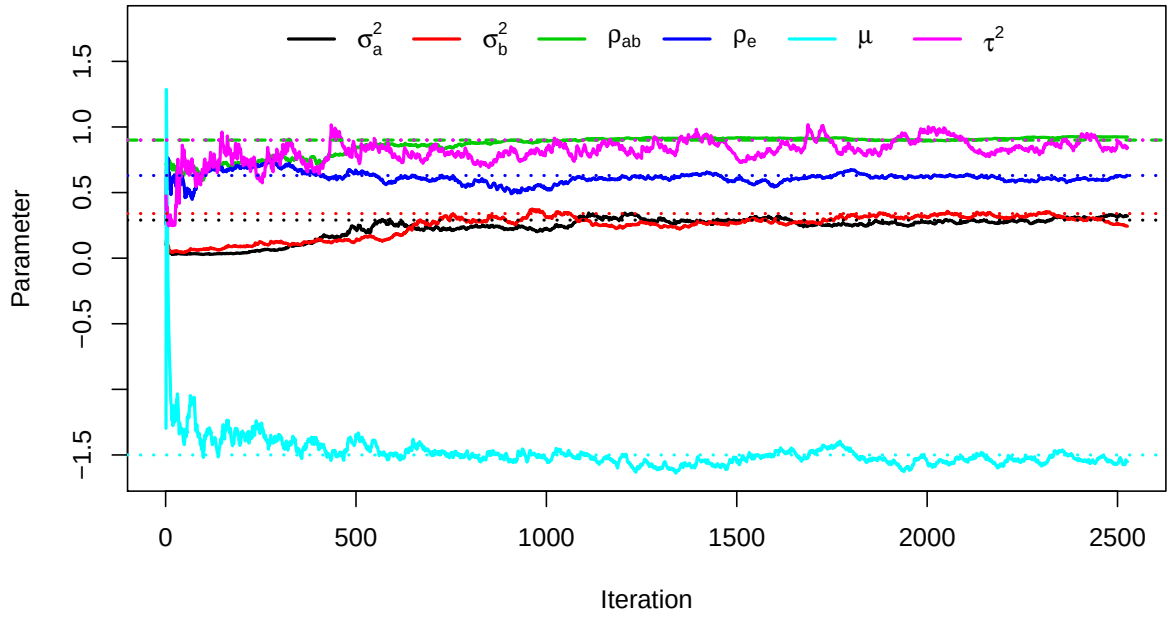


Figure 4.2: Nonadic (9-node) composite likelihood estimation of binary AME models. The dashed line refers to the true values of the parameters

In order to understand how the choice of $q$ may affect the performance of the estimates, we carry out a more thorough simulation study to compare the performance of different $q$-node composite likelihood estimates of the AME models on the same datasets. We generate 20 networks with node sizes $m = 500$ from an AME model given by Equation (4.1) with 1-dimensional multiplicative latent factors using the same values of the parameters $\boldsymbol{\theta} = (\mu, \sigma_a^2, \sigma_b^2, \rho_{ab}, \rho_e, \tau^2) = (-1.50, 0.29, 0.34, 0.9, 0.63, 0.9)$. For each dataset, we estimate the parameter $\boldsymbol{\theta}$ using 4 different $q$-node composite likelihood estimation: triadic, pentadic (5-node), heptadic (7-node) and nonadic composite likelihood.

The stochastic MC-AdaGrad algorithm 4.3.1 was used to optimize the objective function with the base learning rate $\alpha = 0.5$, the number of MCMC samples $S = 1000$ after a burnin

period of 500 iterations, a tolerance $\epsilon = \sqrt{1e-5}$ and the maximum number of iterations as 3000. Similar to the simulation study in Chapter 3, we obtained the starting values using moment estimates based on a rough estimate of the matrix $\mathbf{Z}$. First, we impute the continuous representation $\mathbf{Z}_0$ of the binary network $\mathbf{Y}$ using the $z$-scores of the ranks (defined in Chapter 3). Second, the starting value of $\mu_0$ is calculated using the mean of $\mathbf{Z}_0$, the row effects $a_{0,i}$ takes the row means of $\mathbf{Z}_0 - \mu_0\mathbf{1}\mathbf{1}^T$ and $b_{0,i}$ is from its column means. The starting value of $\rho_{0,ab}$ is then computed as the correlation between the vectors $\mathbf{a}$ and $\mathbf{b}$. Third, we do a singular value decomposition on $\mathbf{Z}_0 - \mu_0\mathbf{1}\mathbf{1}^T - \mathbf{a}_0\mathbf{b}_0^T$ and take the first singular vector as the approximation of $\mathbf{U}_0$ and $\mathbf{V}_0$. Fourth, we take the dyadic pairs $(e_{0,ij}, e_{0,ji})$ from the residual matrix $\mathbf{E}_0 = \mathbf{Z}_0 - \mu_0\mathbf{1}\mathbf{1}^T - \mathbf{a}_0\mathbf{b}_0^T - \mathbf{U}_0\mathbf{V}_0^T$ and calculate its correlation as $\rho_{0,e}$. Fifth, since we require the variance of the error terms to be 1 for identifiability, we need to scale $\mu_0$, the standard deviance terms $\sigma_{0,a}$, $\sigma_{0,b}$ and $\tau_0^2$ with the standard deviance of $\mathbf{e}_0$. Finally, using the inverse transformation of Equations (4.8) to get the starting point of $\boldsymbol{\delta}_0$.

Figure 4.3 shows the box plots that illustrate the distribution of estimates across the 20 datasets, and compares the performance among different choices of $q$ to the Bayes estimates given by the amen package. When $q = 3, 5$, the estimates have large variance and the medians are far from the truth, while for $q = 7, 9$, the estimates are closer to the truth as well as to the Bayes estimates. Also the variance across different datasets is small. This indicates that the triadic/pentadic composite likelihood estimates are not as good as the heptadic/nonadic composite likelihood estimates. Additionally, for small $q$, the chains may be trapped at some local mode for a very long time, and so longer MCMC runs may be necessary. These results suggest that composite likelihood estimation with larger subsets ($q = 7, 9$) provide better converging stochastic gradient descent algorithm and more accurate parameter estimates for binary AME models with 1-dimensional multiplicative latent factors.

**Comparison to Bayes estimates:** In addition to the choice of $q$, the previous simulation also shows that for networks of moderate sizes, the composite likelihood estimates perform similar to the Bayes estimates obtained using the amen package.

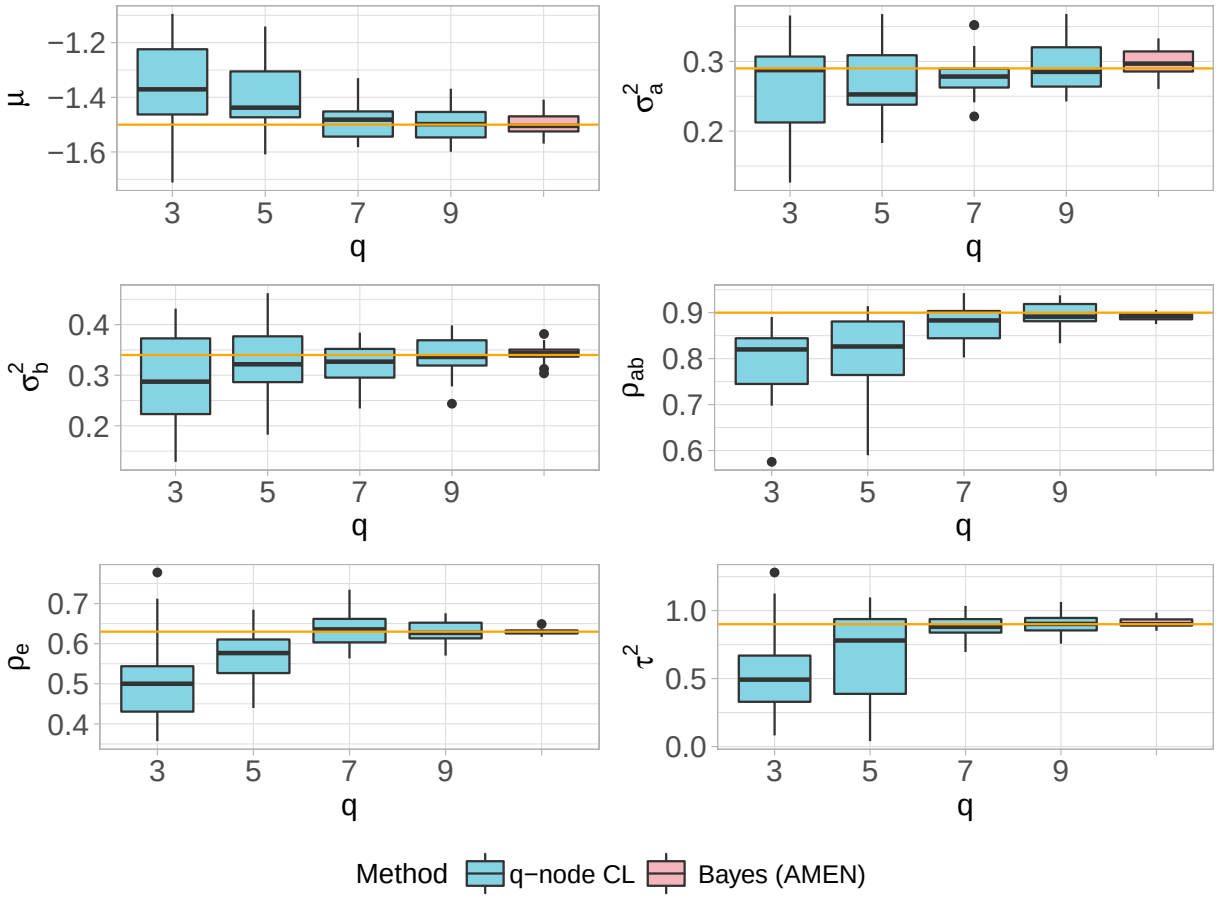


Figure 4.3: Comparison of composite likelihood estimate on different choices of $q$. The box plots show the distribution of estimates across the 20 datasets ($m = 500$). The orange line represents the true values of parameters

For a node size of $m = 500$, Figure 4.4 shows a scatterplot comparing the maximum heptadic composite likelihood estimates and the fully Bayes estimates on the 20 simulated datasets. The parameter estimates from the two methods are concentrated around the identity line, which means that for $m = 500$, both methods provide similar estimates. We perform another simulation study with the exact same settings as above but on the networks of size $m = 5000$. Figure 4.5 shows the box plots that illustrate the distribution of $q$-

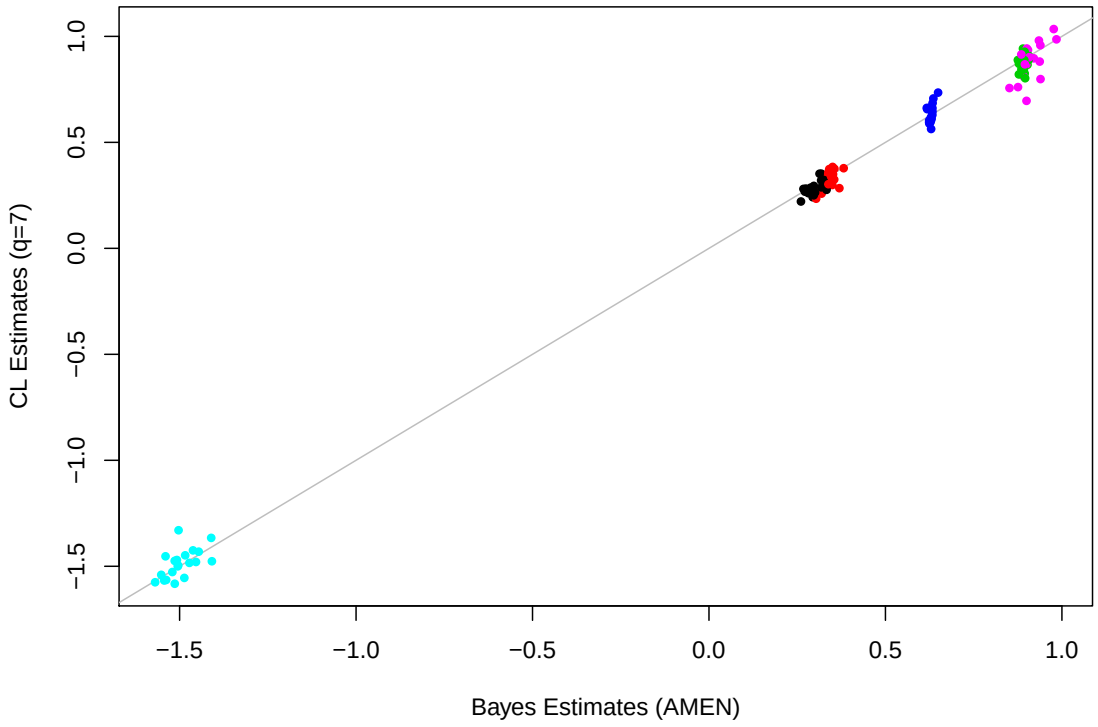


Figure 4.4: Scatterplot of the heptadic composite likelihood estimates versus the full-likelihood Bayes estimates using MCMC on networks of size $m = 500$.

node composite likelihood estimates across the 20 datasets with a node size of $m = 5000$. Similar conclusion can be drawn that composite likelihood estimation with larger subsets ($q = 7, 9$) provide better converging stochastic gradient descent algorithm and more accurate parameter estimates for binary AME models with 1-dimensional multiplicative latent factors. However, in this case, fully Bayesian estimation fails due to a huge time consumption. The MCMC algorithm becomes very slow for large networks. Therefore, we have shown that both composite likelihood estimation and fully Bayesian estimation provide accurate estimates for the AME models on networks of moderate sizes, e.g., $m = 500$. For large networks,

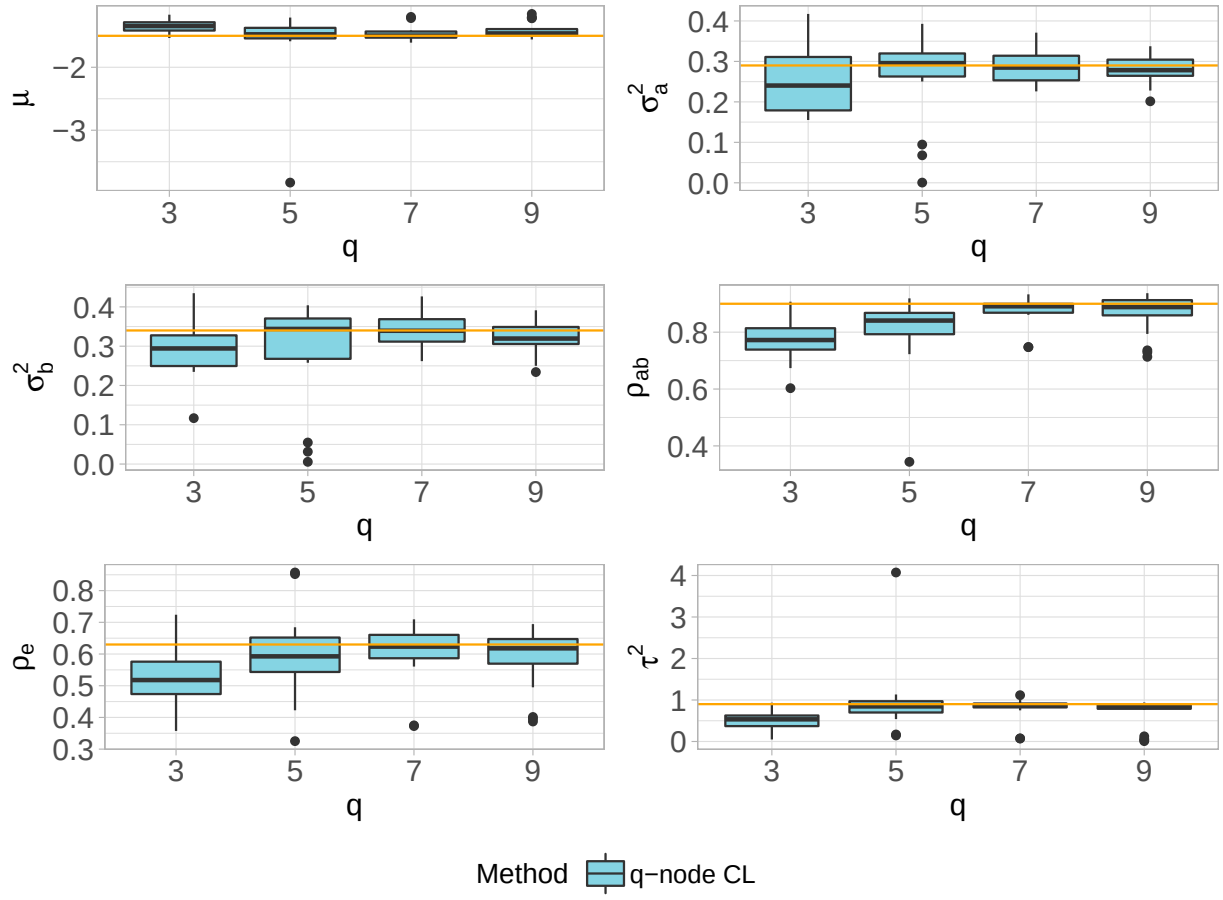


Figure 4.5: Comparison of composite likelihood estimate on different choices of $q$. The box plots show the distribution of estimates across the 20 datasets ($m = 5000$). The orange line represents the true values of parameters

say $m = 5000$, the MCLEs are still reasonably good comparing to the true values of the parameters, but fully Bayesian estimation using the amen package fails to provide estimates within a short amount of time.

**Computational cost:** Theoretically, the number of stochastic gradient descent iterations required to reach the accuracy of $\epsilon$ is $\mathcal{O}(1/\epsilon)$ [Bottou, 2010]. In each iteration, we calculate the gradient using $S$ MCMC samples based on a $q \times q$ subgraph. Therefore, the complexity

of algorithm to get MCLE is $\mathcal{O}(Sq^2/\epsilon)$, which does not depend on the node size $m$ of the full network, i.e., the complexity is $\mathcal{O}(C/\epsilon)$. On the contrary, the MCMC algorithm performed in the amen package updates the network $\mathbf{Z}$ in each iteration, which results in the complexity of $\mathcal{O}(m^2)$ in each iteration. If we perform $S$ iterations of MCMC, the total computational cost of the fully Bayesian estimation is $\mathcal{O}(Sm^2)$, which grows quadratically with the network size.

To see this, we performed a simulation study to compare the composite likelihood estimates with the Bayes estimates using the amen package. One network dataset was generated for each of the node sizes $m = 50, 200, 500, 2000, 5000$ and fitted with the AME model using the two methods. For the full Bayesian method, we ran the function in amen with a burnin period of 1000 and 2000 samples after that, except when $m = 5000$, we set the burnin period as 100 and number of samples 200, and then multiplied the time by 10 as an approximation since the time consumption is too large. For each dataset, we also ran MC-AdaGrad algorithm 5 times with different seeds but the same burnin period (500), MCMC sample size for each subgraph (1000), and tolerance ($\sqrt{1e-5}$) to optimize the hepatic composite likelihood. Then for each $m$, we took the average time across the 5 runs as an approximation of the time consumption.

Figure 4.6 shows the comparison on the time consumption. We find that when the network size increases, the speed of heptadic composite likelihood estimation keeps essentially the same while Bayesian estimation costs much more time. Therefore, the composite likelihood estimation might be preferred over the Bayes estimates when the network is large, for example, when $m \geq 2000$.

Based on the above three simulations, we see that both $q$-node composite likelihood estimation and fully Bayesian estimation provide accurate estimates for networks of moderate sizes. For large networks, composite likelihood estimation provides accurate estimates without costing extra time while the fully Bayesian estimation may fail due to the growing computational cost.

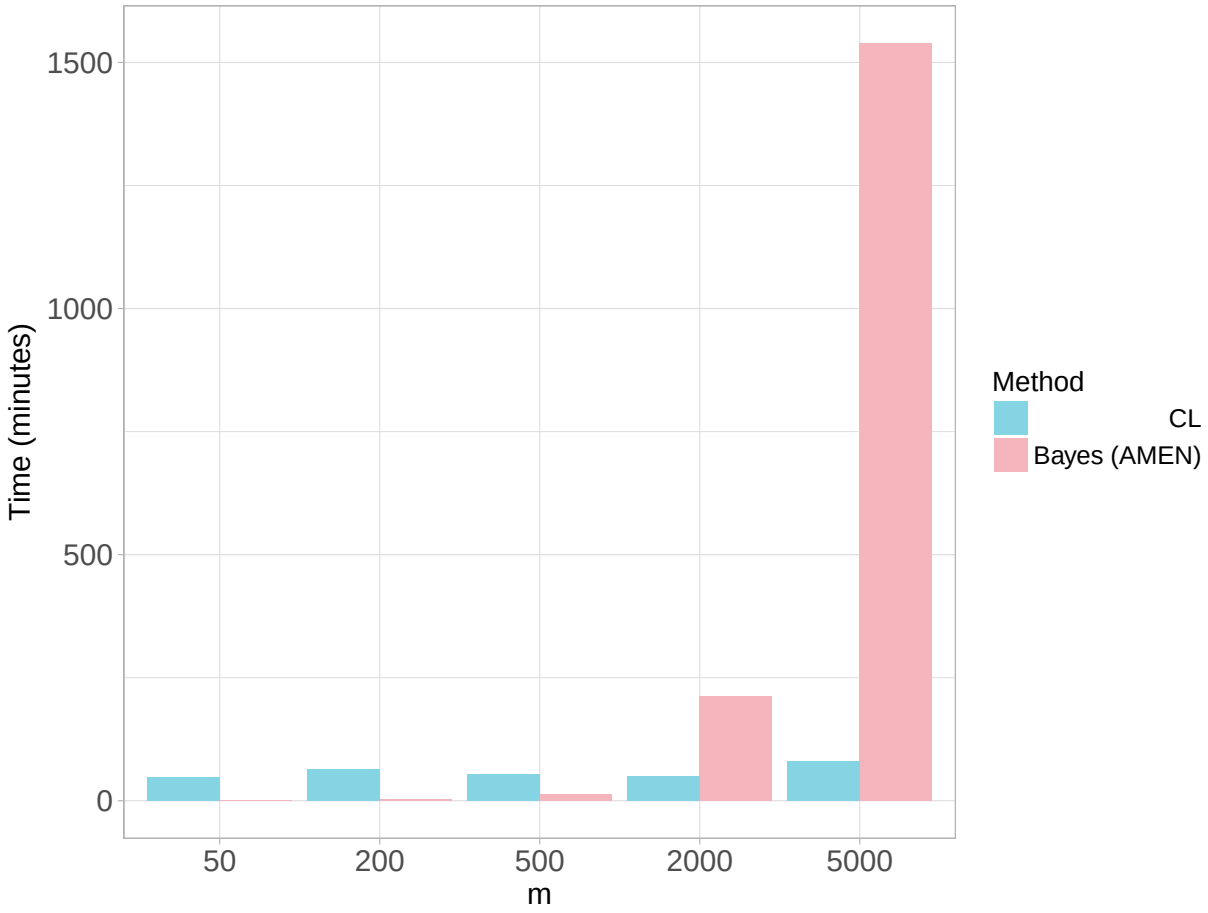


Figure 4.6: Comparison on time consumption between heptadic composite likelihood estimation and fully Bayesian estimation using MCMC.

### *4.4.2 Case study: Slashdot social network*

We now use the composite likelihood estimation method to fit an AME model to an online social network dataset. Slashdot is known as a technology-related news website, which features technology oriented news that is user submitted or editor evaluated. It allows users to tag each other as friends. In February 2009, a friendship network between 82,168 Slashdot users was studied and made available by Leskovec et al. [2009]. There are in total 948,464 undirected connections between those users, so the network is sparse with density $2.8 \times 10^{-4}$.

We fit an AME model with a one-dimensional latent factor to these data. Since the network is $82,168 \times 82,168$ in size, fully Bayesian estimation is infeasible, so we only perform the composite likelihood estimation. Although the network is symmetric by design, we can use our existing model and estimation method for asymmetric networks, but with $\rho_{ab} \equiv \rho_e \equiv 1$. For computational reasons, we set these parameters to be 0.99. Figure 4.7 shows the trajectories of the stochastic MC-AdaGrad algorithm when optimizing a heptadic (7-node) composite likelihood of the binary AME model.

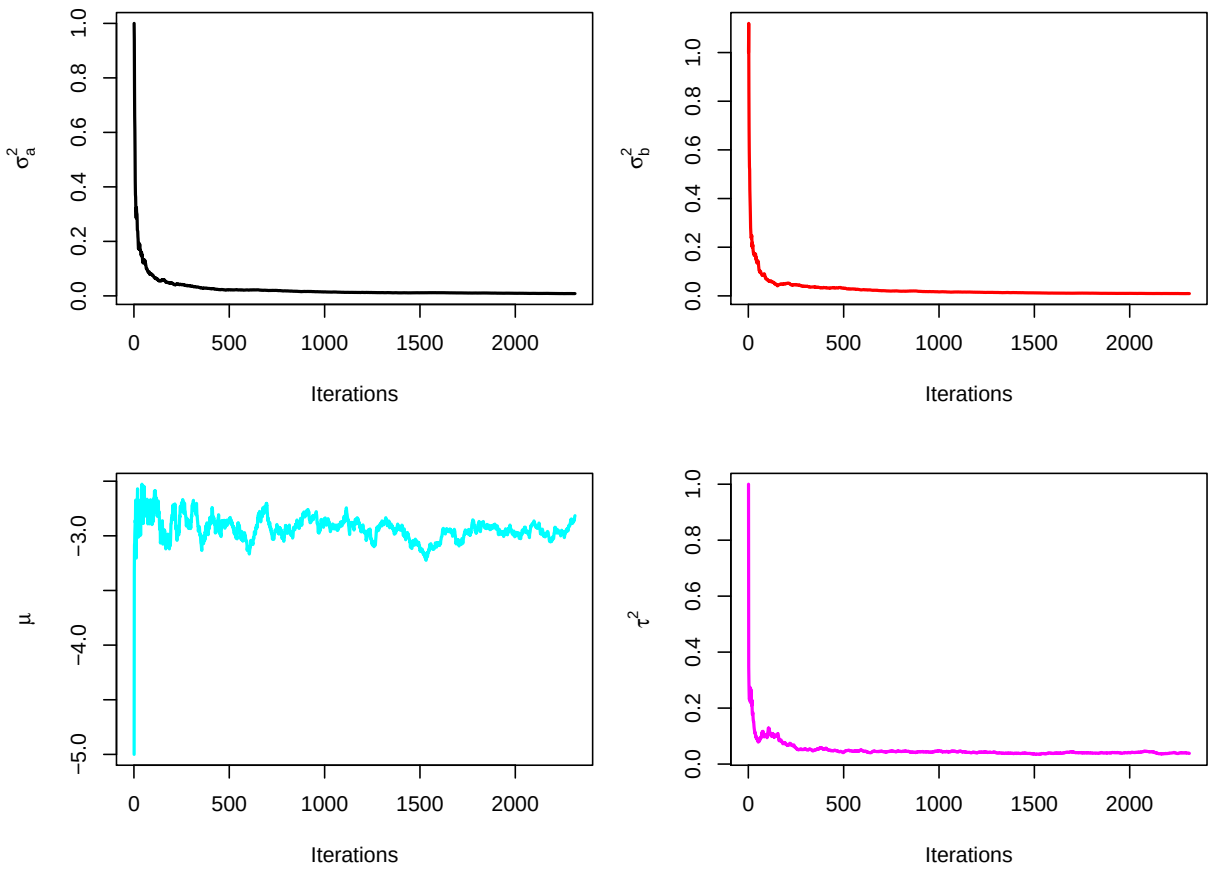


Figure 4.7: Stochastic MC-AdaGrad trajectories for optimizing the composite likelihood of the binary AME model for the Slashdot social network.

In the stochastic MC-AdaGrad algorithm 4.3.1, we set the base learning rate $\alpha = 0.5$, the number of MCMC samples $S = 500$ after a burnin period of 200 iterations. To speed up convergence, we ran the mini-batch version of the algorithm, which means that in each iteration we sample mini-batches (16 in this example) of subgraphs and use their average gradient as the gradient approximation for that step. The chains converged after 2313 iterations with the tolerance as $\sqrt{1e-7}$. The maximum composite likelihood estimates (MCLEs) are shown in Table 4.1. The 95% confidence intervals are obtained using Algorithm 4.3.2 with sample size $C = 100$ and batch size $K = 1024$. The forms of the second derivatives are derived in Appendix A.3.

| Parameter | MCLE | 95% C.I. |
|---|---|---|
| $\mu$ | -2.82 | [-2.78,-2.85] |
| $\sigma_a^2$ | $8.42 \times 10^{-3}$ | $[2.73 \times 10^{-3}, 2.60 \times 10^{-2}]$ |
| $\sigma_b^2$ | $9.43 \times 10^{-3}$ | $[3.20 \times 10^{-3}, 2.79 \times 10^{-2}]$ |
| $\tau^2$ | $3.83 \times 10^{-2}$ | $[2.58 \times 10^{-2}, 5.69 \times 10^{-2}]$ |

Table 4.1: MCLE of the AME model for the Slashdot social network.

As we can see, the estimates of the variance parameters $\sigma_a^2$ and $\sigma_b^2$ are close to each other. Any differences here are due to imprecision noise in the algorithm. In this case, Bayes estimates using the amen package are not obtainable because the dimension of the network is too large. When performing the MCMC using the amen package, it requires updating the $z_{ij}$'s. This means that at each iteration, the algorithm needs at least to simulate a network of size $82,168 \times 82,168$, which is impractical.

## *4.5 Discussion*

In this chapter we have extended the SRM to a wider variety of exchangeable latent factors models, i.e., the AME models. By including a multiplicative effect of latent factors, the marginal distribution of $\mathbf{z}$ is no longer Gaussian. So the Gibbs sampling method used to

approximate the gradients of the SRM of Chapter 3 is not applicable. We proposed a new Gibbs sampling method that generated MC samples of the joint distribution of $\mathbf{z}$ and the latent factors $\mathbf{u}$, $\mathbf{v}$, which provides an approximation to the likelihood gradients.

Another challenge of the latent factor model is that the information provided by triadic composite likelihood estimation does not appear to be sufficient given the increase in the number of parameters. Therefore, we developed a $q$-node composite likelihood estimation with $q > 3$. When $q > 3$, it is challenging to enumerate all isomorphic subgraphs and write the composite likelihood as a weighted sum of a small number of marginal likelihoods. Instead we proposed a stochastic MC-AdaGrad algorithm to optimize the $q$-node composite likelihood. At each iteration, one subgraph of node size $q$ is selected and the parameters are updated based on that subgraph. We illustrated the performance of this approach with a simulation study, and showed that for AME models, a small $q$ results in poor convergence, while larger values of $q$, for example, $q = 7$, provide reasonable estimates comparable to those obtained from a fully Bayesian approach. We have also shown with a real data application that for large networks, estimates can be obtained in a scalable way using composite likelihood estimation while the Bayesian method using the amen package fail in providing an estimate.

Since the reasonable choices of $q$ depend on the model, it might be interesting to consider an aggregated composite likelihood by using the marginal distribution of all subgraphs with node size no larger than $Q$. In this case, the objective becomes

$$\tilde{\ell}^{\text{agg}}(\boldsymbol{\theta} : \mathbf{y}) = \sum_{q=1}^{Q} \sum_{k=1}^{\binom{m}{q}} \log \left[ p(\mathbf{y}_{m_k} | \boldsymbol{\theta}) \right]. \tag{4.11}$$

A stochastic MC-AdaGrad algorithm would randomly select subgraphs with different dimensions at each iteration. By doing this, we would avoid the task of having to specify a particular value of $q$.

So far, we have discussed how composite likelihood estimation works on exchangeable models for large binary network data. In the next chapter, covariates will be introduced to the network models, resulting in non-exchangeability. We will show that the algorithm

introduced in this chapter, which proceeds by random selection of single subgraphs, will still be applicable for the non-exchangeable case.

# Chapter 5

# COMPOSITE LIKELIHOOD ESTIMATION OF BINARY NON-EXCHANGEABLE NETWORK MODELS

## *5.1 Overview*

In the previous chapters we used a triadic composite likelihood to estimate the parameters in binary network models such as the SRM, and extended it to a $q$-node composite likelihood for more complicated models, such as the AME models. Both of the models are exchangeable, i.e., the distribution of the network $p(\mathbf{Y}|\boldsymbol{\theta})$ being invariant to the permutation of the nodes.

In many cases, however, we may want to consider a network model that involves some covariates. For instance, in Chapter 2, we showed an example of a student friendship network that included data on individual-level delinquency status. In that example, we model the network as a function of the delinquency. Any model that includes observed node-level attributes is non-exchangeable. In general, the network model with covariates for binary network can be represented as follows:

$$\begin{aligned} y_{ij} &= 1(z_{ij} > 0),\ i, j \in \{1, \cdots, m\},\ i \neq j, \\ z_{ij} &= \boldsymbol{\beta}^T \mathbf{x}_{ij} + f(\mathbf{u}_i, \mathbf{v}_j) + e_{ij}, \\ \begin{pmatrix} \mathbf{u}_i \\ \mathbf{v}_i \end{pmatrix}, \cdots, \begin{pmatrix} \mathbf{u}_m \\ \mathbf{v}_m \end{pmatrix} &\overset{i.i.d.}{\sim} N_{2r}\left(\mathbf{0}, \boldsymbol{\Sigma}_{uv}(\boldsymbol{\phi})\right), \\ \mathbf{e} = \mathrm{vec}(\mathbf{E}) &\sim N_{m^2}(\mathbf{0}, \boldsymbol{\Sigma}(\boldsymbol{\phi})), \end{aligned} \tag{5.1}$$

where $\mathbf{u}_i$ and $\mathbf{v}_i$ are the $r$-dimensional latent factors that are associated with node $i$ and $\mathbf{E}$ is the matrix representation of the error term. The term $\mathbf{x}_{ij}$ is a $D$-dimensional covariate, and $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_D)^T$ is the corresponding vector of parameters that describes the relationship between $\mathbf{x}_{ij}$ and $y_{ij}$. The covariate $\mathbf{x}_{ij}$ can include nodal attributes (e.g., the number of delinquent behaviors of individual $i$ or $j$) or dyadic attributes including the network of the

previous time point or other relationships such as the number of mutual friends, etc. Similar to what was described in Chapter 4, the row and column effects are absorbed into the error term $e_{ij}$ and the term $f(\mathbf{u}_i, \mathbf{v}_j)$ represents some function of random effects $\mathbf{u}_i$ and $\mathbf{v}_j$, which is not necessarily linear, for example the stochastic blockmodel of Nowicki and Snijders [2001] and the latent distance model of Hoff et al. [2002].

The goal is to estimate the parameter vector $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\phi})$. Similar to the exchangeable models, the full likelihood $\ell(\boldsymbol{\theta} : \mathbf{Y}) = \log[p(\mathbf{Y}|\boldsymbol{\theta})]$ of the model with covariates is also an intractable integral. In this chapter, we will discuss scalable composite likelihood algorithms for estimating the parameters in these models.

In Section 5.2, we will introduce the composite likelihood estimation for non-exchangeable models, including the approximation of gradients and the algorithm to optimize the composite likelihood. In Section 5.3, a simulation study is presented. We compare the performance of the composite likelihood estimates with the full Bayes estimates using the amen package in R, to illustrate that composite likelihood estimation can be applied to non-exchangeable network models and provide reasonable estimates. In Section 5.4, the composite likelihood estimation algorithm will be applied to two real binary network datasets that include nodal attributes.

## *5.2 Composite Likelihood for Non-Exchangeable Models*

Denoting $\mathbf{Y} \in \{0, 1\}^{m \times m}$ as the binary network, $\mathbf{Z} \in \mathbb{R}^{m \times m}$ as the continuous representation of $\mathbf{Y}$, the matrices $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{m \times r}$ as $r$-dimensional random effects and $\mathbf{X} \in \mathbb{R}^{m \times m \times D}$ as a $d$-dimensional covariate array, we can write the non-exchangeable network model in a matrix form as follows:

$$\begin{aligned} \mathbf{Y} &= \mathbf{1}(\mathbf{Z} > 0) \\ \mathbf{Z} &= \langle \boldsymbol{\beta}, \mathbf{X} \rangle + \mathbf{U}\mathbf{V}^T + \mathbf{E}, \end{aligned} \tag{5.2}$$

with

$$\begin{pmatrix} \mathbf{u}_i \\ \mathbf{v}_i \end{pmatrix}, \cdots, \begin{pmatrix} \mathbf{u}_m \\ \mathbf{v}_m \end{pmatrix} \overset{i.i.d.}{\sim} N_{2r}\left(\mathbf{0}, \mathbf{\Sigma}_{uv}(\boldsymbol{\phi})\right),$$

$$\mathbf{e} = \text{vec}(\mathbf{E}) \sim N_{m^2}(\mathbf{0}, \mathbf{\Sigma}(\boldsymbol{\phi})),$$

where $\mathbf{1}(\cdot)$ is the indicator function that applies element-wise on $\mathbf{Z} > 0$ and $\langle \cdot, \cdot \rangle$ is defined such that $\langle \boldsymbol{\beta}, \mathbf{X} \rangle = \sum_{d=1}^{D} \beta_d \mathbf{X}_d$, with $\mathbf{X}_d$ referring to the $d$-th slice, the $d$-th covariate. The parameters to be estimated are $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\phi})$. The full likelihood can be written as

$$\begin{aligned} \ell\left(\boldsymbol{\theta} : \mathbf{Y}\right) &= \log\left[\int_{S(\mathbf{Y})} p(\mathbf{Z}|\boldsymbol{\theta}) d\mathbf{Z}\right], \\ &= \log\left[\int_{S(\mathbf{Y})} \int_{\mathbb{R}^{2mr}} p(\mathbf{Z}|\mathbf{U}, \mathbf{V}, \boldsymbol{\theta}) p(\mathbf{U}, \mathbf{V}|\boldsymbol{\theta}) d\mathbf{U} d\mathbf{V} d\mathbf{Z}\right], \end{aligned} \tag{5.3}$$

where $S(\mathbf{Y})$ is the support of $\mathbf{Z}$ given $\mathbf{Y}$, i.e., $z_{ij} \leq 0$ if $y_{ij} = 0$ and $z_{ij} > 0$ if $y_{ij} = 1$. This likelihood is an intractable integral and so is its gradient with respect to $\boldsymbol{\theta}$. In this section, we will first discuss how to approximate the gradient of the likelihood, and how the approximation can be used in the optimization of the composite likelihood.

### *5.2.1 Gradient Approximation for Non-exchangeable Models*

Recall that in Chapter 4, we derived the general form of the gradient of the likelihood shown in Equation (5.3) as follows,

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} \ell\left(\boldsymbol{\theta} : \mathbf{Y}\right) &= \nabla_{\boldsymbol{\theta}} \log\left[\int_{S(\mathbf{Y})} \int_{\mathbb{R}^{2r}} p(\mathbf{Z}|\mathbf{U}, \mathbf{V}, \boldsymbol{\theta}) p(\mathbf{U}, \mathbf{V}|\boldsymbol{\theta}) d\mathbf{U} d\mathbf{V} d\mathbf{Z}\right], \\ &= \mathbf{E}\left[\frac{\nabla_{\boldsymbol{\theta}} p(\mathbf{Z}|\mathbf{U}, \mathbf{V}, \boldsymbol{\theta})}{p(\mathbf{Z}|\mathbf{U}, \mathbf{V}, \boldsymbol{\theta})} + \frac{\nabla_{\boldsymbol{\theta}} p(\mathbf{U}, \mathbf{V}|\boldsymbol{\theta})}{p(\mathbf{U}, \mathbf{V}|\boldsymbol{\theta})} | \mathbf{Y}\right], \end{aligned}$$

where the expectation term $E[\cdot|\mathbf{Y}]$ is with respect to the joint distribution $p(\mathbf{Z}, \mathbf{U}, \mathbf{V}|\mathbf{Y})$. This suggests that the gradient can be approximated with a Monte Carlo (MC) sample from this conditional distribution. The process is similar to the one introduced for the AME model in the previous chapter.

**Example 5.2.1.** *Gradient approximation for binary AME models with covariates.*

The AME model for a binary network with covariate array $\mathbf{X}$ can be represented as

$$
\begin{aligned}
\mathbf{Y} &= \mathbf{1}(\mathbf{Z} > 0), \\
\mathbf{Z} &= \langle \boldsymbol{\beta}, \mathbf{X} \rangle + \mathbf{1}\mathbf{a}^T + \mathbf{b}\mathbf{1}^T + \mathbf{U}\mathbf{V}^T + \mathbf{E}, \\
\begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix} &\sim N_{2m}\left(\mathbf{0}, \begin{pmatrix} \sigma_a^2 & \rho_{ab}\sigma_a\sigma_b \\ \rho_{ab}\sigma_a\sigma_b & \sigma_b^2 \end{pmatrix} \otimes \mathbf{I}_m\right), \\
\begin{pmatrix} \mathbf{u}_i \\ \mathbf{v}_i \end{pmatrix}, \cdots, \begin{pmatrix} \mathbf{u}_m \\ \mathbf{v}_m \end{pmatrix} &\overset{i.i.d.}{\sim} N_{2r}\left(\mathbf{0}, \mathbf{I}_2 \otimes \mathrm{diag}(\tau_1^2, \cdots, \tau_r^2)\right), \\
\left\{ \begin{pmatrix} e_{ij} \\ e_{ji} \end{pmatrix} \right\}_{i \neq j} &\overset{i.i.d.}{\sim} N_2\left(\mathbf{0}, \begin{pmatrix} 1 & \rho_e \\ \rho_e & 1 \end{pmatrix}\right).
\end{aligned}
\tag{5.4}
$$

Denoting $\mathbf{y}$, $\mathbf{z}$, $\mathbf{u}$ and $\mathbf{v}$ as the vectorization of the matrices $\mathbf{Y}$, $\mathbf{Z}$, $\mathbf{U}$ and $\mathbf{V}$, and also denoting $\tilde{\mathbf{X}}$ as an $m^2 \times d$ matrix, with the column $\tilde{\mathbf{x}}_s$ as the vectorization of the $s$-th covariate $\mathbf{X}_s$, $s = 1 \cdots, d$, the log-likelihood for Model (5.4) can be written as

$$
\begin{aligned}
\ell(\boldsymbol{\theta} : \mathbf{y}) &= \log\left[\int_{S(\mathbf{y})} p(\mathbf{z}|\boldsymbol{\theta}) d\mathbf{z}\right], \\
&= \log\left[\int_{S(\mathbf{y})} \int_{\mathbb{R}^{2r}} p(\mathbf{z}|\mathbf{u}, \mathbf{v}, \boldsymbol{\theta}) p(\mathbf{u}, \mathbf{v}|\boldsymbol{\theta}) d\mathbf{u} d\mathbf{v} d\mathbf{z}\right],
\end{aligned}
$$

with

$$
\begin{aligned}
\mathbf{z}|\mathbf{u}, \mathbf{v}, \boldsymbol{\theta} &\sim N_{m^2}\left(\tilde{\mathbf{X}}\boldsymbol{\beta} + (\mathbf{V} \otimes \mathbf{U})\mathbf{i}, \boldsymbol{\Sigma}(\boldsymbol{\phi})\right), \\
p(\mathbf{u}, \mathbf{v}|\boldsymbol{\theta}) &\propto \left[\prod_{i=1}^{r} (\tau_i^2)^{-m}\right] \exp\left\{-\frac{1}{2}\mathbf{U}\left[\mathrm{diag}(\tau_1^2, \cdots, \tau_r^2)\right]^{-1}\mathbf{V}^T\right\},
\end{aligned}
$$

where $\mathbf{i}$ is the vectorization of the identity matrix $\mathbf{I}_r$. Notice that the row and column effects $\mathbf{a}$, $\mathbf{b}$ are absorbed into the error term such that the covariance $\boldsymbol{\Sigma}(\boldsymbol{\phi})$ is associated with $\boldsymbol{\phi}_1 = (\sigma_a^2, \sigma_b^2, \rho_{ab}, \rho_e)$ and the variance in $p(\mathbf{u}, \mathbf{v}|\boldsymbol{\theta})$ is only associated with $\boldsymbol{\phi}_2 = (\tau_1^2, \cdots, \tau_r^2)$.

Therefore, the gradients can be calculated as

$$\begin{aligned}
\frac{\partial}{\partial \beta_i}\ell(\boldsymbol{\theta}:\mathbf{y}) &= \mathrm{E}\left[\tilde{\mathbf{x}}_i^T\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)(\mathbf{z}-\tilde{\mathbf{X}}\boldsymbol{\beta}-(\mathbf{V}\otimes\mathbf{U})\mathbf{i})|\mathbf{y}\right], \\
\frac{\partial}{\partial \phi_{1,i}}\ell(\boldsymbol{\theta}:\mathbf{y}) &= \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}(\boldsymbol{\phi}_1)\frac{\partial}{\partial\phi_{1,i}}\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\right) \\
&\quad -\frac{1}{2}\mathrm{E}\left[(\mathbf{z}-\tilde{\mathbf{X}}\boldsymbol{\beta}-(\mathbf{V}\otimes\mathbf{U})\mathbf{i})^T\frac{\partial}{\partial\phi_{1,i}}\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)(\mathbf{z}-\tilde{\mathbf{X}}\boldsymbol{\beta}-(\mathbf{V}\otimes\mathbf{U})\mathbf{i})|\mathbf{y}\right], \\
\frac{\partial}{\partial \tau_i^2}\ell(\boldsymbol{\theta}:\mathbf{y}) &= -\frac{m}{\tau_i^2}-\frac{1}{2}\mathrm{E}\left[\mathbf{U}\frac{\partial\left[\mathrm{diag}(\tau_1^2,\cdots,\tau_r^2)\right]^{-1}}{\partial\tau_i^2}\mathbf{V}^T|\mathbf{y}\right].
\end{aligned} \tag{5.5}$$

To approximate the gradients numerically, we need to sample from the joint distribution of $p(\mathbf{z},\mathbf{u},\mathbf{v}|\mathbf{y})$. We can iteratively sample from the conditional distributions $p(\mathbf{z}|\mathbf{u},\mathbf{v},\mathbf{y})$, $p(\mathbf{u}|\mathbf{v},\mathbf{z})$ and $p(\mathbf{v}|\mathbf{u},\mathbf{z})$ using a Gibbs sampling algorithm, which provides us with approximate Monte Carlo samples from the joint distribution $p(\mathbf{z},\mathbf{u},\mathbf{v}|\mathbf{y})$. Algorithm 5.2.1 describes the steps of the Gibbs sampler.

**Algorithm 5.2.1.** *Gibbs sampling from the joint distribution* $p(\mathbf{z},\mathbf{u},\mathbf{v}|\mathbf{y})$ *of AME models with covariate array* $\mathbf{X}$.

1. Choose a starting point $\mathbf{z}_0$, $\mathbf{u}_0$ and $\mathbf{v}_0$.

2. For $s=1,\cdots,S$:

   2.1 The distribution of $\mathbf{z}^{(s)}|\mathbf{u}^{(s-1)},\mathbf{v}^{(s-1)},\mathbf{y}$ is $N_{m^2}(\tilde{\mathbf{X}}\boldsymbol{\beta}+(\mathbf{V}^{(s-1)}\otimes\mathbf{U}^{(s-1)})\mathbf{i},\boldsymbol{\Sigma}(\boldsymbol{\phi}))$ constrained to the sign of $\mathbf{y}$. We iteratively sample one element at a time using Algorithm 3.3.1 in Chapter 3 except that the conditional mean now depends on $\tilde{\mathbf{X}}\boldsymbol{\beta}$ and $\mathbf{U}$, $\mathbf{V}$. Note that the conditional distribution of $z_{ij}$ is unconstrained if $y_{ij}$ is missing or undefined (as is the case for the diagonal entries of $\mathbf{Y}$).

   2.2 Given $\mathbf{z}^{(s)}$ and $\mathbf{v}^{(s-1)}$, sample $\mathbf{u}^{(s)}$ from the full conditional of $\mathbf{u}^{(s)}|\mathbf{z}^{(s)},\mathbf{v}^{(s-1)}$ given by Equation (A.7) in Appendix A.4.

   2.3 Given $\mathbf{z}^{(s)}$ and $\mathbf{u}^{(s)}$, sample $\tilde{\mathbf{v}}^{(s)}$ from the full conditional of $\tilde{\mathbf{v}}^{(s)}|\mathbf{z}^{(s)},\mathbf{u}^{(s)}$ given by Equation (A.9) in Appendix A.4.

Once we obtain the Gibbs samples from Algorithm 5.2.1 as $\{(\mathbf{z}^{(s)}, \mathbf{u}^{(s)}, \mathbf{v}^{(s)})\}_{s=1}^{S}$, the gradients in Equations (5.5) can be approximated as follows:

$$\begin{aligned}
\widehat{\frac{\partial}{\partial \beta_i}\ell(\boldsymbol{\theta} : \mathbf{y})} &= \frac{1}{S}\sum_{s=1}^{S}\left[\tilde{\mathbf{x}}_i^T \boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\mathbf{w}^{(s)}\right], \\
\widehat{\frac{\partial}{\partial \phi_{1,i}}\ell(\boldsymbol{\theta} : \mathbf{y})} &= \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}(\boldsymbol{\phi}_1)\frac{\partial}{\partial \phi_{1,i}}\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\right) - \frac{1}{2S}\sum_{s=1}^{S}\left[\mathbf{w}^{(s)T}\frac{\partial}{\partial \phi_{1,i}}\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\mathbf{w}^{(s)}\right], \\
\widehat{\frac{\partial}{\partial \tau_i^2}\ell(\boldsymbol{\theta} : \mathbf{y})} &= -\frac{m}{\tau_i^2} - \frac{1}{2S}\sum_{s=1}^{S}\left[\mathbf{U}^{(s)}\frac{\partial\left[\mathrm{diag}(\tau_1^2, \cdots, \tau_r^2)\right]^{-1}}{\partial \tau_i^2}\mathbf{V}^{(s)T}\right],
\end{aligned} \tag{5.6}$$

with $\mathbf{w}^{(s)} \equiv \mathbf{z}^{(s)} - \tilde{\mathbf{X}}\boldsymbol{\beta} - (\mathbf{V}^{(s)} \otimes \mathbf{U}^{(s)})\mathbf{i}$. □

This method of approximating gradients applies in general to latent factors models with covariates. However, as we discussed in Chapter 3, if the dimension of $\mathbf{Y}$ is high, it is impractical to get enough Monte Carlo samples from the joint distribution $p(\mathbf{z}, \mathbf{u}, \mathbf{v}|\mathbf{y})$ in a reasonable time. In the following subsection, we will revisit composite likelihood estimation for nonexchangeable binary network models. Stochastic optimization of a composite likelihood based on subgraphs can be preformed using this approach.

### *5.2.2 Composite Likelihood Estimation*

Depending on the complexity of the model, we may choose different $q$ for $q$-node composite likelihood. Denote $m_k = (i_{k,1}, \cdots, i_{k,q})$, $k = 1, \cdots, \binom{m}{q}$ as the $q$-dimensional subset of the nodes and $\mathbf{y}_{m_k}$ as the vectorization of the subnetwork of nodes $m_k$. The $q$-node composite likelihood for a network model with covariates is

$$\begin{aligned}
\tilde{\ell}^{(q)}(\boldsymbol{\theta} : \mathbf{y}) &= \sum_{k=1}^{\binom{m}{q}} \log\left[p(\mathbf{y}_{m_k}|\mathbf{x}_{m_k}, \boldsymbol{\theta})\right] \\
&= \sum_{k=1}^{\binom{m}{q}} \log\left[\int_{S(\mathbf{y}_{m_k})}\int_{\mathbb{R}^{2r}} p(\mathbf{z}_{m_k}|\mathbf{x}_{m_k}, \mathbf{u}_{m_k}, \mathbf{v}_{m_k}, \boldsymbol{\theta})p(\mathbf{u}_{m_k}, \mathbf{v}_{m_k}|\boldsymbol{\theta})d\mathbf{u}_{m_k}d\mathbf{v}_{m_k}d\mathbf{z}_{m_k}\right].
\end{aligned} \tag{5.7}$$

Because of the presence of the covariates, this composite likelihood cannot be written into a summation of a small number of marginal distributions of isomorphic subgraphs, even for

dyadic or triadic composite likelihood.

Similar to Chapter 4, we use a stochastic gradient descent algorithm to optimize the composite likelihood (5.7). Instead of calculating gradients for all subgraphs, we sample one subgraph at each iteration and estimate the gradient of the composite likelihood using the approximate gradient of the marginal distribution of that subgraph. Even though the model is not exchangeable, this estimate of the gradient is still unbiased. Therefore using it in a stochastic gradient ascent algorithm will guarantee convergence to a local optimum [Bottou, 1998]. The following stochastic MC-AdaGrad algorithm is what we propose for optimizing a $q$-node composite likelihood for binary non-exchangeable network models.

**Algorithm 5.2.2.** *Stochastic MC-AdaGrad algorithm for $q$-node composite likelihood estimation of binary network models with covariates.*

0. Choose a starting point $\boldsymbol{\theta}_0$, a base learning rate $\alpha$, the number of MCMC samples $S$ and a tolerance $\epsilon$.

1. Denote $\hat{g}_i(\boldsymbol{\theta})$ as the approximation of the gradient with respect to the $i$-th parameter given $\boldsymbol{\theta}$. While
$$\max_{i=1,\cdots,p} \left| \frac{\hat{g}_i(\boldsymbol{\theta}_t) - \hat{g}_i(\boldsymbol{\theta}_{t-1})}{\hat{g}_i(\boldsymbol{\theta}_{t-1})} \right| < \epsilon : \tag{5.8}$$

   1.1 Sample one $q$-node subset $k_s$ and obtain the corresponding subgraph $\mathbf{y}_{m_k}$ , generate $S$ MCMC samples from the conditional distribution of $\mathbf{z}$ given $\mathbf{y} = \mathbf{y}_{m_k}$ using Algorithm 5.2.1.

   1.2 Approximate the gradient at $\boldsymbol{\theta}_{t-1}$ as $\hat{\nabla}_{\boldsymbol{\theta}} \tilde{\ell}^{(q)}(\boldsymbol{\theta}_{t-1} : \mathbf{y})$.

   1.3 Update the parameter as $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} + \alpha \mathbf{G}_t^{-1/2} \tilde{\ell}^{(q)}(\boldsymbol{\theta}_{t-1} : \mathbf{y})$, with $\mathbf{G}_t = \mathbf{G}_{t-1} + \hat{\nabla}_{\boldsymbol{\theta}} \tilde{\ell}^{(q)}(\boldsymbol{\theta}_t : \mathbf{y}) \hat{\nabla}_{\boldsymbol{\theta}} \tilde{\ell}^{(q)}(\boldsymbol{\theta}_t : \mathbf{y})^T$ and $\mathbf{G}_1 = \mathbf{I}$.

When implementing the algorithm, we need to perform a parameter transformation to map the constrained parameters $\boldsymbol{\theta}$ to the unconstrained ones of $\boldsymbol{\delta} = (\boldsymbol{\beta}, \gamma_a, \gamma_b, \eta_{ab}, \eta_e, \gamma_1, \cdots, \gamma_r)$.

Formally, the mapping is as follows:

$$\begin{aligned}
\sigma_a^2 &= e^{\gamma_a},\ \gamma_a \in \mathbb{R} \\
\sigma_b^2 &= e^{\gamma_b},\ \gamma_b \in \mathbb{R} \\
\rho_{ab} &= \frac{e^{\eta_{ab}} - 1}{e^{\eta_{ab}} + 1},\ \eta_{ab} \in \mathbb{R} \\
\rho_e &= \frac{e^{\eta_e} - 1}{e^{\eta_e} + 1},\ \eta_e \in \mathbb{R} \\
\tau_i^2 &= e^{\gamma_i},\ \gamma_i \in \mathbb{R},\ i = 1, \cdots, r.
\end{aligned} \tag{5.9}$$

We obtain the gradient with respect to the parameter $\boldsymbol{\delta}$ by using the chain rule. Then we can use Algorithm 5.2.2 to get the maximum composite likelihood estimate (MCLE) $\hat{\boldsymbol{\delta}}$ and apply the transformation of Equations (5.9) to obtain the estimate $\hat{\boldsymbol{\theta}}$ at convergence.

## *5.3 Simulation Study*

A simulation study was performed to compare the estimates obtained by optimizing composite likelihood to those provided by the functions in the amen package. We generate 20 networks for each of the node sizes $m = 50, 200, 500$ from an AME model with covariates given by Equation (5.4) with 1-dimensional multiplicative latent factors using the values of parameters $\boldsymbol{\theta} = (\beta_0, \beta_1, \sigma_a^2, \sigma_b^2, \rho_{ab}, \rho_e, \tau^2) = (-1.50, 1.00, 0.29, 0.34, 0.9, 0.63, 0.9)$. For each value of $m$, a covariate array $\mathbf{X}$ is generated by sampling from the uniform distribution U$[0, 1]$ and taking the logarithm. In the previous chapter, we showed that for the AME models, a heptadic ($q = 7$) composite likelihood provided reasonable parameter estimates. Therefore, for each dataset, we estimate the parameter $\boldsymbol{\theta}$ using both heptadic (7-node) composite likelihood estimation and the fully Bayes method using the amen package.

The stochastic MC-AdaGrad algorithm 5.2.2 was used to optimize the objective function with the base learning rate $\alpha = 0.5$, the number of MCMC samples $S = 500$ after a burnin period of 200 iterations, and a tolerance $\epsilon = 0.005$. Similar to the simulation study in Chapter 4, we obtained the starting values using moment estimates based on a rough estimate of the matrix $\mathbf{Z}$. First, we impute the continuous representative $\mathbf{Z}_0$ of the binary network $\mathbf{Y}$ using

the $z$-score of the rank (defined in Chapter 3). Second, the starting value of $\beta_0$ and $\beta_1$ is obtained by taking the coefficients after regressing $\mathbf{Z}_0$ on $\mathbf{X}$, the row effects $a_{0,i}$ takes the row means of $\mathbf{Z}_0 - \beta_0 \mathbf{1}\mathbf{1}^T - \beta_1 \mathbf{X}$ and $b_{0,i}$ is from its column means. The starting value of $\rho_{0,ab}$ is then computed as the correlation between the vectors $\mathbf{a}$ and $\mathbf{b}$. Third, we do a singular value decomposition on $\mathbf{Z}_0 - \beta_0 \mathbf{1}\mathbf{1}^T - \beta_1 \mathbf{X} - \mathbf{a}_0 \mathbf{b}_0^T$ and take the first singular vector as the approximation of $\mathbf{U}_0$ and $\mathbf{V}_0$. Fourth, we take the dyadic pairs $(e_{0,ij}, e_{0,ji})$ from the residual matrix $\mathbf{E}_0 = \mathbf{Z}_0 - \beta_0 \mathbf{1}\mathbf{1}^T - \beta_1 \mathbf{X} - \mathbf{a}_0 \mathbf{b}_0^T - \mathbf{U}_0 \mathbf{V}_0^T$ and calculate its correlation as $\rho_{0,e}$. Fifth, since we require the variance of the error terms to be 1 for identifiability, we need to scale $\mu_0$, the standard deviance terms $\sigma_{0,a}$, $\sigma_{0,b}$ and $\tau_0^2$ with the standard deviance of $\mathbf{e}_0$. Finally, the transformation of Equations (5.9) are applied to get the starting point of $\boldsymbol{\delta}_0$.

Figure 5.1 presents the box plots of the parameter estimates across the 20 datasets under different network dimensions. The orange horizontal lines give the true values of the parameters. As the network size increases, the variances of the composite likelihood estimates around the true values decrease. This means that the performance of composite likelihood estimates is roughly similar to the fully Bayes estimates using MCMC on the binary network models with a covariate array when the network size is large.

For a node size of $m = 500$, Figure 5.2 shows a scatterplot comparing the maximum composite likelihood estimates and the fully Bayes estimates from the 20 simulated datasets. The parameter estimates from the two methods are concentrated around the identity line, which means that for large $m$, both methods provide similar estimates. Again for networks with small or moderate sizes, we suggest using Bayesian methods such as MCMC to estimate the parameters. When $m$ is large, say $m \geq 1000$, the results suggest that composite likelihood estimates are almost the same as the Bayes estimates. However the runtime of MCMC increases with $m$, while the runtime to the composite likelihood estimation algorithm is essentially constant as a function of the network size.

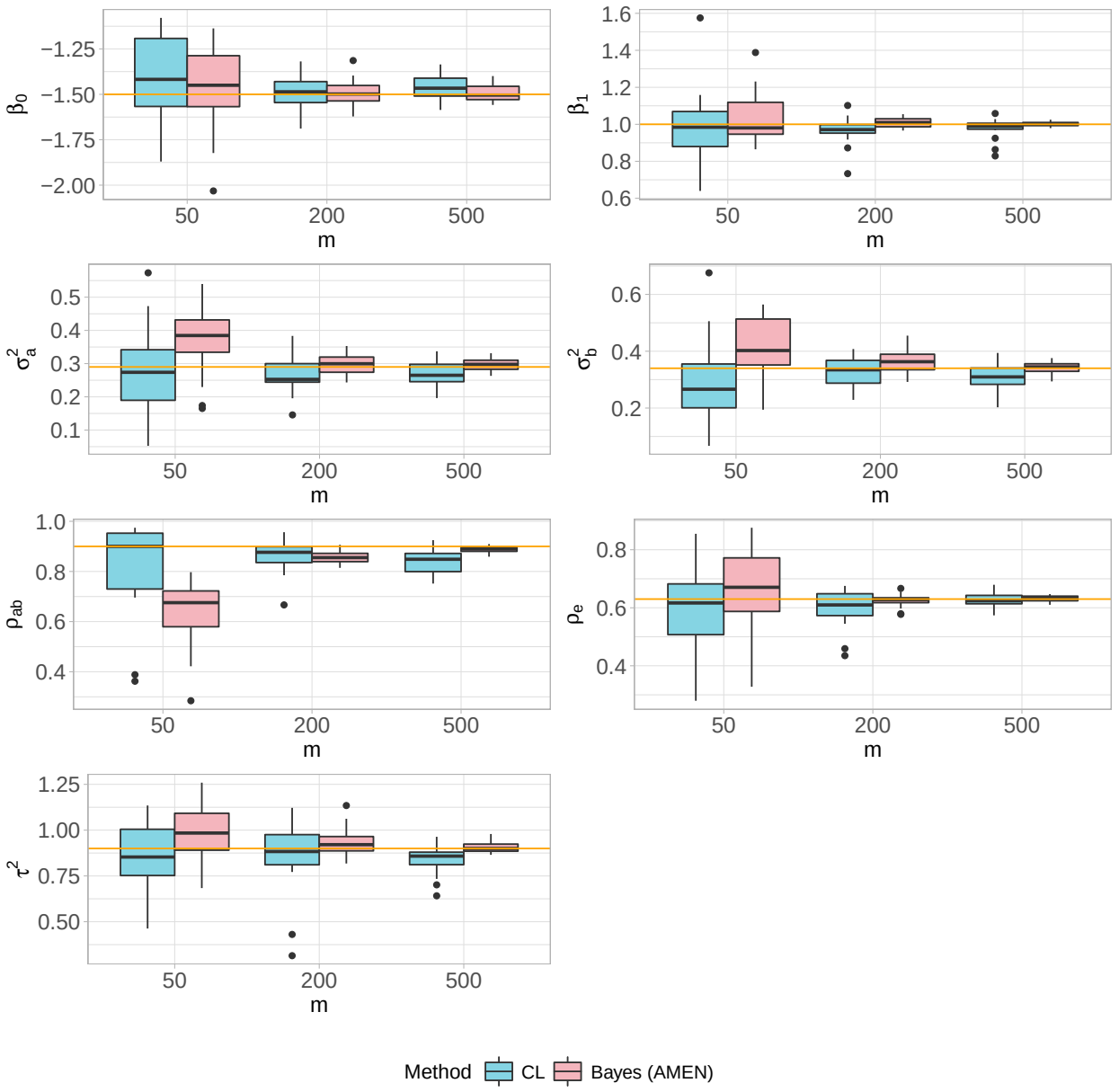


Figure 5.1: Comparison between the composite likelihood estimates and the full Bayesian method using MCMC. The box plots show the distribution of estimates across the 20 datasets. The orange line represents the true values of parameters.

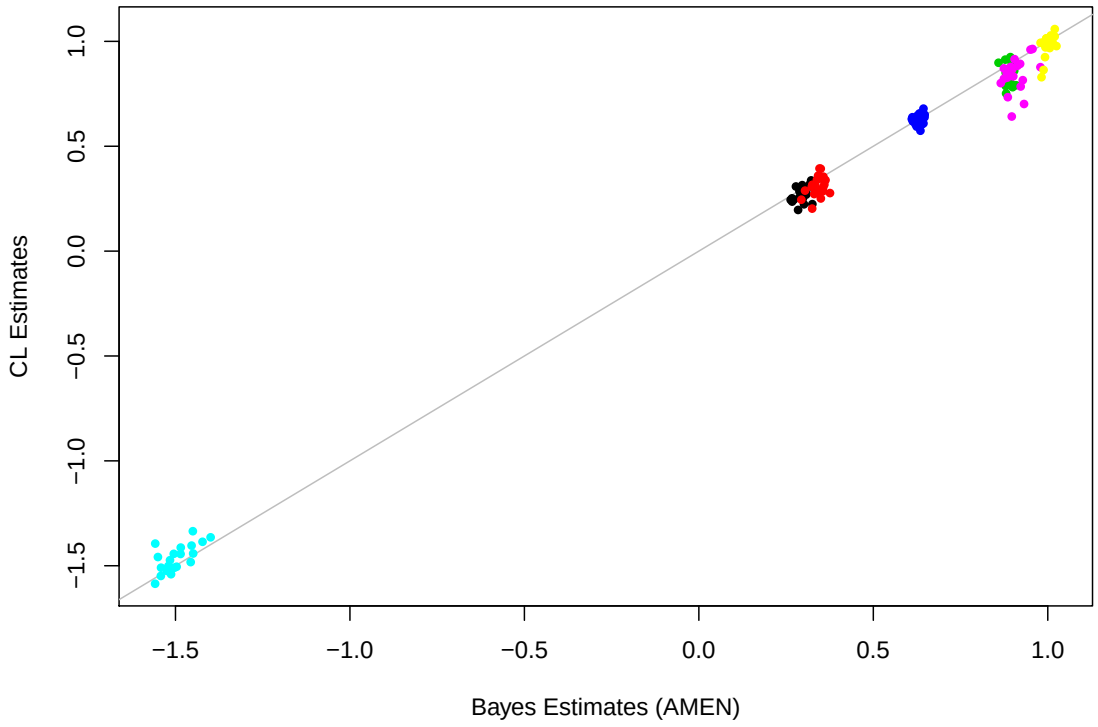


Figure 5.2: Scatterplot of the composite likelihood estimates versus the full-likelihood Bayes estimates using MCMC.

## *5.4 Case Study*

In this section, we obtain composite likelihood estimates of parameters in an AME model, bases on two network datasets. The first network is of moderate size, and the second one is very large. By using the two examples, we show that composite likelihood estimation may outperform fully Bayesian estimation in two aspects. One is in terms of model fit, and the other is scalability.

### *5.4.1 Email network*

The first case study is an email network, which records the incoming and outgoing emails between 1005 members of a European research institution. The link $y_{ij}$ takes the value 1 if person $i$ sent person $j$ at least one email within a period from October 2003 to May 2005 (18 months), and 0 otherwise. The dataset also contains some community memberships of the nodes. Each person belongs to exactly one of 42 departments at the research institute. The covariate matrix $\mathbf{X}$ is defined such that $x_{ij} = 1$ if person $i$ and person $j$ are from the same department. The data are described in Yin et al. [2017] and Leskovec et al. [2007].

We apply both 12-node composite likelihood estimation and Bayes estimation on the AME model of the email network with one covariate. The estimates are listed in Table 5.1. As we can see, the two sets of parameter estimates are not the same and the difference is non-negligible for some parameters.

| Method | $\beta_0$ | $\beta_1$ | $\sigma_a^2$ | $\sigma_b^2$ | $\rho_{ab}$ | $\rho_e$ | $\tau^2$ |
|---|---|---|---|---|---|---|---|
| MCLE | -2.88 | 1.76 | 0.44 | 0.35 | 0.97 | 0.96 | 0.20 |
| Bayes Est. (amen) | -3.38 | 1.97 | 0.89 | 0.59 | 0.95 | 0.92 | 0.46 |

Table 5.1: Point MCLEs of the AME model with one covariate for the email network.

Although we do not expect the two results to be exactly the same since the MCLE does not optimize the full likelihood, the apparent difference inspires us to investigate which one fits the data better. Therefore, we simulate 100 datasets using each of the estimates and perform a goodness-of-fit study. For each estimate, we calculate five metrics of each simulated network and compare the distribution over the 100 simulations to the metrics of the observed data. The metrics include the density of the network, the standard deviations of the out-degrees and the in-degrees, the dyadic and triadic dependence (reciprocity and transitivity).

Figure 5.3 shows the boxplots of the distribution of the five goodness-of-fit metrics (cal-

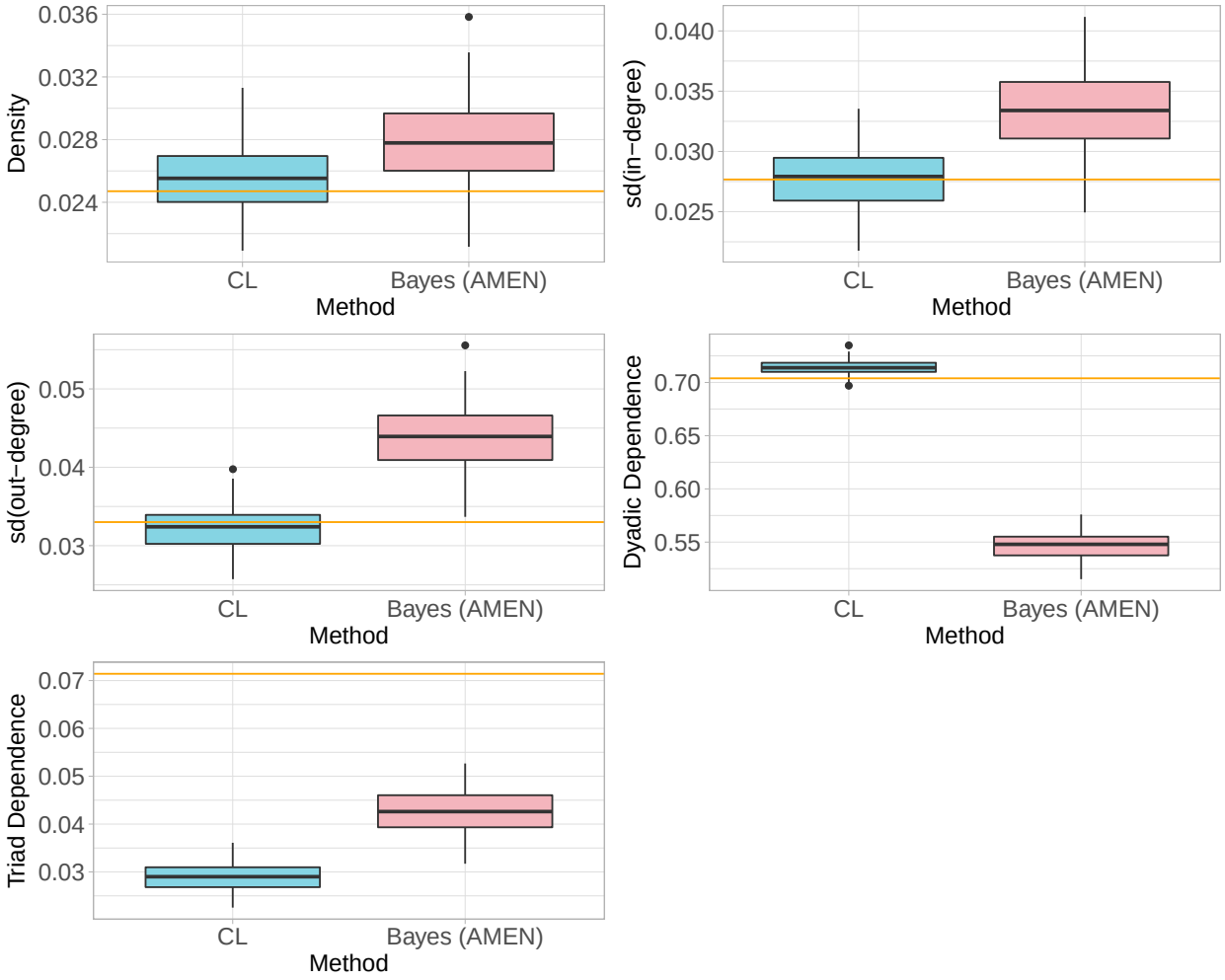


Figure 5.3: Goodness-of-fit diagnosis of the MCLE and the Bayes estimates

culated using the amen package) over the 100 simulated networks. The orange line indicates the value of the corresponding metric in the observed email network. The maximum composite likelihood estimate produces simulated networks that more closely resemble the actual data than the fully Bayes estimate in all metrics except the triadic dependence, where both estimates perform poorly. Probably the dataset needs to be fitted with a model with higher order multiplicative effect, i.e., $r > 1$. This indicates that MCLE performs better in terms of some goodness-of-fit statistics, as compared to the fully Bayes estimates using the amen

package.

### *5.4.2 YouTube*

We now use composite likelihood estimation to fit an AME model with one covariates to a very large YouTube network. Zafarani and Liu [2009] crawled the information of 1,138,499 users from YouTube (http://www.youtube.com), a video-sharing website on which users can upload, share, and view videos. The network data include the friendship information on 1,138,499 users, 2,990,443 undirected connections, and a density of $4.6 \times 10^{-6}$. In the dataset, the benchmark group membership of each user is also given. Some users may not belong to any groups while others could have more than one membership. We consider a $1,138,499 \times 1,138,499$ covariate $\mathbf{X}$ with the element $x_{ij} = 1$ indicating user $i$ and $j$ share at least one group membership.

We fit an AME model with a one-dimensional latent factor to these data. Since the network is large in size, fully Bayesian estimation is impractical, so we only perform the composite likelihood estimation. Although the network is symmetric by design, we can use our existing model and estimation method for asymmetric networks, but with $\rho_{ab} \equiv \rho_e \equiv 1$. For computational reasons, we set these parameters to be 0.99.

In the stochastic MC-AdaGrad algorithm 5.2.2, we set the base learning rate $\alpha = 0.5$, the number of MCMC samples $S = 500$ after a burnin period of 200 iterations. To speed up convergence, we ran the mini-batch version of the algorithm, with means that in each iteration, we sample mini-batches (16 in this example) of subgraphs and use their average gradient as the gradient approximation for that step. The chains converged after 1,046 iterations with the tolerance as $\sqrt{1e-6}$. The maximum composite likelihood estimates (MCLEs) are shown in Table 5.2. The 95% confidence intervals are obtained using Algorithm 4.3.2 in Chapter 4 with sample size $C = 100$ and batch size $K = 1024$. The forms of the second derivatives are derived in Appendix A.5. As we can see, the estimates for $\beta_1$ is significantly positive, which indicates significant evidence of a relationship between group membership and friendship links. This implies that people in the same group are more likely

to connect with each other than connect to those outside their groups. The estimates of the variance parameters $\sigma_a^2$ and $\sigma_b^2$ are close to each other. Any differences here are due to imprecision noise in the algorithm.

| Parameter | MCLE | 95% C.I. |
|---|---|---|
| $\beta_0$ | -3.06 | [-3.03,-3.09] |
| $\beta_1$ | 0.69 | [0.19, 1.18] |
| $\sigma_a^2$ | $1.57 \times 10^{-2}$ | $[7.21 \times 10^{-3}, 3.43 \times 10^{-2}]$ |
| $\sigma_b^2$ | $1.85 \times 10^{-2}$ | $[9.62 \times 10^{-3}, 3.55 \times 10^{-2}]$ |
| $\tau^2$ | $3.45 \times 10^{-2}$ | $[2.21 \times 10^{-2}, 5.36 \times 10^{-2}]$ |

Table 5.2: Point MCLEs of the AME model for the Slashdot social network.

In this case, the Bayes estimates from the amen package are not obtainable because the dimension of the network is too large. When performing the MCMC using the amen package, it requires updating the $z_{ij}$'s. This means that in each iteration, the algorithm needs at least to simulate a network of size $1,138,499 \times 1,138,499$, which is impractical.

## *5.5 Discussion*

In this chapter we have extended the composite likelihood estimation algorithm of Chapter 4 to non-exchangeable models for binary network data, for example, models with covariates. The existence of covariates breaks the exchangeability, which means that the composite likelihood cannot be written as a simple summation of a small number of marginal distributions of isomorphic subgraphs as in Chapter 3, even for the dyadic or triadic composite likelihood.

Similar to Chapter 4, we proposed a stochastic MC-AdaGrad algorithm to optimize the composite likelihood. We carried out a simulation study to compare its performance with the Bayes estimates on binary AME models with covariates, and showed that composite likelihood estimation provides estimates as accurate as the fully Bayes estimates for moderate sized networks.

We also applied the composite likelihood estimation to two real networks, where we saw the following two phenomena. The first is a case when the composite likelihood estimates provided better goodness-of-fit metrics than the Bayes estimates. The second is when the network size is very large. In this case, composite likelihood estimation can still be implemented while the fully Bayesian estimation cannot be practically implemented for $m \geq 2000$ based on the results in Chapter 4.

# BIBLIOGRAPHY

Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9(Sep):1981–2014, 2008.

James H. Albert and Siddhartha Chib. Bayesian analysis of binary and polychotomous response data. *J. Amer. Statist. Assoc.*, 88(422):669–679, 1993. ISSN 0162-1459. URL http://links.jstor.org/sici?sici=0162-1459(199306)88:422<669:BAOBAP>2.0.CO;2-T&origin=MSN.

Arthur U. Asuncion, Qiang Liu, Alexander T. Ihler, and Padhraic Smyth. Learning with Blocks: Composite Likelihood and Contrastive Divergence. *Aistats*, 9:33–40, 2010. ISSN 15324435.

Francesco Bartolucci, Maria Francesca Marino, and Silvia Pandolfi. Composite likelihood inference for hidden Markov models for dynamic networks. (67242), 2015.

Vladimir Batagelj and Andrej Mrvar. Pajek-program for large network analysis. *Connections*, 21(2):47–57, 1998.

Douglas Bates, Martin Mächler, Benjamin M Bolker, and Steven C Walker. Fitting linear mixed-effects models using lme4. *arXiv:1406.5823v1[stat.CO]23*, pages 1 – 51, 2014. ISSN 0092-8615. doi: 10.1177/009286150103500418.

Julian Besag. Statistical Analysis of Non-Lattice Data. *Journal of the Royal Statistical Society. Series D*, 24(3):179–195, 1975.

Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz, and Michael Ward. Icews coded event data, 2016. URL http://dx.doi.org/10.7910/DVN/28075.

Léon Bottou. Online Learning and Stochastic Approximations. *On-line learning in neural networks*, pages 1–34, 1998.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

Yan Chen, Qing Liu, and Peter ZG Qian. Sequential sampling enhanced composite likelihood approach to estimation of social intercorrelations in large-scale networks. 2014.

D. R. Cox and N. Reid. Miscellanea A note on pseudolikelihood constructed from marginal densities. *Biometrika*, 91(3):729–737, 2004. ISSN 00063444. doi: 10.1093/biomet/91.3.729.

Daniele Durante and David B. Dunson. Bayesian dynamic financial networks with time-varying predictors. *Statistics and Probability Letters*, 93:19–26, 2014. ISSN 01677152.

Nial Friel. Bayesian inference for Gibbs random fields using composite likelihoods. *arXiv.org*, stat.CO(2011):1–10, 2012. URL http://dl.acm.org/citation.cfm?id=2429795.

Steve Hanneke, Wenjie Fu, and Eric Xing. Discrete Temporal Models of Social Networks. *Electronic Journal of Statistics*, 4:585–605, 2010. ISSN 1935-7524. doi: 10.1214/09-EJS548. URL http://arxiv.org/abs/0908.1258.

Patrick Heagerty and Subhash Lele. A Composite Likelihood Approach to Binary Spatial Data. *Journal of the American Statistical Association*, 1459(September):1099–1111, 1998. ISSN 0162-1459. doi: 10.1080/01621459.1998.10473771. URL http://www.jstor.org/stable/2669853.

Peter D. Hoff. Modeling homophily and stochastic equivalence in symmetric relational data. *Nips*, pages 657–664, 2008. URL http://arxiv.org/abs/0711.1146.

Peter D. Hoff. Multiplicative latent factor models for description and prediction of social networks. *Computational and Mathematical Organization Theory*, 15(4):261–272, 2009a. ISSN 1381298X. doi: 10.1007/s10588-008-9040-4.

Peter D. Hoff. *A First Course in Bayesian Statistical Methods*. 2009b. doi: 10.1007/978-0-387-92407-61.

Peter D Hoff, Adrian E Raftery, and Mark S Handcock. Latent Space Approaches to Social Network Analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002. ISSN 0162-1459. doi: 10.1198/016214502388618906.

David R Hunter, Mark S Handcock, Carter T Butts, Steven M Goodreau, and Martina Morris. ergm: A package to fit, simulate and diagnose exponential-family models for networks. *Journal of statistical software*, 24(3):nihpa54860, 2008.

Stefan Knecht, Julia Reinholz, and Peter Kenning. The spread of obesity in a social network. *The New England journal of medicine*, 357(18):1866–1867; author reply 1867–1868, 2007. ISSN 0028-4793. doi: 10.1056/NEJMc072478.

Pavel N. Krivitsky and Mark S. Handcock. A separable model for dynamic networks. *J. R. Stat. Soc. Ser. B. Stat. Methodol.*, 76(1):29–46, 2014. ISSN 1369-7412. doi: 10.1111/rssb.12014. URL http://dx.doi.org/10.1111/rssb.12014.

Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1 (1):2, 2007.

Jure Leskovec, Kevin J Lang, Anirban Dasgupta, and Michael W Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.

Bruce G Lindsay. Composite Likelihood Methods. *Contemporary Mathematics*, 80:221–239, 1988.

Krzysztof Nowicki and Tom a. B Snijders. Estimation and Prediction for Stochastic Blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087, 2001. ISSN 0162-1459. doi: 10.1198/016214501753208735.

Ari Pakman and Liam Paninski. Exact Hamiltonian Monte Carlo for Truncated Multivariate Gaussians. *Journal of Computational and Graphical Statistics*, 8600(February): 130610142855008, 2013. ISSN 1061-8600. doi: 10.1080/10618600.2013.788448. URL http://www.tandfonline.com/doi/abs/10.1080/10618600.2013.788448.

Walter W. Piegorsch and George Casella. Empirical Bayes Estimation for Logistic Regression and Extended Parametric Regression Models. *Journal of Agricultural, Biological, and Environmental Statistics*, 1(2):231–249, 1996. ISSN 10857117. doi: 10.2307/1400367.

José C Pinheiro and Edward C Chao. Efficient Laplacian and adaptive Gaussian quadrature algorithms for multilevel generalized linear mixed models. *Journal of Computational and Graphical Statistics*, 15(1):58–81, 2006. ISSN 1061-8600. doi: 10.1198/106186006X96962.

Didier Renard, Geert Molenberghs, and Helena Geys. A pairwise likelihood approach to estimation in multilevel probit models. *Computational Statistics and Data Analysis*, 44 (4):649–667, 2004. ISSN 01679473. doi: 10.1016/S0167-9473(02)00263-3.

Håvard Rue, Sara Martino, and Nicolas Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 71(2):319–392, 2009. ISSN 13697412. doi: 10.1111/j.1467-9868.2008.00700.x.

Purnamrita Sarkar and Andrew W Moore. Dynamic social network analysis using latent space models. *ACM SIGKDD Explorations Newsletter*, 7(2):31–40, 2005.

Robert Schall. Estimation in Generalized Linear Models with random effects. *Biometrics*, 78(4):719–727, 1991.

Daniel K Sewell and Yuguo Chen. Latent Space Models for Dynamic Networks. *Journal of the American Statistical Association*, (April):37–41, 2015. doi: 10.1080/01621459.2014.988214.

T. a B Snijders, Gerhard G. van de Bunt, and C. E G Steglich. Introduction to stochastic actor-based models for network dynamics. *Social Networks*, 32(1):44–60, 2010. ISSN 03788733. doi: 10.1016/j.socnet.2009.02.004.

T.A.B. Snijders, C.E.G. Steglich, and M. Schweinberger. Modeling the co-evolution of networks and behavior. *Longitudinal models in the behavioral and related sciences*, pages 41–71, 2007.

Tom AB Snijders. Models for longitudinal network data. *Models and methods in social network analysis*, 1:215–247, 2005.

Julien Stoehr. Calibration of conditional composite likelihood for Bayesian inference on Gibbs random fields. pages 1–16, 2015. ISSN 15337928.

Cristiano Varin and Claudia Czado. A mixed autoregressive probit model for ordinal longitudinal data. *Biostatistics*, 11(1):127–138, 2010. ISSN 14654644. doi: 10.1093/biostatistics/kxp042.

Cristiano Varin, Nancy Reid, and David Firth. An overview of composite likelihood methods. *Statistica Sinica*, 21(2008):5–42, 2011. ISSN 10170405. URL http://www3.stat.sinica.edu.tw/statistica/j21n1/J21N11/J21N11.html.

Michael D Ward, John S Ahlquist, and Arturas Rozenas. Gravity's rainbow: a dynamic latent space model for the world trade network. *Network Science*, 1(01):95–118, 2013.

Halbert White. A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica: Journal of the Econometric Society*, pages 817–838, 1980.

Eric P. Xing, Wenjie Fu, and Le Song. A state-space mixed membership blockmodel for dynamic network tomography. *Ann. Appl. Stat.*, 4(2):535–566, 2010. ISSN 1932-6157. doi: 10.1214/09-AOAS311. URL http://dx.doi.org/10.1214/09-AOAS311.

Hao Yin, Austin R Benson, Jure Leskovec, and David F Gleich. Local higher-order graph clustering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 555–564. ACM, 2017.

R. Zafarani and H. Liu. Social computing data repository at ASU, 2009. URL http://socialcomputing.asu.edu.

# Appendix A

## *A.1 Gibbs sampler for probit MCR*

When $y_{ij,t}$'s are ordinal-valued, we model the network with a probit link and assume $y_{ij,t} = f(z_{ij,t})$. The Gibbs sampler for the probit MCR then include the Steps 1-4 as described in Section 2.3.2 and also an addition step to update the $z_{ij,t}$'s.

Denote

$$\begin{cases} \mathbf{h}_{i,t+1}^{-j} \equiv \mathbf{x}_{i,t+1} - \boldsymbol{\theta}_i - \mathbf{A}^T\mathbf{x}_{i,t} - \mathbf{C}\mathbf{X}_{-j\cdot,t}^T\mathbf{z}_{i,-j,t}, \\ \mathbf{h}_{j,t+1}^{-i} \equiv \mathbf{x}_{j,t+1} - \boldsymbol{\theta}_j^X - \mathbf{A}^T\mathbf{x}_{j,t} - \mathbf{C}\mathbf{X}_{-i\cdot,t}^T\mathbf{z}_{-i,j,t}, \\ \mathbf{R}_t \equiv \mathbf{M}_t + \mathbf{X}_t\mathbf{H}\mathbf{X}_t^T, \end{cases}$$

where $\mathbf{X}_{-i\cdot,t}$ refers to the sub-matrix of $\mathbf{X}_t$ with the $i$-th row removed and $\mathbf{M}_t$ refers to the matrix of $\{\mu_{ij}\}$. Then the equations related to $z_{ij,t}$ include

$$\begin{aligned} z_{ij,t} &= \alpha z_{ij,t-1} + r_{ij,t-1} + \epsilon_{ij,t}, \\ z_{ij,t+1} &= \alpha z_{ij,t} + r_{ij,t} + \epsilon_{ij,t+1}, \\ \mathbf{h}_{i,t+1}^{-j} &= z_{ij,t}\mathbf{C}\mathbf{x}_{i,t} + \mathbf{e}_{i,t+1}, \\ \mathbf{h}_{j,t+1}^{-i} &= z_{ij,t}\mathbf{C}\mathbf{x}_{j,t} + \mathbf{e}_{j,t+1}. \end{aligned} \tag{A.1}$$

Given the prior $z_{ij,t} \sim \mathrm{N}(\mu_0, \sigma_0^2)$, the full conditional distribution of $z_{ij,t}$ is given by $\mathrm{N}(\mu_{\mathrm{ij,t}}, \sigma_{\mathrm{ij,t}}^2)$, where

$$\sigma_{ij,t}^2 = \left(\frac{1+\alpha^2}{\sigma^2} + \mathbf{x}_{i,t}^T\mathbf{C}^T\boldsymbol{\Sigma}^{-1}\mathbf{C}\mathbf{x}_{i,t} + \mathbf{x}_{j,t}^T\mathbf{C}^T\boldsymbol{\Sigma}^{-1}\mathbf{C}\mathbf{x}_{j,t} + \frac{1}{\sigma_0^2}\right)^{-1},$$

$$\mu_{ij,t} = \sigma_{ij,t}^2\left(\frac{\alpha z_{ij,t-1} + r_{ij,t-1} + \alpha(z_{ij,t+1} - r_{ij,t})}{\sigma^2} + \mathbf{x}_{i,t}^T\mathbf{C}^T\boldsymbol{\Sigma}^{-1}\mathbf{h}_{i,t+1}^{-j} + \mathbf{x}_{j,t}^T\mathbf{C}^T\boldsymbol{\Sigma}^{-1}\mathbf{h}_{j,t+1}^{-i} + \frac{\mu_0}{\sigma_0^2}\right).$$

However, we cannot directly sample from the full conditional distribution due to the restriction of $y_{ij,t}$. Luckily, we only need to restrict our sampling to the interval $[z_{ij,t}^-, z_{ij,t}^+]$,

rather than change the full conditionals. The idea for getting the intervals is that the upper bound cannot exceed the minimum value among all the entries of $\mathbf{Z}$ whose corresponding entries in $\mathbf{Y}$ is higher than $y_{ij,t}$. Similarly, the lower bound is determined by the maximum of those with values in $\mathbf{Y}$ lower than $y_{ij,t}$, i.e.,

$$\begin{cases} z^+_{ij,t} = \min\{z_{kl,s} : y_{kl,s} > y_{ij,t}; k, l \in \{1, \cdots, m\}, k \neq l, s \in \{1, \cdots, n\}, (k, l, s) \neq (i, j, t)\}, \\ z^-_{ij,t} = \max\{y_{kl,s} : y_{kl,s} < y_{ij,t}; k, l \in \{1, \cdots, m\}, k \neq l, s \in \{1, \cdots, n\}, (k, l, s) \neq (i, j, t)\}. \end{cases}$$

This method is known as rank likelihood, which is introduced in Hoff [2009b]. In this book, the author also presents an alternative way that works for ordinal data with ranks $\{1, \cdots, q\}$. Using this method, we need to update the thresholds $\{h_0, \cdots, h_q\}$ during the Gibbs sampler procedure. Assume a prior for all the thresholds and during the iteration, the threshold $h_s$ is sampled according to the interval $[h^-_s, h^+_s]$, where $h^+_s = \min\{y_{kl,s} : y_{kl,s} = s + 1\}$ and $h^-_s = \max\{y_{kl,s} : y_{kl,s} = s\}$.

To estimate the parameters and latent variables in the model with both ordinal networks and ordinal nodal attributes, we can still follow the Gibbs sampler procedure discussed here, along with an extra step to update $w_{i,k,t}$. In each iteration, we can sample a new point for $w_{i,k,t}$ from its full conditional distribution with restrict to interval $[w^-_{i,k,t}, w^+_{i,k,t}]$. The boundaries/thresholds can be obtained using rank likelihood or sampled from their full conditionals.

### *A.2 Full Conditionals of* $\mathbf{u}$ *and* $\mathbf{v}$ *in the AME Model*

In the AME model, we can write the model of $\mathbf{z}$ as follows:

$$\begin{aligned} \mathbf{z} &= \mu\mathbf{1} + (\mathbf{V} \otimes \mathbf{I}_m)\,\mathbf{u} + \boldsymbol{\epsilon} \\ &= \mu\mathbf{1} + (\mathbf{I}_m \otimes \mathbf{U})\,\tilde{\mathbf{v}} + \boldsymbol{\epsilon}, \end{aligned} \tag{A.2}$$

where $\boldsymbol{\epsilon} \sim (\mathbf{0}, \boldsymbol{\Sigma}(\boldsymbol{\phi}))$, and $\tilde{\mathbf{v}}$ is the vectorization of $\mathbf{V}^T$. Given $\mathbf{z}$ and $\mathbf{V}$, we have the following models that are associated with $\mathbf{u}$:

$$\begin{aligned} \mathbf{z}|\mathbf{u}, \mathbf{v} &\sim \mathrm{N}\left(\mu\mathbf{1} + (\mathbf{V} \otimes \mathbf{I}_m)\,\mathbf{u}, \boldsymbol{\Sigma}(\boldsymbol{\phi})\right), \\ \mathbf{u} &\sim \mathrm{N}\left(\mathbf{0}, \mathrm{diag}(\tau_1^2, \cdots, \tau_R^2) \otimes \mathbf{I}_m\right). \end{aligned}$$

This indicates that the full conditional of $\mathbf{u}$ is

$$\begin{aligned} p(\mathbf{u}|\mathbf{z},\mathbf{v}) &\propto p(\mathbf{z}|\mathbf{u},\mathbf{v})p(\mathbf{u}) \\ &\propto \exp\{-\frac{1}{2}(\mathbf{z}-\mu\mathbf{1}-(\mathbf{V}\otimes\mathbf{I}_m)\,\mathbf{u})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\phi})(\mathbf{z}-\mu\mathbf{1}-(\mathbf{V}\otimes\mathbf{I}_m)\,\mathbf{u}) \\ &\qquad -\frac{1}{2}\mathbf{u}^T(\mathrm{diag}(\tau_1^2,\cdots,\tau_R^2)\otimes\mathbf{I}_m)^{-1}\mathbf{u}\} \\ &\propto \exp\{(\mathbf{u}-\boldsymbol{\mu}_u)^T\boldsymbol{\Sigma}_u^{-1}(\mathbf{u}-\boldsymbol{\mu}_u)\}, \end{aligned} \tag{A.3}$$

where

$$\begin{aligned} \boldsymbol{\Sigma}_u &= \left[(\mathbf{V}\otimes\mathbf{I}_m)^T\,\boldsymbol{\Sigma}^{-1}(\boldsymbol{\phi})\,(\mathbf{V}\otimes\mathbf{I}_m)+(\mathrm{diag}(\tau_1^2,\cdots,\tau_R^2)\otimes\mathbf{I}_m)^{-1}\right]^{-1}, \\ \boldsymbol{\mu}_u &= \boldsymbol{\Sigma}_u^{-1}\,(\mathbf{V}\otimes\mathbf{I}_m)^T\,\boldsymbol{\Sigma}^{-1}(\boldsymbol{\phi})(\mathbf{z}-\mu\mathbf{1}). \end{aligned} \tag{A.4}$$

Therefore during the Gibbs sampling, we update $\mathbf{u}$ by sampling from the full conditional distribution $\mathrm{N}(\boldsymbol{\mu}_u,\boldsymbol{\Sigma}_u)$, with $\boldsymbol{\mu}_u$ and $\boldsymbol{\Sigma}_u$ defined in Equations (A.8). Similarly, the full conditional of $\tilde{\mathbf{v}}$ is given as

$$\begin{aligned} &\tilde{\mathbf{v}}|\mathbf{z},\mathbf{u}\sim\mathrm{N}(\boldsymbol{\mu}_v,\boldsymbol{\Sigma}_v), \\ &\boldsymbol{\Sigma}_v = \left[(\mathbf{I}_m\otimes\mathbf{U})^T\,\boldsymbol{\Sigma}^{-1}(\boldsymbol{\phi})\,(\mathbf{I}_m\otimes\mathbf{U})+(\mathrm{diag}(\tau_1^2,\cdots,\tau_R^2)\otimes\mathbf{I}_m)^{-1}\right]^{-1}, \\ &\boldsymbol{\mu}_v = \boldsymbol{\Sigma}_v^{-1}\,(\mathbf{I}_m\otimes\mathbf{U})^T\,\boldsymbol{\Sigma}^{-1}(\boldsymbol{\phi})(\mathbf{z}-\mu\mathbf{1}). \end{aligned} \tag{A.5}$$

### *A.3 Second Derivative of the Composite Likelihood of the AME Model*

Since the covariance $\boldsymbol{\Sigma}(\boldsymbol{\phi})$ is associated with $\boldsymbol{\phi}_1 = (\sigma_a^2,\sigma_b^2,\rho_{ab},\rho_e)$ and the variance in $p(\mathbf{u},\mathbf{v}|\boldsymbol{\theta})$ is only associated with $\boldsymbol{\phi}_2=(\tau_1^2,\cdots,\tau_r^2)$. Therefore, the second derivatives can be

calculated as

$$
\begin{aligned}
\frac{\partial^2}{\partial\mu^2}\ell(\boldsymbol{\theta}:\mathbf{y}) &= -2\times\mathbf{1}^T\boldsymbol{\Lambda}\mathbf{1}+2\mathrm{Cov}\left[\mathbf{1}^T\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\mathbf{w}|\mathbf{y}\right],\\
\frac{\partial^2}{\partial\mu\partial\phi_{1,i}}\ell(\boldsymbol{\theta}:\mathbf{y}) &= 2\mathrm{E}\left[\mathbf{1}^T\frac{\partial}{\partial\phi_{1,i}}\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\mathbf{w}|\mathbf{y}\right]-\mathrm{Cov}\left[\mathbf{1}^T\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\mathbf{w},\mathbf{w}^T\frac{\partial}{\partial\phi_{1,i}}\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\mathbf{w}|\mathbf{y}\right],\\
\frac{\partial^2}{\partial\phi_{1,i}\partial\phi_{1,j}}\ell(\boldsymbol{\theta}:\mathbf{y}) &= \mathrm{tr}\left(\frac{\partial}{\partial\phi_{1,i}}\boldsymbol{\Sigma}(\boldsymbol{\phi}_1)\frac{\partial}{\partial\phi_{1,j}}\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\right)+\mathrm{tr}\left(\boldsymbol{\Sigma}(\boldsymbol{\phi}_1)\frac{\partial^2}{\partial\phi_{1,i}\partial\phi_{1,j}}\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\right)\\
&\quad+\mathrm{E}\left[\mathbf{w}^T\frac{\partial^2}{\partial\phi_{1,i}\partial\phi_{1,j}}\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\mathbf{w}|\mathbf{y}\right]\\
&\quad-\frac{1}{2}\mathrm{E}\left[\mathbf{w}^T\frac{\partial}{\partial\phi_{1,i}}\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\mathbf{w},\mathbf{w}^T\frac{\partial}{\partial\phi_{1,j}}\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\mathbf{w}|\mathbf{y}\right],\\
\frac{\partial^2}{\partial\mu\partial\tau_i^2}\ell(\boldsymbol{\theta}:\mathbf{y}) &= -\mathrm{Cov}\left[\mathbf{1}^T\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\mathbf{w},\mathbf{U}\frac{\partial\left[\mathrm{diag}(\tau_1^2,\cdots,\tau_r^2)\right]^{-1}}{\partial\tau_i^2}\mathbf{V}^T|\mathbf{y}\right],\\
\frac{\partial^2}{\partial\phi_{1,i}\partial\tau_i^2}\ell(\boldsymbol{\theta}:\mathbf{y}) &= -\frac{1}{2}\mathrm{Cov}\left[\mathbf{w}^T\frac{\partial}{\partial\phi_{1,j}}\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\mathbf{w},\mathbf{U}\frac{\partial\left[\mathrm{diag}(\tau_1^2,\cdots,\tau_r^2)\right]^{-1}}{\partial\tau_i^2}\mathbf{V}^T|\mathbf{y}\right],\\
\frac{\partial^2}{\partial\tau_i^2\partial\tau_j^2}\ell(\boldsymbol{\theta}:\mathbf{y}) &= \frac{m}{\tau_i^4}\times 1_{i=j}+\mathrm{E}\left[\mathbf{U}\frac{\partial^2\left[\mathrm{diag}(\tau_1^2,\cdots,\tau_r^2)\right]^{-1}}{\partial\tau_i^2\partial\tau_j^2}\mathbf{V}^T|\mathbf{y}\right]\\
&\quad-\frac{1}{2}\mathrm{E}\left[\mathbf{U}\frac{\partial\left[\mathrm{diag}(\tau_1^2,\cdots,\tau_r^2)\right]^{-1}}{\partial\tau_i^2}\mathbf{V}^T,\mathbf{U}\frac{\partial\left[\mathrm{diag}(\tau_1^2,\cdots,\tau_r^2)\right]^{-1}}{\partial\tau_j^2}\mathbf{V}^T|\mathbf{y}\right],
\end{aligned}
$$

where $\mathbf{w}=\mathbf{z}-\mu\mathbf{1}-(\mathbf{V}\otimes\mathbf{U})\mathbf{i}$.

### *A.4 Full Conditionals of* u *and* v *in the AME Model with Covariate*

In the AME model with covariates, we can write the model of $\mathbf{z}$ as follows:

$$
\begin{aligned}
\mathbf{z} &= \tilde{\mathbf{X}}\boldsymbol{\beta}+(\mathbf{V}\otimes\mathbf{I}_m)\,\mathbf{u}+\boldsymbol{\epsilon}\\
&= \tilde{\mathbf{X}}\boldsymbol{\beta}+(\mathbf{I}_m\otimes\mathbf{U})\,\tilde{\mathbf{v}}+\boldsymbol{\epsilon},
\end{aligned}
\tag{A.6}
$$

where $\boldsymbol{\epsilon}\sim(\mathbf{0},\boldsymbol{\Sigma}(\boldsymbol{\phi}))$, and $\tilde{\mathbf{v}}$ is the vectorization of $\mathbf{V}^T$. Given $\mathbf{z}$ and $\mathbf{V}$, we have the following models that are associated with $\mathbf{u}$:

$$
\begin{aligned}
\mathbf{z}|\mathbf{u},\mathbf{v} &\sim N_{m^2}\left(\tilde{\mathbf{X}}\boldsymbol{\beta}+(\mathbf{V}\otimes\mathbf{I}_m)\,\mathbf{u},\boldsymbol{\Sigma}(\boldsymbol{\phi})\right),\\
\mathbf{u} &\sim N_{mr}\left(\mathbf{0},\mathrm{diag}(\tau_1^2,\cdots,\tau_R^2)\otimes\mathbf{I}_m\right).
\end{aligned}
$$

This indicates that the full conditional of $\mathbf{u}$ is

$$\begin{aligned} p(\mathbf{u}|\mathbf{z},\mathbf{v}) &\propto p(\mathbf{z}|\mathbf{u},\mathbf{v})p(\mathbf{u}) \\ &\propto \exp\{-\frac{1}{2}(\mathbf{z}-\tilde{\mathbf{X}}\boldsymbol{\beta}-(\mathbf{V}\otimes\mathbf{I}_m)\,\mathbf{u})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\phi})(\mathbf{z}-\tilde{\mathbf{X}}\boldsymbol{\beta}-(\mathbf{V}\otimes\mathbf{I}_m)\,\mathbf{u}) \\ &\qquad -\frac{1}{2}\mathbf{u}^T(\mathrm{diag}(\tau_1^2,\cdots,\tau_R^2)\otimes\mathbf{I}_m)^{-1}\mathbf{u}\} \\ &\propto \exp\{(\mathbf{u}-\boldsymbol{\mu}_u)^T\boldsymbol{\Sigma}_u^{-1}(\mathbf{u}-\boldsymbol{\mu}_u)\}, \end{aligned} \tag{A.7}$$

where

$$\begin{aligned} \boldsymbol{\Sigma}_u &= \left[(\mathbf{V}\otimes\mathbf{I}_m)^T\,\boldsymbol{\Sigma}^{-1}(\boldsymbol{\phi})\,(\mathbf{V}\otimes\mathbf{I}_m)+(\mathrm{diag}(\tau_1^2,\cdots,\tau_R^2)\otimes\mathbf{I}_m)^{-1}\right]^{-1}, \\ \boldsymbol{\mu}_u &= \boldsymbol{\Sigma}_u^{-1}\,(\mathbf{V}\otimes\mathbf{I}_m)^T\,\boldsymbol{\Sigma}^{-1}(\boldsymbol{\phi})(\mathbf{z}-\tilde{\mathbf{X}}\boldsymbol{\beta}). \end{aligned} \tag{A.8}$$

Therefore during the Gibbs sampling, we update $\mathbf{u}$ by sampling from the full conditional distribution $N_{mr}(\boldsymbol{\mu}_u,\boldsymbol{\Sigma}_u)$, with $\boldsymbol{\mu}_u$ and $\boldsymbol{\Sigma}_u$ defined in Equations (A.8). Similarly, the full conditional of $\tilde{\mathbf{v}}$ is given as

$$\begin{aligned} &\tilde{\mathbf{v}}|\mathbf{z},\mathbf{u}\sim N_{mr}(\boldsymbol{\mu}_v,\boldsymbol{\Sigma}_v), \\ &\boldsymbol{\Sigma}_v = \left[(\mathbf{I}_m\otimes\mathbf{U})^T\,\boldsymbol{\Sigma}^{-1}(\boldsymbol{\phi})\,(\mathbf{I}_m\otimes\mathbf{U})+(\mathrm{diag}(\tau_1^2,\cdots,\tau_R^2)\otimes\mathbf{I}_m)^{-1}\right]^{-1}, \\ &\boldsymbol{\mu}_v = \boldsymbol{\Sigma}_v^{-1}\,(\mathbf{I}_m\otimes\mathbf{U})^T\,\boldsymbol{\Sigma}^{-1}(\boldsymbol{\phi})(\mathbf{z}-\tilde{\mathbf{X}}\boldsymbol{\beta}). \end{aligned} \tag{A.9}$$

### *A.5 Second Derivative of the Composite Likelihood of the AME Model with Covariates*

Since the covariance $\boldsymbol{\Sigma}(\boldsymbol{\phi})$ is associated with $\boldsymbol{\phi}_1 = (\sigma_a^2,\sigma_b^2,\rho_{ab},\rho_e)$ and the variance in $p(\mathbf{u},\mathbf{v}|\boldsymbol{\theta})$ is only associated with $\boldsymbol{\phi}_2=(\tau_1^2,\cdots,\tau_r^2)$. Therefore, the second derivatives can be

calculated as

$$
\begin{aligned}
\frac{\partial^2}{\partial\beta_i\beta_j}\ell(\boldsymbol{\theta}:\mathbf{y}) &= -2\times\mathbf{x}_i^T\boldsymbol{\Lambda}\mathbf{x}_j + 2\mathrm{Cov}\left[\mathbf{x}_i^T\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\mathbf{w}, \mathbf{x}_j^T\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\mathbf{w}|\mathbf{y}\right], \\
\frac{\partial^2}{\partial\beta_i\partial\phi_{1,i}}\ell(\boldsymbol{\theta}:\mathbf{y}) &= 2\mathrm{E}\left[\mathbf{x}_i^T\frac{\partial}{\partial\phi_{1,i}}\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\mathbf{w}|\mathbf{y}\right] - \mathrm{Cov}\left[\mathbf{x}_i^T\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\mathbf{w}, \mathbf{w}^T\frac{\partial}{\partial\phi_{1,i}}\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\mathbf{w}|\mathbf{y}\right], \\
\frac{\partial^2}{\partial\phi_{1,i}\partial\phi_{1,j}}\ell(\boldsymbol{\theta}:\mathbf{y}) &= \mathrm{tr}\left(\frac{\partial}{\partial\phi_{1,i}}\boldsymbol{\Sigma}(\boldsymbol{\phi}_1)\frac{\partial}{\partial\phi_{1,j}}\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\right) + \mathrm{tr}\left(\boldsymbol{\Sigma}(\boldsymbol{\phi}_1)\frac{\partial^2}{\partial\phi_{1,i}\partial\phi_{1,j}}\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\right) \\
&\quad + \mathrm{E}\left[\mathbf{w}^T\frac{\partial^2}{\partial\phi_{1,i}\partial\phi_{1,j}}\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\mathbf{w}|\mathbf{y}\right] \\
&\quad - \frac{1}{2}\mathrm{E}\left[\mathbf{w}^T\frac{\partial}{\partial\phi_{1,i}}\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\mathbf{w}, \mathbf{w}^T\frac{\partial}{\partial\phi_{1,j}}\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\mathbf{w}|\mathbf{y}\right], \\
\frac{\partial^2}{\partial\beta_i\partial\tau_i^2}\ell(\boldsymbol{\theta}:\mathbf{y}) &= -\mathrm{Cov}\left[\mathbf{x}_i^T\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\mathbf{w}, \mathbf{U}\frac{\partial\left[\mathrm{diag}(\tau_1^2,\cdots,\tau_r^2)\right]^{-1}}{\partial\tau_i^2}\mathbf{V}^T|\mathbf{y}\right], \\
\frac{\partial^2}{\partial\phi_{1,i}\partial\tau_i^2}\ell(\boldsymbol{\theta}:\mathbf{y}) &= -\frac{1}{2}\mathrm{Cov}\left[\mathbf{w}^T\frac{\partial}{\partial\phi_{1,j}}\boldsymbol{\Lambda}(\boldsymbol{\phi}_1)\mathbf{w}, \mathbf{U}\frac{\partial\left[\mathrm{diag}(\tau_1^2,\cdots,\tau_r^2)\right]^{-1}}{\partial\tau_i^2}\mathbf{V}^T|\mathbf{y}\right], \\
\frac{\partial^2}{\partial\tau_i^2\partial\tau_j^2}\ell(\boldsymbol{\theta}:\mathbf{y}) &= \frac{m}{\tau_i^4}\times 1_{i=j} + \mathrm{E}\left[\mathbf{U}\frac{\partial^2\left[\mathrm{diag}(\tau_1^2,\cdots,\tau_r^2)\right]^{-1}}{\partial\tau_i^2\partial\tau_j^2}\mathbf{V}^T|\mathbf{y}\right] \\
&\quad - \frac{1}{2}\mathrm{E}\left[\mathbf{U}\frac{\partial\left[\mathrm{diag}(\tau_1^2,\cdots,\tau_r^2)\right]^{-1}}{\partial\tau_i^2}\mathbf{V}^T, \mathbf{U}\frac{\partial\left[\mathrm{diag}(\tau_1^2,\cdots,\tau_r^2)\right]^{-1}}{\partial\tau_j^2}\mathbf{V}^T|\mathbf{y}\right],
\end{aligned}
$$

where $\mathbf{w} = \mathbf{z} - \langle\boldsymbol{\beta}, \mathbf{X}\rangle - (\mathbf{V}\otimes\mathbf{U})\mathbf{i}$ and $\mathbf{x}_i$ is the vectorization of the $i$-th covariate matrix.

# VITA

Yanjun He grew up in Shanghai, China. She went to Peking University in Beijing, China for a Bachelors in Mathematics and Economics. She obtained a Masters in Statistics from the University of Washington in December 2014. In March 2018, she obtained a PhD in Statistics from the University of Washington.