Output Analysis of Monte Carlo Methods with Applications to Networks and Functional Approximation

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Haema Nilakanta

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Galin L. Jones, Adviser

February, 2020

Acknowledgements

First and foremost, I have to express gratitude to my advisor, Dr. Galin Jones. He accepted me as his student while I was still trying to find my bearings in the department and gave me the guidance, room, and support to think about problems in a different way. He also reaffirmed in me the importance of the Oxford comma.

I am also grateful to my dissertation committee: Drs. Snigdanshu Chatterjee, Adam Rothman, and Katerina Marcoulides for providing me insightful and helpful comments to improve my research. I would also not be able to complete a dissertation if it weren't for the tremendous help of the School of Statistics' office staff, namely Taryn and Taylor.

It's the people at the School of Statistics who were key in my decision to join the program, and it's the people who helped me both succeed and enjoy my time here. There are too many to name, but I can't express enough how grateful I am for the friendships I've made and that, I hope, will continue beyond this campus. Thank you: Adam, Chenglong, Christina, Dan, Dootika, James, Karl Oskar, Kaibo, Lindsey, Megan, Mitch, Sakshi, Sarah, Sanhita, Si, Sijia, and Xuetong.

I'm also grateful for the endless encouragement from Jessica Oster and Poojya Anantharam. And to Paul Beamer whose partnership and support has been immeasurable. Also to my brother Sidhartha, sister-in-law Cristina, and niece Asha: thank you for your support and bringing such joy to the last few years of my program.

No achievements happen alone and any of mine are because of my parents. Ma and Dad, I can't thank you enough for all your support, guidance, and encouragement. I hope I've made you proud.

Dedication

To my parents

Abstract

The overall objective of the Monte Carlo method is to use data simulated in a computer to learn about complex systems. This is a highly flexible approach and can be applied in a variety of settings. For instance, Monte Carlo methods are used to estimate network properties or to approximate functions. Although the use of these methods in such cases is common, little to no work exists on assessing the reliability of the estimation procedure. Thus, the contribution of this work lies in further developing methods to better address the reliability of Monte Carlo estimation, particularly with respect to estimating network properties and approximating functions.

In network analysis, there exist many networks which can only be studied via sampling methods due to the scale or complexity of the network, access limitations, or the population of interest is hard to reach. In such cases, the application of random walk-based Markov chain Monte Carlo (MCMC) methods to estimate multiple network features is common. However, the reliability of these estimates has been largely ignored. We consider and further develop multivariate MCMC output analysis methods in the context of network sampling to directly address the reliability of the multivariate estimation. This approach yields principled, computationally efficient, and broadly applicable methods for assessing the Monte Carlo estimation procedure.

We also study the Monte Carlo estimation reliability in approximating functions using Importance Sampling. Although we focus on approximating difficult to compute density and log-likelihood functions, we develop a general framework for constructing simultaneous confidence bands that could be applied in other contexts. In addition, we propose a correction to improve the reliability of the log-likelihood function estimation using the Monte Carlo Likelihood Approximation approach.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The overall objective of the Monte Carlo method is to learn about some system or solve a problem by simulation. It is a general, highly flexible, and powerful approach to solve a variety of problems across various disciplines (Owen, 2013).

We can use Monte Carlo methods to estimate several features of interest, such as but not limited to: population means, quantiles, probabilities, and density functions. Instead of obtaining data through an experiment, the Monte Carlo method first generates realizations of a random variable through computer simulation and then uses these realizations to estimate features of interest.

That means there are two core components to implement the Monte Carlo method: 1) designing the algorithm or sampling scheme to simulate data and 2) using this simulated data to reliably estimate features of interest. A fundamental concern when using any Monte Carlo method is the reliability of the procedure. Since there is a simulation step, each time the method is run, there will be slightly different results. So how can we be assured that the results we obtain are reliable? To ensure reliability of the results, further attention needs to be given to calculating the error in the estimation. The contribution of this work lies in further developing methods to better address the reliability of Monte Carlo approximation, particularly with respect to estimating network properties and approximating functions.

We first present an introduction to Monte Carlo methods. Although an incredibly vast field, we highlight fundamental topics in the area. The remaining of this chapter is organized as follows: In Section 1 we introduce the Monte Carlo method with notation. In Section 2 we discuss various methods to simulate a representative sample from a system of interest. In cases when more simple Monte Carlo approaches fail, we can use Markov chain Monte Carlo, which we introduce in Section 3. Then in Section 4 we set up how to use the Monte Carlo sample for estimation. With an in depth discussion of Monte Carlo estimation using Importance Sampling in Section 5 and its extensions into functional approximation in Section 6. We then discuss output analysis in Section 7 for both independent and correlated samples, in particular the work done thus far on assessing Monte Carlo error and using it to determine simulation effort. Lastly, in Section 8, we provide a road map for the remainder of this thesis and how in subsequent chapters we will apply and extend reliability measures of these Monte Carlo methods with applications to networks and functional approximation.

1.1 Introduction to Monte Carlo Methods

Suppose the distribution F has support $\mathcal{X}$ with probability function (pf) $f(x)$. The goal of Monte Carlo methods is to estimate $\theta \in \mathbb{R}^p$, a p-dimensional vector of fixed, unknown features of F, where $p \geq 1$.

Most often we can express these features of interest in the form of expectations and quantiles. For example, let $g : \mathcal{X} \to \mathbb{R}$ and define

$$\mu = \mathbb{E}_F[g(X)] = \int_{\mathcal{X}} g(x)F(dx) \tag{1.1}$$

where μ may be a feature of interest. Another feature we can estimate is quantiles. Let $Y = g(X)$ and let F_Y denote the distribution of Y. If $q \in (0, 1)$, the qth quantile is

$$\xi_q = F_Y^{-1}(q) = \inf\{y : F_Y(y) \geq q\} \tag{1.2}$$

where F^{-1} is the quantile function.

The first task of Monte Carlo methods is the simulation effort to obtain m representative samples from a *target distribution*, F. In many situations it is challenging to sample from this target directly so we employ different sampling strategies. After we obtain these samples, we use them to estimate the features of interest of F.

Throughout we will let m denote the size of the simulation effort, that is, the Monte Carlo sample size while we will use n to denote the sample size associated with the original statistical setting (e.g., observed data size). In addition, we will specify the use of "multivariate" either with respect to simulation or estimation. Multivariate simulation refers to when we simulate $X \in \mathbb{R}^d$ where $d > 1$. Multivariate estimation refers to when we estimate a vector of features, $\theta \in \mathbb{R}^p$ where $p > 1$. This distinction is important as it will shape the sampling, estimation, or analysis procedures we use.

Before we begin to introduce the various ways to draw samples from any target F, there are two fundamental theorems we will use routinely and are worth mentioning early on: Taylor's theorem (Casella and Berger, 2002) and the Multivariate Delta Method (Lehmann, 1999). We assume knowledge of Taylor's theorem but restate a form of the Multivariate Delta Method below:

Theorem 1.1.1. *Suppose that* $Y_m = (Y_{m1}, Y_{m2}, \ldots, Y_{mk_1})$ *is a sequence of random vectors with population mean vector* μ_Y *and* $k_1 \times k_1$ *positive definite covariance matrix* Σ *such that as* $m \to \infty$,

$$\sqrt{m}(Y_m - \mu_Y) \xrightarrow{d} N_{k_1}(0, \Sigma).$$

Let $g : \mathbb{R}^{k_1} \to \mathbb{R}^{k_2}$, *if all the* k_2 *entries of* g *have continuous partial derivatives at* μ_Y *then, as* $m \to \infty$,

$$\sqrt{m}(g(Y_m) - g(\mu_Y)) \xrightarrow{d} N_{k_2}(0, \nabla g(\mu_Y)^T \Sigma \nabla g(\mu_Y))$$

where $\nabla g(\mu_Y)$ *is the* $k_2 \times k_1$ *matrix of partial derivatives of* g *evaluated at* μ_Y.

1.2 Simulating a Representative Sample

Monte Carlo methods rely on the ability to draw independent and identically distributed (iid) samples from a standard Uniform distribution (abbreviated as $U(0,1)$). We consider several fundamental methods for using these $U(0,1)$ realizations to obtain realizations of a random variable from a non-uniform target distribution, F. If we were able to directly draw iid samples from F, we refer to this as *classical, direct, simple*, or *crude Monte Carlo* (Owen, 2013). Unless mentioned, assume we cannot sample directly from the target distribution.

1.2.1 Inversion Method

Suppose $X \sim F$ and the quantile function, F^{-1}, is known. If $U \sim U(0,1)$ then $X = F^{-1}(U)$. Therefore to obtain $X_1, \ldots, X_m \overset{iid}{\sim} F$, we can first simply draw $U_1, \ldots, U_m \overset{iid}{\sim} U(0,1)$ and compute $X_t = F^{-1}(U_t)$ for $t = 1, \ldots, m$.

Although straightforward, the inversion method requires a known form of F^{-1}. In many cases this is not possible.

1.2.2 Accept-Reject Algorithm

Suppose we do not know F^{-1}, but we can sample from a more convenient distribution F^* with support containing $\mathcal{X}$ and corresponding pf f^*. We refer to F^* as the *proposal distribution*. Suppose,

$$M = \sup_{x \in \mathcal{X}} \frac{f(x)}{f^*(x)} < \infty. \tag{1.3}$$

Then the Accept-Reject algorithm is,

1. Generate $X \sim F^*$ and $U \sim U(0,1)$ independently. Denote the observed values x and u respectively.
2. If $u < \frac{f(x)}{Mf^*(x)}$, then accept x as a draw from F; otherwise reject and return to step 1.

Notice that the probability of accepting a draw is,

$$Pr\left(U \leq \frac{f(X)}{Mf^*(X)}\right) = \mathbb{E}\left[\mathbb{E}\left(\mathbb{I}\left(U \leq \frac{f(X)}{Mf^*(X)}\right)|X\right)\right] = \mathbb{E}\left[\frac{f(X)}{Mf^*(X)}\right] = \frac{1}{M}.$$

Therefore, smaller values of M lead to a more efficient algorithm. To achieve a small M, the proposal should be chosen so that it is similar to $f(x)$, particularly in its tails. The algorithm can also be used if the normalizing constants of f and f^* are unknown.

1.2.3 Linchpin-Variable

In multivariate simulation, i.e., $X \subseteq \mathbb{R}^d$ where $d > 1$, the Accept-Reject algorithm can be difficult. Suppose we can split $X = (Y, Z)$ and let $f(y, z)$ be the pf of interest such that $y \in \mathbb{R}^{d_1}$, $z \in \mathbb{R}^{d_2}$, where $d = d_1 + d_2$. If,

$$f(y, z) = f_{Y|Z}(y|z)f_Z(z)$$

where we can sample from the distribution of $Y|Z$, we say Z is the *linchpin variable* (Huber, 2016). Then to generate realizations from F we can use the Linchpin-Variable algorithm:

1. Draw $Z \sim F_Z$, denote the observed value as z
2. Generate Y from $F_{Y|Z=z}$

The first step may require an additional Accept-Reject. Notice, the algorithm is only useful if it is easier to sample from F_Z then it is from F directly.

1.3 MCMC

There are also many situations where drawing iid samples from F simply is not possible. This either may be due to the complexity of the target distribution or it may be prohibitively expensive to collect a set of iid samples. Alternatively, we can turn to another Monte Carlo method, namely Markov Chain Monte Carlo (MCMC). MCMC

has skyrocketed in popularity especially with more readily available computational resources. It is often associated with Bayesian statistics but is used in many statistical applications (Brooks et al., 2011).

The first task of MCMC still remains to simulate a representative sample from F. Now instead of collecting iid samples from this target, we draw a correlated sample which is a realization of a Markov chain. Notice, Classical Monte Carlo is actually a special case of MCMC where the Markov chain is an iid sequence from the target distribution. Before we discuss popular MCMC methods we first define Markov chains and introduce some basic properties.

1.3.1 Basic Introduction to Markov Chains

Consider a stochastic process $\{X_0^*, X_1^*, \ldots\}$ defined over a general state space, $\mathcal{X}$ and let $\mathcal{B}(\mathcal{X})$ be the associated sigma-algebra. This sequence is a *Markov chain* if it satisfies the *Markov property*,

$$Pr(X_{m+1}^* = x_{m+1} | X_m^* = x_m, X_{m-1}^* = x_{m-1}, \ldots, X_0^* = x_0) = Pr(X_{m+1}^* = x_{m+1} | X_m^* = x_m).$$

Then the *Markov transition kernel*, P, is a map $P : \mathcal{X} \times \mathcal{B}(\mathcal{X}) \to [0, 1]$ where for all $A \in \mathcal{B}(\mathcal{X})$,

$$P(x, A) = Pr(X_m^* \in A | X_{m-1}^* = x)$$

such that $P(\cdot, A)$ is measurable on $\mathcal{X}$ and for all $x \in \mathcal{X}$, $P(x, \cdot)$ is a probability measure on $\mathcal{B}(\mathcal{X})$.

Then

$$P^m(x, A) = Pr(X_m^* \in A | X_0^* = x)$$

is the m-step transition probability.

The Markov chain, $\{X_m^*\}$, on $\mathcal{X}$ is *reversible* with respect to distribution F, or F-symmetric, if for all $x, x' \in \mathcal{X}$,

$$F(dx)P(x, dx') = F(dx')P(x', dx). \tag{1.4}$$

In addition, if

$$\int_{\mathcal{X}} F(dx)P(x, dx')dx = F(dx'),$$

then F is a *stationary* or *invariant distribution*. By integrating both sides of (1.4), we see that if the chain is F-symmetric, then F is the stationary distribution. Notice, the converse does not hold.

We say P is *ϕ-irreducible* if for every $x \in \mathcal{X}$ and $A \subseteq \mathcal{B}(\mathcal{X})$ such that $\phi(A) > 0$, there exists an $n \in \mathbb{N}$ such that $P^n(x, A) > 0$. If P is F-symmetric and ϕ-irreducible, then P is F-irreducible (Meyn and Tweedie, 1993).

A Markov chain, $\{X_m^*\}$, is *aperiodic* if there does not exist at least two disjoint subsets $A_1, \ldots, A_d \subseteq \mathcal{X}$ such that $F(A_i) > 0$ for $i = 1, \ldots, d$ and $P(x, A_{i+1}) = 1$ for all $x \in A_i, 1 \leq i \leq d-1$ and $P(x, A_1) = 1$ for all $x \in A_d$. If $N = (A_1 \cup A_2 \cup \cdots \cup A_d)^c$, then $F(N) = 0$. Otherwise the chain is *periodic*.

Recall the goal in MCMC is to obtain a representative sample from F. However, even if a Markov chain has stationary distribution, F, it may fail to converge to it (Roberts and Rosenthal, 2004). Therefore, we want to ensure we only sample from chains where convergence is guaranteed. One way we measure how close the current state of the chain is from its stationary distribution is with the *total variation distance*. Let ν_1 and ν_2 be probability measures, then the total variation distance is defined as,

$$||\nu_1(\cdot) - \nu_2(\cdot)|| = \sup_{A \in \mathcal{B}(\mathcal{X})} |\nu_1(A) - \nu_2(A)|.$$

If P is F-irreducible with invariant distribution F, then P is *Harris Recurrent* if for all $A \in \mathcal{B}(\mathcal{X})$ with $F(A) > 0$ and $\forall\ x \in \mathcal{X}$, $Pr(\inf\{m \geq 1 : X_m \in A\} < \infty | X_0 = x) = 1$.

If the Markov chain has the following four conditions (referred to as $\mathfrak{C}$):

(i) has stationary distribution F,

(ii) is F-irreducible,

(iii) is aperiodic, and

(iv) is Harris recurrent,

then the initial position of the chain is irrelevant, the chain will explore all areas of the state space, and will converge to its stationary distribution as the number of steps goes to infinity. More specifically,

$$\lim_{m\to\infty} ||P^m(x,\cdot) - F(\cdot)|| = 0.$$

The next obvious question then is, how does one construct such a chain to simulate a representative sample from F? The most well-known approach is the Metropolis-Hastings algorithm which we present below.

1.3.2 Metropolis-Hastings

The Metropolis-Hastings algorithm was first proposed by Metropolis et al. (1953) and later generalized by Hastings (1970). The goal is to construct a Markov chain $\{X_m^*\}$ with stationary distribution F and corresponding density $f(x)$. Define $h(x)$ as the *unnormalized density* and c as the *normalizing constant* where $f(x) = h(x)/c$ such that $\int_{\mathcal{X}} h(x)dx = c < \infty$.

Let ν be a measure on $\mathcal{B}(\mathcal{X})$ and Q a Markov transition kernel such that

$$Q(x,\cdot) = q(x,y)\nu(dy).$$

In addition, define the *Hastings ratio* as

$$r(x,y) = \frac{h(y)q(y,x)}{h(x)q(x,y)}. \tag{1.5}$$

Notice, we could have used the normalized density in line (1.5), but the normalizing constants would have canceled out.

Then the Metropolis-Hastings algorithm is as follows:

1. Choose an initial position $X_0^* = x_0$.

2. At the mth step given $X^*_{m-1} = x$ generate a proposal Y from $Q(x, \cdot)$ and independently $U \sim U(0, 1)$, denote observed values as y and u respectively.

3. If $u < r(x, y)$ then accept $X^*_m = y$ otherwise $X^*_m = x$.

4. Repeat steps 2-3 M times to obtain M samples.

The third step is known as the *Metropolis rejection* step, not to be confused with the Accept-Reject algorithm. Notice the difference, in the Accept-Reject algorithm, if we reject a draw, we keep drawing samples until we get one we accept, discarding any rejected draws along the way. In Metropolis-Hastings, if we reject a draw, then we keep the old state as the new.

The Metropolis-Hastings algorithm produces a Markov chain that is reversible with respect to F and therefore ensures F is the stationary distribution (see Liu, 2008; Roberts and Rosenthal, 2004). Moreover, Mengersen and Tweedie (1996) show that the Markov chain is F-irreducible if

$$F(y) > 0 \Rightarrow q(x, y) > 0 \quad \text{for all } x \in \mathcal{X}$$

and aperiodic if $F(x)$ and $Q(x, y)$ are positive and continuous for all $x, y \in \mathcal{X}$.

Notice in the Metropolis-Hastings algorithm, there are two user-specified items: 1) the starting position and 2) the proposal density. If F is positive everywhere and the user chooses a proposal Q that is also positive and continuous for all $x, y \in \mathcal{X}$, then the chain will converge to F.

Some common choices of proposal density include: a symmetric Metropolis where $q(x, y) = q(y, x)$, a random walk where $q(x, y) = q|y - x|$, an independence sampler where $q(x, y) = q(y)$ does not depend on the current state, or the Langevin algorithm (Roberts and Rosenthal, 2004). We implement a version of the Metropolis-Hastings algorithm in Chapter 2.

1.3.3 Gibbs Sampler

Another popular MCMC method is the Gibbs Sampler; a composition of Metropolis-Hastings updates. The difference in this algorithm is it relies on the full conditionals and always accepts a proposed move. Just as we saw earlier with the Linchpin-Variable algorithm (Section 1.2.3), if we can split up the target density of interest and can sample from the conditionals: $F_{Y|Z}$ and $F_{Z|Y}$, then we can use the Gibbs Sampler. The Gibbs Sampler works as follows, choose an initial position $(Y_0 = y_0, Z_0 = z_0)$ then one iteration given $(Y_m = y_m, Z_m = z_m)$ is:

1. Draw $Y_{m+1} \sim Y|Z = z_m$.
2. Then draw $Z_{m+1} \sim Z|Y = y_{m+1}$.

Advantages and Disadvantages of MCMC

MCMC is useful when we cannot draw iid samples from the density of interest, works well for complicated distributions in high dimensions, and it is straightforward to implement algorithms like Metropolis-Hastings.

The disadvantage is we collect correlated non-identically distribted samples, so calculating variances become more complicated. Also, convergence only happens asymptotically. Lastly, since the samples are correlated, MCMC requires more samples to achieve the same level of accuracy versus Classical Monte Carlo.

1.4 Monte Carlo Estimation

After obtaining a representative sample from the target distribution, the next step is estimation. Fundamentally, Monte Carlo estimation works because of the strong law of large numbers (SLLN). Consider Classical Monte Carlo where μ as defined in line (1.1) exists, i.e., $\int_{\mathcal{X}} |g(x)|F(dx) < \infty$. Suppose $X_1, X_2, \ldots, X_m \overset{iid}{\sim} F$. We will refer to the observed values $x_1, x_2, \ldots, x_m$ as the Monte Carlo sample. The Monte Carlo estimator

of μ is simply the sample mean $\mu_m = \frac{1}{m}\sum_{t=1}^{m} g(X_t)$. And by the SLLN, as $m \to \infty$,

$$\mu_m \to \mu \quad \text{almost surely.} \tag{1.6}$$

Notice that μ_m is an unbiased estimator of μ with variance $m^{-1}\mathbb{E}_F[(g(X)-\mu)^2]$, where we can simply take the sample mean to approximate the expectation. Moreover, if $\mathbb{E}_F g^2 < \infty$, there is a Central Limit Theorem (CLT) for μ_m; that is for $\sigma^2 = Var_F g$, as $m \to \infty$,

$$\sqrt{m}(\mu_m - \mu) \xrightarrow{d} N(0, \sigma^2). \tag{1.7}$$

Similarly, if we want to estimate ξ_q, define $Y_t = g(X_t)$ for $t = 1, \ldots, m$ and $Y_{m(1)}, \ldots, Y_{m(m)}$ as the order statistics from the observed m length Monte Carlo sample. We can use the following estimator: $\xi_{m,q} = Y_{m(j)}$, where $j - 1 < mq \leq j$ which is strongly consistent for ξ_q (Serfling, 2009). Hence, using the Monte Carlo sample for estimation is straightforward.

Even with an MCMC sample, when the Markov chain meets assumptions $\mathfrak{C}$, we can use sample means because of the Ergodic theorem. For any $g : \mathcal{X} \to \mathbb{R}$ where $\mathbb{E}_F|g(X^*)| < \infty$ a Markov chain LLN, known as the Ergodic theorem (Birkhoff, 1931), holds that as, $m \to \infty$,

$$\mu_m^* = \frac{1}{m}\sum_{t=0}^{m-1} g(X_t^*) \to \mathbb{E}_F g \quad \text{with probability 1.}$$

For a CLT, we also need to know the rate of convergence. Let $M(x)$ be a nonnegative function and $\gamma(m)$ a nonnegative decreasing function on $\mathbb{Z}_+$ such that

$$||P^m(x, \cdot) - F(\cdot)|| \leq M(x)\gamma(m).$$

We say the Markov chain is *geometrically ergodic* if $\gamma(m) = t^m$ for some $t < 1$ and *uniformly ergodic* if $M(x)$ is also bounded for all x. Notice, uniform ergodicity is a stronger condition and implies geometric ergodicity.

If the Markov chain meets assumptions $\mathfrak{C}$ and either: the chain is geometrically ergodic and $\mathbb{E}_F|g(X^*)|^{2+\delta} < \infty$ for some $\delta > 0$ or the chain is uniformly ergodic and $\mathbb{E}_F g(X^*)^2 <$

∞, then for any starting point, a Markov chain CLT exists; except now the form of the asymptotic variance in the CLT is more complicated because of the correlated samples (Jones, 2004). We return to the explicit form of the variance in Section 1.7.

Thus far in order to estimate a vector of features of F, we have first spent time trying to simulate a representative sample from this target distribution. Sometimes, though, it is not beneficial to try and simulate from F. Situations like these require Importance Sampling.

1.5 Importance Sampling

In many situations either we cannot draw samples directly from F, as seen before, or there are large unimportant areas of F that we do not need to sample from (i.e., will provide little information about μ so do not need to waste simulation effort sampling from these areas). This leads us to use *Importance Sampling* (IS). Kahn and Harris (1951) first proposed IS when working on a physics problem estimating when nuclear particles would penetrate shields. They stated there are areas of the shield with low probability of the particle transferring through, so to focus instead on areas with higher probability instead of waiting on those by chance. The idea behind importance sampling then is to focus on regions of importance and correct the estimate to account for oversampling from these areas (Liu, 2008).

We consider the two most common forms of IS: Simple Importance Sampling and Weighted Importance Sampling.

1.5.1 Simple Importance Sampling

Define $\tilde{F}$ to be some other distribution with pdf $\tilde{f}$ and support $\tilde{\mathcal{X}}$ where $\mathcal{X} \subseteq \tilde{\mathcal{X}}$. Then,

$$\begin{aligned}\mathbb{E}_F[g(X)] &= \int_{\mathcal{X}} g(x)f(x)dx \\ &= \int_{\tilde{\mathcal{X}}\cap\mathcal{X}} g(x)f(x)dx + \int_{\tilde{\mathcal{X}}\cap\mathcal{X}^C} g(x)f(x)dx\end{aligned}$$

$$
\begin{aligned}
&= \int_{\tilde{\mathcal{X}}} g(x)f(x)dx \\
&= \int_{\tilde{\mathcal{X}}} g(x)\frac{f(x)}{\tilde{f}(x)}\tilde{f}(x)dx \\
&= \mathbb{E}_{\tilde{F}}\left[g(X)\frac{f(X)}{\tilde{f}(X)}\right].
\end{aligned}
$$

We refer to $\tilde{f}$ as the *importance density* and $\tilde{F}$ the *importance distribution*. We do not need to worry about dividing by zero, since $\mathcal{X} \subseteq \tilde{\mathcal{X}}$. Then the Simple Importance (SIS) estimator of μ is for $X_1, X_2, \ldots, X_m \overset{iid}{\sim} \tilde{F}$,

$$
\mu_m = \frac{1}{m}\sum_{t=1}^{m}\frac{g(X_t)f(X_t)}{\tilde{f}(X_t)}. \tag{1.8}
$$

Theorem 1.5.1. *Define μ_m as in (1.8). Then μ_m is an unbiased estimator of μ. Let $\sigma^2 = Var_{\tilde{F}}\left[\frac{g(X)f(X)}{\tilde{f}(X)}\right] < \infty$ then the variance of μ_m is $\sigma^2/m = m^{-1}\mathbb{E}_{\tilde{F}}\left[(g(X)f(X)/\tilde{f}(X) - \mu)^2\right]$.*

Proof. The SIS estimator, μ_m, is unbiased for μ since,

$$
\begin{aligned}
\mathbb{E}_{\tilde{F}}[\mu_m] &= \mathbb{E}_{\tilde{F}}\left[\frac{1}{m}\sum_{t=1}^{m}\frac{g(X_t)f(X_t)}{\tilde{f}(X_t)}\right] \\
&= \mathbb{E}_{\tilde{F}}\left[\frac{g(X)f(X)}{\tilde{f}(X)}\right] \\
&= \int_{\tilde{\mathcal{X}}}\frac{g(x)f(x)}{\tilde{f}(x)}\tilde{f}(x)dx \\
&= \int_{\mathcal{X}} g(x)f(x)dx = \mu.
\end{aligned}
$$

The variance of the estimator follows directly. Let $\sigma^2 = Var_{\tilde{F}}\left[\frac{g(X)f(X)}{\tilde{f}(X)}\right]$ then,

$$
Var_{\tilde{F}}[\mu_m] = \frac{1}{m}Var_{\tilde{F}}\left[\frac{g(X)f(X)}{\tilde{f}(X)}\right] = \frac{1}{m}\mathbb{E}_{\tilde{F}}\left[(g(X)f(X)/\tilde{f}(X) - \mu)^2\right] = \frac{\sigma^2}{m}.
$$

□

If $\sigma^2/m < \infty$, then by the CLT we have $\sqrt{m}(\mu_m - \mu) \xrightarrow{d} N(0, \sigma^2)$ as $m \to \infty$.

Ensuring a finite second moment of μ_m is nontrivial and is largely dependent on the variance of the *importance weights*, $w(x) = \frac{f(x)}{\tilde{f}(x)}$. Moreover, the variance of the estimator is sensitive to the choice of $\tilde{f}$. Notice for $x \in \tilde{\mathcal{X}}$, if $\tilde{f}(x)$ is much smaller than $f(x)$, $w(x)$ may be large. If we want an effective importance density for various $g(x)$'s we want an $\tilde{f}$ to keep the weights from exploding, i.e., avoid light tailed importance densities. Ideally, we choose $\tilde{f}$ with tails that envelope f. In fact, if we can bound the weights such that $w(x) \leq c$ for all x, then we can bound the variance.

Additionally, there is a dimension effect in SIS. For example, if each component of x is independent under f and $\tilde{f}$, then we can write $w(x) = f(x)/\tilde{f}(x) = \prod_{j=1}^{d} f(x_j)/\tilde{f}(x_j)$. The variance of μ_m will be driven largely by the variance in the weights. To see this, first notice,

$$\mathbb{E}_{\tilde{F}}[w(X)] = \int_{\tilde{\mathcal{X}}} \frac{f(x)}{\tilde{f}(x)} \tilde{f}(x)dx = \int_{\mathcal{X}} f(x)dx = 1.$$

Then,

$$\mathbb{E}_{\tilde{F}}[w(X)^2] = \prod_{j=1}^{d} \mathbb{E}_{\tilde{F}}[w(X_j)^2] = \prod_{j=1}^{d} (Var_{\tilde{F}}(w(X_j)) + 1).$$

Since $Var_{\tilde{F}}(\mu_m) = m^{-1}(E_{\tilde{F}}[g(X)^2 w(X)^2] - \mu^2)$, we see $g(x)$ plays a role, however as the dimension of $\mathcal{X}$ grows, the variance will grow exponentially with d. So in order to control the variance of μ_m, $\tilde{f}$ will need to be chosen such that the variances of the weights decreases as dimension increases.

Keeping these properties in mind, there exists an ideal SIS importance density such that μ_m has zero variance. This density is,

$$\tilde{f}(x)_{opt} = \frac{g(x)f(x)}{\int g(x)f(x)dx}. \tag{1.9}$$

But sampling from $\tilde{f}(x)_{opt}$ implies we know μ! Hence in practically relevant situations the ideal is not possible. Finding an effective importance density is non-trivial (Liu, 2008).

Moreover, if the goal is to estimate $Pr(X \in A) = \int_A f(x)dx$, then our estimate should

satisfy basic probability rules such as $\hat{Pr}(X \in A) = 1 - \hat{Pr}(X \notin A)$ but these are not guaranteed by SIS. Another limitation of SIS is in order to compute μ_m we must know the normalizing constants of f and $\tilde{f}$.

1.5.2 Weighted Importance Sampling

In Weighted Importance Sampling (WIS), also known as normalized importance sampling, the importance weights, $w(x)$, only need to be known up to a constant of proportionality. Let $\tilde{f}(x) = \tilde{h}(x)/\tilde{c}$ and $w^*(x) = h(x)/\tilde{h}(x)$. Then if $X_t \overset{iid}{\sim} \tilde{F}, t = 1, \ldots, m$, (or from the unnormalized distribution) the WIS estimator of μ is a ratio of sample means,

$$\hat{\mu}_m = \frac{\frac{1}{m}\sum_{t=1}^{m} g(X_t)w^*(X_t)}{\frac{1}{m}\sum_{t=1}^{m} w^*(X_t)} = \frac{\sum_{t=1}^{m} g(X_t)w^*(X_t)}{\sum_{t=1}^{m} w^*(X_t)}. \tag{1.10}$$

Typically we know the normalizing constant of $\tilde{f}$, although it is not required in WIS, but the normalizing constant of f may not be easily available. Additionally notice, $w^*(x) > 0$ since $\mathcal{X} \subseteq \tilde{\mathcal{X}}$, $h(x) \geq 0$, and $\tilde{h}(x) > 0$ for all $x \in \tilde{\mathcal{X}}$. Therefore in (1.10) we will not divide by zero.

Theorem 1.5.2. *Let $X \sim \tilde{F}$, define $G = g(X)w(X)$ and $W = w(X)$, then the WIS estimator, $\hat{\mu}_m$, as defined in* (1.10) *converges with probability 1 to μ as $m \to \infty$. If $Var_{\tilde{F}} G < \infty$ and $Var_{\tilde{F}} W < \infty$, then $\hat{\mu}_m$ has variance $m^{-1}\mathbb{E}_{\tilde{F}}[(G - \mu W)^2]$.*

Proof. We first rewrite $\hat{\mu}_m$ as,

$$\frac{\frac{1}{m}\sum_{t=1}^{m} g(X_t)w^*(X_t)}{\frac{1}{m}\sum_{t=1}^{m} w^*(X_t)} = \frac{\frac{1}{m}\sum_{t=1}^{m} g(X_t)\frac{h(X_t)/c}{\tilde{h}(X_t)/\tilde{c}}}{\frac{1}{m}\sum_{t=1}^{m} \frac{h(X_t)/c}{\tilde{h}(X_t)/\tilde{c}}} = \frac{\frac{1}{m}\sum_{t=1}^{m} g(X_t)\frac{f(X_t)}{\tilde{f}(X_t)}}{\frac{1}{m}\sum_{t=1}^{m} \frac{f(X_t)}{\tilde{f}(X_t)}} = \frac{\frac{1}{m}\sum_{t=1}^{m} g(X_t)w(X_t)}{\frac{1}{m}\sum_{t=1}^{m} w(X_t)}.$$

Let $\bar{G} = m^{-1}\sum_{t=1}^{m} g(X_t)w(X_t)$ and $\bar{W} = m^{-1}\sum_{t=1}^{m} w(X_t)$. Then by the SLLN, $\bar{G} \to \mu$ with probability (w.p.) 1 as $m \to \infty$. Similarly, by the SLLN, $\bar{W} \to \mathbb{E}_{\tilde{F}}[W] = 1$ w.p. 1

as $m \to \infty$. Using Slutsky's theorem we have w.p. 1 as $m \to \infty$,

$$\frac{\bar{G}}{\bar{W}} \to \mu.$$

To solve for the variance we can use the delta method. Recall the $Var_{\tilde{F}}(G) = \sigma^2$ and define $\tau^2 = Var_{\tilde{F}}(W)$.

Since all draws are iid, $\mathbb{E}_{\tilde{F}}(\bar{G}) = \mu, Var_{\tilde{F}}(\bar{G}) = \sigma^2/m, \mathbb{E}_{\tilde{F}}(\bar{W}) = 1$, and $Var_{\tilde{F}}(\bar{W}) = \tau^2/m$.

Define $\rho = Corr_{\tilde{F}}(G, W)$ then, $Cov_{\tilde{F}}(G, W) = \rho\sigma\tau$. Since $\sigma^2, \tau^2 < \infty$, as $m \to \infty$,

$$\sqrt{m}\left(\begin{pmatrix}\bar{G}\\ \bar{W}\end{pmatrix} - \begin{pmatrix}\mu\\ 1\end{pmatrix}\right) \xrightarrow{d} N_2\left(\begin{pmatrix}0\\ 0\end{pmatrix}, \begin{pmatrix}\sigma^2 & \rho\sigma\tau\\ \rho\sigma\tau & \tau^2\end{pmatrix}\right).$$

Using the delta method where the function is $g_{DM}((a, b)) = a/b$ where $b \neq 0$,

$$\sqrt{m}\left(\frac{\bar{G}}{\bar{W}} - \mu\right) \xrightarrow{d} N(0, \sigma^2 - 2\mu\rho\sigma\tau + \mu^2\tau^2).$$

We can rewrite the variance term as,

$$\begin{aligned}
\sigma^2 - 2\mu\rho\sigma\tau + \mu^2\tau^2 &= \sigma^2 + \mu^2 - 2\mu\rho\sigma\tau - 2\mu^2 + \mu^2\tau^2 + \mu^2 \\
&= \sigma^2 + \mu^2 - 2\mu(\rho\sigma\tau + \mu) + \mu^2(\tau^2 + 1) \\
&= \mathbb{E}_{\tilde{F}}(G^2) - 2\mu(Cov_{\tilde{F}}(G, W) + \mathbb{E}_{\tilde{F}}(G)\mathbb{E}_{\tilde{F}}(W)) + \mu^2\mathbb{E}_{\tilde{F}}(W^2) \\
&= \mathbb{E}_{\tilde{F}}(G^2) - 2\mu\mathbb{E}_{\tilde{F}}(GW) + \mu^2\mathbb{E}_{\tilde{F}}(W^2) \\
&= \mathbb{E}_{\tilde{F}}((G - \mu W)^2)
\end{aligned}$$

□

Owen (2013) further shows that to easily compute the estimated variance of the WIS estimator we can use the following where $\tilde{w}_t = w^*(X_t)/\sum_{t=1}^m w^*(X_i)$ is the tth normalized weight such that,

$$\widehat{Var}(\hat{\mu}_m) = \sum_{t=1}^{m} \tilde{w}_t^2 (g(X_t) - \hat{\mu}_m)^2. \tag{1.11}$$

It is worth noting that $\hat{\mu}_m$ is a biased estimator of μ.

Proposition 1.5.3. *Bias*$(\hat{\mu}_m) = m^{-1}(\mu\tau^2 - \rho\sigma\tau) + O(m^{-2})$.

Proof. We use the Taylor expansion around the function $g_{DM}(a, b) = a/b$. This implies,

$$\nabla g_{DM} = \begin{pmatrix} 1/b & -a/b^2 \end{pmatrix} \quad \text{and} \quad \nabla^2 g_{DM} = \begin{pmatrix} 0 & -1/b^2 \\ -1/b^2 & 2a/b^3 \end{pmatrix}.$$

Then evaluating around point $(\mu, 1)$,

$$\begin{aligned} \frac{\bar{G}}{\bar{W}} = g_{DM}(\bar{G}, \bar{W}) &= g_{DM}(\mu, 1) + \nabla g_{DM}(\mu, 1) \begin{pmatrix} \bar{G} - \mu \\ \bar{W} - 1 \end{pmatrix} \\ &\quad + \frac{1}{2} \begin{pmatrix} \bar{G} - \mu \\ \bar{W} - 1 \end{pmatrix}^T \nabla^2 g_{DM}(\mu, 1) \begin{pmatrix} \bar{G} - \mu \\ \bar{W} - 1 \end{pmatrix} + O(m^{-2}) \\ &= \mu + (\bar{G} - \mu\bar{W}) + (-\bar{G}\bar{W} + \bar{G} + \mu\bar{W}^2 - \mu\bar{W}) + O(m^{-2}). \end{aligned}$$

Taking the expectation under $\tilde{F}$,

$$\begin{aligned} \mathbb{E}_{\tilde{F}} \left(\frac{\bar{G}}{\bar{W}} \right) &= \mathbb{E}_{\tilde{F}}(\mu) + \underbrace{\mathbb{E}_{\tilde{F}}(\bar{G} - \mu\bar{W})}_{0} + \mathbb{E}_{\tilde{F}}(-\bar{G}\bar{W} + \bar{G} + \mu\bar{W}^2 - \mu\bar{W}) + O(m^{-2}) \\ &= \mu - \mathbb{E}_{\tilde{F}}(\bar{G}\bar{W}) + \mathbb{E}_{\tilde{F}}(\bar{G}) \cdot 1 + \mu[\mathbb{E}_{\tilde{F}}(\bar{W}^2) - 1^2] + O(m^{-2}) \\ &= \mu - \mathbb{E}_{\tilde{F}}(\bar{G}\bar{W}) + \mathbb{E}_{\tilde{F}}(\bar{G})\mathbb{E}_{\tilde{F}}(\bar{W}) + \mu[\mathbb{E}_{\tilde{F}}(\bar{W}^2) - \mathbb{E}_{\tilde{F}}(\bar{W})^2] + O(m^{-2}) \\ &= \mu - Cov_{\tilde{F}}(\bar{G}, \bar{W}) + \mu Var_{\tilde{F}}(\bar{W}) + O(m^{-2}) \\ &= \mu - \frac{\rho\sigma\tau}{m} + \frac{\mu\tau^2}{m} + O(m^{-2}) \end{aligned}$$

□

So how close the WIS estimator is to the truth depends on the variance of the importance weights, although this bias diminishes quickly as the Monte Carlo sample size increases. In WIS, there is no $\tilde{f}_{opt}$ that yields a zero-variance, however the optimal importance density to minimize bias and variance would be proportional to $|g(x) - \mu|f(x)$ (Owen, 2013), which again, is impractical.

Although $\hat{\mu}_m$ is a biased estimator, it addresses some limitations of μ_m. For example, we do not need normalizing constants of f or $\tilde{f}$. Additionally, the WIS estimator does not violate the complement rule.

1.5.3 Umbrella Sampling

Finding an effective importance distribution is a challenging task. Torrie and Valleau (1977) first suggest the idea of *umbrella sampling*, also known as mixture importance sampling (Owen, 2013). The idea is take a finite mixture, say $\tilde{f}_1, \tilde{f}_2, \ldots, \tilde{f}_s$, of densities that span the support of f such that,

$$\tilde{f}(x) = \sum_{i=1}^{s} \delta_i \tilde{f}_i \quad \text{where } \sum_{i=1}^{s} \delta_i = 1. \tag{1.12}$$

Mixtures of unimodal densities are good starting points to estimate multi-modal target densities. They also allow for creating importance densities with not too light of tails. (Owen, 2013).

Advantageous and Disadvantages of IS

The advantages of IS is that is highly flexible since we can frame most features of interest as expectations, usable in multivariate simulation, and simple to implement. Also, compared to the Accept-Reject algorithm, IS is possible even if the $\sup_x f(x)/\tilde{f}(x)$ is unknown. Moreover, in some cases, using IS can yield an estimator with smaller variance than if we were to sample from the target directly (Liu, 2008). The disadvantage, whether using SIS or WIS, is the difficulty in selecting the importance distribution and consequently efficiency of the procedure.

1.6 IS in functional approximation

IS is also useful to approximate functions. Functional approximation occurs in a variety of settings. In Bayesian analysis, researchers study marginal posterior densities that

may be multi-modal or in functional regression sometimes the outcome of interest is a function or the covariates are functions themselves. Similarly, functional estimation is found in frequentist settings often in the form of estimating likelihood functions in order to do likelihood-based inference (Geyer, 1994; Geyer and Thompson, 1992).

We focus on estimating density and log-likelihood functions using importance sampling. More specifically, consider the density $f_\theta : \mathbb{R}^d \to \mathbb{R}$ with $d \geq 1$. Let $\mathsf{X} \subseteq \mathcal{X} \subseteq \mathbb{R}^d$. We want to estimate $f_\theta(x)$ for all $x \in \mathsf{X}$. Or consider $\Theta \subseteq \mathbb{R}^d$, we want to estimate the likelihood function, $L(\theta|x)$, or the log-likehood function, $\ell(\theta|x)$, for all $\theta \in \Theta$.

Focusing on a compact set rather than the whole support is practically relevant since often the goal is to understand some function over a smaller set as we are usually interested in modes or tails of densities or likelihoods. For example, instead of being just interested in maximums or minimums of functions, if we can get a good estimate of a density or a likelihood function over the neighborhood of the optimum, then we can make better inference about the precision in the estimation.

1.6.1 Density Estimation

Estimating pdfs is a well-researched field. Below is a brief discussion of popular functional estimation techniques. In all cases below, the goal is to estimate a smooth continuous density f.

Common Density Estimation Techniques

Histograms

Histograms are the most elementary form of non-parametric density estimation. Given some observed data $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$ where $d \geq 1$, we divide the data into B equal sized bins. The simplest way to break into bins is to create B $d-$dimensional hypercubes, h_n^d, each side with length h_n. We write h_n since the side length is dependent on the sample size where as $n \to \infty, h_n \to 0$. Define $Vol(B)$ as the volume of each hypercube.

Then for bins $B_1, B_2, \ldots, B_B$,

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{I(X_i \in B_j)}{Vol(B)}.$$

So the histogram method calculates the proportion of observations in each hypercube. As the sample size grows and the bins become smaller, a finer estimate of the density forms. Although conveniently simple, the histogram method does have several limitations. The accuracy of the estimate depends on the size of the bins, it is not a smooth function, and as d increases the number of bins will grow exponentially, hence requiring many samples to estimate f.

Kernel Density Estimation

Kernel density estimation (KDE) is another non-parametric method to estimate a pdf. Again, given $X_1, X_2, \ldots, X_n \in \mathbb{R}^d$ we can estimate the density with

$$\hat{f}(x) = \frac{1}{nh_n^d} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h_n}\right). \tag{1.13}$$

In the above equation, K is a smooth function known as the *kernel function* where $K : \mathbb{R}^d \to \mathbb{R}$ and $\int K(t)dt = 1$. Typically K is chosen to be symmetric with mean 0. And $h_n > 0$ is the *bandwidth* that controls how smooth the kernel density estimator is. The larger h_n is, the smoother the estimator $\hat{f}$ and vice versa. Using KDE, the estimated density is a smooth function that better scales than the histogram method as d grows. There has been a wide set of research to determine the optimal h_n for certain kernel functions, a common choice of kernel is the Gaussian kernel. KDE is well studied for different kernel functions in terms of consistency, bias, and variance (Chen, 2017). Kernel methods are known to not work well at boundaries of regions or at the tails of bounded distributions (Jones, 1993).

Basis Functions or Splines

We can also estimate a density function by representing it as a linear combination of basis functions (Wasserman, 2006). Let $\phi_1, \phi_2, \ldots$ be an orthonormal basis for a density f such that

$$f(x) = \sum_{j=1}^{\infty} \beta_j \phi_j(x)$$

where $\beta_j = \int_{\mathcal{X}} f(x)\phi_j(x)dx$. Then a basis estimate of f is,

$$\hat{f}(x) = \sum_{j=1}^{b} \hat{\beta}_j \phi_j(x) \quad \text{where } \hat{\beta}_j = \frac{1}{n}\sum_{i=1}^{n} \phi_j(X_i).$$

The number of basis terms, b, can be determined through cross validation.

Rao-Blackwellisation

Rao-Blackwellisation is a method to estimate marginal densities from joint posterior densities. The approach was first introduced in Wei and Tanner (1990). For simplicity consider a bivariate joint density, $f(x, y)$, with support $\mathsf{X} \times \mathsf{Y}$ where $m_X(x)$ and $m_Y(y)$ represent the marginals with supports X and Y respectively. The goal is to estimate m_X. Notice then,

$$m_X(x) = \int_{\mathsf{Y}} f(x, y)dy = \int_{\mathsf{Y}} f_{X|Y}(x|y)m_Y(y)dy = \mathbb{E}_{m_Y}[f_{X|Y}(x|y)].$$

If $Y_1, \ldots, Y_m \sim m_Y$ then we can estimate the marginal of X since for each x,

$$\hat{m}_X(x) = \frac{1}{m}\sum_{t=1}^{m} f(x|y_t) \to m_X(x) \text{ almost surely as } m \to \infty.$$

The limitation of Rao-Blackwellisation is it requires known forms of the conditional and ability to simulate from the conditional distribution.

IS in Density Estimation

We now introduce two ways we can use IS to estimate density functions.

IS density estimation with histograms

Say the goal is to estimate the pdf $f(x)$ over a compact set $\mathsf{X} \subset \mathbb{R}$. Notice, we can approximate the density at any point $x \in \mathsf{X}$ using a small $\omega > 0$ where F denotes the distribution of X as,

$$f(x) \approx \frac{F(x+\omega) - F(x-\omega)}{2\omega} = \frac{Pr(x-\omega < X < x+\omega)}{2\omega}. \tag{1.14}$$

Then define a g function as,

$$g_\omega(y, x) = \frac{I(y-\omega < x < y+\omega)}{2\omega}. \tag{1.15}$$

If the full form of $f(x)$ was known, there would be no need to do density estimation. However, if the normalizing constant is unknown, one option is to use WIS,

$$\hat{f}_m(x) = \frac{\frac{1}{m}\sum_{t=1}^{m} g_\omega(x, X_t) h(X_t)/\tilde{f}(X_t)}{\frac{1}{m}\sum_{t=1}^{m} h(X_t)/\tilde{f}(X_t)}. \tag{1.16}$$

The form of (1.16) closely follows the histogram method. Therefore the ω in equation (1.15) should be chosen such that as m gets larger, ω should get smaller. This approach has the same limitations as the histogram method in that it does not produce a smooth estimate nor does it scale well with dimension.

Rubin Importance-Sampling Algorithm

Another approach that uses IS for marginal density estimation was proposed in Rubin's comment to Tanner and Wong (1987)'s data augmentation paper. Gelfand and

Smith (1990) refer to this approach as the Rubin Importance-Sampling algorithm. This method essentially marries IS with Rao-Blackwellisation.

Again for simplicity, assume a bivariate case as described in Section 1.6.1 where the goal is to estimate m_X. Notice,

$$m_X(x) = \int_{\mathsf{Y}} f_{X|Y}(x|y) m_Y(y) dy = \int_{\mathsf{Y}} f_{X|Y}(x|y) \left(\int_{\mathsf{X}} f(x,y) dx \right) dy.$$

Suppose we do not know m_Y (as is typically the case in applications), only have the unnormalized form, $h(x,y)$, of the joint density, and know the full form of the conditional $f(x|y)$. Define an importance distribution $\tilde{m}_Y$ with support containing Y. Then define the importance weights,

$$w(X_t, Y_t) = \frac{h(X_t, Y_t)}{f(X_t|Y_t)\tilde{m}(Y_t)}. \tag{1.17}$$

First draw $Y_t \sim \tilde{m}_Y$, then using that Y_t draw an $X_t \sim f(X_t|Y_t)$. Then,

$$\hat{m}_X(x) = \frac{\frac{1}{m}\sum_{t=1}^{m} f(x|Y_t) w(X_t, Y_t)}{\frac{1}{m}\sum_{t=1}^{m} w(X_t, Y_t)}. \tag{1.18}$$

If we also have the full form for $f(y|x)$, then we can estimate $m_Y(y)$ similarly. We can directly expand this set up to include more than two variables as Gelfand and Smith (1990) show.

1.6.2 Log-likelihood Estimation

Another application of IS is estimating a likelihood function. In many situations the likelihood function is not fully known either because the integral is intractable, as in Generalized Linear Mixed Effects Models (GLMMs), or normalizing constants are unknown, as is the case for Exponential Random Graph Models (ERGMs). Typically, interest lies in finding the argument that maximizes the likelihood function. In terms of estimating the entire likelihood function directly though, we use Monte Carlo Likelihood Approximation (MCLA) (Geyer and Thompson, 1992).

MCLA is the extension of IS to likelihood estimation. It was first introduced by Geyer and Thompson (1992) to estimate models when the normalizing constant is unknown. It was further developed to handle missing data by Guo and Thompson (1992) and the theory justifying its use was established by Geyer (1994).

MCLA for unknown normalizing constant

Suppose $f_\theta(x) = h_\theta(x)/c(\theta)$ is a pdf. As usual, F is the distribution, $h_\theta(x)$ is the unnormalized density, and $c(\theta)$ is the normalizing constant such that $c(\theta)$ is nonzero and finite for each $\theta \in \Theta$. If $c(\theta) = \int_{\mathcal{X}} h_\theta(x)dx < \infty$, then the log likelihood function is

$$\ell(\theta|x) = \log(h_\theta(x)) - \log(c(\theta)). \tag{1.19}$$

Often $c(\theta)$ involves an intractable integral. If it is low dimensional, we may try to solve it with numerical integration techniques, e.g., adaptive Gaussian quadrature. However, when the integral is more complex, we can use Monte Carlo methods. Consider another pdf $\tilde{f}(x)$ that is free of θ, where the support of $\tilde{f}(x)$ contains the support of f for all $\theta \in \Theta$ and $\tilde{F}$ is its distribution. Then,

$$c(\theta) = \int \frac{h_\theta(x)}{\tilde{f}(x)} \tilde{f}(x)dx = \mathbb{E}_{\tilde{F}}\left[\frac{h_\theta(x)}{\tilde{f}(x)}\right]. \tag{1.20}$$

Draw $X_1, X_2, \ldots, X_m \overset{iid}{\sim} \tilde{F}$, then a direct SIS estimator of $c(\theta)$ is

$$c_m(\theta) = \frac{1}{m}\sum_{t=1}^{m} \frac{h_\theta(X_t)}{\tilde{f}(X_t)}. \tag{1.21}$$

By the SLLN as $m \to \infty$, $c_m(\theta) \to c(\theta)$ with probability 1 for all $\theta \in \Theta$. Then we can estimate the log likelihood as,

$$\ell_m(\theta|x) = \log(h_\theta(x)) - \log\left(\frac{1}{m}\sum_{t=1}^{m} \frac{h_\theta(X_t)}{\tilde{f}(X_t)}\right). \tag{1.22}$$

Recall, we have the form of $h_\theta(x)$. Therefore, we can obtain an estimate of the MLE as the θ that maximizes equation (1.22), i.e.,

$$\hat{\theta} = \arg\max_{\theta \in \Theta} \ell_m(\theta|x). \tag{1.23}$$

Another approach is to estimate the *relative likelihood*. Consider another density, not required to be from the same parametric family, indexed by the parameter ψ, such that $f_\psi(x) = h_\psi(x)/c(\psi)$. Then the relative likelihood is

$$\begin{aligned} r(\theta, \psi) = \ell(\theta|x) - \ell(\psi|x) &= \log(h_\theta(x)) - \log(h_\psi(x)) - \log(c(\theta)) + \log(c(\psi)) \\ &= \log\left(\frac{h_\theta(x)}{h_\psi(x)}\right) - \log\left(\frac{c(\theta)}{c(\psi)}\right), \end{aligned}$$

where

$$\frac{c(\theta)}{c(\psi)} = \int \frac{c(\theta) f_\theta(x)}{c(\psi) f_\psi(x)} f_\psi(x) dx = \int \frac{h_\theta(x)}{h_\psi(x)} f_\psi(x) dx = \mathbb{E}_{F_\psi}\left[\frac{h_\theta(x)}{h_\psi(x)}\right].$$

Then for a fixed ψ we can estimate the relative likelihood drawing a sample $X_1, X_2, \ldots, X_m$ iid from $F_\psi(x)$ as

$$r_m(\theta, \psi) = \log\left(\frac{h_\theta(x)}{h_\psi(x)}\right) - \log\left(\frac{1}{m}\sum_{t=1}^{m} \frac{h_\theta(X_t)}{h_\psi(X_t)}\right). \tag{1.24}$$

We can then estimate the MLE as

$$\hat{\theta} = \arg\max_{\theta \in \Theta} r_m(\theta, \psi). \tag{1.25}$$

In both situations this estimator of the MLE is known as the MC-MLE. Geyer (1994) proved that as long as $c(\theta)$ is continuous then the estimated Monte Carlo log likelihood converges to the exact log likelihood for any single parameter value almost surely. Additionally, if the parameter space is compact, or can be made compact, the MC-MLE converges to the true MLE almost surely (Geyer, 1994).

Applications to Network Modeling

Exponential Random Graph Models (ERGMs) are one way to model network data. Suppose Y is a network with n many nodes and y is the observed network (a more in-depth introduction to networks is in Chapter 2). Define $g(y)$ to be a vector of b natural statistics of the network, β a $b-$dimensional vector of natural parameters, and $\mathcal{Y}$ the set of all possible networks with n nodes. Then the probability mass function for observing the network Y is,

$$P_\beta(Y = y) = \frac{1}{c(\beta)} exp\left\{\beta^T g(y)\right\} \quad \text{for } y \in \mathcal{Y}, \tag{1.26}$$

where the normalizing constant $c(\beta)$ is,

$$c(\beta) = \sum_{\mathcal{Y}} exp\left\{\beta^T g(y)\right\}. \tag{1.27}$$

If we consider all the different types of networks we can observe with n nodes, where the simplest case is when the link between nodes has no direction, the number of potential networks is $2^{\binom{n}{2}}$. This makes solving for the normalizing constant in equation (1.27) virtually impossible for networks with 7 or more nodes. If we can instead estimate $c(\beta)$, then we can obtain an estimate of the log-likelihood function. We can then estimate the MLE and use the MC-MLE for inference. This is the overarching approach used in the popular `statnet` package suite available in `R` to model social networks (Handcock et al., 2008). In order to sample from equation (1.26), Handcock et al. (2008) follow the steps proposed in Geyer and Thompson (1992). They first use MCMC to sample networks by starting from a known network and at each step modifying the network by either adding or deleting an edge at random and then calculating the statistics of that modified structure. Therefore each draw is a b-dimensional vector of statistics about the observed network at that step. Then using the relative likelihood approach, they use the MCMC sample to construct the MCMC-MLE estimates.

MCLA for Missing Data Models

The missing data we consider occur in models that either have *latent variables* or *random effects*. Consider the following setting, $Y \in \mathbb{R}^n$ is the response vector, $X \in \mathbb{R}^{n\times p}$ is the design matrix of observed predictor variables, $\beta \in \mathbb{R}^p$ the coefficient vector, $Z \in \mathbb{R}^{n\times q}$ is the model matrix for the random effects, $U \in \mathbb{R}^q$ is a vector of unobservable random effects, $\nu \in \mathbb{R}^\kappa$ is a vector of variance components such that each one is non-negative, D is a variance matrix dependent on ν, such that U has a density only dependent on the vector ν, i.e. $f_\nu(u)$. Unless otherwise stated, all the off-diagonal elements of D are 0, i.e., the random effects are independent of one another. The distribution of the response given the random effects is $f_\beta(y|u)$. Define $\theta = (\beta^T, \nu^T)^T \in \mathbb{R}^{p+\kappa}$. Then the joint density is:

$$f_\theta(y,u) = f_\beta(y|u)f_\nu(u). \tag{1.28}$$

The likelihood function is the likelihood of θ given the observed data. That is,

$$L(\theta|y) = f_\theta(y) = \int_{\mathbb{R}^q} f_\theta(y,u)du. \tag{1.29}$$

The log-likelihood function directly follows,

$$\ell(\theta|y) = \log(L(\theta|y)) = \log\left(\int_{\mathbb{R}^q} f_\theta(u,y)du\right). \tag{1.30}$$

Again, the difficulty in calculating this likelihood is in the evaluation of the $q-$dimensional integral. As before, define some other density $\tilde{f}(u)$ that is free of θ. Then,

$$\ell(\theta|y) = \log\left(\int \frac{f_\beta(y|u)f_\nu(u)}{\tilde{f}(u)}\tilde{f}(u)du\right) = \log\left(\mathbb{E}_{\tilde{F}}\left[\frac{f_\beta(Y|U)f_\nu(U)}{\tilde{f}(U)}\right]\right). \tag{1.31}$$

Which implies that a valid SIS estimator would be, for $U_1, \ldots, U_m$ iid from $\tilde{F}$,

$$\ell_m(\theta|y) = \log\left(\frac{1}{m}\sum_{t=1}^{m}\frac{f_\beta(y|U_t)f_\nu(U_t)}{\tilde{f}(U_t)}\right). \tag{1.32}$$

Similarly the WIS estimator is

$$\hat{\ell}_m(\theta|y) = \log\left(\frac{\frac{1}{m}\sum_{t=1}^{m}\frac{f_\beta(y|U_t)f_\nu(U_t)}{\tilde{f}(U_t)}}{\frac{1}{m}\sum_{t=1}^{m}\frac{f_\nu(U_t)}{\tilde{f}(U_t)}}\right). \tag{1.33}$$

In equation (1.33) we can replace densities $f_\nu(U)$ or $\tilde{f}(U)$ with their unnormalized forms.

Similar to density estimation as in Section 1.6.1, equations (1.32) and (1.33) give a functional approximation for all θ. Geyer (1994) proved that as long as a Wald-like integrability condition is met, the MCLA converges to the true log likelihood function for GLMMs with unnormalized densities. In addition, the estimated profile likelihoods converge to the exact profile likelihoods almost surely, with no additional regularity conditions needed. Knudson (2016) has implemented MCLA for GLMMs in the R package `glmm` where the response data is either Binomial or Poisson with Normally distributed random effects. The package estimates parameters by maximizing the estimated log likelihood function. It also calculates the value, gradient vector, and Hessian matrix of the MCLA at the MC-MLEs and uses the Hessian to calculate standard errors for the MC-MLEs. Park and Haran (2019) use importance sampling to find MLEs for latent Gaussian random field models and apply the results of Knudson (2016) to construct standard errors and confidence intervals.

1.6.3 Bias in Functional Approximation

In functional approximation we are interested in estimating some curve, be it a density or likelihood. This implies some type of smoothing. However, whenever we do smoothing we have to balance the bias-variance trade off. Wasserman (2006) discusses how in nonparametric inference, a function $r(x)$ is estimated with $\hat{r}(x)$ that largely depends on its mean $\bar{r}(x) = \mathbb{E}[\hat{r}(x)]$. Where the problem is by the SLLN we have that $\hat{r}(x) \to \mathbb{E}[\hat{r}(x)]$, but $\mathbb{E}[\hat{r}(x)] \neq r(x)$. In that, if you construct a confidence interval around this $\hat{r}(x)$ it will not be centered around the true $r(x)$ due to a smoothing bias of $\bar{r}(x) - r(x)$. With large enough sample size, this bias will be minimal. However in settings like non-parametric regression, even with large sample sizes the bias term does not dimish

quickly. One suggestion then is to calculate the bias in estimation and correct for it accordingly (Cummins et al., 2001).

Separately in the study of Markov Random Fields, authors have also addressed bias but with respect to estimating the partition function, i.e, the normalizing constant of these models (see Burda et al., 2015; Liu et al., 2015).

So in functional approximation it is important to address the bias in the estimation procedure. We address and correct for bias in our functional approximation using IS, particularly in the log-likelihood approximation. We discuss this further in Chapter 3.

1.7 Output Analysis

The second task of any Monte Carlo method is using the representative sample from the target distribution for inference, i.e., output analysis. This encompasses constructing point estimates, standard errors, confidence intervals, confidence regions, and the like. We have already introduced point estimates in Section 1.4. At this stage we can now further discuss the two types of estimation we make: univariate or multivariate. First though, we must introduce Monte Carlo error and its tie with simulation effort.

1.7.1 Monte Carlo Error and Simulation Effort

Suppose F is still the target distribution, now let $g : \mathcal{X} \to \mathbb{R}^p$ where $\theta = \mathbb{E}_F g \in \mathbb{R}^p$ is a vector of features of interest of F, and let θ_m be the Monte Carlo estimator. By the LLN, $\theta_m \approx \theta$ for large m. However, no matter how large m, there will be an unknown *Monte Carlo error*, $\theta_m - \theta$. Notice this Monte Carlo error is due to the "pseudo randomness of computer simulation, rather than randomness of real world phenomena" (Brooks et al., 2011, Chapter 1). Other error may be introduced by using some observed data but is unrelated to the Monte Carlo error, hence why we distinguish the error introduced by the Monte Carlo procedure.

An obvious question is when is m sufficiently large so that the approximation is good? Another way to think about this is, what does m have to be so that the estimation

is reliable in the sense that if we run the procedure again we will not obtain broadly different estimates (Flegal et al., 2008a)?

There are two main approaches to this question. The first is a fixed-time approach which pre-specifies a Monte Carlo sample size and terminates when it is achieved. The second is a random-time approach where the simulation is terminated based on some criterion regarding the quality of the Monte Carlo approximation.

Calculation of θ_m alone, and hence the fixed-time approach, is an incomplete solution. Indeed θ_m will be more valuable if we attach a measure of the Monte Carlo error. Thus a solution to the problem will be a point estimate θ_m and a measure of the Monte Carlo error. In the next sections we discuss how to calculate the Monte Carlo error when samples are iid versus correlated for both univariate and multivariate estimation. We also discuss how to create confidence intervals or regions and how to use them to determine simulation effort.

1.7.2 IID Output Analysis

As long as we draw iid samples from the target distribution, we can use classical large-sample frequentist methods to assess the quality of the estimation. These include sample means and a CLT (for finite variance).

If $X_1, \ldots, X_m \overset{iid}{\sim} F$ where $\theta_m = m^{-1} \sum_{t=1}^{m} g(X_t)$ then as $m \to \infty$,

$$\sqrt{m}(\theta_m - \theta) \xrightarrow{d} N_p(0, \Sigma) \tag{1.34}$$

where Σ is the $p \times p$ positive definite covariance matrix.

Univariate

Let $\theta_{i,m}$ and θ_i denote the ith components of the vectors θ_m and θ where $\theta_i = \mathbb{E}_F g_i$ where $g = (g_1, \ldots, g_p)$. Also, let $\sigma_i^2 = Var_F g_i$ be the ith diagonal element of Σ and $\sigma_{i,m}^2$ the sample variance. The Monte Carlo standard error of $\theta_{i,m}$ is then $\sigma_{i,m}/\sqrt{m}$. We can

construct the usual $(1-\alpha)\%$ confidence intervals for the point estimates with,

$$\theta_{i,m} \pm z_{1-\alpha/2}\frac{\sigma_{i,m}}{\sqrt{m}} \tag{1.35}$$

where $z_{1-\alpha/2}$ is the $(1-\alpha/2)$ quantile from the standard Normal distribution. With the usual CLT conditions, we could also construct the intervals using the Student's t distribution, but as addressed in Owen (2013), since Monte Carlo applications usually use large m, there is no practical difference between t-based confidence intervals since the only component that changes in line (1.35) is replacing $z_{1-\alpha/2}$ by $t_{1-\alpha/2,m-1}$ which converges to $z_{1-\alpha/2}$ as $m \to \infty$.

To figure out what m we need for a specific margin of error, E, or interval width, we can use simple algebra to solve that depending on the sample variance,

$$m \geq \left(\frac{z_{1-\alpha/2}\sigma_{i,m}}{E}\right)^2.$$

Multivariate

Notice we are using one Monte Carlo sample to estimate multiple features. Therefore, there may be a relationship between these p features of interest that we overlook if we just conduct univariate analysis. Therefore, we can refer back to the CLT in equation 1.34 to build confidence regions instead of intervals. A $100(1-\alpha)\%$ confidence region is the set of all θ such that

$$(\theta_m - \theta)^T \Sigma^{-1} (\theta_m - \theta) \leq \chi^2_{p,1-\alpha}$$

where $\chi^2_{p,1-\alpha}$ denotes the $(1-\alpha)$ quantile of a Chi-squared distribution with p degrees of freedom.

However, we do not know Σ so replace it with a consistent estimator, Σ_m, such as the sample covariance matrix. Then we can define the confidence region as a p-dimensional ellipsoid centered at θ_m as the set of all θ that satisfy,

$$m(\theta_m - \theta)^T \Sigma_m^{-1} (\theta_m - \theta) \leq \frac{(m-1)p}{(m-p)} F_{1-\alpha,p,m-p}$$

where $F_{1-\alpha,p,m-p}$ denotes the $(1-\alpha)$ quantile of the F-distribution with p and $m-p$ degrees of freedom.

1.7.3 MCMC Output Analysis

Classical iid results do not follow anymore when using correlated MCMC samples, in particular the variance calculations. Now consider $\{X_t^*\}_{t=0}^{m-1}$ is a Markov chain with stationary distribution F. Let $\theta_m^* = m^{-1}\sum_{t=0}^{m-1} g(X_t^*) \in \mathbb{R}^p$ be the MCMC estimator of θ. Assume conditions for the Markov Chain CLT hold, that is,

$$\sqrt{m}(\theta_m^* - \theta) \xrightarrow{d} N_p(0, \Sigma^*) \tag{1.36}$$

where Σ^* is a $p \times p$ positive definite matrix such that

$$\Sigma^* = Var_F(X_0^*) + \sum_{j=1}^{\infty} Cov_F(g(X_0^*), g(X_j^*)) + \sum_{j=1}^{\infty} Cov_F(g(X_j^*), g(X_0^*)). \tag{1.37}$$

Univariate

Let $\theta_{i,m}^*$ be the MCMC estimator of θ_i, then $\theta_{i,m}^*$ differs from $\theta_{i,m}$ as it is a sample mean of correlated draws. By the Ergodic theorem we still have that $\theta_{i,m}^* \to \theta_i$ with probability 1 as $m \to \infty$. Let σ_i^{*2} be the ith diagonal component of Σ^*, then $\sigma_i^*/\sqrt{m}$ is the Monte Carlo error of $\theta_{i,m}^*$. In addition, as $m \to \infty$,

$$\sqrt{m}(\theta_{i,m}^* - \theta_i) \xrightarrow{d} N(0, \sigma_i^{*2}). \tag{1.38}$$

Notice by equation 1.37, $\sigma_i^{*2} \neq Var_F g(X_0^*)$. In the univariate setting, equation 1.37 simplies to

$$\sigma_i^{*2} = Var_F g_i(X_0^*) + 2\sum_{j=1}^{\infty} Cov_F(g_i(X_0^*), g_i(X_j^*)).$$

As a result, calculating σ_i^* has garnered great attention (see, e.g., Flegal et al., 2010; Geyer, 1992; Jones et al., 2006a). Using a consistent estimator of σ_i^{*2}, we can then

construct $(1-\alpha)\%$ confidence intervals with,

$$\theta^*_{i,m} \pm t_* \frac{\sigma^*_i}{\sqrt{m}} \tag{1.39}$$

where t_* corresponds to the appropriate Student's t quantile.

In order to determine when to stop the simulation process, Jones et al. (2006a) implement a fixed-width sequential stopping rule that stops simulation when the width of the confidence interval for each component i is small. More specifically, for a desired interval half width ϵ_i for component i, sampling is terminated after collecting some $m^* \geq 0$ samples when,

$$t_* \frac{\sigma^*_{i,m}}{\sqrt{m}} + \epsilon_i \mathbb{I}(m \leq m^*) \leq \epsilon_i. \tag{1.40}$$

We need to have enough samples to obtain a good estimate of σ^{*2}_i, hence the specification of an m^*, i.e., a minimum simulation effort. The resulting m is the Monte Carlo sample size for reliable estimation, not for guaranteeing convergence of the Markov chain to its stationary distribution (Vats et al., 2019). Therefore, for each component i there is a different termination time.

The fixed-width sequential stopping rule is one approach for determining when to end the simulation. Another approach uses the effective sample size, which is the number of iid samples needed for an equivalent Monte Carlo standard error. Let $\sigma^2_i = Var_F g_i(X_0)$. Then the univariate effective sample size (ESS) is,

$$ESS_i = m \left(\frac{\sigma_i}{\sigma^*_i} \right)^2. \tag{1.41}$$

Meaning, one can pre-specify an ESS and run the sampling process until that ESS is achieved.

Multivariate

Similar to the iid case, if we analyze the components of θ^*_m separately, we ignore any potential dependency. Being so, we return to the CLT in equation (1.36) where we now

want a consistent estimator of Σ^*.

Estimation of Σ^* is nontrivial and has attracted a significant research interest (Andrews, 1991; Chen and Seila, 1987; Dai and Jones, 2017; Hobert et al., 2002; Jones et al., 2006b; Kosorok, 2000; Liu and Flegal, 2018a,b; Seila, 1982; Vats and Flegal, 2018; Vats et al., 2019, 2018). Some approaches to estimate Σ^* include multivariate batch means, spectral variance estimators, and multivariate initial sequence estimators. Further details of each of these methods are found in their corresponding papers. In Chapter 2, we go into greater detail about multivariate batch means.

Assuming we have a consistent estimator of the covariance matrix, denoted as Σ^*_m, then,

$$(\theta^*_m - \theta)^T (\Sigma^*_m)^{-1} (\theta^*_m - \theta) \xrightarrow{d} \text{Hotelling's } T^2_{p,q}$$

where q is determined by the estimation procedure used for Σ^*_m and the Hotelling's T distribution is further discussed in Chapter 2.

To determine when to stop the sampling procedure one way is to directly extend the MCMC univariate output analysis results to multivariate estimation. For instance, we can define a common termination time as the Monte Carlo sample size that satisfies the fixed-width sequential stopping times for all components. Then we can do some type of multiple correction to construct the intervals. However this approach can be challenging when p is moderately large and it ignores the cross-correlation among the features of interest.

Vats et al. (2019) instead develop a multivariate MCMC output analysis framework. In their work they suggest a *relative standard deviation fixed-volume sequential stopping rule* that stops the sampling procedure once the volume of the confidence region is relatively small, indicating that the Monte Carlo error is small compared to the variation in the target distribution. An equivalent termination rule they note is to stop when a minimum multivariate effective sample size has been reached. The details of both of these approaches are in Chapter 2.

1.8 Reliability of Monte Carlo Methods

In the remainder of this thesis we use Monte Carlo methods to estimate network properties and approximate functions. The use of Monte Carlo is these cases is not novel. However, little to no work exists on assessing the reliability of the estimation. This dissertation tries to start filling that gap. More specifically, in Chapter 2 we look to first extend the multivariate MCMC output analysis tools by Vats et al. (2019) in the context of network property estimation. We generalize parts of the output analysis framework to work with popular network sampling algorithms. We follow in Chapter 3 by assessing the Monte Carlo error in functional approximation using IS. Although we focus the approximation on density and log-likelihood functions, we develop a general framework for constructing simultaneous confidence bands that could be applied in other contexts. In addition, we propose a correction to improve reliability of MCLA. In both chapters we conclude with several examples highlighting the importance of further studying and analyzing the Monte Carlo estimation.

Chapter 2

Ensuring Reliable Monte Carlo Estimates of Network Properties

Networks are a powerful way to represent relational data or systems of interconnected elements. Typically modern day networks are enormous or challenging to construct and therefore difficult to study by brute force. In particular, much of the network literature has focused on complete network data (Kolaczyk, 2009; Scott, 2017; Wasserman and Faust, 1994), but in many practically relevant settings, the network is either sufficiently large or complicated to be fully enumerated. In such cases, traditional survey sampling methods, e.g., simple random sampling, are not practical due to the absence of a sampling frame. Alternatively, one can collect an approximately uniform sample of nodes from a network by using Monte Carlo methods to traverse the structure in a nondeterministic manner. Features of interest can then be estimated using sample statistics. A particular focus within the network sampling literature is on traversing networks with MCMC methods such as random walk-based algorithms. These methods, also known as link trace sampling, have been used on large online social networks, like Facebook, to estimate average connectedness and clustering coefficients (Gjoka et al., 2011) or on hard to reach populations, such as individuals at high risk for HIV, to estimate disease prevalence (Thompson, 2017). While the use of MCMC methods to estimate network features is well known, the quality of estimation with these Monte Carlo samples has not

been directly addressed in a computationally effective way. We contribute to this area by considering and further developing multivariate MCMC output analysis methods in the context of network sampling that directly address the reliability of estimation.

The rest of this chapter is organized as follows. We begin Section 1 with a brief introduction to network science, highlighting its mathematical foundations and interdisciplinary nature. Then in Section 2 we elaborate on why we need methods to sample networks without sampling frames and introduce basic network notation, terminology, and network features of interest. We follow in Section 3 and 4 with an overview of Markov chains on graphs and introduce output analysis tools to determine multivariate MCMC estimation reliability. In Section 5 we further develop these output analysis tools in the context of network sampling, providing three examples of its use in Section 6 on a simple simulated high school social network to illustrate the concepts and progressively move to more complicated, larger networks. Finally, we conclude with a discussion in Section 7.

2.1 Introduction

A Brief Introduction to Network Science

The study of networks, known as networks science, network analysis, or social network analysis, is not a novel discipline. In the 1730s, Euler wrote about the now famous Köenigsberg bridge problem (Newman, 1953). In this exercise, Euler considered how to travel across an island by traversing all of its seven bridges only once. He proved that completing such a task is impossible and that in order to reach all parts of the island, you must travel across at least one bridge more than once. With this simple yet groundbreaking solution, Euler is often cited as starting the fields of graph theory and topology. Many years later another mathematician, D. Köing, was credited with developing the formal details of graph theory (König, 1950); building the mathematical foundations of network science. The growth of network science continued in physics with the study of electrical circuits and in chemistry with molecular structure (Kolaczyk, 2009). Sociology also had its own contributions to the discipline when Jacob Moreno

introduced more precise techniques to account for social configurations (Moreno and Jennings, 1938). By the 1960s, mathematicians Erdös and Renyi had published several articles on random graphs (Erdos and Rényi, 1960), which led to the development of a probabilistic framework in network science. Around the same time there were also advances in operations research and computer science that allowed researchers to study network-driven problems that involved, to name a few, transportation, trading, and supply chain systems (Kolaczyk, 2009).

Of particular importance in the last 10-15 years is the growth of Online Social Networks (OSNs), like Facebook, Twitter, and LinkedIn (Ellison et al., 2007). These networks provide an online platform for individuals to not only connect with other individuals, but also other companies, government entities, and organizations (and vice versa). These connections range in purpose from friendship to employment opportunities to information dissemination. As a result, OSNs generate an immense amount of information, creating some of the largest sources of relational data available. This wealth of information has spurred a new wave of interdisciplinary social network research (Kumar et al., 2010). An in depth history and breakdown of network and social network analysis can be found in several introductory network analysis textbooks, see e.g., Jackson and Watts (2002); Luke and Harris (2007); Wasserman and Faust (1994), and Kolaczyk (2017).

Despite the advances in network science, approaches on how to sample and estimate network properties is still an open area of research. We focus on studying methods used on networks without a sampling frame.

2.2 Network Sampling Without a Sampling Frame

Having access to an entire network dataset is often impractical. As a result, we typically use sample data instead to study and infer properties of the network. Sampling from a network may seem to be straightforward, but obtaining a sample and using it to estimate network parameters is a nontrivial task (Scott, 2017). A sampling frame is a list of elements or people in the population from which a sample is taken. It is common that we do not have a sampling frame for the network we are interested in.

For example, the sampling frame for an OSN is a complete list of all the users in the network. We would need to sample from a network with no sampling frame when 1) the network is enormous, 2) there are access or privacy limitations, or 3) the network is of a non-identifiable population.

1. Size: When networks are massive (e.g., OSNs like Facebook or Twitter, where Facebook reported ∼2 billion monthly active users in March 2019 (Facebook, 2019)), it is computationally infeasible to enumerate, retrieve, or study the full dataset.

2. Privacy: Due to some OSNs' privacy limitations (e.g., Facebook), it is impossible to access the full network data unless you work at the companies themselves. However, even if full access to large OSNs is available, using the entire network data still does not eliminate the size challenge.

3. Non-identifiable populations: Certain hard-to-reach, hidden, or vulnerable populations, like the homeless, injection drug users, undocumented immigrants, or HIV-positive individuals, cannot be reached with traditional survey sampling methods as they often face barriers which make it impracticable to construct a sampling frame for them. To better understand these populations, researchers instead opt to sample from their social networks (Heckathorn and Cameron, 2017).

To handle the absence of a sampling frame, one can collect an approximately uniform sample of nodes from a network by using Monte Carlo methods to traverse the structure in a nondeterministic manner. Features of interest can then be estimated using sample statistics. Constructing MCMC sampling algorithms to efficiently traverse a network is not trivial and is an active area of research. As a result, there has been substantial work on comparing various MCMC sampling methods for networks, but the comparisons usually only consider the properties of univariate point estimates, computation speed (i.e., clock time or percent of network sampled), or the difference in empirical distributions using the Kullback-Leibler divergence, Kolmogorov-Smirnov D-statistic, or the total variation distance (see, among others, Ahmed et al., 2014; Avrachenkov et al., 2018; Blagus et al., 2017; Gile and Handcock, 2010; Gjoka et al., 2011; Joyce, 2011; Lee et al., 2012, 2006; Leskovec and Faloutsos, 2006; Li et al., 2015; Salamanos

et al., 2017; Wang et al., 2011; Zhou et al., 2016). Typically the goal is to estimate many network features based on one Monte Carlo sample, while comparisons typically focus on univariate summaries. That is, the multivariate nature of the estimation problem has been broadly ignored.

Moreover, separate from the natural variability in the data, the estimates produced by these Monte Carlo methods are also subject to additional Monte Carlo error. In that, different runs of the algorithm or nodes traversed will result in different estimates, where the algorithm used will impact the quality of the estimation. Of course, if the Monte Carlo sample sizes are large enough, then the differences in run estimates will be negligible. This, then raises the question, how large is large enough? That is, how large does the Monte Carlo sample need to be so that the estimates are trustworthy?

The current tools used in the network sampling literature to determine when to terminate the sampling process are insufficient. Popular methods rely on the use of so-called convergence diagnostics (Cowles and Carlin, 1996; Gelman and Rubin, 1992; Geweke, 1992; Heidelberger and Welch, 1983), but none of these methods make any attempt to assess the quality of estimation (Flegal et al., 2008b; Jones et al., 2006a). Moreover, these diagnostics have been shown to stop the sampling process prematurely (Jones et al., 2006a; Vats and Knudson, 2018). Another common approach is to study the running mean plot and determine the point at which it stabilizes to find approximately when the estimates have settled (Gjoka et al., 2011; Lee et al., 2006; Lu and Li, 2012; Ribeiro and Towsley, 2010). This approach is inadequate since its interpretation is subject to how much one zooms in on a section of the plot.

Although the network sampling literature on Monte Carlo estimation reliability is relatively sparse, Avrachenkov et al. (2016); Chiericetti et al. (2016); Lee et al. (2006); Salamanos et al. (2017), and Wang et al. (2011) considered the relative error or normalized root mean squared error of sample estimates from various sampling methods. However, neither approach takes into account the multivariate nature of the problem nor tries to calculate the sample variance from the correlated sampling procedure. In addition, Mohaisen et al. (2010) and Zhou et al. (2016) discuss the theoretical mixing time of the sampling algorithms they propose, although theoretically valid they are impractical to implement. Recently Rohe et al. (2019) introduced a threshold for design

effects to find a minimum sample size applicable to Respondent Driven Sampling (RDS) with multiple recruitments, a type of branching process used on networks to estimate one network feature.

An extensive amount of work exists in the network literature on sampling hard-to-reach populations with RDS (see Gile and Handcock, 2010). We do not consider RDS in this work as it is a specific type of sampling without replacement that does not align with methods we study. In short, to the best of our knowledge, we are not aware of any other work that directly address the reliability of the multivariate estimation with these MCMC network samples.

In this chapter we consider and further develop multivariate MCMC output analysis methods (see e.g. Flegal et al., 2015; Vats et al., 2019, 2018) in the context of network sampling with respect to two MCMC algorithms: a simple random walk and a random walk-based version of the Metropolis-Hastings algorithm. This approach yields principled, computationally efficient, and broadly applicable methods for assessing the reliability of the Monte Carlo estimation procedure. In particular, we construct and compare network parameter estimates, effective sample sizes, coverage probabilities, and stopping rules.

Before we present these random walk sampling algorithms in greater detail, we first introduce some basic network notation and terminology.

2.2.1 Notation

A network, or graph, is a relational structure comprised of two elements: a set of nodes or vertices (used interchangeably), and a set of vertex pairs representing edges or ties (i.e., a relationship between two nodes). Formally, let V denote a non-empty countable set of nodes, $E \subseteq V \times V$ denote the set of edges between the vertices, and $G = (V, E)$ denote the network. Define the network size, n, to be the set cardinality of V. Similarly, n_e is the number of edges in the graph. If there is an edge from node i to node j we write this as $i \to j$ and as $i \nrightarrow j$ if there is no connection between them. We can equivalently

think of the network G as an $n \times n$ binary adjacency matrix Y, where each element

$$y_{ij} = \begin{cases} 1 & \text{if } i \to j \\ 0 & \text{if } i \nrightarrow j. \end{cases}$$

These ties can either be directed (from $i \to j$) or can be undirected and therefore mutual ($i \leftrightarrow j$). We only consider networks with no self loops, i.e., $y_{ii} = 0$ for all $i \in V$. If Y represents an undirected network, then Y is a square symmetric matrix. These edges can also be weighted, such that

$$y_{ij} = \begin{cases} w_{ij} & \text{if } i \to j \\ 0 & \text{if } i \nrightarrow j \end{cases}$$

where $w_{ij} \in \mathbb{R}$. If $w_{ij} > 0$ this indicates a positive or strong relationship and $w_{ij} < 0$ indicates a negative or adversarial one.

In the binary network (where $y_{ij} = 0$ or 1) we denote the connections or number of ties for any node i as d_i, also referred to as the degree of node i. For a directed graph, there is a distinction between the incoming versus outgoing degree, whereas for undirected graphs, d_i is simply all connections for node i that are both incoming and outgoing, so $d_i = \sum_{j=1}^{n} y_{ij}$.

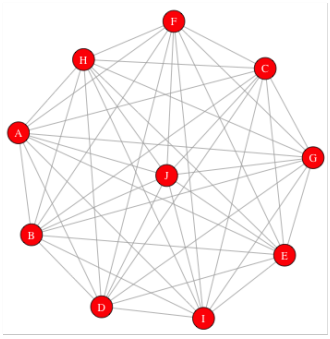
There are a few other topological features that are unique to graphs that may impact sampling.

- Connectivity: is a measure of how many edges exist in the graph versus the number possible (see Figure 2.1 for examples). Connectivity is also referred to as network *density*, which is simply the ratio of the number of observed edges to the number of possible edges, where for an undirected graph, $density = n_e / \binom{n}{2} \in [0, 1]$. A graph is called *completely connected* if all possible edges between any two nodes exists, and is called *empty* if there are zero observed edges. Likewise, a *dense* graph is one where the number of observed edges is close to the maximum number possible and a *sparse* graph is one where few edges exist. The thresholds for what defines dense and sparse vary by context. In addition, a *well-connected* graph is

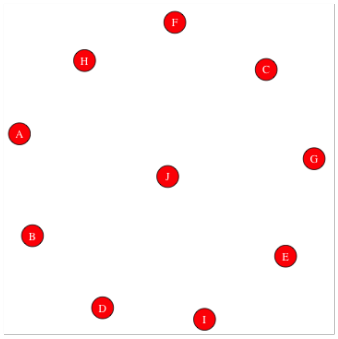
one where there is a path from any node i to any other node j in a finite number of steps. The condition of a graph being well-connected is of utmost importance for random walk sampling methods.

- Clustering: is when a set of nodes have a large number of edges between them but are sparsely connected to nodes outside of this set (Figure 2.2a); so that the network appears to be a bunch of sparsely connected groups.
- Bipartite: a bipartite graph is when there are two distinct sets of nodes that have no edges within the sets, but have edges between them (Figure 2.2b).
- Degree Assortativity: is the tendency of nodes of similar degree to connect to one another (Newman, 2003). A network is *assortative* when high degree nodes generally connect with other high degree nodes, and low degree nodes generally connect with other low degree nodes. The network is *disassortative* when high degree nodes in general connect with low degree nodes and vice versa. Degree assortativity, or ρ, ranges between -1 (disassortative) and 1 (assortative) and is interpreted similar to correlation, where $\rho = 0$ implies there is no relationship between degree and an edge existing between two nodes. The level of assortativity in a network can have a profound affect on its topology, see Noldus and Van Mieghem (2015) for more details.

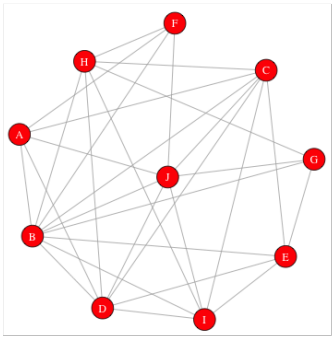
Unless otherwise noted G is an undirected, unweighted, well-connected, and non-bipartite graph.



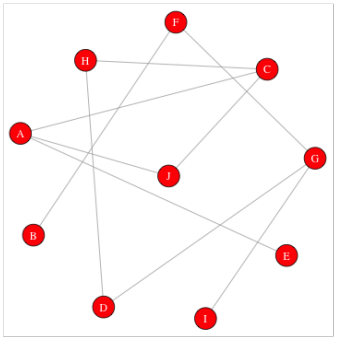
(a) Completely connected graph



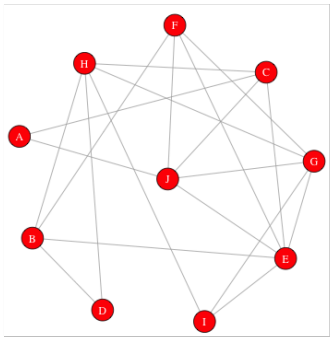
(b) Empty graph



(c) Dense graph

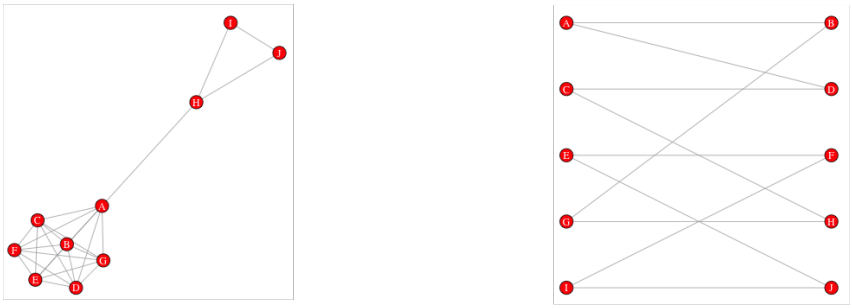


(d) Sparse graph



(e) Well-connected graph

Figure 2.1: Examples of different connectivity on a network with 10 nodes.



(a) Two clustered graph **(b)** Bipartite graph

Figure 2.2: Examples of different connection types on a network with 10 nodes.

2.2.2 Node Level Attributes

1. **Degree**

   As introduced earlier, degree is the number of connections of a node. If the node has no edges or connections then the degree is zero, i.e., it is an isolate. Degree plays an important role in networks as it is a measure of connectivity and can indicate a node's importance; where the higher the degree the more important a node. Node degree also plays a role in how communities operate, ideas spread, and how individuals communicate with one another (Newman et al., 2002).

2. **Triples**

   A triple is a set of three nodes connected by two edges, or in another words an open triangle. Consider the diagram in Figure 2.3. Here node A has one triple, since A is connected to B and A is connected to C, which we can write as BAC. If A were connected as well to node D, then it would have three triples (BAC, BAD, CAD). Notice then, the number of triples for a node A is $\binom{d_A}{2}$, where d_A stands for the degree of node A. Triples are another measure of connectivity in a graph and are used to understand the level of clustering in a network.

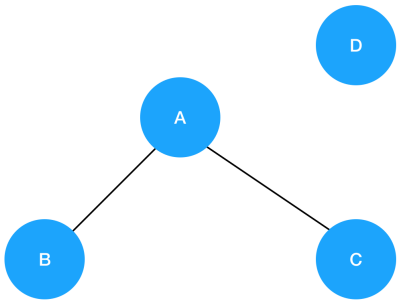


Figure 2.3: Node A has one triple since $A \leftrightarrow B$ and $A \leftrightarrow C$.

3. **Triangles**

   Triangles are one of the most important measures in a network (Watts and Strogatz, 1998). A triangle is a set of three nodes fully connected (see Figure 2.4). Triangles represent interconnectedness and transitivity of network connections. If $A \leftrightarrow B$ and $A \leftrightarrow C$ then how likely is $B \leftrightarrow C$? These transitive connections are prevalent and important in many networks. For instance, there is a common belief that friends of friends are likely friends themselves. Or, in patient sharing networks among physicians, triangles occur when physicians treat common patients, creating systems of care. Not only do triangles represent network transitivity, but they are also used to detect naturally forming communities (Durak et al., 2012). In combination with triples, triangles are used to construct a measure of clustering.

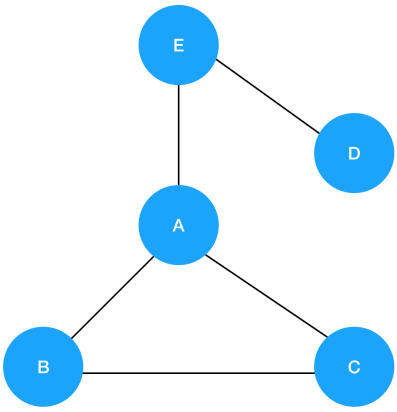


Figure 2.4: Node A has one triangle since $A \leftrightarrow B$, $A \leftrightarrow C$ and $B \leftrightarrow C$.

4. **Other nodal attributes**

   Nodes can have other attributes of interest. In social networks, if each node represents an individual, these nodal attributes could be demographic information like age, sex, education, income, or disease status. In business networks, attributes may be company size, yearly revenue, etc. These attributes are critical as they may play a role in network topology.

2.2.3 Network features

The network features of interest can be expressed as the mean of a function over the entire network. We focus on means over nodal attributes, although it is possible to consider means over edge attributes. More formally, suppose $g : V \to \mathbb{R}^p$ where p is the number of features of interest and let λ be the uniform distribution on V. Then, if $X \sim \lambda$, we want to calculate the p-dimensional mean vector

$$E_\lambda[g(X)] = \frac{1}{n} \sum_{v \in V} g(v). \tag{2.1}$$

It will be notationally convenient to denote $E_\lambda[g(X)] = \mu_g$ and we will use both interchangeably. Specific network features of interest might include: mean degree, degree

distribution, mean clustering coefficient, and proportion of nodes with specific nodal attributes, e.g., proportion of female users in an online social network.

Computing μ_g is often difficult in practically relevant applications and hence we turn to MCMC. Part of the popularity of random walks on networks is that they are finite state space Markov chains and thus have a well-established theory. We introduce some properties of Markov chains on graphs in greater detail in the next section.

2.3 Markov Chains on Graphs

A random walk (RW) on an undirected, non-bipartite, and well-connected graph G is a finite state space Markov chain. The RW is defined on the finite state space since the walk travels on the finite vertex set, V. Suppose we take a RW on G, then the transition probability of moving from any node i on the $t-1^{\text{th}}$ step to a neighboring node j on the t^{th} step is,

$$p_{ij} = \mathbb{P}(V_t = j | V_{t-1} = i) = \frac{1}{d_i}.$$

Notice, this is a *first order* Markov chain since conditioned on the current position the next move does not depend on the previously visited nodes. Moreover, this walk is *time-homogeneous* since the transition probabilities of traveling between nodes remain the same across time, i.e., p_{ij} does not depend on t. We can define a matrix, $P \in [0,1]^{n \times n}$ as the transition probability matrix of this walk, where the network size is n and $P(i,j) = p_{ij}$. Let $P(i,\cdot)$ denote the i^{th} row of this matrix, then this row is the distribution $P(i,\cdot)$ which means P is stochastic and

$$\sum_{j \in V} P(i,j) = 1 \quad \text{for all } i \in V.$$

Since G is well-connected and undirected, P is irreducible. If G was not well-connected, there would be some node(s) that would not be reachable from any other node in a finite number of steps. This would imply that P is not irreducible, and the results we present below would not be applicable.

Let $\mathcal{T}(i) := \{t \geq 1 : P^t(i,i) > 0\}$, then $\mathcal{T}(i)$ is the set of the number of steps when

it is possible for the walk to return to the starting node i. Since P is irreducible, the gcd $\mathcal{T}(i) = \text{gcd}\ \mathcal{T}(j)\ \forall\ i, j \in V$ (Levin et al., 2009). In addition, because G is non-bipartite and undirected, $\mathcal{T}(i)$ contains both even and odd cycles. Therefore, the gcd $\mathcal{T}(i) = 1\ \forall\ i \in V$ which implies the chain is aperiodic. In addition, the RW on G is a detailed balanced Markov chain, where for a probability distribution, λ on V,

$$\lambda(i)P(i,j) = \lambda(j)P(j,i) \quad \text{for all } i, j \in V.$$

Therefore, λ is the unique stationary distribution for the irreducible and aperiodic P, where $\lambda \in \mathbb{R}^n$ such that $\lambda P = \lambda$ and $\lambda > 0$ (Levin et al., 2009). Intuitively this means if we let the walk go long enough, there is some stable probability of visiting any node in the graph.

Then Birkhoff's Ergodic theorem follows (Birkhoff, 1931; Bremaud, 2010). Although already introduced in Chapter 1, we rewrite the Theorem here using the network notation.

Theorem 2.3.1. *(Birkhoff, 1931) For a RW $\{V_0, V_1, \ldots, V_{m-1}, \ldots\}$ on a well-connected undirected graph with function $g : V \to \mathbb{R}^p$. Let $\{X_t\} = \{g(V_t)\}$. Define $\mu_g = E_\lambda g(V)$ and $\mu_m = m^{-1} \sum_{t=0}^{m-1} g(V_t)$. If $E_\lambda |g(V)| < \infty$ then,*

$$\mu_m \to \mu_g \quad \textit{with probability 1 as } m \to \infty.$$

Thus estimation of μ_g is straightforward; simulate m steps of the Markov chain and use the sample mean. However, quality of estimation depends on the Monte Carlo sample size, m, since for a finite m there will be an unknown Monte Carlo error, $\mu_m - \mu_g$. We can begin to assess this error through the CLT introduced in Chapter 1 (see e.g. Aldous et al., 1997; Jones, 2004; Vats et al., 2019). Again, using the network notation,

Theorem 2.3.2. *(Jones, 2004; Vats, 2017) Following the definitions in Theorem 2.3.1, there exists a $p \times p$ positive definite matrix, Σ such that as $m \to \infty$*

$$\sqrt{m}(\mu_m - \mu_g) \xrightarrow{d} N_p(0, \Sigma)$$

where $\Sigma = Var_\lambda(X_1) + \sum_{t=1}^{\infty} [Cov_\lambda(X_0, X_t) + Cov_\lambda(X_0, X_t)^T]$.

Since the chain is on the finite state space V, it is uniformly ergodic (Aldous et al., 1997). If $\|\cdot\|$ denotes the standard Euclidean norm, then, given our assumptions on the Markov chain, the remaining requirement for Theorem 2.3.2 is that $E_\lambda[\|g\|^2] < \infty$, which typically will hold.

The matrices Σ and $\Lambda := \text{Var}_\lambda(g(V_0))$ will be fundamental to the remainder of this chapter. Estimating Λ is straightforward using the sample covariance, denoted Λ_m, but estimating Σ is difficult. As introduced in Chapter 1, there are several approaches to estimate Σ, such as spectral variance estimators, but these are computationally demanding especially with large Monte Carlo sample sizes (Liu and Flegal, 2018b). Due to computational feasibility, we will only consider the method of batch means, which we present below.

2.3.1 Multivariate Batch Means

Let $\{X_t,\, t \geq 0\} = \{g(V_t),\, t \geq 0\}$ and set $m = a_m b_m$ where a_m is the number of batches and b_m is the batch size. For $k = 0, \ldots, a_m - 1$ set

$$\bar{X}_k := b_m^{-1} \sum_{t=0}^{b_m - 1} X_{k b_m + t}.$$

Then $\bar{X}_k$ is the mean vector for batch k and the estimator of Σ is

$$\Sigma_m = \frac{b_m}{a_m - 1} \sum_{k=0}^{a_m - 1} (\bar{X}_k - \mu_m)(\bar{X}_k - \mu_m)^T.$$

For Σ_m to be positive definite, $a_m > p$. It is common to choose $a_m = \lfloor m^{1/2} \rfloor$ or $a_m = \lfloor m^{1/3} \rfloor$ where $a_m > p$ is met. Batch means produces a strongly consistent estimator of Σ (Vats et al., 2019) under conditions similar to those required for Theorem 2.3.2 and is implemented in the `mcmcse` R package (Flegal et al., 2015).

2.4 MCMC Output Analysis

It would be natural to use the CLT and Σ_m to form asymptotically valid confidence regions for μ_g. The volume of the confidence region could then be used to describe the precision in the estimation and, indeed, this sort of procedure has been advocated (Jones et al., 2006a). More specifically, if $T^2_{1-\alpha,p,q}$ denotes the $1-\alpha$ quantile of a Hotelling's T-squared distribution where $q = a_m - p$, then a $100(1-\alpha)\%$ confidence ellipsoid for μ_h is the set

$$C_\alpha(m) = \{\mu_g \in \mathbb{R}^p : m(\mu_m - \mu_g)^T \Sigma_m^{-1}(\mu_m - \mu_g) < T^2_{1-\alpha,p,q}\}.$$

The volume of the ellipsoid is given by

$$\text{Vol}(C_\alpha(m)) = \frac{2\pi^{p/2}}{p\Gamma(p/2)} \left(\frac{T_{1-\alpha,p,q}}{m}\right)^{p/2} |\Sigma_m|^{1/2}.$$

We can get the respective $T_{1-\alpha,p,q}$ by using its relationship with the F-distribution. For a random walk of m steps with p estimates using the mBM estimator, a_m = number of batches and $q = a_m - p$,

$$T_{1-\alpha,p,q} = \frac{pq}{q-p+1} F_{1-\alpha,p,q}.$$

One could then terminate a simulation when the volume is sufficiently small, indicating that our Monte Carlo error is sufficiently low. However, the fixed-volume approach is difficult to implement even when p is small (Flegal et al., 2015; Glynn and Whitt, 1992; Vats et al., 2019).

An alternative is to terminate the simulation when the volume is small compared to the generalized variance of the limiting normal distribution in the CLT (Wilks, 1932), that is, if $|\cdot|$ denotes determinant, small compared to $|\Lambda|$. The intuition is that when the Monte Carlo error is small compared to the variation in the target distribution, then it is safe to stop. More formally, letting $m^* > 0$ and $\epsilon > 0$ be given, then we terminate the simulation at the random time $T_{SD}(\epsilon)$ defined as,

$$T_{SD}(\epsilon) = \inf\left\{m \geq 0 : \text{Vol}(C_\alpha(m))^{1/p} + \epsilon|\Lambda_m|^{1/2p} I(m < m^*) + m^{-1} \leq \epsilon|\Lambda_m|^{1/2p}\right\}.$$

The role of m^* is to require some minimum simulation effort; at least it should be such that both Λ_{m^*} and Σ_{m^*} are positive definite.

We can connect $T_{SD}(\epsilon)$ to effective sample size,

$$\text{ESS} = m \left[\frac{|\Lambda|}{|\Sigma|} \right]^{1/p} \tag{2.2}$$

and naturally estimated with

$$\widehat{\text{ESS}} = m \left[\frac{|\Lambda_m|}{|\Sigma_m|} \right]^{1/p}. \tag{2.3}$$

When $p = 1$, the ESS is simply the univariate ESS (Gong and Flegal, 2016) and when there is no correlation in the chain, ESS $= m$ since $\Sigma = \Lambda$. Vats et al. (2019) showed that upon rearrangement, the $T_{SD}(\epsilon)$ relates to a minimum ESS, i.e., a termination time when the ESS is larger than a lower bound.

$$\widehat{ESS} \geq \left[\left(\frac{2\pi^{p/2}}{p\Gamma(p/2)} \right)^{1/p} (T_{1-\alpha,p,q})^{1/2} + \frac{|\Sigma_m|^{-1/2p}}{m^{1/2}} \right]^2 \frac{1}{\epsilon^2} \approx \frac{2^{2/p}\pi}{(p\Gamma(p/2))^{2/p}} (T_{1-\alpha,p,q}) \frac{1}{\epsilon^2}.$$

This lower bound is a function of the chain length, m, through q and thus difficult to determine before the simulation. The authors point out though, as $m \to \infty$, $T^2_{p,q} \to \chi^2_p$ which implies,

$$\widehat{ESS} \geq \frac{2^{2/p}\pi}{(p\Gamma(p/2))^{2/p}} \frac{\chi^2_{1-\alpha,p}}{\epsilon^2}.$$

Using this relationship, one can set a minimum number of effective samples for the respective ϵ and α. In addition, after sampling and retrieving some W effective samples, this value can be compared to the minimum ESS under that α and ϵ to measure precision in the Monte Carlo estimation.

We substantially broaden the application of these methods by use of the delta method and show that the application of the delta method does not impact the ESS as long as the number of estimators remains the same.

Proposition 2.4.1. *Consider the setting under Theorem 2.3.2 where the effective sample size of the Markov chain is* $ESS = m\left(\frac{|\Lambda|}{|\Sigma|}\right)^{1/p}$. *Let* $g_{DM} : \mathbb{R}^p \to \mathbb{R}^{p'}$ *that has a derivative* $\nabla g_{DM}(a)$ *at* $a \in \mathbb{R}^p$ *then* $ESS_{g_{DM}} = m\left(\frac{|\nabla g_{DM}^T \Lambda \nabla g_{DM}|}{|\nabla g_{DM}^T \Sigma \nabla g_{DM}|}\right)^{\frac{1}{p'}}$. *If* $p' = p$, *the effective sample size of* $g_{DM}(\mu_m)$ *is* $ESS_{g_{DM}} = ESS$.

Proof. If $\{X_t\}$ was a series of independent draws from λ then by the CLT,

$$\sqrt{m}(\mu_m - \mu_g) \xrightarrow{d} N_p(0, \Lambda) \quad \text{where } \Lambda = Var_\lambda X_1.$$

But if the draws are correlated then,

$$\sqrt{m}(\mu_m - \mu_g) \xrightarrow{d} N_p(0, \Sigma) \quad \text{where } \Sigma = \Lambda + \sum_{t=1}^{\infty}[Cov_\lambda(X_0, X_t) + Cov_\lambda(X_0, X_t)^T].$$

Then the ESS of the correlated draws is, $\text{ESS} = m\left(\frac{|\Lambda|}{|\Sigma|}\right)^{1/p}$.

For $g_{DM} : \mathbb{R}^p \to \mathbb{R}^{p'}$ that has a derivative $\nabla g_{DM}(a)$ at $a \in \mathbb{R}^p$, by the multivariate delta method, if there was no correlation between the draws,

$$\sqrt{m}(g_{DM}(\mu_m) - g_{DM}(\mu_g)) \xrightarrow{d} N_{p'}(0, \nabla g_{DM}^T \Lambda \nabla g_{DM}).$$

And if there was correlation,

$$\sqrt{m}(g_{DM}(\mu_m) - g_{DM}(\mu_g)) \xrightarrow{d} N_k(0, \nabla g_{DM}^T \Sigma \nabla g_{DM}).$$

Then the ESS after the multivariate delta method is

$$\text{ESS}_{g_{DM}} = m\left(\frac{|\nabla g_{DM}^T \Lambda \nabla g_{DM}|}{|\nabla g_{DM}^T \Sigma \nabla g_{DM}|}\right)^{\frac{1}{p'}}.$$

If $p' = p$, $\nabla g_{DM}, \Lambda, \Sigma \in \mathbb{R}^{p \times p}$, which implies,

$$\text{ESS}_{g_{DM}} = m\left(\frac{|\nabla g_{DM}^T \Lambda \nabla g_{DM}|}{|\nabla g_{DM}^T \Sigma \nabla g_{DM}|}\right)^{\frac{1}{p}} = m\left(\frac{|\nabla g_{DM}^T||\Lambda|\nabla g_{DM}|}{|\nabla g_{DM}^T||\Sigma|\nabla g_{DM}|}\right)^{\frac{1}{p}} = m\left(\frac{|\Lambda|}{|\Sigma|}\right)^{\frac{1}{p}} = \text{ESS}.$$

If $p' \neq p$ then we cannot break apart the determinant since $\nabla g_{DM} \in \mathbb{R}^{p \times p'}$. □

2.5 Two MCMC Sampling Methods

We will consider two random walk-based MCMC methods, a simple random walk (SRW) and a Metropolis-Hastings (MH) algorithm with a simple random walk proposal. MH is constructed to have its invariant distribution as λ, the uniform distribution over nodes. However, SRW has a different invariant distribution, necessitating the use of importance sampling in estimation.

The SRW is described as follows. If the current state is i, then the transition probability of moving to node j is

$$P(i,j)^{SRW} = \begin{cases} \frac{1}{d_i} & \text{if } j \text{ is a neighbor of } i \\ 0 & \text{otherwise.} \end{cases}$$

Diaconis and Stroock (1991) have studied the SRW in great detail on completely connected and well-connected graphs. They developed bounds for the second eigenvalue of this Markov chain on a well-connected, non bipartite, and undirected graph as a function of the graph topology. Their results allow us to bound the convergence rate of the walk and theoretically study its performance on different graph structures. The stationary distribution of the SRW is $\lambda(i) = \frac{d_i}{2n_e}$, which is not the ideal uniform.

Proposition 2.5.1. *The stationary distribution, λ, of a simple random walk on a well-connected, non-bipartite, and undirected graph, $G = (V, E)$ where $n = |V|$ and $n_e = |E|$ is $\lambda(i) = d_i/2n_e$ for all $i \in V$.*

Proof. Let $\lambda \in \mathbb{R}^n$ where $\lambda(i) = d_i/2n_e$. Then,

$$\begin{aligned} \lambda P^{SRW} &= \frac{1}{2n_e} \begin{pmatrix} d_1 & d_2 & \cdots & d_n \end{pmatrix} \begin{pmatrix} \frac{y_{11}}{d_1} & \frac{y_{12}}{d_1} & \cdots & \frac{y_{1n}}{d_1} \\ \frac{y_{21}}{d_2} & \frac{y_{22}}{d_2} & \cdots & \frac{y_{2n}}{d_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{y_{n1}}{d_n} & \frac{y_{n2}}{d_n} & \cdots & \frac{y_{nn}}{d_n} \end{pmatrix} \\ &= \frac{1}{2n_e} \begin{pmatrix} \sum_{i=1}^n y_{i1} & \sum_{i=1}^n y_{i2} & \cdots & \sum_{i=1}^n y_{in} \end{pmatrix} \\ &= \frac{1}{2n_e} \begin{pmatrix} d_1 & d_2 & \cdots & d_n \end{pmatrix} \end{aligned}$$

$$= \lambda$$

Therefore λ is the stationary distribution for the SRW and the probability of being at node i is a function of the node degree and total number of edges in the graph. □

Notice then, the SRW samples come from a more convenient distribution that is not the target but used for estimation, hence the connection with importance sampling.

Gjoka et al. (2011) suggested using a Metropolis-Hastings algorithm with SRW as the proposal distribution. The target distribution is $\lambda(i) = 1/n$, the uniform distribution over the nodes. If j is a neighbor of i the acceptance probability of moving from i to j is,

$$\alpha(i,j) = \min\left(1, \frac{\lambda(j)q(j,i)}{\lambda(i)q(i,j)}\right) = \min\left(1, \frac{\frac{1}{nd_j}}{\frac{1}{nd_i}}\right) = \min\left(1, \frac{d_i}{d_j}\right),$$

so the rejection probability is $1 - \alpha(i,j)$. Then the transition probability of moving from node i to j if j is a neighbor of i is

$$P(i,j)^{MH} = q(i,j)\alpha(i,j) = \frac{1}{d_i}\min\left(1, \frac{d_i}{d_j}\right).$$

Hence the MH transition probability is,

$$P(i,j)^{MH} = \begin{cases} \frac{1}{d_i}\min\left(1, \frac{d_i}{d_j}\right) & \text{if } j \text{ is a neighbor of } i \\ 1 - \sum_{k \neq i} \frac{1}{d_i}\min\left(1, \frac{d_i}{d_k}\right) & \text{if } j = i \\ 0 & \text{otherwise.} \end{cases}$$

The resulting stationary distribution is $\lambda(i) = \frac{1}{n}$ for all $i \in V$. The upfront benefit of the MH sampling algorithm is the output of "ready-to-use" data to compute the network averages. To the best of our knowledge, no bound on the second eigenvalue of the MH walk as a function of the graph topology has been developed.

2.6 Application of Monte Carlo Methods for Network Descriptive Statistics and Inference

We focus on estimating popular network features, these include: mean degree, degree distribution, mean clustering coefficient, and mean of nodal attributes. For a given node v, let d_v be the degree, t_v be the number of triangles, and a categorical attribute, x_v, (e.g. race) having c levels $x(1), x(2), \ldots, x(c)$. We keep these estimators general as one can easily see that the list can be expanded to include more estimators of interest. In terms of the notation from the previous section, we want to estimate μ_g where

$$g(v) = (d_v, \mathbb{I}(d_v = k), 2t_v\mathbb{I}(d_v \geq 2)/d_v(d_v - 1), \mathbb{I}(x_v = x(c)))^T. \tag{2.4}$$

When using MH, estimation proceeds by using μ_m. When using SRW, estimation will proceed using importance sampling with

$$\mu_m^{SRW} = \frac{\sum_{t=0}^{m-1} \left[\frac{g(V_t)}{d_{V_t}}\right]}{\sum_{t=0}^{m-1} \frac{1}{d_{V_t}}}.$$

Other names for this approach include reweighted random walk or respondent driven sampling as MCMC (Avrachenkov et al., 2016; Gjoka et al., 2011; Goel and Salganik, 2009; Salganik and Heckathorn, 2004). To find the form of the CLT we use a transformed version of g.

Namely, let $g^*(v) = 1/d_v(1, \mathbb{I}(d_v = k), 2t_v\mathbb{I}(d_v \geq 2)/d_v(d_v - 1), \mathbb{I}(x_v = x(c)))^T$ so that if

$$\mu_m^* = \frac{1}{m}\sum_{t=0}^{m-1} g^*(V_t),$$

then, by the CLT, we have, as $m \to \infty$,

$$\sqrt{m}(\mu_m^* - \mu_{g^*}) \to \mathrm{N}(0, \Sigma^*).$$

We then apply the delta method with $g_{DM}(a,b,c,d) = (1/a, b/a, c/a, d/a)^T$ so that

$$\nabla g_{DM} = \begin{pmatrix} -1/a^2 & -b/a^2 & -c/a^2 & -d/a^2 \\ 0 & 1/a & -0 & 0 \\ 0 & 0 & 1/a & 0 \\ 0 & 0 & 0 & 1/a \end{pmatrix},$$

to obtain, via the delta method, that, as $m \to \infty$,

$$\sqrt{m}(g_{DM}(\mu_m^*) - g_{DM}(\mu_{g^*})) \to \mathrm{N}\left(0, [\nabla g_{DM}(\mu_{g^*})]^T \Sigma^* [\nabla g_{DM}(\mu_{g^*})]\right)$$

and we can estimate the asymptotic variance with

$$[\nabla g(\mu_m^*)]^T \Sigma_m^* [\nabla g(\mu_m^*)].$$

Again, the goal is to obtain not only estimates of these network properties but also measures on the reliability of those estimates with respect to coverage probabilities and number of samples required for a desired confidence level and relative precision.

We now consider the algorithms and output analysis methods described above as applied to three social networks. We begin with a simple example to illustrate the concepts and progressively move to more complicated, larger networks.

2.7 Examples

To demonstrate the applicability of this work we look into classic cases in the literature: (1) a simulated network based on Ad-Health data (Handcock et al., 2008; Resnick et al., 1997), (2) a college Facebook friendship network (Traud et al., 2008), and (3) the Friendster network to showcase its use on large scale graphs. These three cases allow us to demonstrate the effectiveness of the output analysis methods.

2.7.1 High School Social Network Data

The `faux.magnolia.high` social network is in the `ergm` R package (Handcock et al., 2008; Resnick et al., 1997). It is a simulation of a within-school friendship network representative of those in the southern United States. All edges are undirected and we removed 1022 nodes out of 1461 to ensure a well-connected graph. This resulting social network has 439 nodes (students) and 573 edges (friendships). Other nodal attributes besides structural are grade, race, and sex. Population parameters are in Tables 2.1 and 2.2.

	Min	25%	Median	Mean	75%	Max
Degree	1.00	1.00	2.00	2.61	4.00	8.00
Triples	0.00	0.00	1.00	3.15	6.00	28.00
Triangles	0.00	0.00	0.00	0.90	1.00	10.00
Clustering Coefficient	0.00	0.00	0.00	0.13	0.20	1.00

Table 2.1: Population parameters of well-connected `faux.magnolia.high` social network.

Grade	Mean	SD				
	9.42	1.62				
Sex	Male	Female				
%	42.82	57.18				
Race	White	Black	Asian	Hisp	NatAm	Other
%	79.73	12.07	3.19	3.19	1.37	0.46

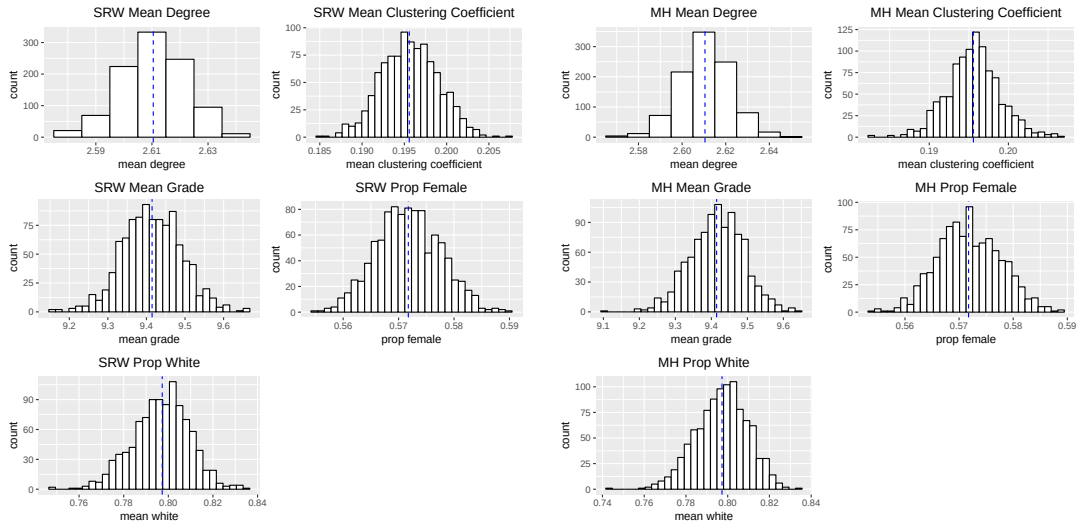
Table 2.2: Other population parameters of well-connected `faux.magnolia.high` social network.

We ran a single chain of both the SRW and MH walks on this network with random starting nodes repeating this 1000 times independently, constructing estimates for the mean degree, mean clustering coefficient, mean grade, proportion of females, and proportion of students who identified as white. The minimum ESS for $p = 5$, $\epsilon = 0.05$, and $\alpha = 0.05$ is 10363. We also constructed the 95% confidence region and used the

corresponding volume to determine the termination time using the relative fixed-volume sequential stopping rule with multivariate batch means with the square root batch size, $\epsilon = 0.05$, and $m^* = 10{,}000$. At this random terminating point we also noted the univariate mean estimates, multivariate effective sample size, and the number of unique nodes visited by the termination step.

Results

The univariate estimates from both the SRW and MH are in Figure 2.5 and Table 2.3.



(a) Mean estimates from the SRW at termination.

(b) Mean estimates from the MH at termination.

Figure 2.5: Mean estimates from SRW and MH on well-connected `faux.magnolia.high` network. Replications = 1000. Blue dashed line indicates population quantity.

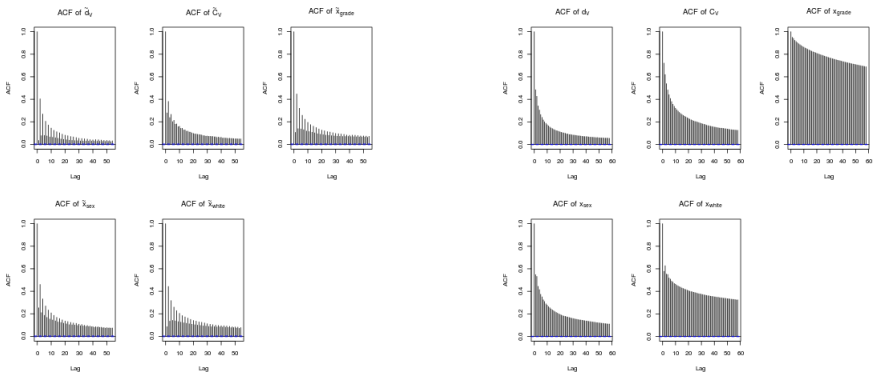
Type	Degree	Clustering coeff	Grade	Prop female	Prop white
Truth	2.6105	0.1956	9.4146	0.5718	0.7973
SRW	2.6106 (0.0004)	0.1956 (0.0001)	9.4145 (0.0026)	0.5716 (0.0157)	0.7969 (0.0127)
MH	2.6103 (0.0004)	0.1956 (0.0001)	9.4158 (0.0024)	0.5719 (0.0157)	0.7973 (0.0127)

Table 2.3: Mean estimates from SRW and MH on the well-connected `faux.magnolia.high` network at termination time. Replications = 1000 and standard errors in parentheses.

All SRW samples terminated on average around 341,000 steps (average computer run time 425 seconds) whereas the MH samples did not achieve the stopping criterion until around 689,115 steps on average (average computer run time 352 seconds). Results are shown in Table 2.4. Since the network is relatively small, all runs of the two sampling methods captured all the nodes in the network. The mean acceptance rate of the MH samples was 0.29. Auto correlation function (ACF) plots for the five estimates from one terminated chain of the SRW and MH are shown in Figure 2.6.

	Termination Step	ESS	Unique Nodes	$T(\epsilon = 0.05)$
SRW	341190 (452.481)	10639.67 (3.113)	439 (0)	0.0497 (0)
MH	689115 (698.090)	10550.03 (2.273)	439 (0)	0.0498 (0)

Table 2.4: Termination time, effective sample size, unique nodes sampled by termination for $\epsilon = 0.05$, and $T(\epsilon = 0.05)$ at termination step on the well-connected `faux.magnolia.high` network. Replications = 1000 and standard errors are indicated in parentheses.



(a) ACF plots from SRW. **(b)** ACF plots from MH.

Figure 2.6: ACF plots from one terminated chain of SRW and MH on `faux.magnolia.high` network.

2.7.2 NYU Facebook Data

The New York University (NYU) Facebook (FB) dataset is a snapshot of anonymized Facebook data from the NYU student population in 2005 (Traud et al., 2008). Nodes are NYU FB users and edges are online friendships. The data was obtained directly from FB and is a complete set of users at NYU at the time. Other nodal attributes in this data are: gender, class year, major, high school, and residence. Some nodes had missing attribute data, so we created a new category for missing data called "Not Reported" (NR). The full NYU FB dataset contains 21,679 nodes (users) and 715,715 undirected edges (online friendships). We only considered the largest well-connected component, NYU WC FB, which has 21,623 users and 715,673 undirected edges. The population parameters of this network are in Table 2.5. We estimated the mean degree, clustering coefficient, proportion of female users, and proportion of users with major = 209.

	Min	25%	Median	Mean	75%	Max
Degree	1.00	21.00	50.00	66.20	93.00	2315.00
Triples	0.00	210.00	1225.00	4666.47	4278.00	2678455.00
Triangles	0.00	39.00	197.00	502.24	598.00	39402.00
Clustering Coefficient	0.00	0.10	0.15	0.19	0.23	1.00

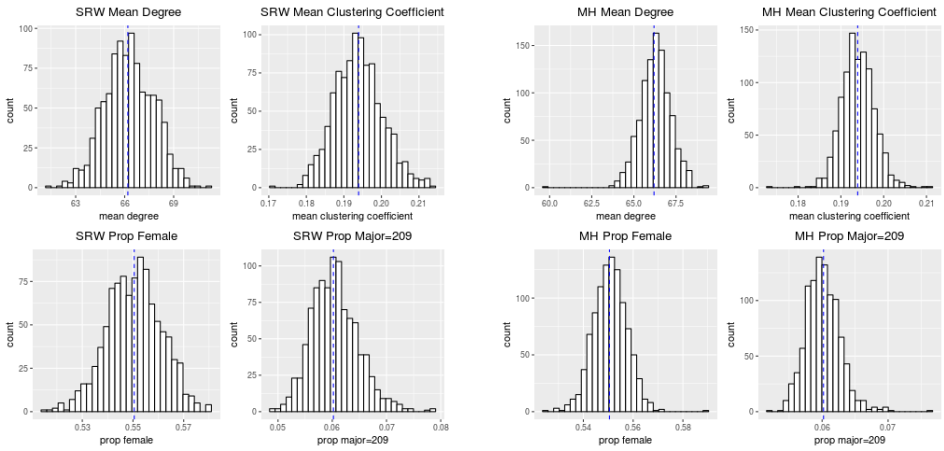
Gender	Female	Male	NR
%	55.05	37.39	7.57
Major	209	Other	NR
%	6.02	77.82	16.16

Table 2.5: Population parameters of well-connected NYU FB social network, NR = Not Reported. $n = 21623$, $n_e = 715673$.

Again we ran a single chain of both the SRW and MH on this network with random starting nodes, repeating this 1000 times independently, constructing the 95% confidence region and determining the termination time with the square root batch size, $\epsilon = 0.05$ and $m^* = 10{,}000$. The minimum ESS for $p = 4, \epsilon = 0.05$, and $\alpha = 0.05$ is 9992.

Results

The univariate network mean estimates are noted in Figure 2.7 and Table 2.6. The mean degree estimate from the SRW and MH on average both slightly overestimate the true mean degree. Otherwise, the estimates from both the SRW and MH algorithms are close to the true means.



(a) Mean estimates from the SRW at termination

(b) Mean estimates from the MH at termination

Figure 2.7: Mean estimates from SRW and MH on NYU WC FB at termination. Replications = 1000. Blue dashed line indicates population quantity.

Type	Degree	Clustering coeff	Prop female	Prop Major=209
Truth	66.1955	0.1939	0.5505	0.0602
SRW	66.2708 (0.04714)	0.1939 (2e-04)	0.5504 (0.01573)	0.0605 (0.00754)
MH	66.2803 (0.02853)	0.194 (0.00012)	0.5508 (0.0157)	0.0601 (0.0075)

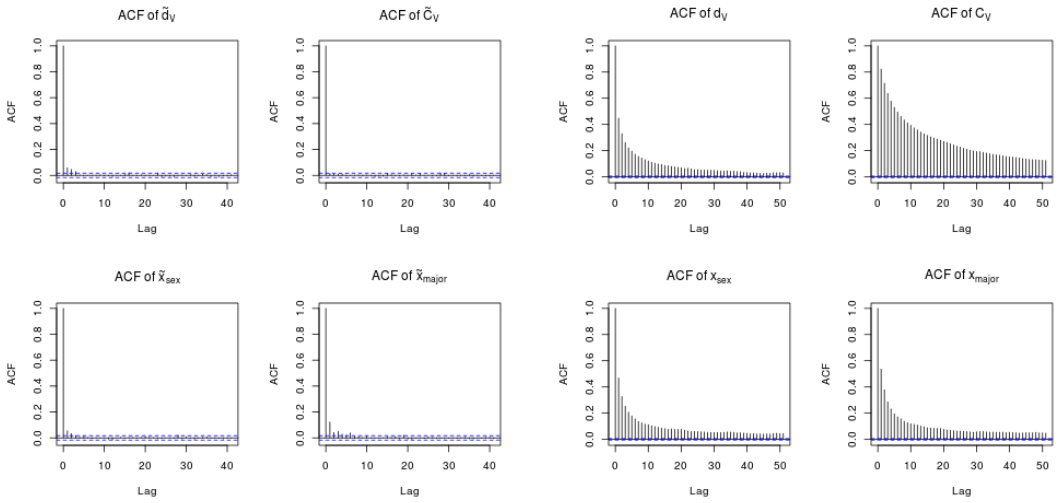
Table 2.6: Mean estimates from SRW and MH on NYC WC FB at termination time. Replications = 1000 and standard errors in parentheses.

	Termination Step	ESS	Coverage Prob	Unique Nodes	$T(\epsilon = 0.05)$
SRW	14676.78 (51.02)	10558.7 (25.36)	0.938 (0.002)	8703.88 (17.55)	0.048 (0.00)
MH	85948.61 (416.40)	6824.317 (11.38)	0.91 (0.003)	16790.81 (19.96)	0.049 (0.00)

Table 2.7: Termination times, effective sample size, coverage probabilities, number of unique nodes sampled by termination time for $\epsilon \leq 0.05$, and $T(\epsilon = 0.05)$ at termination for NYU WC FB. Replications = 1000 and standard errors in parentheses.

All SRW samples terminated on average around 14,700 steps (average computer run time 8.1 seconds) whereas among the MH samples terminated on average by 86,000 steps (average computer run time 30.9 seconds), see Table 2.7. The mean acceptance rate of the MH walks was 0.5621. ACF plots for one chain of both the SRW and MH

are shown in Figure 2.8.



(a) ACF plots from SRW. **(b)** ACF plots from MH.

Figure 2.8: ACF plots from one chain of SRW and MH on NYU WC FB network.

2.7.3 Friendster Data

The Friendster dataset is hosted on the Stanford Large Network Dataset (SNAP) web site (Leskovec and Sosič, 2016). Friendster was an online social gaming and social networking site, where members had user profiles and could link to one another. Friendster also allowed users to form groups which other members could join. The SNAP-hosted Friendster dataset is the largest well-connected component of the induced subgraph of nodes that belonged to at least one group or were connected to other nodes that belonged to at least one group. This social network has 65,608,366 nodes (users) and 1,806,067,135 undirected edges (friendships). There are no other nodal attributes in this data. We estimated the mean degree and mean clustering coefficient.

Implementation

We ran 100 chains of length 100,000 from random starting nodes. To find these random starting nodes we generated random numbers and searched if it existed in the network. If it existed, the sample began at this node, if not we generated another random number until it was accepted. During the sampling procedure we collected the visited nodes id, neighborhood, and calculated its degree. Running all 100 independent chains on five cores, took around 80 minutes for the SRW samples and 116 minutes for the MH samples. After completing the walks, we queried the file again to count the number of triangles for each visited node. Counting triangles is a computationally expensive step, so we only computed triangles on the chains up to length 10,000. Therefore, the multivariate results we present are on shorter chains of length 10,000, but we also present full 100,000 results on the univariate estimate of mean degree.

Shorter chain results

Results are in Figure 2.9 and Tables 2.8 and 2.9. The mean degree estimate from both the SRW and MH is around 55 with more variability in the MH samples and the mean clustering coefficient for both algorithms is around 0.16.

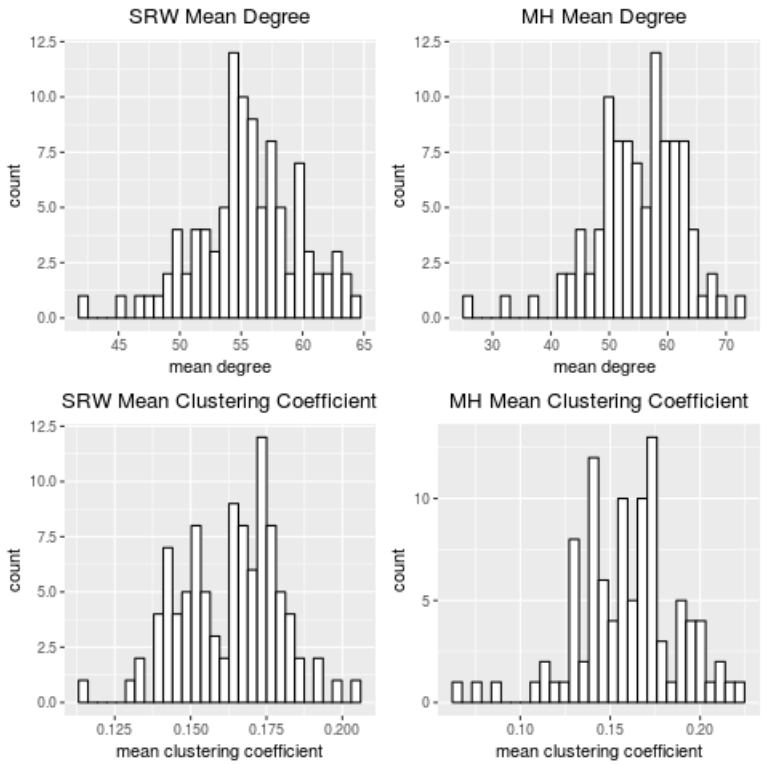


Figure 2.9: Mean estimates from SRW and MH walks on the Friendster network for 1e4 length chains. Replications = 100.

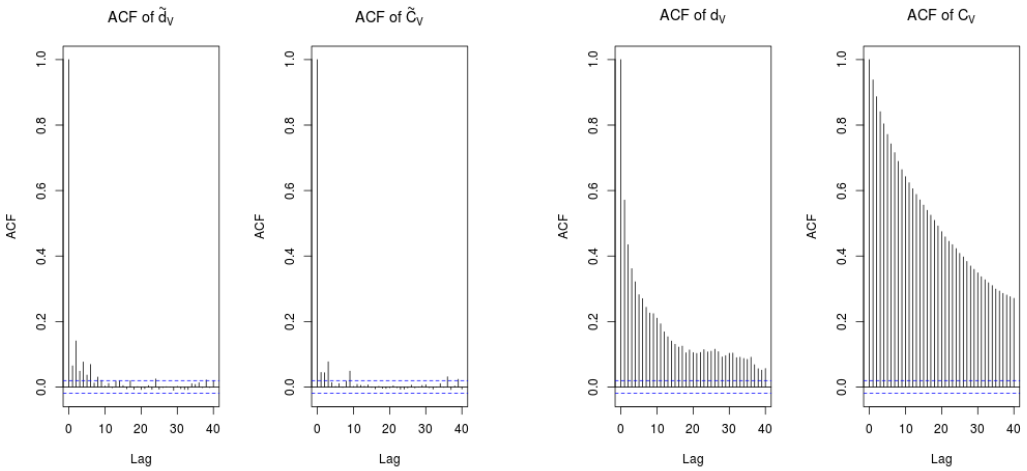
Type	Degree	Clustering coeff
SRW	55.51 (0.414)	0.163 (0.002)
MH	54.97 (0.765)	0.159 (0.009)

Table 2.8: Mean estimates from the SRW and MH on Friendster network with chain length 1e4. Replications = 100 and standard errors in parentheses.

The striking difference between the SRW and MH is in the effective sample size and number of unique nodes captured. The MH walks on average collect only around 25% of the unique nodes that the SRW does. And in the multivariate ESS, the MH on average is less than 20% of the SRW. The mean acceptance rate in the MH walks was 0.2904. The minimum ESS for $p = 2, \epsilon = 0.05$, and $\alpha = 0.05$ is 7530, where none of the simulations achieved the minimum ESS for reliable estimation by 1e4 steps, which implies more samples are needed. ACF plots for one chain are shown in Figure 2.10.

	$T(\epsilon = 0.05)$	ESS	Unique Nodes
SRW	0.058 (0.0004)	3865.95 (212.399)	9797 (2.096)
MH	0.0985 (0.0002)	462.918 (6.467)	2437 (27.023)

Table 2.9: Multivariate: $T_{SD}(\epsilon = 0.05)$, effective sample size, and number of unique nodes sampled by 1e4 steps in Friendster network. Replications = 100 and standard errors in parentheses.



(a) ACF plots from SRW. **(b)** ACF plots from MH.

Figure 2.10: ACF plots from one 1e4 chain of SRW and MH on Friendster network.

Full chain results

If we consider estimating the mean degree of the 1e5 chains, we see the mean degree estimates from the SRW and MH walks are again similar. Likewise, the ESS and number of unique nodes are on starkly different scales (Figure 2.11 and Table 2.10). We use the result from Proposition 2.4.1, with $p = 1$, $g(x) = 1/x$ and the square root batch means estimation to calculate the univariate ESS. The mean acceptance rate of of the MH walks was 0.2905. ACF plots for one chain are shown in Figure 2.12.

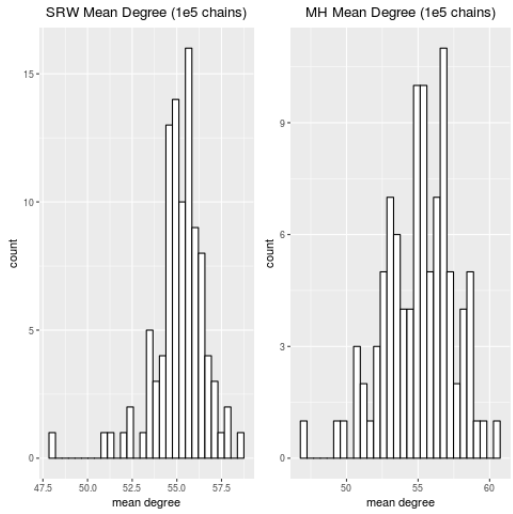
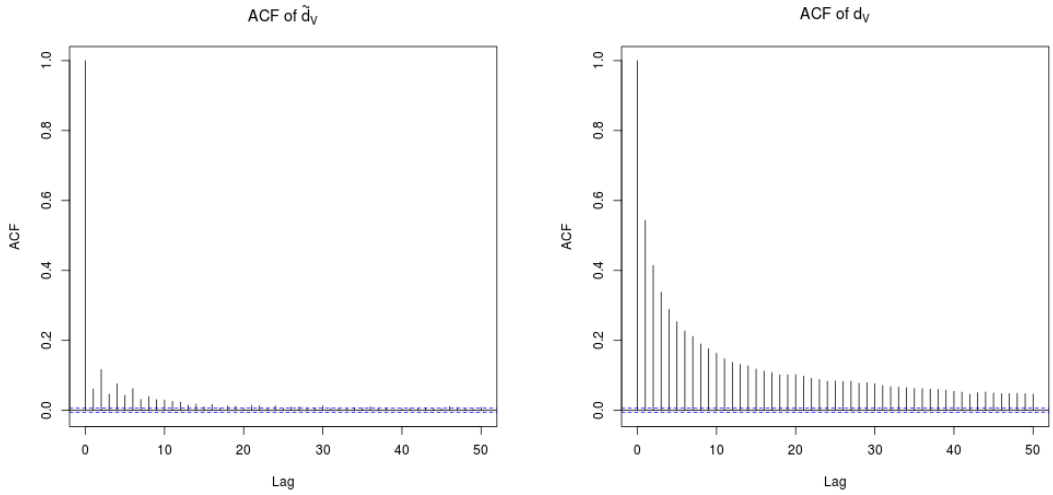


Figure 2.11: Mean estimates from SRW and MH walks on the Friendster network for 1e5 length chains. Replications = 100.

	Degree	ESS	Unique Nodes
SRW	55.15 (0.149)	36229 (1408.53)	97474 (14.124)
MH	55.07 (0.245)	6002 (53.507)	24477 (91.33)

Table 2.10: Univariate: mean degree, effective sample size, and number of unique nodes sample by 1e5 steps for $\epsilon = 0.05$ for Friendster network. Replications = 100 and standard errors in parenthesis.



(a) ACF plots from SRW. **(b)** ACF plots from MH.

Figure 2.12: ACF plots from one 1e5 chain of SRW and MH on Friendster network.

2.7.4 Summary of results

Consistently across all three networks, the SRW was more efficient than the MH, either with respect to the termination time to achieve the stopping criterion or with respect to the effective sample size. Our results confirm what other authors have found in univariate settings (Avrachenkov et al., 2016; Gjoka et al., 2011). In addition, as clearly indicated in the histograms, repeated runs of the algorithms obtained slightly different estimates. However when the minimum effective sample size or stopping criterion was reached, the variation in these estimates was small. This further emphasizes that prior to running the algorithms on any of these networks, a researcher can determine the simulation effort required via the minimum ESS. Once that minimum ESS has been reached, researchers will have an approximately $100(1-\alpha)\%$ confidence with precision ϵ for the p many estimates (as shown in Table 2.11).

p	Conf level	ϵ	Minimum ESS
5	95%	0.05	10363
4	95%	0.05	9992
2	95%	0.05	7530

Table 2.11: Minimum ESS required for p estimated features at a $100(1-\alpha)\%$ confidence level and threshold level ϵ.

2.8 Discussion

The use of MCMC methods on networks without sampling frames to estimate multiple features is well known. However, the error associated with the estimation in the multivariate setting has not been studied closely. We contribute to the literature by further developing multivariate MCMC output analysis methods in the context of network sampling that directly addresses the reliability of the multivariate estimation.

We support existing findings that the MH is less efficient than the SRW in univariate estimation and extend the results to a multivariate setting. We have also extended the MCMC output analysis framework more generally so that it can be applied to other MCMC sampling algorithms. If a researcher plans to use an MCMC method to collect a sample, they can now find the minimum number of effective samples they should collect before they terminate the sampling procedure. Moreover, they have the tools to assess the reliability of the inference they make from that sample.

By using tools like the minimum effective sample size or stopping rules in their work, researchers can have greater confidence in the consistency and reproducibility of their results. This reduces the chance of outlier results or non-reproducible estimates due to insufficient Monte Carlo sample sizes.

There are multiple extensions of this work that could benefit from further research. First, it would be interesting to extend this research to handle edge sampling algorithms to estimate network edge properties. In addition, we focused on binary networks, so generalizing the framework to work on weighted networks that convey relationship strength or weakness would be useful. Another extension is to develop these methods to work on directed networks. The most practically beneficial extension, though, may

be to use these reliable estimation tools, such as minimum effective sample size, in the context of RDS. However, the assumptions required for the output analysis tools are not met in RDS, therefore further work is required to apply the methods we propose.

Chapter 3

Reliability in Density and Log-Likelihood Function Approximation Using IS

3.1 Introduction

In Section 1.6 we introduced IS to approximate density and log-likelihood functions. We did not address, however, the reliability of the estimation procedure. In fact, in the Monte Carlo literature there is little to no work on the quality of functional approximation from IS. We know there is computational uncertainty with the estimates we obtain, so we must address the confidence in the functional approximation. In other words, how reliable is the estimation? We begin assessing the reliability of the IS functional approximation using confidence bands to capture the simulation variability. There are two ways to create such bands: pointwise or simultaneous. We introduce both approaches below, how they differ, and why it is advantageous to use simultaneous bands for functional approximation.

3.1.1 Pointwise vs. Simultaneous Confidence Bands

A $100(1-\alpha)\%$ pointwise confidence band is one such that we create $100(1-\alpha)\%$ confidence intervals around each x at which we estimate $f_\theta(x)$. Then we stitch together the upper and lower bounds respectively to construct a confidence band. For each $x \in \mathcal{X}$, each confidence interval will cover the true $f_\theta(x)$ with probability $1-\alpha$. More formally, define $w(x)$ to be half width for each $x \in \mathcal{X}$ such that,

$$Pr(\hat{f}_\theta(x) - w(x) \leq f_\theta(x) \leq \hat{f}_\theta(x) + w(x)) = 1 - \alpha.$$

As the number of x's we evaluate the function over grows, the proportion of pointwise confidence bands that contain the true function will tend to zero. This is because it increases the chance that at least one interval does not cover the true function. Multiple correction procedures, such as Bonferroni or Scheffé, are sometimes used with pointwise confidence bands to counter this, but are overly conservative when the number of corrections is large.

A $100(1-\alpha)\%$ simultaneous confidence band has simultaneous coverage probability of $1-\alpha$. That is,

$$Pr(\hat{f}_\theta(x) - w(x) \leq f_\theta(x) \leq \hat{f}_\theta(x) + w(x) \text{ for all } x \in \mathcal{X}) = 1 - \alpha.$$

In most cases the simultaneous confidence band is wider than the pointwise with the same coverage probability, and hence has better simultaneous coverage. Consider estimating a density which is a mixture of three Normal densities. We estimate this density over 30 grid points from -4 to 9. Details of the estimation procedure are in Example 1 (Section 3.5.1). We complete 500 independent replications of this estimation procedure using a Monte Carlo sample of size 20,000 from a mixture of two Student's t-distributions to build 90% pointwise and simultaneous confidence bands. The coverage probability, or proportion of confidence bands that contained the entire function over this range, was 0.006 for the pointwise (not adjusting for multiplicity) and 0.890 for the simultaneous. One run of this estimation procedure with the confidence bands is shown in Figure 3.1.

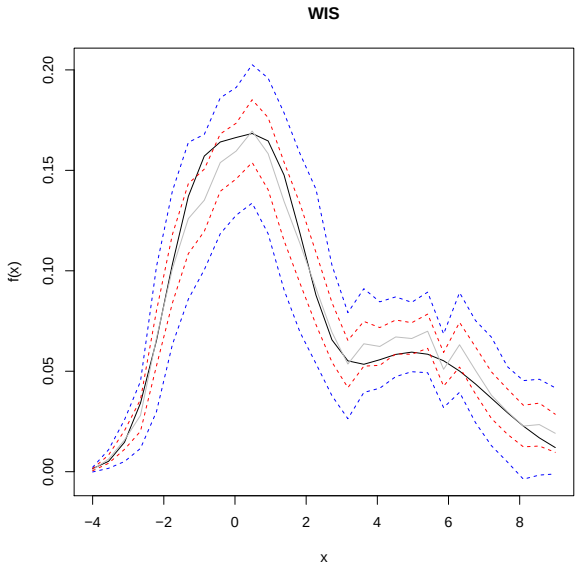


Figure 3.1: Simulation results from one replication of estimating a mixture of Normal densities. True mixture density in solid black line and estimated density in solid grey line. 90% pointwise band in dashed red and simultaneous in dashed blue.

Overview of Remaining Chapter

The outline of the remaining of this chapter is as follows: in Section 2 we introduce the parametric bootstrap procedure that we employ to construct the simultaneous confidence bands. Then in Section 3 we introduce how to construct these confidence bands with SIS. We also propose a bias correction for log-likelihood estimation to improve coverage probabilities thereby improving reliability of the estimation procedure. In Section 4 we do the same as before but in context of WIS, addressing how the bands can also be used for density estimation when WIS is applicable. Finally, in Section 5 we conclude with examples, two density estimation and two log-likelihood estimation examples, highlighting the use of the methods we develop in this chapter to assess the reliability of the estimation procedure. Throughout we assume all Monte Carlo samples are iid.

3.2 Parametric Bootstrap Procedure

The main tool we use to construct simultaneous confidence bands and regions is the parametric bootstrap procedure. Suppose $f_m(y) \in \mathbb{R}^k$ is an estimate of $f(y)$ and $\mathbb{E}[f_m(y)] = f(y)$. Assume that, as $m \to \infty$,

$$\sqrt{m}(f_m(y) - f(y)) \xrightarrow{d} N_k(0, \Sigma) \tag{3.1}$$

where Σ is positive definite. Define Σ_m as the sample covariance matrix, requiring large enough m so that Σ_m is positive definite. Then we can create a $(1-\alpha)$ simultaneous confidence band by simulating from a multivariate Normal centered at $f_m(y)$ with covariance matrix Σ_m/m truncating the tails so that the observations reside within a $(1-\alpha)$ region. To do so we first simulate m many $Z \sim N_k(0, I)$ such that $z^T z < \chi^2_{1-\alpha,k}$ where $\chi^2_{1-\alpha,k}$ represents the $(1-\alpha)$ quantile for the Chi-squared distribution with k degrees of freedom. We then do a Cholesky decomposition of the estimated covariance matrix, $\Sigma_m/m = LL^T$ and use L in the form $U = f_m(y) + Lz$. Then U is from the desired truncated Normal. Finally, we construct the simultaneous confidence bands by using the observed minimum and maximum at each point in y from these m iid sampled U values.

Degras (2017) also suggest using simultaneous confidence bands when modeling functional data. Although their approach uses the same idea of the parametric bootstrap procedure, their fundamental problem is different. They are interested in modeling functional data using data directly from the function of interest. Meanwhile, in the problems we consider, we do not have data directly from the function of interest (e.g., density or mixed effects model). In separate work, Montiel Olea and Plagborg-Møller (2019) also construct simultaneous confidence bands for general non-linear models, like vector autoregressions used in economics. They take a bootstrap or Bayesian approach to construct sup-t confidence bands with finite sample credibility. The plug-in sup-t implementation more closely resembles the parametric approach we take. It uses a delta method argument and draws samples directly from the desired centered multivariate Normal distribution. However, in constructing the simultaneous bands, instead of using

the sample covariance matrix directly as we do, the plug-in sup-t procedure uses empirical quantiles from the draws and pointwise standard errors to construct intervals to be linked together to make the band. Meanwhile, the bootstrap or Bayesian approach Montiel Olea and Plagborg-Møller (2019) advocate draws many y's from a bootstrap distribution or posterior distribution and then applies the function of interest, which in our case we do not assume to know fully, to obtain the empirical quantiles for the simultaneous bands.

3.3 SIS Confidence Bands

In order to use the parametric bootstrap procedure introduced above, we first need to frame the problem in the same way as equation (3.1). Once we set the SIS estimation appropriately, we can use the parametric bootstrap procedure to obtain the simultaneous confidence bands.

We first introduce how to construct these confidence bands for log-likelihood estimation for missing data models. Although, the framework is general enough it can be modified easily for log-likelihood estimation in missing normalizing constant models. Recall with IS, we obtain a functional approximation over the whole set, but for implementation, we choose specific values over which to evaluate the estimated function. The first step we take is to write out the asymptotic distribution of SIS estimator.

3.3.1 For log-likelihood estimation in missing data models

Recall the missing data model setup in Section 1.6.2. Suppose Θ is a compact subset of $\mathbb{R}^{(p+\kappa)}$ and $\theta_j \in \Theta$ for $j = 1, \ldots, k$ where $k \geq 1$. Let U be a $q-$dimensional vector from $\tilde{F}$, define $G_j = f_{\theta_j}(y|U)f_{\theta_j}(U)/\tilde{f}(U)$, and $L_j = L(\theta_j|y)$. Let $U_1, U_2, \ldots, U_m$ be iid samples from $\tilde{F}$ and $\theta = (\theta_1, \ldots, \theta_k)$. Define $\bar{G}_j = m^{-1} \sum_{t=1}^{m} f_{\theta_j}(y|U_t)f_{\theta_j}(U_t)/\tilde{f}(U_t)$, $L_m(\theta) = (\bar{G}_1, \bar{G}_2, \ldots, \bar{G}_k)^T$, and $L(\theta) = (L_1, L_2, \ldots, L_k)^T$ where $\mathbb{E}_{\tilde{F}}[L_m(\theta)] = L(\theta)$ then,

$$\sqrt{m}(L_m(\theta) - L(\theta)) \xrightarrow{d} N_k(0, \Sigma). \tag{3.2}$$

where

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \rho_{12}^{\dagger}\sigma_1\sigma_2 & \cdots & \rho_{1k}^{\dagger}\sigma_1\sigma_k \\ \rho_{12}^{\dagger}\sigma_1\sigma_2 & \sigma_2^2 & \cdots & \rho_{2k}^{\dagger}\sigma_2\sigma_k \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{k1}^{\dagger}\sigma_1\sigma_k & \rho_{k2}^{\dagger}\sigma_2\sigma_k & \cdots & \sigma_k^2 \end{pmatrix},$$

where $\rho_{ij}^{\dagger} = Corr(G_i, G_j)$. We can estimate Σ with the sample covariance matrix, Σ_m, since all draws are iid. This requires m is large enough such that Σ_m is positive definite. At this stage we can use the parametric bootstrap procedure to build confidence bands for likelihood estimation. Since we want to estimate the entire log-likelihood function, we need to apply the log function to the vector of sample means, $L_m(\theta)$, requiring the delta method.

Let $\ell_m(\theta) = \log(L_m(\theta)) \in \mathbb{R}^k$ and $\ell(\theta) = \log(L(\theta))$. Then,

$$\sqrt{m}(\ell_m(\theta) - \mathbb{E}_{\tilde{F}}[\ell_m(\theta)]) \xrightarrow{d} N_k(0, \Sigma_{DM}), \tag{3.3}$$

where $\Sigma_{DM} = \nabla_{DM}^T \Sigma \nabla_{DM}$ where ∇_{DM} is a $k \times k$ diagonal matrix with diagonal $(1/L_1, 1/L_2, \ldots, 1/L_k)$. To estimate Σ_{DM} we can use a function of consistent estimators. Let $\hat{\nabla}_{DM} = diag(1/\bar{G}_1, 1/\bar{G}_2, \ldots, 1/\bar{G}_k)$, then $\hat{\Sigma}_{DM} = \hat{\nabla}_{DM}^T \Sigma_m \hat{\nabla}_{DM}$.

Now using the parametric bootstrap procedure we can construct the $(1-\alpha)$ simultaneous confidence bands by simulating m many $Z \sim N_k(0, I)$ that fall below the $\chi^2_{1-\alpha,k}$ threshold. We then compute $\hat{\Sigma}_{DM} = LL^T$ and subsequently $U = \ell_m(\theta) + Lz$ and link together the observed minimum and maximum at each point in θ from these m iid sampled U values.

For Log-Likelihood Estimation with Unknown Normalizing Constant

Recall we do not use SIS for density estimation, since SIS implies a known-normalizing constant in the density of interest. We can use SIS in MCLA for an unknown normalizing constant though. In this situation, we use SIS to first estimate the normalizing constant. Again for $\theta = (\theta_1, \ldots, \theta_k)$ now consider that $G_j^{(c)} = h_{\theta_j}(U)/\tilde{f}(U), c(\theta_j) = \mathbb{E}_{\tilde{F}} G_j^{(c)}$, $\bar{G}_j^{(c)} = m^{-1}\sum_{t=1}^m h_{\theta_j}(U_t)/\tilde{f}(U_t)$, $c_m(\theta) = (\bar{G}_1^{(c)}, \ldots, \bar{G}_k^{(c)})^T$, and $c(\theta) = (c(\theta_1), \ldots, c(\theta_k))^T$.

Since $c_m(\theta)$ is a vector of iid sample means we have by the CLT that, as $m \to \infty$,

$$\sqrt{m}(c_m(\theta) - c(\theta)) \xrightarrow{d} N_k(0, \Sigma^{(c)})$$

where $\Sigma^{(c)}$ is the corresponding positive definite covariance matrix. Since $\ell_m(\theta_j|y) = \log(h_{\theta_j}(y)) - \log c_m(\theta_j)$ we first apply the log function to the estimated normalizing constant,

$$\sqrt{m}(\log(c_m(\theta)) - \log(c(\theta))) \xrightarrow{d} N_k(0, \Sigma^{(c)}_{DM})$$

where $\Sigma^{(c)}_{DM} = \nabla^T_{c,DM}\Sigma^{(c)}\nabla_{c,DM}$ where $\nabla_{c,DM}$ is a $k \times k$ diagonal matrix with diagonal $(1/c(\theta_1), 1/c(\theta_2), \ldots, 1/c(\theta_k))$. Define $\log h(\theta) = (\log h_{\theta_1}(y), \ldots, \log h_{\theta_k}(y))$ as a constant vector. Then,

$$\sqrt{m}(\underbrace{(\log h(\theta) - \log(c_m(\theta))}_{\ell_m(\theta)} - \underbrace{(\log h(\theta) - \log(c(\theta)))}_{\ell(\theta)}) \xrightarrow{d} N_k(0, \Sigma^{(c)}_{DM})$$

and we can build the confidence bands accordingly. A similar set up is possible for creating bands using the relative likelihood for a fixed ψ.

3.3.2 Underestimation in Missing Data Models

Similar to the univariate case introduced in Section 1.5.1, the CLT only holds if Σ or Σ_{DM} are invertible. That implies a well informed choice of importance distribution such that, as discussed in Section 1.5.1, the variance of the importance weights do not explode with light tailed densities. In the next section we argue that after applying the log function and when the choice of importance distribution is not the ideal, there is a large chance that the SIS estimator underestimates the true log-likelihood function in missing data models. To account for this underestimation, we propose a correction that goes to zero as the Monte Carlo sample size grows to infinity, since as $m \to \infty$ the estimated function will converge to the true function for all θ. Therefore, the correction term must also diminish accordingly.

First we provide a general argument as to why we might be underestimating. Recall $\theta = (\beta, \nu)^T$ and for simplicity we write $f_\theta(y|u) = f_\beta(y|u)$, $f_\theta(u) = f_\nu(u)$, and $f_\theta(y, u) =$

$f_\beta(y|u)f_\nu(u)$.

In SIS, $U_t \overset{iid}{\sim} \tilde{F}$ for $t = 1, \ldots, m$ where,

$$L_m(\theta|y) = \frac{1}{m}\sum_{t=1}^{m} \frac{f_\theta(y, U_t)}{\tilde{f}(U_t)} \quad \text{and} \quad \ell_m(\theta|y) = \log\left(\frac{1}{m}\sum_{t=1}^{m} \frac{f_\theta(y, U_t)}{\tilde{f}(U_t)}\right). \tag{3.4}$$

Since all samples are iid, SIS generates an unbiased estimator,

$$\mathbb{E}_{\tilde{F}}[L_m(\theta|y)] = \frac{1}{m}\sum_{t=1}^{m} \mathbb{E}_{\tilde{F}}\left[\frac{f_\theta(y, U_t)}{\tilde{f}(U_t)}\right] = \mathbb{E}_{\tilde{F}}\left[\frac{f_\theta(y, U)}{\tilde{f}(U)}\right] = \int_{\mathcal{U}} \frac{f_\theta(y, u)}{\tilde{f}(u)}\tilde{f}(u)du = L(\theta|y).$$

But by Jensen's inequality we have that,

$$\mathbb{E}_{\tilde{F}}[\ell_m(\theta|y)] = \mathbb{E}_{\tilde{F}}[\log L_m(\theta|y)] \leq \log \mathbb{E}_{\tilde{F}}[L_m(\theta|y)] = \ell(\theta|y).$$

On average $\ell_m(\theta|y)$ will be less than or equal to $\ell(\theta|y)$ for all θ. But by the continuous mapping theorem since log is a continuous function, as $m \to \infty$, for all θ,

$$L_m(\theta|y) \to L(\theta|y) \text{ almost surely} \quad \Rightarrow \quad \log(L_m(\theta|y)) \to \log(L(\theta|y)) \text{ almost surely.}$$

As $m \to \infty$, on average $\ell_m(\theta|y)$ will approach $\ell(\theta|y)$ from below, suggesting that we may be able to add a correction term to $\ell_m(\theta|y)$ to get a better estimate of the true log likelihood.

In addition, define the importance weights as $w_\theta(u) = f_\theta(u)/\tilde{f}(u)$. Recall the support of $\tilde{f}$ contains f_θ, i.e., $\mathcal{X} \subseteq \tilde{\mathcal{X}}$. Therefore,

$$w_\theta(u) = \begin{cases} (1, \infty) & \text{when } f_\theta(u) > \tilde{f}(u) \\ 1 & \text{when } f_\theta(u) = \tilde{f}(u) \\ (0, 1) & \text{when } f_\theta(u) < \tilde{f}(u) \end{cases}$$

If $\tilde{f}$ is not the ideal density, $\tilde{f}$ would more often oversample areas less important in $f_\theta(u)$ and under sample areas of more importance in $f_\theta(u)$. In particular, if the likelihood is

unimodal or has one unique maximum, the maximum at $\hat{\theta}$ will be underestimated, i.e., there will be more small $w_{\hat{\theta}}(u)$'s than large. Therefore, SIS is likely to underestimate the likelihood by the argument above and further exacerbate the underestimation in the log likelihood because of Jensen's inequality.

As discussed in Chapter 1, bias in functional estimation has been noted before. Hence we calculate the bias so that we can correct the estimate accordingly.

Concentration Analysis

In KDE, one measure of estimation precision is to do a concentration analysis to ascertain how close the estimated density is to the true density. We will use a similar technique for the likelihood.

Treating Likelihood as a Density

Suppose we treat the likelihood as a density. The two conditions required are: 1) likelihood is always non-negative and 2) that it integrates to 1 (i.e. has a finite integral),

$$\int L(\theta|y)d\theta = c < \infty.$$

Condition 1 is met since $L(\theta|y) = \int f_\theta(y|u)f(u|\theta)du$ is an integral of positive terms and therefore will always be positive. Condition 2 is not necessarily met. Consider the following toy example.

Example 3.3.1. *$Y_i \overset{ind}{\sim} Poisson(u_i)$ and $u_i \overset{iid}{\sim} Exp(\lambda)$ for $i = 1, \ldots, n$. Then,*

$$\int_{\mathbb{R}^+} L(\lambda|y)d\lambda = \int_0^\infty \lambda^n (1+\lambda)^{-(n+\sum_{i=1}^n y_i)} d\lambda = \infty.$$

Therefore, this likelihood cannot be made into a density over the whole support.

Let Θ be a compact set of the parameter space. Then

$$\int_{\Theta} L(\theta|y)d\theta = c' < \infty.$$

and $p(\theta) = L(\theta|y)/c'$ would be a density over Θ. So if we could draw θ's from a distribution resembling $p(\theta)$, then we could use KDE to estimate $p(\theta) \propto L(\theta|y)$. De Valpine (2004) suggested a Monte Carlo Kernel Likelihood method, where they suggest first drawing θ's from some prior density. As this approach already exists in the literature we instead try to find a probabilistic upper bound on the estimate and use that to shape a correction.

Finding a Probabilistic Upper Bound for Likelihood Estimation

Consider the probability that the estimated likelihood, or log-likelihood, overestimates the true value for any $\theta \in \Theta$. We drop the conditional on y for brevity. We first state a more general result.

Proposition 3.3.1. *Consider any function $f(\theta) > 0$ for all $\theta \in \Theta$ and a Monte Carlo estimator $f_m(\theta)$ such that $\mathbb{E}[f_m(\theta)] = f(\theta)$ for all θ. If $a > 0$, then,*

$$Pr(f_m(\theta) > f(\theta) + a) < \left(1 + \frac{a}{f(\theta)}\right)^{-1}$$

and

$$Pr(\log[f_m(\theta)] > \log[f(\theta)] + a) < e^{-a}$$

for any $\theta \in \Theta$.

Proof. Let $a > 0$. Then,

$$\begin{aligned} Pr(f_m(\theta) > f(\theta) + a) &< \frac{\mathbb{E}_{\tilde{F}}[f_m(\theta)]}{f(\theta) + a} \quad \text{by Markov's inequality} \\ &= \frac{f(\theta)}{f(\theta) + a} \end{aligned}$$

$$= \left(\frac{f(\theta) + a}{f(\theta)} \right)^{-1}$$
$$= \left(1 + \frac{a}{f(\theta)} \right)^{-1}.$$

And,

$$\begin{aligned} Pr(\log[f_m(\theta)] > \log[f(\theta)] + a) &= Pr(e^{\log[f_m(\theta)]} > e^{\log[f(\theta)]+a}) \\ &= Pr(f_m(\theta) > f(\theta)e^a) \\ &< \frac{\mathbb{E}_{\tilde{F}} f_m(\theta)}{f(\theta)e^a} \quad \text{by Markov's inequality} \\ &= \frac{f(\theta)}{f(\theta)e^a} \\ &= e^{-a} \end{aligned}$$

□

Corollary 3.3.1.1. *If $a > 0$, then the probability that that the SIS likelihood estimator overestimates the true likelihood is,*

$$Pr(L_m(\theta) > L(\theta) + a) < \left(1 + \frac{a}{L(\theta)} \right)^{-1}$$

and the probability that that the SIS log-likelihood estimator overestimates the true log-likelihood is

$$Pr(\log[L_m(\theta)] > \log[L(\theta)] + a) < e^{-a}$$

for any $\theta \in \Theta$.

Proof. Set $f(\theta) = L(\theta)$ and $f_m(\theta) = L_m(\theta)$ as defined in equation 3.4, then the results hold by direct application of Proposition 3.3.1. □

Thus it is unlikely for both the SIS estimator to overestimate the likelihood and log-likelihood function. In fact, the closer $L(\theta)$ is to zero, the less likely it is that $L_m(\theta)$ overestimates by some constant a, since as $L(\theta) \to 0 \Rightarrow (1 + a/L(\theta))^{-1} \to 0$. Notice, $L(\theta)$ is typically small when the dimension of the data and random effects is large. It

is in these situations of high and crossed dimensions when the likelihood function is difficult to compute in the first place. Moving forward we will not concern ourselves with the chance of overestimation. We consider finding a probabilistic bound on the difference between the estimate and the truth. Recall Bernstein's Inequality (Bennett, 1962; Bernstein, 1927).

Theorem 3.3.2. *(Bernstein's Inequality) Suppose $X_1, X_2, \ldots, X_n$ are iid with mean μ, $Var(X_i) \leq \sigma^2 < \infty$, and $|X_i| \leq M < \infty$, then for $\bar{X} = n^{-1} \sum_{i=1}^{n} X_i$ and for every $\epsilon > 0$,*

$$P(|\bar{X} - \mu| > \epsilon) \leq 2exp\left\{ \frac{-n\epsilon^2}{2\sigma^2 + 2M\epsilon/3} \right\}. \tag{3.5}$$

Notice this inequality only deals with a sample mean of iid bounded random variables. We use Bernstein's Inequality to prove the next result.

Theorem 3.3.3. *Assume for each y and θ, $f_\theta(y|u) \in [0,1]$ and define $L_m(\theta)$ as in equation (3.4). Let $M = \sup_\theta w_\theta(U) < \infty$ and $\tau^2 = \sup_\theta Var_{\tilde{F}} w_\theta(U) < \infty$. Then for every $\epsilon > 0$,*

$$Pr(L(\theta) - L_m(\theta) > \epsilon) \leq 2 \exp\left\{ \frac{-m\epsilon^2}{2\tau^2 + 2M\epsilon/3} \right\}. \tag{3.6}$$

For a fixed Monte Carlo sample size m, to achieve an upper bound probability of $\alpha \in (0,1)$ set

$$\epsilon = \frac{\frac{2}{3}M\log(\alpha/2) - \sqrt{\frac{4}{9}M^2(\log(\alpha/2))^2 - 8m\tau^2\log(\alpha/2)}}{-2m}. \tag{3.7}$$

Proof. See Appendix A.1. □

Corollary 3.3.3.1. *The minimum Monte Carlo sample size needed for confidence level $(1-\alpha)$ and precision ϵ is,*

$$m = -\epsilon^{-2}\log(\alpha/2)(2\tau^2 + M\epsilon/3). \tag{3.8}$$

Proof. See Appendix A.1.1. □

To estimate the probability we can use the following statistics from the m samples,

$$M_m = \sup_{\theta\in\Theta} \max_{1\le t\le m} w_\theta(u_t) \quad \text{and} \quad \tau_m^2 = \sup_{\theta\in\Theta} \widehat{Var}(w_\theta(u_t)).$$

We saw in simulations that ϵ may be so large relative to $L_m(\theta)$ that it destroys the curvature in the likelihood estimation. So instead of adding the correction to the likelihood, can we add the correction to the log likelihood estimate. Recall from Corollary 3.3.1.1, for any $a > 0$, the $Pr(L_m(\theta) > L(\theta) + a) < (1 + a/L(\theta))^{-1}$ and $Pr(\ell_m(\theta) > \ell(\theta) + a) < e^{-a}$.

Corollary 3.3.3.2. *If $L_m(\theta) \le L(\theta)$ and if either (1), $1 \le L_m(\theta)$ or (2.i.) $L_m(\theta) < 1$ and*

$$1 - \alpha \le Pr(\ell(\theta) - \ell_m(\theta) \le \log(\epsilon + 1))$$

then,

$$Pr(\ell(\theta) - \ell_m(\theta) > \log(\epsilon + 1)) \le 2\exp\left\{\frac{-m\epsilon^2}{2\tau^2 + 2M\epsilon/3}\right\}. \tag{3.9}$$

Otherwise if (2.ii.) $L_m(\theta) < 1$ and

$$1 - \alpha > Pr(\ell(\theta) - \ell_m(\theta) \le \log(\epsilon + 1))$$

then,

$$Pr(\ell(\theta) - \ell_m(\theta) > \log(\epsilon + 1)) \ge 2\exp\left\{\frac{-m\epsilon^2}{2\tau^2 + 2M\epsilon/3}\right\}. \tag{3.10}$$

Proof. See Appendix A.1.2 □

Case (2.ii.) occurs when the scale of $L(\theta)$ and $L_m(\theta)$ is quite small, in which $L(\theta)/L_m(\theta) >> L(\theta) - L_m(\theta)$. This is more likely in situations when the dimension of the random effects is large. This further motivates a need for a correction when the scale of the likelihood is small.

There may also be underestimation after applying the log function.

Theorem 3.3.4. *Define $\bar{G}_j, L_j$, and ℓ_j as in Section 3.3.1. Using the importance density $\tilde{f}$, the SIS estimator of the log likelihood, $\ell_m(\theta_j)$, has bias $\mathbb{E}_{\tilde{F}}[\ell_m(\theta_j)] - \ell(\theta_j) = \sigma_j^2/(2mL_j^2) + O(m^{-2})$.*

Proof. See Appendix A.2. □

SIS log-likelihood correction

Combining the results from Theorem 3.3.3 and Theorem 3.3.4, the estimated log-likelihood for any $\theta_j \in \Theta$ is

$$\ell_m^*(\theta_j|y) = \log\left(\frac{1}{m}\sum_{t=1}^{m}\frac{f_{\theta_j}(y|u)f_{\theta_j}(u)}{\tilde{f}(u)}\right) + \log(\epsilon+1) + \frac{\hat{\sigma}_j^2}{2mL_{j,m}^2}. \tag{3.11}$$

3.4 WIS Confidence Bands

The WIS estimator, even in the simple univariate estimation case, is the ratio of two sample means. This implies we can use the same approach we took to build the bands for SIS functional estimation. We will first need to write out the asymptotic distribution of the sample means and then apply the delta method to find the distribution of the ratio.

Again we first introduce constructing the WIS confidence bands with respect to log-likelihood estimation in the missing data models. The framework is general enough so that it can be easily used for density estimation as well.

3.4.1 In log-likelihood estimation for missing data models

Let $U \sim \tilde{F}$ (or the unnormalized distribution) and define $G_j^* = f_{\theta_j}(y|U)h_{\theta_j}(U)/\tilde{h}(U), W_j^* = h_{\theta_j}(U)/\tilde{h}(U)$, and $c_j = c(\theta_j)/\tilde{c}$.

Then,

$$\sqrt{m}\left(\begin{pmatrix}\bar{G}_1^* \\ \bar{W}_1^* \\ \bar{G}_2^* \\ \bar{W}_2^* \\ \vdots \\ \bar{G}_k^* \\ \bar{W}_k^*\end{pmatrix} - \begin{pmatrix}c_1 L_1 \\ c_1 \\ c_2 L_2 \\ c_2 \\ \vdots \\ c_k L_k \\ c_k\end{pmatrix}\right) \xrightarrow{d} N_{2k}\left(0, \Sigma^*\right) \tag{3.12}$$

where $Var_{\tilde{F}}(W_j) = \tau_j^2, Corr_{\tilde{F}}(G_j, G_{j'}) = \rho_{jj'}^{\dagger}, Corr_{\tilde{F}}(W_j, W_{j'}) = \rho_{jj'}^{\ddagger}$, and $Corr_{\tilde{F}}(G_j, W_{j'}) = \rho_{jj'}$. When $j = j', \rho_{jj}^{\dagger} = \rho_{jj}^{\ddagger} = 1$. Then,

$$\Sigma^* = \begin{pmatrix} c_1^2\sigma_1^2 & c_1^2\rho_{11}\sigma_1\tau_1 & c_1c_2\rho_{12}^{\dagger}\sigma_1\sigma_2 & \cdots & c_1c_k\rho_{1k}\sigma_1\tau_k \\ c_1^2\rho_{11}\sigma_1\tau_1 & c_1^2\tau_1^2 & c_1c_2\rho_{12}\tau_1\sigma_2 & \cdots & c_1c_k\rho_{1k}^{\ddagger}\tau_1\tau_k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_kc_1\rho_{k1}^{\dagger}\sigma_k\sigma_1 & c_kc_1\rho_{k1}\sigma_k\tau_1 & c_kc_2\rho_{k2}^{\dagger}\sigma_k\sigma_2 & \cdots & c_k^2\rho_{kk}\sigma_k\tau_k \\ c_kc_1\rho_{k1}\tau_k\sigma_1 & c_kc_1\rho^{\ddagger}\tau_k\tau_1 & c_kc_2\rho_{k2}\tau_k\sigma_2 & \cdots & c_k^2\tau_k^2 \end{pmatrix}_{2k\times 2k}.$$

We apply the delta method where $g_{DM} : \mathbb{R}^{2k} \to \mathbb{R}^k$ such that,

$$g_{DM}\begin{pmatrix}\bar{G}_1 \\ \bar{W}_1 \\ \bar{G}_2 \\ \bar{W}_2 \\ \vdots \\ \bar{G}_k \\ \bar{W}_k\end{pmatrix} = \begin{pmatrix}\bar{G}_1^*/\bar{W}_1^* \\ \bar{G}_2^*/\bar{W}_2^* \\ \vdots \\ \bar{G}_k^*/\bar{W}_k^*\end{pmatrix} = \begin{pmatrix}\bar{G}_1/\bar{W}_1 \\ \bar{G}_2/\bar{W}_2 \\ \vdots \\ \bar{G}_k/\bar{W}_k\end{pmatrix} = \hat{L}_m(\theta).$$

The Jacobian matrix evaluated on the mean vector $(c_1L_1, c_1, c_2L_2, c_2, \ldots, c_kL_k, c_k)$ is,

$$\nabla g = \begin{pmatrix} \frac{1}{c_1} & 0 & 0 & \cdots & 0 & 0 \\ -\frac{c_1L_1}{c_1^2} & 0 & 0 & \cdots & 0 & 0 \\ 0 & \frac{1}{c_2} & 0 & \cdots & 0 & 0 \\ 0 & -\frac{c_2L_2}{c_2^2} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & \frac{1}{c_k} \\ 0 & 0 & 0 & \cdots & 0 & -\frac{c_kL_k}{c_k^2} \end{pmatrix}_{2k\times k}.$$

Put together,

$$\sqrt{m}(\hat{L}_m(\theta) - \mathbb{E}_{\tilde{F}}[\hat{L}_m(\theta)]) \xrightarrow{d} N_k(0, \nabla g^T \Sigma^* \nabla g). \tag{3.13}$$

Again, we can use consistent estimators in place to get an estimate of the covariance matrix. Just as in the SIS case, for log-likelihood estimation we take one further delta method step to apply to the log function to $\hat{L}_m(\theta)$ (we could combine both steps into one delta method argument, but for the sake of clarity we break it up into two steps).

This implies,

$$\sqrt{m}(\hat{\ell}_m(\theta) - \mathbb{E}_{\tilde{F}}[\hat{\ell}_m(\theta)]) \xrightarrow{d} N_k(0, \nabla_{DM}^T(\nabla g^T \Sigma^* \nabla g)\nabla_{DM}). \tag{3.14}$$

Then we can estimate $\Sigma_{DM}^* = \nabla_{DM}^T(\nabla g^T \Sigma^* \nabla g)\nabla_{DM}$ with a function of consistent estimators, i.e., $\hat{\Sigma}_{DM}^* = \hat{\nabla}_{DM}^T(\hat{\nabla} g^T \Sigma_m^* \hat{\nabla} g)\hat{\nabla}_{DM}$. Then to construct the confidence bands we can once again draw m multivariate standard Normal samples that fall below the Chi-squared threshold. Decompose $\hat{\Sigma}_{DM}^* = LL^T$, compute $U = \hat{\ell}_m(\theta) + Lz$, and link together the observed minimum and maximum at each point in θ from these m iid sampled U values.

In Density Estimation

Since we do not apply the the log function to the ratio of sample means for density estimation using WIS, we do not need to take the last delta method step. This means

we can use the parametric bootstrap at stage (3.13).

3.4.2 Underestimation in Missing Data Models

By similar arguments as in Section 3.3.2 we can build a correction for the WIS log-likelihood estimator.

Theorem 3.4.1. *Assume for each y and θ, $f_\theta(y|u) \in [0,1]$ and define $\hat{L}_m(\theta)$ as the exponent of equation (1.33). In addition, define $w^*_\theta(U) = h_\theta(U)/\tilde{h}(U)$ and $\tilde{w}^*_\theta(U) = w^*_\theta(U)c(\theta)/\tilde{c}$. Let $\tilde{M} = \sup_\theta \tilde{w}^*_\theta(U) < \infty$ and $\tilde{\tau}^2 = \sup_\theta Var_{\tilde{F}}\tilde{w}^*_\theta(U) < \infty$. Then for every $\tilde{\epsilon} > 0$,*

$$Pr(L(\theta) - \hat{L}_m(\theta) > \log(\tilde{\epsilon}+1)) \leq 2\exp\left\{\frac{-m\tilde{\epsilon}^2}{2\tilde{\tau}^2 + 2\tilde{M}\tilde{\epsilon}/3}\right\}. \tag{3.15}$$

For a finite Monte Carlo sample size m, to achieve an upper bound probability of $\alpha \in (0,1)$, $\tilde{\epsilon}$ should be the same as in equation 3.7 replacing M and τ^2 with $\tilde{M}$ and $\tilde{\tau}^2$ respectively.

Proof. See Appendix A.3. □

Corollary 3.4.1.1. *The minimum Monte Carlo sample size needed for confidence level $(1-\alpha)$ and precision $\tilde{\epsilon}$ is,*

$$m = -\tilde{\epsilon}^{-2}\log(\alpha/2)(2\tilde{\tau}^2 + \tilde{M}\tilde{\epsilon}/3). \tag{3.16}$$

Proof. See Appendix A.3.1. □

Again to estimate the probability, in place of $\tilde{M}$ and $\tilde{\tau}^2$ we can use the following statistics from the m samples where $\bar{w}^*_\theta = \frac{1}{m}\sum_{t=1}^{m} w^*_\theta(u_t)$,

$$\tilde{M}_m = \sup_{\theta\in\Theta}\max_{1\leq t\leq m}\frac{w^*_\theta(u_t)}{\bar{w}^*_\theta} \quad \text{and} \quad \tilde{\tau}^2_m = \sup_{\theta\in\Theta}\hat{Var}\left(\frac{w^*_\theta(u_t)}{\bar{w}^*_\theta}\right).$$

Likewise, we can add the bias correction term to the log-likelihood instead.

Corollary 3.4.1.2. *If $\hat{L}_m(\theta) \leq L(\theta)$ and if either (1), $1 \leq \hat{L}_m(\theta)$ or (2.i.) $\hat{L}_m(\theta) < 1$ and*

$$1 - \alpha \leq Pr(\ell(\theta) - \hat{\ell}_m(\theta) \leq \log(\tilde{\epsilon} + 1))$$

then,

$$Pr(\ell(\theta) - \hat{\ell}_m(\theta) > \log[\tilde{\epsilon} + 1]) \leq 2 \exp\left\{ \frac{-m\tilde{\epsilon}^2}{2\tilde{\tau}^2 + 2\tilde{M}\tilde{\epsilon}/3} \right\}. \tag{3.17}$$

Otherwise if (2.ii.) $\hat{L}_m(\theta) < 1$ and

$$1 - \alpha > Pr(\ell(\theta) - \hat{\ell}_m(\theta) \leq \log[\tilde{\epsilon} + 1])$$

then,

$$Pr(\ell(\theta) - \hat{\ell}_m(\theta) > \log[\tilde{\epsilon} + 1]) \geq 2 \exp\left\{ \frac{-m\tilde{\epsilon}^2}{2\tilde{\tau}^2 + 2\tilde{M}\tilde{\epsilon}/3} \right\}, \tag{3.18}$$

which motivates a minimum correction.

Proof. See Appendix A.3.2 □

Similar to the SIS log-likelihood approximation, there may be underestimation from applying the log function. However, recall the WIS estimator is also biased.

Theorem 3.4.2. *Define $\bar{G}_j, \bar{W}_j, L_j$, and ℓ_j as in Section 3.4.1. Using the importance density $\tilde{f}$, the WIS estimator of the log-likelihood, $\hat{\ell}_m(\theta_j)$, has bias $\mathbb{E}_{\tilde{F}}[\hat{\ell}_m(\theta_j)] - \ell(\theta_j) = \frac{1}{2m}\left(\tau_j^2 - \frac{\sigma_j^2}{L_j^2}\right) + O(m^{-2})$.*

Proof. See Appendix A.4. □

WIS log-likelihood correction

Combining the results from Theorem 3.4.1 and Theorem 3.4.2 the estimated log-likelihood for any $\theta_j \in \Theta$ is,

$$\hat{\ell}^*_m(\theta_j|y) = \log\left(\frac{\frac{1}{m}\sum_{t=1}^{m}\frac{f_\beta(y|U_t)f_\nu(U_t)}{\tilde{f}(U_t)}}{\frac{1}{m}\sum_{t=1}^{m}\frac{f_\nu(U_t)}{\tilde{f}(U_t)}}\right) + \log(\tilde{\epsilon}+1) - \frac{1}{2m}\left(\hat{\tau}_j^2 - \frac{\hat{\sigma}_j^2}{\hat{L}_{j,m}^2}\right). \tag{3.19}$$

Taking into Account the Dimension of Random Effects and Bandwidth

Other than τ and $\tilde{\tau}$ being a function of a $q-$dimensional vector, the dimension of the random effects never comes into play in equations 3.11 or 3.19. In addition, except in the introduction to the parametric bootstrap method, we have not addressed the number and spacing of points at which we are evaluating the function over. In other words, the grid points on which to evaluate the approximated function.

Through simulations we have seen that as the number of independent random effects, q, grows, so does the severity of underestimation. This result also connect to corollaries 3.3.3.2 and 3.4.1.2. When both y or u are in high dimension, the likelihood is typically on a small scale. Then the ratio between $L(\theta)$ and $L_m(\theta)$ or $\hat{L}_m(\theta)$ might be much larger than the difference ($L(\theta) - L_m(\theta)$ or $L(\theta) - \hat{L}_m(\theta)$). Intuitively this makes sense since all draws are independent, we are oversampling areas not as important to $f_\theta(u)$ more often and taking their product and hence getting smaller weights. We also expect that as we approximate the likelihood function on a finer grid (smaller bandwidth), we should capture the likelihood function over that set more closely.

We consider adjusting ϵ and $\tilde{\epsilon}$ for a finite m. Let $\theta \in \mathbb{R}$ and suppose we are estimating the log-likelihood function over the compact set $[a, b]$. Consider a grid of θ values, $a \leq \theta_1 < \theta_2 < \cdots < \theta_k \leq b$ and the goal is to estimate $\ell_j = \ell(\theta_j|y)$ for $j = 1, \ldots, k$. Define the bandwidth as $h_{bw} = (b-a)/k$ and

$$\epsilon^* = q\sqrt{h_{bw}}\epsilon = q\sqrt{\frac{b-a}{k}}\frac{\frac{2}{3}M\log(\alpha/2) - \sqrt{\frac{4}{9}M^2(\log(\alpha/2))^2 - 8m\tau^2\log(\alpha/2)}}{-2m}.$$

We multiply by q since we saw the greater the number of random effects, the more underestimation. And we chose the square root of bandwidth as motivated by rates of convergence in KDE. Notice that as either $m \to \infty$ or $k \to \infty, \epsilon^* \to 0$.

Another way to adjust is to shift $\log(\epsilon + 1)$ such that we add on $q\sqrt{h_{bw}}\log(\epsilon + 1)$. Similarly, as $m \to \infty$ or $k \to \infty$ the correction goes to 0. We explore both corrections in the simulations and explore the same type of modifications for $\tilde{\epsilon}$.

Wasserman (2006) also briefly discuss the uncertainty introduced by the bandwidth parameter in their estimation and consider doing a type of Bonferroni correction to improve coverage. However, as discussed in Section 3.1.1, such corrections are overly conservative when the number of corrections is large hence why we are constructing simultaneous confidence bands to avoid this type of multiplicity correction.

3.5 Examples

To highlight the construction and use of simultaneous confidence bands for density and log-likelihood functional approximation we present four examples below. Two density and two log-likelihood estimation examples.

In Example 1 we estimate a known mixture density using IS density estimation with histograms, where we study the coverage probability of these bands. Whereas, in Example 2, we use Rubin-Importance Sampling to estimate two unknown posterior densities from a two-parameter model. In the log-likelihood functional approximation examples, we estimate a known log-likelihood function in Example 3 and unknown in Example 4. In the latter two examples we also apply the corrections we presented in this chapter to improve the coverage probabilities of the simultaneous confidence bands. All the examples present output analysis results of the IS functional approximation to study the Monte Carlo estimation reliability.

3.5.1 Example 1: Normal Mixture Density Estimation

Suppose we want to estimate the density of a mixture of three Normal densities. Let $N(\mu, \sigma^2)$ denote a univariate Normal density with mean μ and variance σ^2. Then X comes from

$$m_X = \frac{1}{3}(N(1,1) + N(5,5) + N(-1,1)). \tag{3.20}$$

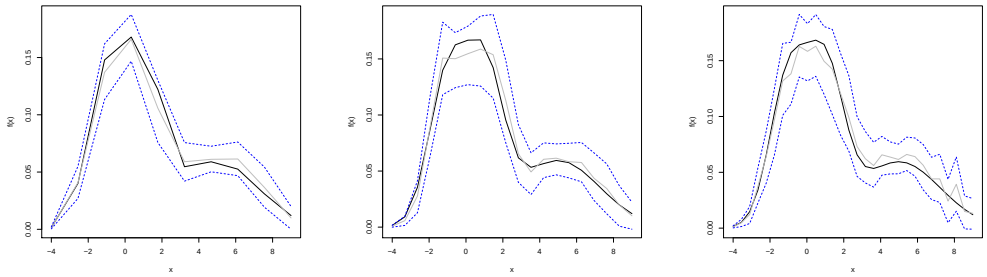
Suppose we do not know the normalizing constants of these densities nor do we know the proportion each mixture contributes. To estimate the true density instead we can first use the histogram like approach as detailed in Section 1.6.1.

Consider $\tilde{f}$ to be a mixture of two Student's t-densities. The first distribution is centered at zero with two degrees of freedom, the second density is shifted with a mean of 5 also with two degrees of freedom. Then using various Monte Carlo sample sizes we estimate the density across k equally spaced grid points from -4 to 9. The value of ω was chosen to be the space between any two points divided by $m^{(1/4)}$, this was guided by the optimal bandwidth when using the Gaussian kernel and performed well in simulations.

Repeating this 500 times independently, we obtained the following results as shown in Table 3.1. One run of the simulation is shown in Figure 3.2 for $m = 1e4, 2e4$, and $4e4$ when $k = 10, 20$, and 30 respectively. The coverage probability is the proportion of times across the 500 replicates that the simultaneous confidence band captured the entire true function over the range from -4 to 9.

m	k	ω	Cov Prob
1e4	10	0.144	0.884 (0.014)
1e4	20	0.068	0.882 (0.014)
2e4	20	0.058	0.904 (0.013)
2e4	30	0.038	0.890 (0.014)
4e4	30	0.032	0.892 (0.013)
4e4	40	0.026	0.890 (0.014)
5e4	80	0.011	0.844 (0.016)

Table 3.1: Coverage probabilities of 90% simultaneous confidence bands over 500 independent replications using the IS histogram approach to estimate a density that is a mixture of three Normal densities. Standard errors are in parentheses. m = Monte Carlo sample size, k = number of grid points equally spaced from -4 to 9.



(a) $m = 1e4, k = 10$ **(b)** $m = 2e4, k = 20$ **(c)** $m = 4e4, k = 30$

Figure 3.2: Estimating a mixture of Normal densities with the IS histogram approach using different Monte Carlo sample sizes and number of grid points. Solid black line indicates true density value, solid gray line is estimated density, and dashed blue lines indicate the 90% simultaneous confidence band.

3.5.2 Example 2: Marginal Posterior Density Estimation

Tanner and Wong (1987) first introduced a one-parameter posterior density estimation problem in the context of a genetic linkage model. In this model, animals were multinomially distributed into one of four groups with certain cell probabilities determined by some unknown parameter θ. Given some observed data, y, the task was to estimate the posterior probability of θ. Gelfand and Smith (1990) expanded this genetic linkage model into a two-parameter estimation problem. Now suppose $Y = (Y_1, \ldots, Y_5)$ and

$$Y|\theta, \eta \sim mult(n, a_1\theta + b_1, a_2\theta + b_2, a_3\eta + b_3, a_4\eta + b_4, c(1 - \theta - \eta)),$$

where $a_i, b_i \geq 0$ and are known. In addition, $0 < c < 1 - \sum_{i=1}^{4} b_i = a_1 + a_2 = a_3 + a_4 < 1$. Then the remaining unknown parameters are θ and η where $\theta, \eta \geq 0$, and $\theta + \eta \leq 1$. A natural choice of a prior for θ and η is the Dirichlet$(\alpha_1, \alpha_2, \alpha_3)$. Again, the goal is to estimate the marginal posterior distributions θ and η from the joint $p(y|\theta, \eta)\pi(\theta, \eta)$. However, obtaining the desired marginals is not straightforward and involves the calculation of a two-dimensional numerical integral since,

$$p(\theta|y) = \frac{\int p(y|\theta, \eta)\pi(\theta, \eta)d\eta}{\int\int p(y|\theta, \eta)\pi(\theta, \eta)d\eta d\theta}.$$

Instead, as first proposed in Tanner and Wong (1987), we take a data augmentation step to reformat the problem into a split cell multinomial, $X = (X_1, \ldots, X_9)$ where,

$$X \sim mult(n, a_1\theta, b_1, a_2\theta, b_2, a_3\eta, b_3, a_4\eta, b_4, c(1-\theta-\eta)).$$

Since the Dirichlet is a conjugate prior of the multinomial and marginals of a Dirichlet are Betas,

$$\begin{aligned}
(\theta, \eta)|X &\sim Dirichlet(X_1 + X_3 + \alpha_1, X_5 + X_7 + \alpha_2, X_9 + \alpha_3), \\
\theta|X &\sim Beta(X_1 + X_3 + \alpha_1, X_5 + X_7 + X_9 + \alpha_2 + \alpha_3), \\
\eta|X &\sim Beta(X_5 + X_7 + \alpha_2, X_1 + X_3 + X_9 + \alpha_1 + \alpha_3), \\
\theta|(X, \eta) &\sim \theta|X \text{ scaled to } (0, 1-\eta) \text{ and}, \\
\eta|(X, \theta) &\sim \eta|X \text{ scaled to } (0, 1-\theta).
\end{aligned}$$

Let $Y_1 = X_1 + X_2$, $Y_2 = X_3 + X_4$, $Y_3 = X_5 + X_6$, $Y_4 = X_7 + X_8$, $Y_5 = X_9$, and define $Z = (X_1, X_3, X_5, X_7)$. Specifying (Y, Z) is equivalent to having X. Moreover,

$$Z|(Y, \theta, \eta) \sim \prod_{i=1,3,5,7} Bin\left(Y_i, \frac{a_i\theta}{a_i\theta + b_i}\right),$$

which is easy to sample from. Gelfand and Smith (1990) use this setting to compare the performance of a Gibbs sampler, a Substitution sampler, and the Rubin Importance-Sampling algorithm. We focus on the results from the Rubin Importance-Sampling algorithm. We also create simultaneous confidence bands for the posterior marginal density estimates.

Recall in the Rubin Importance-Sampling algorithm, we have the form of the desired conditional distribution needed for the Rao-Blackwellisation step. We just need to obtain a sample from the joint distribution which requires WIS.

Suppose we observe $Y = (Y_1, \ldots, Y_5) = (14, 1, 1, 1, 5)$ as a sample from the multinomial distribution $mult(22, \frac{1}{4}\theta + \frac{1}{8}, \frac{1}{4}\theta, \frac{1}{4}\eta, \frac{1}{4}\eta + \frac{3}{8}, \frac{1}{2}(1-\theta-\eta))$ and $\pi(\theta, \eta) \sim Dirichlet(1, 1, 1)$. This implies from the general context above: $a_1 = a_2 = a_3 = a_4 = \frac{1}{4}$, $b_1 = \frac{1}{8}, b_2 = b_3 = 0, b_4 = \frac{3}{8}, c = \frac{1}{2}$, and $\alpha_1 = \alpha_2 = \alpha_3 = 1$. Since $b_2 = b_3 = 0$, we can split Y into a 7-cell

multinomial $X = (X_1, \ldots, X_7)$ such that,

$$X|(\theta,\eta) \sim mult\left(22, \frac{1}{4}\theta, \frac{1}{8}, \frac{1}{4}\theta, \frac{1}{4}\eta, \frac{1}{4}\eta, \frac{3}{8}, \frac{1}{2}(1-\theta-\eta)\right)$$

where $Y_1 = X_1 + X_2, Y_2 = X_3, Y_3 = X_4, Y_4 = X_5 + X_6, Y_5 = X_7$, and $Z = (X_1, X_5)$. Then as before,

$$\begin{aligned}
(\theta,\eta)|X &\sim Dirichlet(X_1 + X_3 + 1, X_4 + X_5 + 1, X_7 + 1),\\
\theta|X &\sim Beta(X_1 + X_3 + 1, X_4 + X_5 + X_7 + 2),\\
\eta|X &\sim Beta(X_4 + X_5 + 1, X_1 + X_3 + X_7 + 2),\\
\theta|(X,\eta) &\sim \theta|X \text{ scaled to } (0, 1-\eta) \text{ and},\\
\eta|(X,\theta) &\sim \eta|X \text{ scaled to } (0, 1-\theta).
\end{aligned}$$

If we obtain samples from a distribution resembling $f(X,\theta,\eta) = f(Y,Z,\theta,\eta)$ we can use the ratio estimator in equation (1.18) to obtain estimates of the posterior marginals.

More specifically, for $t = 1, \ldots, m$ draw,

$$Z_t \sim Z|Y = Bin\left(Y_1, \frac{1}{2}\right) Bin\left(Y_4, \frac{1}{2}\right) \tag{3.21}$$

$$\eta_t \sim \eta|(Z_t, Y) \tag{3.22}$$

$$\theta_t \sim \theta|(\eta_t, Z_t, Y) \tag{3.23}$$

thus getting a sample from (X_t, θ_t, η_t). In this case the only importance distribution we need to specify is $Z|Y$. Then the importance weights are,

$$w_t = w(X_t, \theta_t, \eta_t) = \frac{p(Y, Z_t|\theta_t, \eta_t)\pi(\theta_t, \eta_t)}{f(\theta_t|\eta_t, Z_t, Y)f(\eta_t|Z_t, Y)\tilde{p}(Z_t|Y)}. \tag{3.24}$$

We choose $\tilde{p}(Z_t|Y)$ as the product of the two Binomials as originally used in Gelfand

and Smith (1990). Then the estimated marginals are,

$$\hat{p}_m(\theta|y) = \frac{\frac{1}{m}\sum_{t=1}^{m} f(\theta|Z_t, Y, \eta_t)w_t}{\frac{1}{m}\sum_{t=1}^{m} w_t} \quad \text{and} \quad \hat{p}_m(\eta|y) = \frac{\frac{1}{m}\sum_{t=1}^{m} f(\eta|Z_t, Y, \theta_t)w_t}{\frac{1}{m}\sum_{t=1}^{m} w_t}. \tag{3.25}$$

We estimate the posterior marginal distributions over the grids $\theta = (0.01, 0.1, 0.2, \ldots, 0.9, 0.99)$ and $\eta = (0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, .5, 0.9)$ with different Monte Carlo sample sizes. Using the estimated density, we calculate the area under the curve to estimate the cumulative probabilities at those grid points. Results are in Tables 3.2, 3.3, 3.4, and 3.5. One run of the simulation when $m = 1000$ is shown in Figure 3.3. Each of the 500 replications was run independently for the different Monte Carlo sample sizes.

θ	$m = 1e3$	$m = 1e4$	$m = 1e5$	$m = 1e6$
0.01	0.001 (0, 0.002)	0.001 (0, 0.002)	0.001 (0, 0.002)	0.001 (0, 0.001)
0.10	0.031 (0, 0.063)	0.03 (0.015, 0.046)	0.031 (0.024, 0.037)	0.031 (0.028, 0.033)
0.20	0.223 (0.139, 0.306)	0.222 (0.19, 0.254)	0.222 (0.211, 0.233)	0.222 (0.218, 0.226)
0.30	0.821 (0.695, 0.945)	0.82 (0.775, 0.865)	0.82 (0.805, 0.836)	0.82 (0.815, 0.825)
0.40	1.881 (1.758, 2.003)	1.881 (1.837, 1.926)	1.882 (1.866, 1.897)	1.882 (1.877, 1.887)
0.50	2.8 (2.715, 2.885)	2.801 (2.769, 2.832)	2.801 (2.79, 2.812)	2.801 (2.797, 2.804)
0.60	2.617 (2.476, 2.759)	2.618 (2.566, 2.669)	2.618 (2.6, 2.635)	2.618 (2.612, 2.623)
0.70	1.35 (1.217, 1.483)	1.35 (1.301, 1.398)	1.349 (1.333, 1.366)	1.349 (1.344, 1.355)
0.80	0.271 (0.231, 0.312)	0.271 (0.256, 0.286)	0.271 (0.266, 0.276)	0.271 (0.269, 0.273)
0.90	0.007 (0.005, 0.008)	0.007 (0.006, 0.007)	0.007 (0.006, 0.007)	0.007 (0.007, 0.007)
0.99	0 (0, 0)	0 (0, 0)	0 (0, 0)	0 (0, 0)

Table 3.2: Average $\hat{p}_m(\theta|y)$ at specific values, averaged over 500 independent replications at different Monte Carlo sample sizes. Average minimum and maximum values of simultaneous confidence band at grid point in parentheses.

θ	$m = 1e3$	$m = 1e4$	$m = 1e5$	$m = 1e6$
0.01	0.00	0.00	0.00	0.00
0.10	0.00	0.00	0.00	0.00
0.20	0.01	0.01	0.01	0.01
0.30	0.07	0.07	0.07	0.07
0.40	0.20	0.20	0.20	0.20
0.50	0.43	0.43	0.43	0.43
0.60	0.71	0.71	0.71	0.71
0.70	0.91	0.91	0.91	0.91
0.80	0.99	0.99	0.99	0.99
0.90	1.00	1.00	1.00	1.00
0.99	1.00	1.00	1.00	1.00

Table 3.3: Average estimated cumulative probability at given θ using $\hat{p}_m(\theta|y)$ over 500 independent replications at different Monte Carlo sample sizes.

η	$m = 1e3$	$m = 1e4$	$m = 1e5$	$m = 1e6$
0.01	1.958 (1.861, 2.055)	1.958 (1.923, 1.993)	1.958 (1.946, 1.97)	1.958 (1.954, 1.961)
0.05	5.553 (5.373, 5.732)	5.553 (5.488, 5.617)	5.553 (5.53, 5.575)	5.552 (5.545, 5.56)
0.10	5.325 (5.261, 5.388)	5.325 (5.302, 5.348)	5.325 (5.317, 5.333)	5.325 (5.322, 5.327)
0.15	3.711 (3.666, 3.755)	3.711 (3.695, 3.727)	3.711 (3.706, 3.717)	3.711 (3.709, 3.713)
0.20	2.222 (2.149, 2.294)	2.222 (2.196, 2.248)	2.222 (2.213, 2.231)	2.222 (2.219, 2.225)
0.25	1.202 (1.135, 1.269)	1.202 (1.178, 1.226)	1.202 (1.194, 1.21)	1.202 (1.199, 1.205)
0.30	0.599 (0.55, 0.649)	0.599 (0.581, 0.617)	0.599 (0.593, 0.605)	0.599 (0.597, 0.601)
0.50	0.018 (0.013, 0.023)	0.018 (0.016, 0.02)	0.018 (0.017, 0.019)	0.018 (0.018, 0.018)
0.90	0 (0, 0)	0 (0, 0)	0 (0, 0)	0 (0, 0)

Table 3.4: Average $\hat{p}_m(\eta|y)$ at specific values, averaged over 500 independent replications at different Monte Carlo sample sizes. Average minimum and maximum values of simultaneous confidence band at grid point in parentheses.

η	$m = 1e3$	$m = 1e4$	$m = 1e5$	$m = 1e6$
0.01	0.00	0.00	0.00	0.00
0.05	0.15	0.15	0.15	0.15
0.10	0.42	0.42	0.42	0.42
0.15	0.65	0.65	0.65	0.65
0.20	0.80	0.80	0.80	0.80
0.25	0.88	0.88	0.88	0.88
0.30	0.93	0.93	0.93	0.93
0.50	0.99	0.99	0.99	0.99
0.90	0.99	0.99	0.99	0.99

Table 3.5: Average estimated cumulative probability at given η using $\hat{p}_m(\eta|y)$ over 500 independent replications at different Monte Carlo sample sizes.

For both the marginal posterior estimates of θ and η we see that over the range of the Monte Carlo sample sizes, the general shape of the estimated density remains similar, hence same estimated cumulative probability values. However, the estimation precision improves as evident in the width of the confidence band.

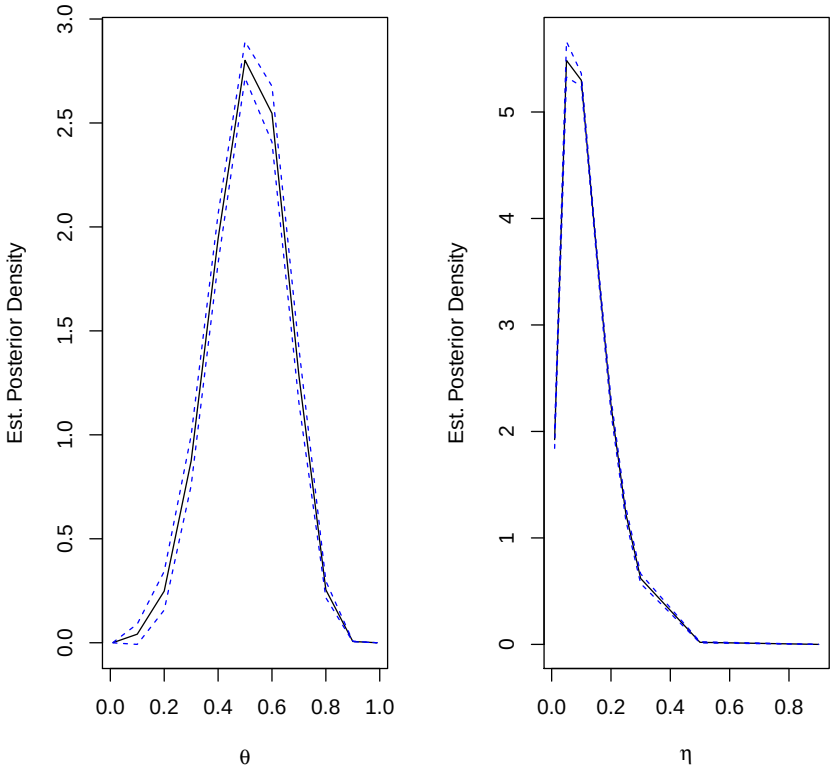


Figure 3.3: Estimated posterior marginals for θ and η when $m = 1000$ for one replication. Estimated density in black, 90% simultaneous confidence band in dashed blue.

Gelfand and Smith (1990) note that the estimation via Rubin's importance sampling is poor and highly variable and mention it is also highly sensitive to the choice of importance distribution. However, it is worth mentioning, their Monte Carlo samples sizes were 40 and 200 respectively.

3.5.3 Example 3: Poisson-Exponential Model

Consider a hierarchal model where

$$Y_i \overset{ind}{\sim} Poisson(u_i)$$

$$u_i \overset{iid}{\sim} Exp(\theta)$$

for $i = 1, \ldots, n$. The goal is to estimate $\ell(\theta|y)$ where

$$\begin{aligned} L(\theta|y) = \int f(y|u) f_\theta(u) du &= \prod_{i=1}^{n} \int_0^\infty \frac{e^{-u_i} u_i^{y_i}}{y_i!} \theta e^{-\theta u_i} du_i \\ &= \prod_{i=1}^{n} \frac{\theta}{y_i!} \int_0^\infty \underbrace{u_i^{y_i} e^{-u_i(1+\theta)}}_{\propto Gamma(y_i+1, 1+\theta)} du_i \\ &= \theta^n \prod_{i=1}^{n} \frac{1}{y_i!} (1+\theta)^{(y_i+1)} (y_i!) \\ &= \theta^n (1+\theta)^{-(n+\sum_{i=1}^n y_i)}. \end{aligned}$$

Hence,

$$\ell(\theta|y) = n \log(\theta) - \left(n + \sum_{i=1}^{n} y_i \right) \log(1+\theta). \tag{3.26}$$

We are mainly interested in the mode of this likelihood (where the true MLE occurs at $1/\bar{y}$). So we take an iterative approach as motivated by Geyer and Thompson (1992)'s suggestion to find the MC-MLE. For the importance distribution, we use a mixture of three easy to sample from distributions that broadly cover $\mathbb{R}^+$.

1. $\tilde{F}_1 = \text{Exponential}(\lambda)$
2. $\tilde{F}_2 = \text{Gamma}(\alpha, \beta)$
3. $\tilde{F}_3 = \text{Normal}_+(\mu, \sigma^2)$

such that,

$$\tilde{f}(u) = \frac{1}{3}(\tilde{f}_1(u|\lambda) + \tilde{f}_2(u|\alpha, \beta) + \tilde{f}_3(u|\mu, \sigma^2)).$$

Where for each draw, an $n-$dimensional vector is sampled from either $\tilde{F}_1, \tilde{F}_2, or \tilde{F}_3$.

It is easy to maximize equation 3.26 to obtain the MLE. Therefore, we set the initial values of $\lambda, \alpha, \beta, \mu$, and σ such that the means of the mixture distributions are approximately around the MLE.

Additionally, as discussed in Section 1.5.1, as the dimension of the sample grows, the variability in the weights needs to become smaller. Therefore, we set the initial values for the importance distribution as:

- $\lambda_1 = 1/\bar{y}$
- $\alpha_1 = 1/(\beta_1 \bar{y})$, $\beta_1 = n^{1/3}$
- $\mu_1 = \lceil (1/\bar{y}) \rceil, \sigma_1 = n^{1/3}$

For each interation, we ran the procedure for m steps and checked across all samples from the iteration what the maximum $w_\theta(u_t)$ for each θ was. We then identified

$$\tilde{\theta} = \arg\min_{\theta} |\max_{t} w_\theta(u_t) - 1|.$$

If the maximum weight for any θ is close to 1, then $\tilde{f}$ is close to f_θ at that point. So distributions centered at $\tilde{\theta}$ may be better. Then keeping λ_1 fixed, we update:

- $\alpha_1 = \tilde{\theta}/\beta_1$, $\beta_1 = n^{1/3}$
- $\mu_2 = \tilde{\theta}, \sigma_2 = n^{1/3}$

At this point we also reduce the window over which we are estimating the likelihood function by half its width, centered at the MC-MLE, and increase m by 10^4. We do this process one more time, for a total of three iterations. Only on the third iteration after the samples are drawn and estimates made do we apply the different bias corrections. Notice, in this example $n = q =$ dimension of the random effects. We consider four cases:

Correction	SIS	WIS
`correct0`	No correction	No correction
`correct1`	$\log(\epsilon + 1) + \frac{\hat{\sigma}_j^2}{2mL_{j,m}^2}$	$\log(\tilde{\epsilon} + 1) - \frac{1}{2m}\left(\hat{\tau}_j^2 - \frac{\hat{\sigma}_j^2}{\hat{L}_{j,m}^2}\right)$
`correct2`	$\log(\epsilon^* + 1) + \frac{\hat{\sigma}_j^2}{2mL_{j,m}^2}$	$\log(\tilde{\epsilon}^* + 1) - \frac{1}{2m}\left(\hat{\tau}_j^2 - \frac{\hat{\sigma}_j^2}{\hat{L}_{j,m}^2}\right)$
`correct3`	$n\sqrt{h_{bw}}\log(\epsilon + 1) + \frac{\hat{\sigma}_j^2}{2mL_{j,m}^2}$	$n\sqrt{h_{bw}}\log(\tilde{\epsilon} + 1) - \frac{1}{2m}\left(\hat{\tau}_j^2 - \frac{\hat{\sigma}_j^2}{\hat{L}_{j,m}^2}\right)$

Table 3.6: Correction types for log-likelihood estimation at θ_j.

`correct0` makes no adjustment to the estimate, `correct1` is motivated by theory, and `correct2` and `correct3` are empirically motivated. The amount of upward correction increases with each correction type. Thus, there is a potential to overestimate the true function after applying the correction.

We simulate data for $n = 15, 25, 50$ and estimate the log-likelihood function over 10 grid points. Below are the simultaneous coverage probabilities at each stage. Recall no corrections were applied in iterative stages one and two. We constructed 90% confidence bands. Results are in Tables 3.7 through 3.13. Since the grid points over which we evaluated the estimated function changed with each replication, the average grid point values by the third iteration are in the caption of each table. One run of the simulation when $n = 50$ and $m = 1e5$ is shown in Figure 3.4.

Table 3.7: $n = 15, m = 1e3$

Iteration	Type	CovProb	MLE
1	SIS_correct0	0.038 (0.009)	0.761 (0.009)
1	WIS_correct0	0.004 (0.003)	0.769 (0.003)
2	SIS_correct0	0.734 (0.02)	0.804 (0.02)
2	WIS_correct0	0.826 (0.017)	0.803 (0.017)
3	SIS_correct0	0.9 (0.013)	0.793 (0.013)
3	SIS_correct1	0.944 (0.01)	0.794 (0.01)
3	SIS_correct2	0.93 (0.011)	0.794 (0.011)
3	SIS_correct3	0.916 (0.012)	0.794 (0.012)
3	WIS_correct0	0.908 (0.013)	0.794 (0.013)
3	WIS_correct1	0.93 (0.011)	0.799 (0.011)
3	WIS_correct2	0.908 (0.013)	0.799 (0.013)
3	WIS_correct3	0.9 (0.013)	0.799 (0.013)

True $\hat{\theta} = 0.789$. Average θ values evaluated on by third iteration: $\theta = (0.51, 0.6, 0.69, 0.78, 0.87, 0.96, 1.05, 1.139, 1.229, 1.319)$ over 500 replications.

Table 3.8: $n = 15, m = 1e4$

Iteration	Type	CovProb	MLE
1	SIS_correct0	0.256 (0.02)	0.686 (0.02)
1	WIS_correct0	0.008 (0.004)	0.69 (0.004)
2	SIS_correct0	0.804 (0.018)	0.808 (0.018)
2	WIS_correct0	0.882 (0.014)	0.809 (0.014)
3	SIS_correct0	0.922 (0.012)	0.789 (0.012)
3	SIS_correct1	0.956 (0.009)	0.789 (0.009)
3	SIS_correct2	0.972 (0.007)	0.789 (0.007)
3	SIS_correct3	0.962 (0.009)	0.789 (0.009)
3	WIS_correct0	0.922 (0.012)	0.789 (0.012)
3	WIS_correct1	0.954 (0.009)	0.793 (0.009)
3	WIS_correct2	0.958 (0.009)	0.793 (0.009)
3	WIS_correct3	0.95 (0.01)	0.793 (0.01)

True $\hat{\theta} = 0.789$. Average θ values evaluated on by third iteration: $\theta = (0.508, 0.598, 0.688, 0.779, 0.869, 0.959, 1.05, 1.14, 1.23, 1.321)$ over 500 replications.

Table 3.9: $n = 25, m = 1e4$

Iteration	Type	CovProb	MLE
1	SIS_correct0	0.018 (0.006)	1.222 (0.006)
1	WIS_correct0	0.008 (0.004)	1.22 (0.004)
2	SIS_correct0	0.436 (0.022)	1.164 (0.022)
2	WIS_correct0	0.526 (0.022)	1.165 (0.022)
3	SIS_correct0	0.664 (0.021)	1.163 (0.021)
3	SIS_correct1	0.732 (0.02)	1.164 (0.02)
3	SIS_correct2	0.88 (0.015)	1.164 (0.015)
3	SIS_correct3	0.838 (0.016)	1.164 (0.016)
3	WIS_correct0	0.672 (0.021)	1.163 (0.021)
3	WIS_correct1	0.614 (0.022)	1.166 (0.022)
3	WIS_correct2	0.754 (0.019)	1.166 (0.019)
3	WIS_correct3	0.712 (0.02)	1.166 (0.02)

True $\hat{\theta} = 1.136$. Average θ values evaluated on by third iteration: $\theta = (0.697, 0.806, 0.916, 1.025, 1.135, 1.244, 1.354, 1.463, 1.573, 1.682)$ over 500 replications.

Table 3.10: $n = 25, m = 5e4$

Iteration	Type	CovProb	MLE
1	SIS_correct0	0.076 (0.012)	1.192 (0.012)
1	WIS_correct0	0 (0)	1.192 (0)
2	SIS_correct0	0.562 (0.022)	1.156 (0.022)
2	WIS_correct0	0.628 (0.022)	1.158 (0.022)
3	SIS_correct0	0.79 (0.018)	1.146 (0.018)
3	SIS_correct1	0.814 (0.017)	1.149 (0.017)
3	SIS_correct2	0.906 (0.013)	1.149 (0.013)
3	SIS_correct3	0.892 (0.014)	1.149 (0.014)
3	WIS_correct0	0.786 (0.018)	1.145 (0.018)
3	WIS_correct1	0.736 (0.02)	1.153 (0.02)
3	WIS_correct2	0.826 (0.017)	1.153 (0.017)
3	WIS_correct3	0.82 (0.017)	1.153 (0.017)

True $\hat{\theta} = 1.136$. Average θ values evaluated on by third iteration: $\theta = (0.686, 0.796, 0.906, 1.016, 1.126, 1.236, 1.346, 1.456, 1.566, 1.676)$ over 500 replications.

Table 3.11: $n = 50, m = 1e5$

Iteration	Type	CovProb	MLE
1	SIS_correct0	0 (0)	0.965 (0)
1	WIS_correct0	0 (0)	0.963 (0)
2	SIS_correct0	0 (0)	0.917 (0)
2	WIS_correct0	0 (0)	0.917 (0)
3	SIS_correct0	0.004 (0.003)	0.907 (0.003)
3	SIS_correct1	0.018 (0.006)	0.906 (0.006)
3	SIS_correct2	0.068 (0.011)	0.906 (0.011)
3	SIS_correct3	0.196 (0.018)	0.906 (0.018)
3	WIS_correct0	0.006 (0.003)	0.906 (0.003)
3	WIS_correct1	0.008 (0.004)	0.908 (0.004)
3	WIS_correct2	0.054 (0.01)	0.908 (0.01)
3	WIS_correct3	0.272 (0.02)	0.908 (0.02)

True $\hat{\theta} = 0.893$. Average θ values evaluated on by third iteration: $\theta = (0.512, 0.613, 0.713, 0.814, 0.915, 1.015, 1.116, 1.217, 1.317, 1.418)$ over 500 replications.

Table 3.12: $n = 50, m = 5e5$

Iteration	Type	CovProb	MLE
1	SIS_correct0	0 (0)	0.95 (0)
1	WIS_correct0	0 (0)	0.949 (0)
2	SIS_correct0	0.002 (0.002)	0.929 (0.002)
2	WIS_correct0	0 (0)	0.931 (0)
3	SIS_correct0	0.03 (0.008)	0.909 (0.008)
3	SIS_correct1	0.036 (0.008)	0.909 (0.008)
3	SIS_correct2	0.124 (0.015)	0.909 (0.015)
3	SIS_correct3	0.278 (0.02)	0.909 (0.02)
3	WIS_correct0	0.028 (0.007)	0.909 (0.007)
3	WIS_correct1	0.026 (0.007)	0.909 (0.007)
3	WIS_correct2	0.084 (0.012)	0.909 (0.012)
3	WIS_correct3	0.31 (0.021)	0.909 (0.021)

True $\hat{\theta} = 0.893$. Average θ values evaluated on by third iteration: $\theta = (0.512, 0.614, 0.717, 0.819, 0.921, 1.023, 1.125, 1.227, 1.329, 1.431)$ over 500 replications.

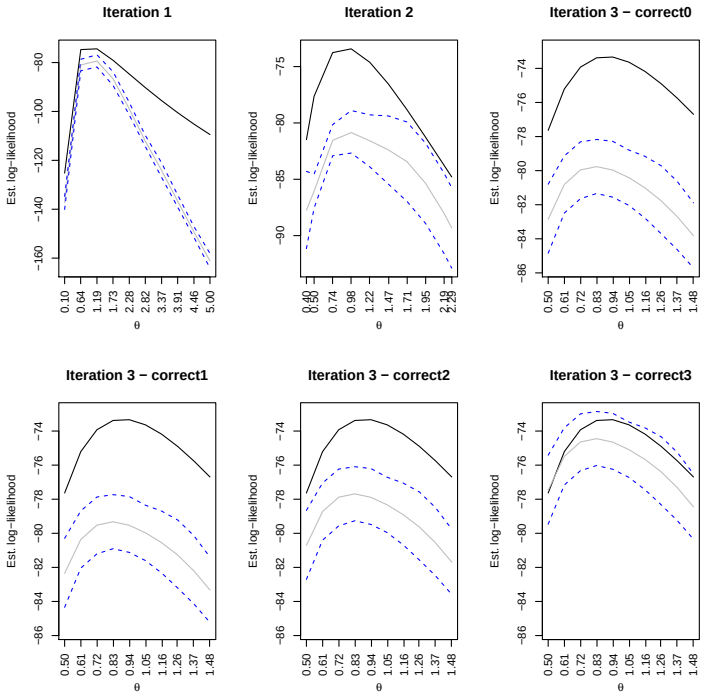
Table 3.13: $n = 50, m = 1e6$

Iteration	Type	CovProb	MLE
1	SIS_correct0	0 (0)	0.972 (0)
1	WIS_correct0	0 (0)	0.971 (0)
2	SIS_correct0	0.006 (0.003)	0.913 (0.003)
2	WIS_correct0	0.008 (0.004)	0.913 (0.004)
3	SIS_correct0	0.024 (0.007)	0.9 (0.007)
3	SIS_correct1	0.044 (0.009)	0.901 (0.009)
3	SIS_correct2	0.146 (0.016)	0.901 (0.016)
3	SIS_correct3	0.306 (0.021)	0.901 (0.021)
3	WIS_correct0	0.024 (0.007)	0.9 (0.007)
3	WIS_correct1	0.018 (0.006)	0.9 (0.006)
3	WIS_correct2	0.11 (0.014)	0.9 (0.014)
3	WIS_correct3	0.282 (0.02)	0.9 (0.02)

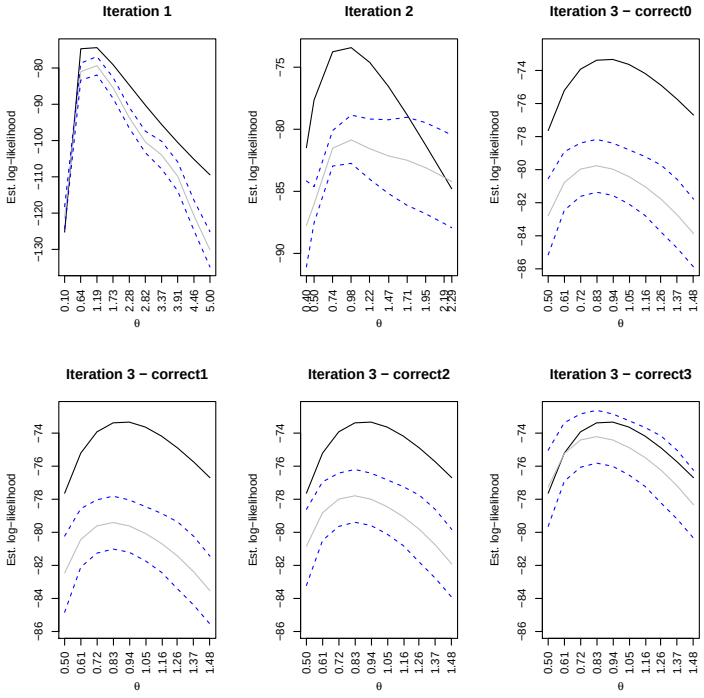
True $\hat{\theta} = 0.893$. Average θ values evaluated on by third iteration: $\theta = (0.512, 0.613, 0.713, 0.814, 0.914, 1.014, 1.115, 1.215, 1.316, 1.416)$ over 500 replications.

When $n = 15$, the estimated log-likelihood using SIS or WIS, reaches the 90% coverage probability by the third iteration without any correction. However as n increases, hence

the dimension of the random effects in this example, the coverage probability without corrections is below 90%. Although, with the empirically motivated corrections (2 and 3), when the dimension of the random effects is not as large, for example when $n = 25$, `correct3` may over correct the estimate, hence reducing coverage. This implies, there may be an argument for when one of the corrections is more advantageous as a function of the random effect dimension.



(a) SIS estimate.



(b) WIS estimate.

Figure 3.4: One run of estimated log-likelihood from iterative procedure when $m = 1e5$. True log-likelihood in solid black line, estimated log-likelihood in solid gray, and 90% simultaneous confidence bands in dashed blue.

3.5.4 Example 4: Salamander Data

The Salamander data is based on a 1986 experiment where University of Chicago scientists tried to determine if salamanders of the same type were more or less likely to mate with one another. The experiment and accompanying dataset was presented in McCullagh and Nelder (1989). The dataset has been studied intensely and is considered a benchmark set for GLMMs (Knudson, 2016). Karim and Zeger (1992) first fit a statistical model to this data with random effects which they referred to as "Model A". In this model, two populations of salamanders are considered, "Rough Butt" and "White Side" named based on their location of origin. There are four combinations for mating pairs: Male Rough Butt with Female Rough Butt (RR), Male Rough Butt with Female White Side (RW), Male White Side with Female Rough Butt (WR), and Male White Side with Female White Side (WW). The fundamental question is: what are the odds that salamanders of the same type will mate with one another versus mating with salamanders of a different type?

The scientists ran the following experiment three times: using two closed groups of 20 salamanders. Each group contained five female Rough Butts, five female White Sides, five male Rough Butts, and five male White Sides. In each trial a female and a male salamander were placed in an isolated space together, then observed whether they mated or not. Each female salamander was part of six trials with male salamanders from her closed group; three trials with male WS and three trials with male RB. Scientists paired salamanders from the same closed group, no inter-group trials were conducted. Overall, 60 trials were conducted on each closed group, where each experiment consisted of 120 trials. Therefore, the overall dataset contains binary responses from 360 trials (Knudson, 2016).

In Karim and Zeger (1992)'s "Model A" there is a random effect for female salamanders, a random effect for male salamanders, a fixed effect predictor for all mating crosses, and the outcome is the binary response if the salamanders mated or not. This means $Y \in \mathbb{R}^{360}, X \in \mathbb{R}^{360\times 4}, \beta \in \mathbb{R}^4, Z \in \mathbb{R}^{360\times 120}$, and $U \in \mathbb{R}^{120}$. Let $x_{i\cdot}$ and $z_{i\cdot}$ denote the ith row of the X and Z matrices respectively. Also let $\beta = (\beta_{RR}, \beta_{RW}, \beta_{WR}, \beta_{WW})^T$, where each component represents the odds of mating for each mating pair. For female

salamanders, the random effects $u_F \overset{iid}{\sim} N(0, \nu_F)$ and the male salamanders random effects $u_M \overset{iid}{\sim} N(0, \nu_M)$, both u_F and u_M are in $\mathbb{R}^{60}$. Then,

$$f_\beta(y|u) = \exp\left(y^T(X\beta + ZU) - \sum_{i=1}^{360} \log(1 + \exp\{x_{i\cdot}\beta + z_{i\cdot}u\})\right) \text{ and}$$
$$f_\nu(u) = \exp\left(-60\log(2\pi) - 30\log\nu_F - 30\log\nu_M - \frac{u_F^T u_F}{2\nu_F} - \frac{u_M^T u_M}{2\nu_M}\right).$$

Which implies for $\theta = (\beta_{RR}, \beta_{RW}, \beta_{WR}, \beta_{WW}, \nu_F, \nu_M)^T$,

$$\begin{aligned} L(\theta) &= \int_{\mathcal{U}} f_\beta(y|u) f_\nu(u) du \\ &= \int_{\mathcal{U}} \exp\Big(y^T(X\beta + ZU) - \sum_{i=1}^{360} \log(1 + \exp\{x_{i\cdot}\beta + z_{i\cdot}u\}) - 60\log(2\pi) \\ &\quad - 30\log\nu_F - 30\log\nu_M - \frac{u_F^T u_F}{2\nu_F} - \frac{u_M^T u_M}{2\nu_M}\Big) du. \end{aligned}$$

The likelihood contains a 120-dimensional integral. Since the experiment was conducted independently among two closed groups with three independent replications, this integral can be split at most into the product of six 20-dimensional integrals. Since this integral is still too complicated for numerical analysis due to crossed random effects, we use MCLA. Building upon the work of Knudson (2016), we further the MCLA procedure by adding a correction to the log likelihood estimate and creating simultaneous confidence bands.

We use the default settings in the `glmm` package which uses the SIS estimator in equation (1.32) to approximate the log-likelihood function. Initial values for θ are from penalized quasi likelihood, the importance distribution is an equal mixture of two multivariate normals and a Student's t-distribution, such that the weights have a finite second moment. In addition, the variance components for the random effects, ν_F and ν_M, are considered distinct.

The code we run for $m = 10^4, 10^5$, and, 5×10^5 independently is,

```
sal = glmm(Mate ~ 0 + Cross,
  random = list(~ 0 + Female, ~ 0 + Male), varcomps.names = c("F", "M"),
```

```
data = salamander,
family.glmm = bernoulli.glmm,
m = m,
debug = TRUE)
```

Table 3.14 presents the MC-MLEs, their 90% CIs (constructed using the observed Fisher information from the MCLA procedure), and the estimated log-likelihood at this MC-MLE.

	$m = 1e4$	$m = 1e5$	$m = 5e5$
$\hat{\beta}_{RR}$	1.197 (0.757, 1.638)	1.021 (0.392, 1.65)	0.993 (0.352, 1.634)
$\hat{\beta}_{RW}$	0.657 (0.228, 1.087)	0.309 (-0.362, 0.979)	0.306 (-0.32, 0.931)
$\hat{\beta}_{WR}$	-2.304 (-2.841, -1.767)	-1.914 (-2.621, -1.207)	-1.914 (-2.634, -1.195)
$\hat{\beta}_{WW}$	1.176 (0.728, 1.623)	0.992 (0.299, 1.686)	0.976 (0.316, 1.635)
$\hat{\nu}_F$	1.785 (1.244, 2.327)	1.347 (0.512, 2.181)	1.305 (0.424, 2.186)
$\hat{\nu}_M$	1.323 (0.924, 1.723)	1.243 (0.327, 2.16)	1.152 (0.383, 1.921)
$\ell_m(\theta_m \mid y)$	-205.99	-207.611	-207.646

Table 3.14: MC-MLEs for different Monte Carlo sample size, m, with 90% confidence intervals in parentheses. Last row is the uncorrected estimated log-likelihood at that MC-MLE.

To implement our additions, we first study the profile log-likelihoods with simultaneous confidence bands. In other words, for a given Monte Carlo sample, we fix five of the six parameters at their respective MC-MLE and estimate the log likelihood function over a grid of the remaining parameter.

For instance, for $m = 10^4, \theta_m = (1.197, 0.657, -2.304, 1.176, 1.785, 1.323)$. To estimate the profile log-likelihood of β_{RR}, we fix the last five elements of θ_m and only vary β_{RR}. We consider a range of grid points equally spaced between $\hat{\beta}_{m,RR} \pm 0.5$ including $\hat{\beta}_{m,RR}$. Then using the m samples, we estimate the log-likelihood function over this range. We do this process for each element of θ, keeping the remaining parameters fixed at their MC-MLE. We also apply the log-likelihood correction proposed earlier. Therefore, for each of the six parameters in θ, we estimate its profile log-likelihood using the three different Monte Carlo sets and apply the four corrections. We also construct the 90% simultaneous confidence bands. In total there are 72 plots. These

plots are in Section 3.5.4 organized by parameter. As clear in the plots, as m increases the curvature in the profile log-likelihood around the maximum is more pronounced and the simultaneous confidence band width is smaller, showing increased certainty at the maximum point.

In addition, for each m we obtained an MC-MLE. Let $m_1 = 10^4, m_2 = 10^5$, and $m_3 = 5 \times 10^5$. Then $\theta_{m_1}, \theta_{m_2}$, and, θ_{m_3} represent their respective MC-MLEs based on the full likelihood. Consider $\theta_{m_1}, \theta_{m_2}$, and, θ_{m_3} are fixed. Now we use the three sets of Monte Carlo samples of size m_1, m_2, and m_3 to estimate the log-likelihood at these three vectors. Then we construct the 90% simultaneous confidence bands.

Since the true log-likelihood function is unknown, we can use these confidence bands as a way to gauge how accurate the estimates are. We know as m approaches infinity, the estimated log-likelihood will converge to the truth for any value of θ. We also compare the effect the bias correction makes on the estimated log-likelihood.

	θ_{m_1}	θ_{m_2}	θ_{m_3}
correct0	-205.99 (-208.4, -203.552)	-207.694 (-208.178, -207.202)	-207.73 (-208.021, -207.438)
correct1	-205.416 (-207.826, -202.979)	-207.582 (-208.066, -207.09)	-207.631 (-207.922, -207.339)
correct2	-203.782 (-206.192, -201.344)	-205.948 (-206.432, -205.456)	-205.997 (-206.288, -205.705)
correct3	-201.099 (-203.508, -198.661)	-203.264 (-203.749, -202.772)	-203.314 (-203.604, -203.021)

Table 3.15: Estimated log-likelihood with 90% bands using $m_1 = 10^4$ dataset.

	θ_{m_1}	θ_{m_2}	θ_{m_3}
correct0	-208.983 (-209.447, -208.519)	-207.611 (-207.756, -207.465)	-207.626 (-207.714, -207.537)
correct1	-208.918 (-209.382, -208.454)	-207.561 (-207.707, -207.416)	-207.577 (-207.665, -207.489)
correct2	-207.758 (-208.222, -207.293)	-206.401 (-206.547, -206.256)	-206.417 (-206.505, -206.329)
correct3	-206.675 (-207.139, -206.21)	-205.318 (-205.464, -205.173)	-205.334 (-205.422, -205.246)

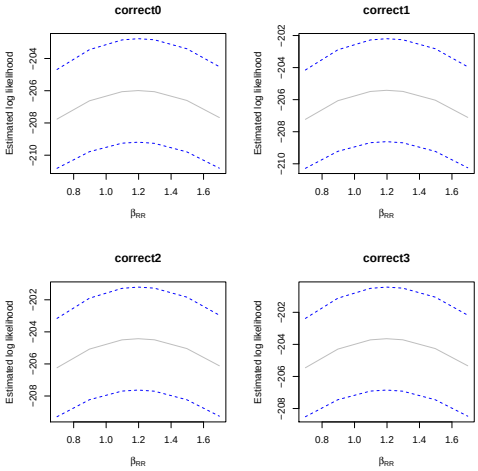
Table 3.16: Estimated log-likelihood with 90% bands using $m_2 = 10^5$ dataset.

	θ_{m_1}	θ_{m_2}	θ_{m_3}
correct0	-209.1 (-209.403, -208.797)	-207.669 (-207.743, -207.596)	-207.646 (-207.696, -207.596)
correct1	-209.077 (-209.379, -208.774)	-207.653 (-207.727, -207.579)	-207.63 (-207.68, -207.58)
correct2	-208.522 (-208.824, -208.219)	-207.098 (-207.172, -207.024)	-207.074 (-207.124, -207.025)
correct3	-208.329 (-208.631, -208.026)	-206.905 (-206.979, -206.831)	-206.881 (-206.931, -206.832)

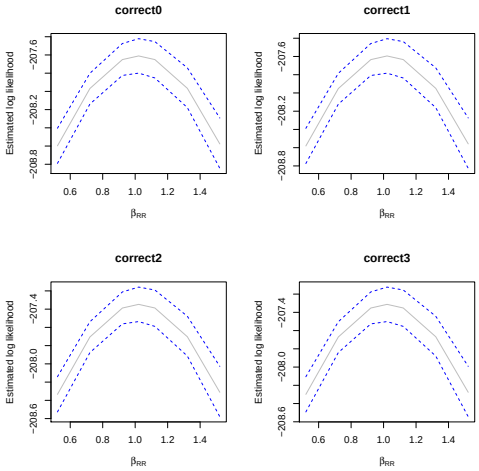
Table 3.17: Estimated log-likelihood with 90% bands using $m_3 = 5 \times 10^5$ dataset.

Since $m_1 < m_2 < m_3$ it is not surprisingly that the bands of around $\ell_{m_1}(\theta_{m_3})$ using the `correct1` bias correction capture $\ell_{m_3}(\theta_{m_3})$. Likewise with $\ell_{m_2}(\theta_{m_3})$ using the `correct1` bias correction capture $\ell_{m_3}(\theta_{m_3})$. The same cannot be said using the `correct2` or `correct3` bias corrections. This may further allude to some type of over-correction that requires further analysis.

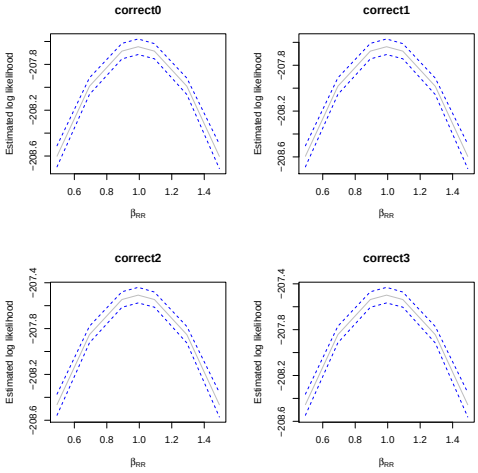
Profile log-likelihood plots



(a) $m = 10^4$.

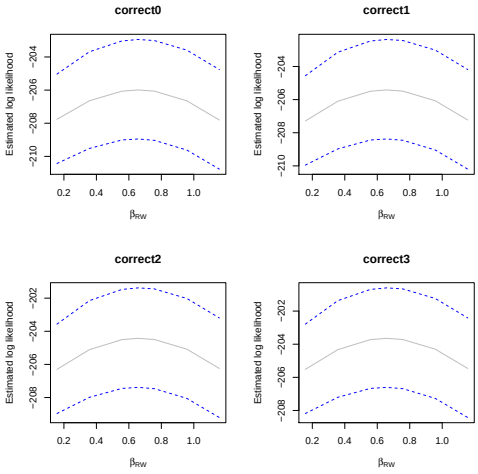


(b) $m = 10^5$.

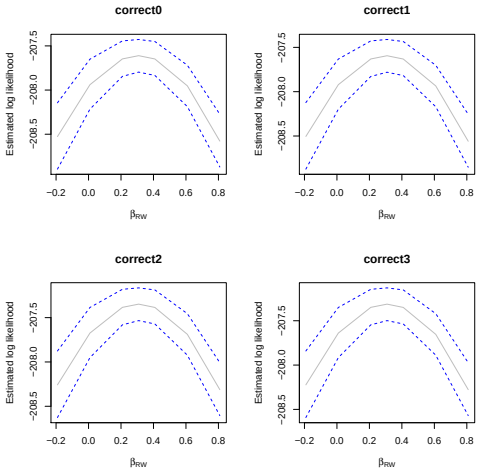


(c) $m = 5 \times 10^5$.

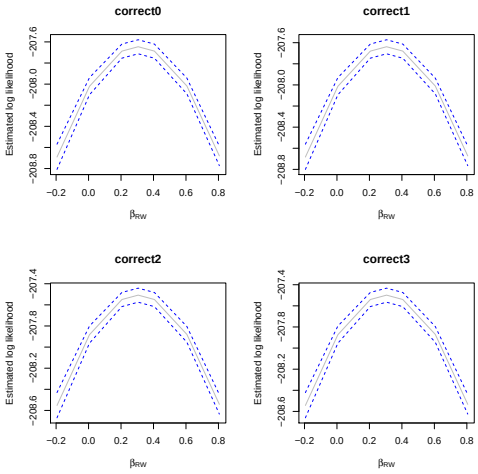
Figure 3.5: Profile log-likelihood for β_{RR}. Estimated profile log-likelihood in solid gray with 90% simultaneous confidence band in dashed blue.



(a) $m = 10^4$.

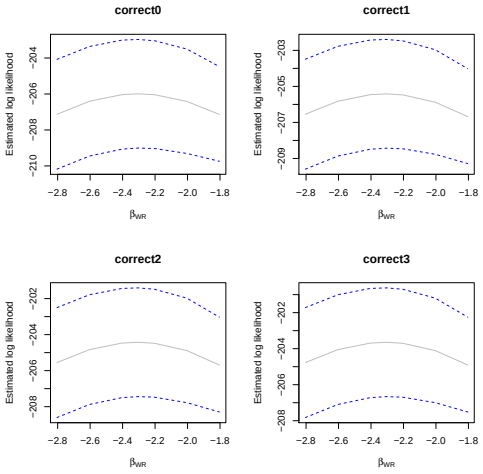


(b) $m = 10^5$.

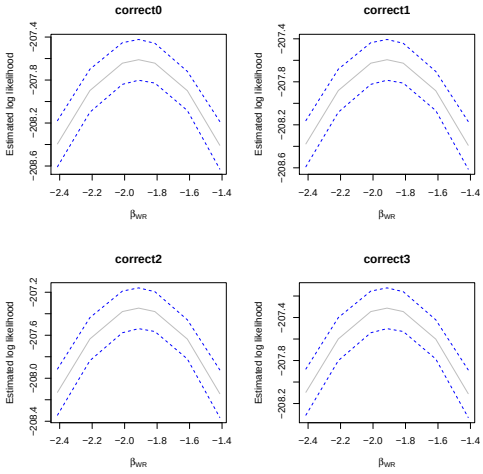


(c) $m = 5 \times 10^5$.

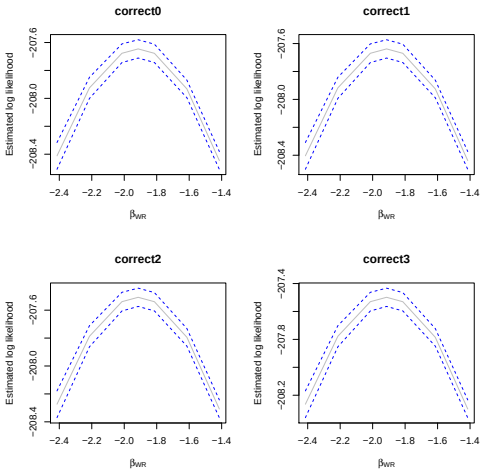
Figure 3.6: Profile log-likelihood for β_{RW}. Estimated profile log-likelihood in solid gray with 90% simultaneous confidence band in dashed blue.



(a) $m = 10^4$.

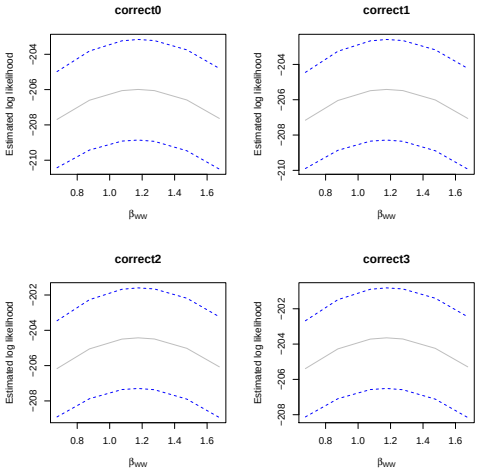


(b) $m = 10^5$.

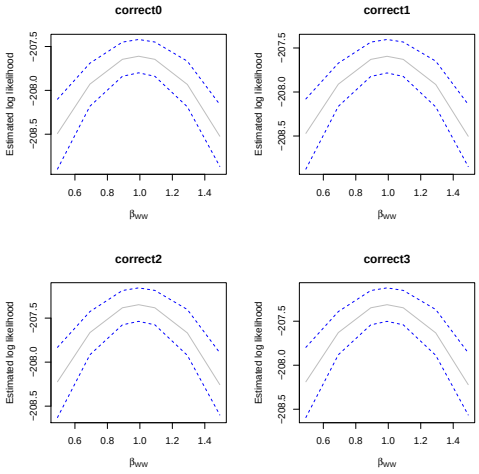


(c) $m = 5 \times 10^5$.

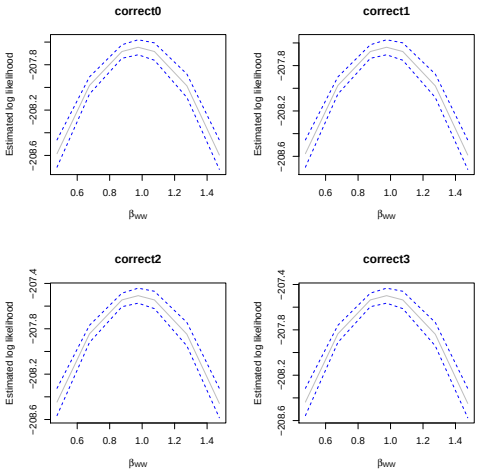
Figure 3.7: Profile log-likelihood for β_{WR}. Estimated profile log-likelihood in solid gray with 90% simultaneous confidence band in dashed blue.



(a) $m = 10^4$.

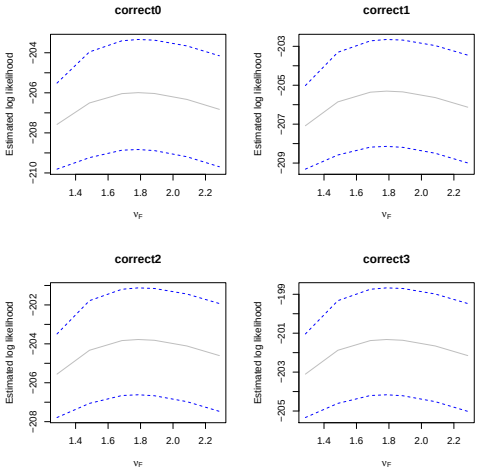


(b) $m = 10^5$.

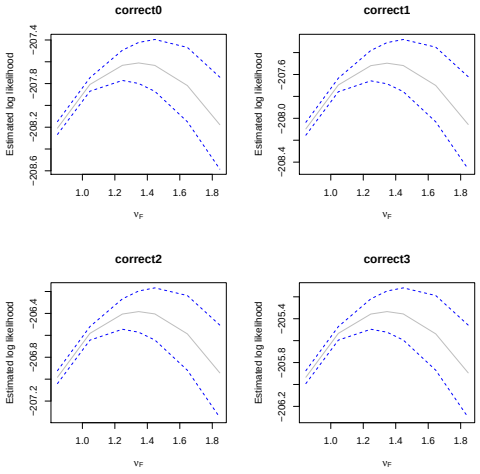


(c) $m = 5 \times 10^5$.

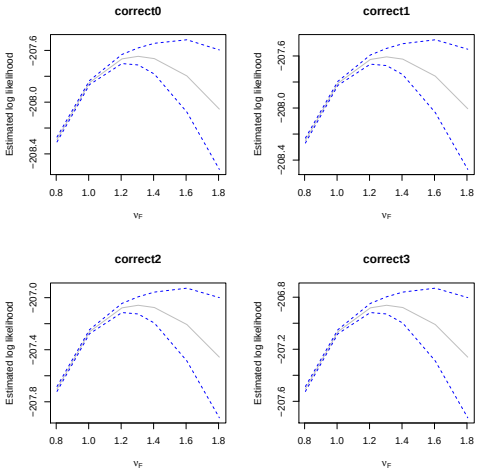
Figure 3.8: Profile log-likelihood for β_{WW}. Estimated profile log-likelihood in solid gray with 90% simultaneous confidence band in dashed blue.



(a) $m = 10^4$.

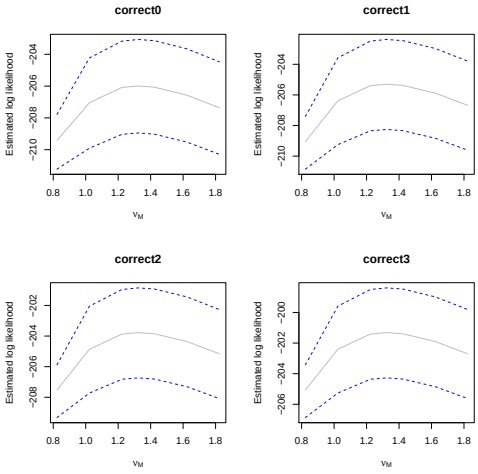


(b) $m = 10^5$.

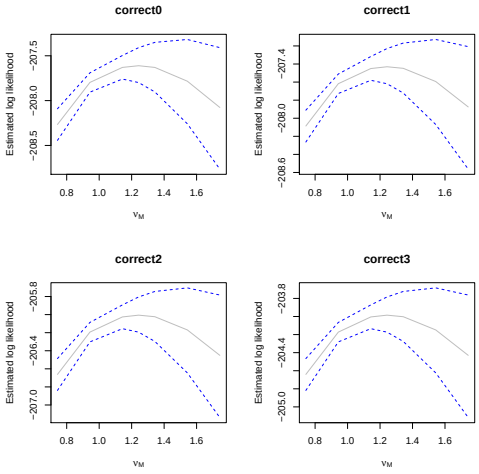


(c) $m = 5 \times 10^5$.

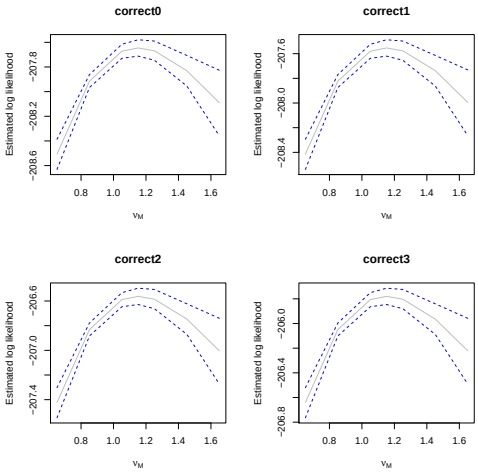
Figure 3.9: Profile log-likelihood for ν_F. Estimated profile log-likelihood in solid gray with 90% simultaneous confidence band in dashed blue.



(a) $m = 10^4$.



(b) $m = 10^5$.



(c) $m = 5 \times 10^5$.

Figure 3.10: Profile log-likelihood for ν_M. Estimated profile log-likelihood in solid gray with 90% simultaneous confidence band in dashed blue.

Profile log-likelihood confidence intervals over a larger grid

Now using a larger of grid of ± 1 including the MC-MLE, we construct the profile log-likelihood confidence intervals. Again, we do this process for each element of θ, keeping the others parameters fixed at their MC-MLE. For example, consider for some m with MC-MLE θ_m we want to construct the 90% profile log likelihood confidence

interval for β_{RR}. Define $\ell(\beta_{RR})$ as its profile log-likelihood function where the remaining parameters are fixed at their MC-MLEs based on the full likelihood. We use Wilk's theorem (Ferguson, 1996) to construct the confidence interval by finding the set of β_{RR} such that $\log(\theta_m) - \chi^2_{1,1-\alpha} < \ell(\beta_{RR})$. Table 3.18 displays the 90% confidence intervals using the profile likelihood method versus the Fisher information.

	CI type	$m = 1e4$	$m = 1e5$	$m = 5e5$
$\hat{\beta}_{RR}$	Estimate	1.197	1.021	0.993
	Fisher	(0.757, 1.638)	(0.392, 1.65)	(0.352, 1.634)
	Profile	(0.997, 1.397)	(0.821, 1.221)	(0.793, 1.193)
$\hat{\beta}_{RW}$	Estimate	0.657	0.309	0.306
	Fisher	(0.228, 1.087)	(-0.362, 0.979)	(-0.32, 0.931)
	Profile	(0.457, 0.857)	(-0.291, 0.509)	(0.106, 0.506)
$\hat{\beta}_{WR}$	Estimate	-2.304	-1.914	-1.914
	Fisher	(-2.841, -1.767)	(-2.621, -1.207)	(-2.634, -1.195)
	Profile	(-2.504, -2.104)	(-2.514, -1.314)	(-2.514, -1.314)
$\hat{\beta}_{WW}$	Estimate	1.176	0.992	0.976
	Fisher	(0.728, 1.623)	(0.299, 1.686)	(0.316, 1.635)
	Profile	(0.976, 1.376)	(0.392, 1.592)	(0.776, 1.176)
$\hat{\nu}_F$	Estimate	1.785	1.347	1.305
	Fisher	(1.244, 2.327)	(0.512, 2.181)	(0.424, 2.186)
	Profile	(1.585, 2.385)	(0.747, 1.947)	(0.705, 1.905)
$\hat{\nu}_M$	Estimate	1.323	1.243	1.152
	Fisher	(0.924, 1.723)	(0.327, 2.16)	(0.383, 1.921)
	Profile	(1.123, 1.523)	(0.643, 1.843)	(0.952, 2.152)

Table 3.18: 90% confidence intervals in parentheses using estimated Fisher information matrix from `glmm` versus 90% profile log likelihood confidence intervals. Regardless of corrections, the profile confidence intervals were the same.

Therefore, the profile likelihood confidence intervals give a tighter interval around the estimated value versus using the Fisher information matrix. This could be used for better inference about the coefficients of interest.

References

Ahmed, N. K., Neville, J., and Kompella, R. (2014). Network sampling: From Static to Streaming Graphs. *ACM Transactions on Knowledge Discovery from Data (TKDD)*.

Aldous, D., Lovász, L., and Winkler, P. (1997). Mixing times for uniformly ergodic Markov chains. *Stochastic Processes and their Applications*.

Andrews, D. (1991). Heteroskedasticity and autocorrelation consistent covariant matrix estimation. *Econometrica*.

Avrachenkov, K., Borkar, V. S., Kadavankandy, A., and Sreedharan, J. K. (2016). Comparison of Random Walk Based Techniques for Estimating Network Averages. In Nguyen, H. T. and Snasel, V., editors, *Computational Social Networks: 5th International Conference, CSoNet 2016, Ho Chi Minh City, Vietnam, August 2-4, 2016, Proceedings*. Springer International Publishing.

Avrachenkov, K., Borkar, V. S., Kadavankandy, A., and Sreedharan, J. K. (2018). Revisiting random walk based sampling in networks: evasion of burn-in period and frequent regenerations. *Computational Social Networks*.

Bennett, G. (1962). Probability inequalities for the sum of independent random variables. *Journal of the American Statistical Association*.

Bernstein, S. (1927). Theory of probability.

Birkhoff, G. D. (1931). Proof of the Ergodic Theorem. *Proceedings of the National Academy of Sciences*.

Blagus, N., Šubelj, L., and Bajec, M. (2017). Empirical comparison of network sampling: How to choose the most appropriate method? *Physica A: Statistical Mechanics and its Applications*.

Bremaud, P. (2010). *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*. Springer Science and Business Media.

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. (2011). *Handbook of Markov Chain Monte Carlo*. CRC press.

Burda, Y., Grosse, R., and Salakhutdinov, R. (2015). Accurate and conservative estimates of MRF log-likelihood using reverse annealing. In *Artificial Intelligence and Statistics*.

Casella, G. and Berger, R. L. (2002). *Statistical Inference*, volume 2. Duxbury Pacific Grove, CA.

Chen, D.-F. R. and Seila, A. F. (1987). Multivariate inference in stationary simulation using batch means. In *Proceedings of the 19th conference on Winter simulation*. ACM.

Chen, Y.-C. (2017). A tutorial on kernel density estimation and recent advances. *Biostatistics & Epidemiology*.

Chiericetti, F., Dasgupta, A., Kumar, R., Lattanzi, S., and Sarlós, T. (2016). On sampling nodes in a network. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee.

Cowles, M. K. and Carlin, B. P. (1996). Markov chain Monte Carlo Convergence Diagnostics: A Comparative Review. *Journal of the American Statistical Association*.

Cummins, D. J., Filloon, T. G., and Nychka, D. (2001). Confidence Intervals for Nonparametric Curve Estimates: Toward More Uniform Pointwise Coverage. *Journal of the American Statistical Association*.

Dai, N. and Jones, G. L. (2017). Multivariate initial sequence estimators in Markov chain Monte Carlo. *Journal of Multivariate Analysis*, 159:184–199.

De Valpine, P. (2004). Monte Carlo state-space likelihoods by weighted posterior kernel density estimation. *Journal of the American Statistical Association*.

Degras, D. (2017). Simultaneous confidence bands for the mean of functional data. *Wiley Interdisciplinary Reviews: Computational Statistics*.

Diaconis, P. and Stroock, D. (1991). Geometric Bounds for Eigenvalues of Markov Chains. *The Annals of Applied Probability*.

Durak, N., Pinar, A., Kolda, T. G., and Seshadhri, C. (2012). Degree Relations of Triangles in Real-world Networks and Graph Models. In *Proceedings of the 21st ACM international conference on Information and Knowledge Management*. ACM.

Ellison, N. B. et al. (2007). Social Network Sites: Definition, History, and Scholarship. *Journal of Computer-Mediated Communication*.

Erdos, P. and Rényi, A. (1960). On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*.

Facebook (2019). Stats. https://newsroom.fb.com/company-info/. Accessed: 2019-06-27.

Ferguson, T. S. (1996). *A course in large sample theory*. Chapman & Hall/CRC Press.

Flegal, J., Hughes, J., and Vats, D. (2015). mcmcse: Monte Carlo standard errors for MCMC. *Riverside, CA and Minneapolis, MN. R package version 1.3.2*.

Flegal, J. M., Haran, M., and Jones, G. L. (2008a). Markov chain Monte Carlo: Can We Trust the Third Significant Figure? *Statistical Science*.

Flegal, J. M., Haran, M., and Jones, G. L. (2008b). Markov chain Monte Carlo: Can we trust the third significant figure? *Statistical Science*.

Flegal, J. M., Jones, G. L., et al. (2010). Batch means and spectral variance estimators in Markov chain Monte Carlo. *The Annals of Statistics*.

Gelfand, A. E. and Smith, A. F. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*.

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences (with discussion). *Statistical Science*.

Geweke, J. (1992). Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments (with discussion). In Bernardo, J. M., Berger, J. O., Dawid, A. P., and Smith, A. F. M., editors, *Bayesian Statistics 4. Proceedings of the Fourth Valencia International Meeting*. Clarendon Press.

Geyer, C. J. (1992). Practical Markov Chain Monte Carlo. *Statistical Science*.

Geyer, C. J. (1994). On the convergence of Monte Carlo maximum likelihood calculations. *Journal of the Royal Statistical Society: Series B (Methodological)*.

Geyer, C. J. and Thompson, E. A. (1992). Constrained Monte Carlo maximum likelihood for dependent data. *Journal of the Royal Statistical Society: Series B (Methodological)*.

Gile, K. J. and Handcock, M. S. (2010). Respondent-driven Sampling: an Assessment of Current Methodology. *Sociological Methodology*.

Gjoka, M., Kurant, M., Butts, C. T., and Markopoulou, A. (2011). Practical Recommendations on Crawling Online Social Networks. *IEEE Journal on Selected Areas in Communications*.

Glynn, P. W. and Whitt, W. (1992). The asymptotic efficiency of simulation estimators. *Operations research*.

Goel, S. and Salganik, M. J. (2009). Respondent-driven sampling as Markov chain Monte Carlo. *Statistics in Medicine*.

Gong, L. and Flegal, J. M. (2016). A Practical Sequential Stopping Rule for High-Dimensional Markov Chain Monte Carlo. *Journal of Computational and Graphical Statistics*.

Guo, S. W. and Thompson, E. (1992). A Monte Carlo method for combined segregation and linkage analysis. *American journal of human genetics*.

Handcock, M. S., Hunter, D. R., Butts, C. T., Goodreau, S. M., and Morris, M. (2008). statnet: Software Tools for the Representation, Visualization, Analysis and Simulation of Network Data. *Journal of Statistical Software*.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*.

Heckathorn, D. D. and Cameron, C. J. (2017). Network sampling: From Snowball and Multiplicity to Respondent-Driven Sampling. *Annual Review of Sociology*.

Heidelberger, P. and Welch, P. D. (1983). Simulation run length control in the presence of an initial transient. *Operations Research*.

Hobert, J. P., Jones, G. L., Presnell, B., and Rosenthal, J. S. (2002). On the applicability of regenerative simulation in Markov chain Monte Carlo. *Biometrika*.

Huber, M. L. (2016). *Perfect Simulation*. Chapman and Hall/CRC.

Jackson, M. O. and Watts, A. (2002). The Evolution of Social and Economic Networks. *Journal of Economic Theory*.

Jones, G. L. (2004). On the Markov chain central limit theorem. *Probability Surveys*.

Jones, G. L., Haran, M., Caffo, B. S., and Neath, R. (2006a). Fixed-width output analysis for Markov chain Monte Carlo. *Journal of the American Statistical Association*.

Jones, G. L., Haran, M., Caffo, B. S., and Neath, R. (2006b). Fixed-Width Output Analysis for Markov Chain Monte Carlo. *Journal of the American Statistical Association*.

Jones, M. C. (1993). Simple boundary correction for kernel density estimation. *Statistics and computing*.

Joyce, J. M. (2011). Kullback-Leibler Divergence. In *International Encyclopedia of Statistical Science*. Springer.

Kahn, H. and Harris, T. E. (1951). Estimation of particle transmission by random sampling. *National Bureau of Standards applied mathematics series*.

Karim, M. R. and Zeger, S. L. (1992). Generalized linear models with random effects; salamander mating revisited. *Biometrics*.

Knudson, C. (2016). *Monte Carlo Likelihood Approximation for Generalized Linear Mixed Models*. PhD thesis, university of minnesota.

Kolaczyk, E. D. (2009). *Statistical Analysis of Network Data: Methods and Models*. Springer New York.

Kolaczyk, E. D. (2017). *Topics at the Frontier of Statistics and Network Analysis*. Cambridge University Press.

König, D. (1950). Theorie der Endlichen und Unendlichen Graphen. *American Mathematical Society*.

Kosorok, M. R. (2000). Monte Carlo error estimation for multivariate Markov chains. *Statistics & probability letters*.

Kumar, R., Novak, J., and Tomkins, A. (2010). Structure and Evolution of Online Social Networks. In *Link Mining: Models, Algorithms, and Applications*. Springer.

Lee, C.-H., Xu, X., and Eun, D. Y. (2012). Beyond Random Walk and Metropolis-Hastings Samplers: Why You Should Not Backtrack for Unbiased Graph Sampling. In *ACM SIGMETRICS Performance Evaluation Review*. ACM.

Lee, S. H., Kim, P.-J., and Jeong, H. (2006). Statistical properties of sampled networks. *Physical Review E*.

Lehmann, E. (1999). *Elements of Large-Sample Theory*. Springer Science & Business Media New York.

Leskovec, J. and Faloutsos, C. (2006). Sampling from Large Graphs. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.

Leskovec, J. and Sosič, R. (2016). SNAP: A General-Purpose Network Analysis and Graph-Mining Library. *ACM Transactions on Intelligent Systems and Technology (TIST)*.

Levin, D. A., Peres, Y., and Wilmer, E. L. (2009). *Markov Chains and Mixing Times*. American Mathematical Soc.

Li, R.-H., Yu, J. X., Qin, L., Mao, R., and Jin, T. (2015). On Random Walk Based Graph Sampling. In *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*. IEEE.

Liu, J. S. (2008). *Monte Carlo Strategies in Scientific Computing*. Springer Science & Business Media.

Liu, Q., Peng, J., Ihler, A., and Fisher III, J. (2015). Estimating the partition function by discriminance sampling. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*. AUAI Press.

Liu, Y. and Flegal, J. (2018a). Optimal mean squared error bandwidth for spectral variance estimators in mcmc simulations. *ArXiv e-prints*.

Liu, Y. and Flegal, J. M. (2018b). Weighted batch means estimators in Markov chain Monte Carlo. *Electron. J. Statist.*

Lu, J. and Li, D. (2012). Sampling online social networks by random walk. In *Proceedings of the First ACM International Workshop on Hot Topics on Interdisciplinary Social Networks Research*. ACM.

Luke, D. A. and Harris, J. K. (2007). Network analysis in public health: history, methods, and applications. *Annual Review of Public Health*.

McCullagh, P. and Nelder, J. (1989). *Generalized Linear Models, Second Edition*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis.

Mengersen, K. L. and Tweedie, R. L. (1996). Rates of convergence of the Hastings and Metropolis algorithms. *The Annals of Statistics*.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*.

Meyn, S. P. and Tweedie, R. L. (1993). *Markov chains and stochastic stability*. Springer London.

Mohaisen, A., Yun, A., and Kim, Y. (2010). Measuring the mixing time of social graphs. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM.

Montiel Olea, J. L. and Plagborg-Møller, M. (2019). Simultaneous confidence bands: Theory, implementation, and an application to SVARs. *Journal of Applied Econometrics*.

Moreno, J. L. and Jennings, H. H. (1938). Statistics of Social Configurations. *Sociometry*.

Newman, J. R. (1953). Leonhard Euler and the Königsberg Bridges. *Scientific American*.

Newman, M. E. (2003). Mixing patterns in networks. *Physical Review E*.

Newman, M. E., Watts, D. J., and Strogatz, S. H. (2002). Random graph models of social networks. *Proceedings of the National Academy of Sciences*.

Noldus, R. and Van Mieghem, P. (2015). Assortativity in Complex Networks. *Journal of Complex Networks*.

Owen, A. B. (2013). *Monte Carlo theory, methods and examples*.

Park, J. and Haran, M. (2019). Reduced-dimensional Monte Carlo Maximum Likelihood for Latent Gaussian Random Field Models. *arXiv preprint arXiv:1910.09711*.

Resnick, M. D., Bearman, P. S., Blum, R. W., Bauman, K. E., Harris, K. M., Jones, J., Tabor, J., Beuhring, T., Sieving, R. E., Shew, M., et al. (1997). Protecting adolescents from harm: Findings from the National Longitudinal Study on Adolescent Health. *Journal of the American Medical Association*.

Ribeiro, B. and Towsley, D. (2010). Estimating and sampling graphs with multidimensional random walks. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM.

Roberts, G. O. and Rosenthal, J. S. (2004). General state space Markov chains and MCMC algorithms. *Probability surveys*.

Rohe, K. et al. (2019). A critical threshold for design effects in network sampling. *The Annals of Statistics*.

Salamanos, N., Voudigari, E., and Yannakoudakis, E. J. (2017). Deterministic graph exploration for efficient graph sampling. *Social Network Analysis and Mining*.

Salganik, M. J. and Heckathorn, D. D. (2004). Sampling and estimation in hidden populations using Respondent-Driven Sampling. *Sociological Methodology*.

Scott, J. (2017). *Social Network Analysis*. Sage.

Seila, A. F. (1982). Multivariate estimation in regenerative simulation. *Operations Research Letters*.

Serfling, R. J. (2009). *Approximation theorems of mathematical statistics*. John Wiley & Sons.

Tanner, M. A. and Wong, W. H. (1987). The calculation of posterior distributions by data augmentation. *Journal of the American statistical Association*.

Thompson, S. K. (2017). Adaptive and Network Sampling for Inference and Interventions in Changing Populations. *Journal of Survey Statistics and Methodology*.

Torrie, G. M. and Valleau, J. P. (1977). Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling. *Journal of Computational Physics*.

Traud, A. L., Kelsic, E. D., Mucha, P. J., and Porter, M. A. (2008). Community Structure in Online Collegiate Social Networks. *arXiv preprint arXiv:0809.0960*.

Vats, D. (2017). *Output Analysis for Markov Chain Monte Carlo*. PhD thesis, University of Minnesota, Twin Cities, Retrieved from the University of Minnesota Digital Conservancy, http://hdl.handle.net/11299/185631.

Vats, D. and Flegal, J. M. (2018). Lugsail lag windows and their application to MCMC. *arXiv:1809.04541*.

Vats, D., Flegal, J. M., and Jones, G. L. (2019). Multivariate Output Analysis for Markov Chain Monte Carlo. *Biometrika*.

Vats, D., Flegal, J. M., Jones, G. L., et al. (2018). Strong consistency of multivariate spectral variance estimators in Markov chain Monte Carlo. *Bernoulli*.

Vats, D. and Knudson, C. (2018). Revisiting the Gelman-Rubin Diagnostic. *arXiv preprint arXiv:1812.09384*.

Wang, T., Chen, Y., Zhang, Z., Xu, T., Jin, L., Hui, P., Deng, B., and Li, X. (2011). Understanding Graph Sampling Algorithms for Social Network Analysis. In *Distributed Computing Systems Workshops (ICDCSW), 2011 31st International Conference*. IEEE.

Wasserman, L. (2006). *All of nonparametric statistics*. Springer Science & Business Media.

Wasserman, S. and Faust, K. (1994). *Social Network Analysis: Methods and Applications*. Cambridge University Press.

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*.

Wei, G. C. and Tanner, M. A. (1990). A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American statistical Association*.

Wilks, S. S. (1932). Certain generalizations in the analysis of variance. *Biometrika*.

Zhou, Z., Zhang, N., Gong, Z., and Das, G. (2016). Faster Random Walks by Rewiring Online Social Networks On-the-Fly. *ACM Transactions on Database Systems (TODS)*.

Appendix A

Proofs for Chapter 3

A.1 Proof of Theorem 3.3.3

Proof. We have $M = \sup_\theta w_\theta(U) < \infty$ and $f_\theta(y|u) \in [0,1]$ for each y and θ. Then,

$$L_m(\theta|y) = \frac{1}{m}\sum_{t=1}^{m} f_\theta(y|u_t)w_\theta(u_t) \leq \frac{1}{m}\sum_{t=1}^{m} w_\theta(u_t) = \bar{W}_\theta \quad \text{where } w_\theta(u_t) \leq M.$$

Also notice,

$$Var_{\tilde{F}}(f_\theta(y|u)w_\theta(u)) \leq Var_{\tilde{F}}(w_\theta(u)) \leq \sup_{\theta\in\Theta} Var_{\tilde{F}}(w_\theta(u)) = \tau^2 < \infty.$$

That is, $L_m(\theta|y)$ is a sample mean of iid samples $f_\theta(y|u_t)w_\theta(u_t)$ each with mean $L(\theta|y)$, finite variance less than or equal to τ^2, and bounded by M. This sets the framework to use Berstein's inequality (Theorem 3.3.2).

Then for every $\epsilon > 0$,

$$Pr(L(\theta) - L_m(\theta) > \epsilon) \leq Pr(|L_m(\theta) - L(\theta)| > \epsilon) \tag{A.1}$$

$$\leq 2\exp\left\{\frac{-m\epsilon^2}{2\tau^2 + 2M\epsilon/3}\right\}. \tag{A.2}$$

Fix $\alpha \in (0,1)$. Then we can solve for ϵ ensuring $Pr(L(\theta) - L_m(\theta) > \epsilon) \leq \alpha$.

$$\begin{aligned} 2\exp\left\{\frac{-m\epsilon^2}{2\tau^2 + 2M\epsilon/3}\right\} &= \alpha \\ &\Rightarrow \frac{-m\epsilon^2}{2\tau^2 + 2M\epsilon/3} = \log(\alpha/2) \\ &\Rightarrow -m\epsilon^2 - \frac{2}{3}M\log(\alpha/2)\epsilon - 2\tau^2\log(\alpha/2) = 0. \end{aligned}$$

Then,

$$\epsilon = \frac{\frac{2}{3}M\log(\alpha/2) \pm \sqrt{\frac{4}{9}M^2(\log(\alpha/2))^2 - 8m\tau^2\log(\alpha/2)}}{-2m}.$$

Notice $\log(\alpha/2) < 0$ since $\alpha \in (0,1)$ and $M, \tau^2 > 0$. Since $\epsilon > 0$ this implies the only always positive solution is,

$$\epsilon = \frac{\frac{2}{3}M\log(\alpha/2) - \sqrt{\frac{4}{9}M^2(\log(\alpha/2))^2 - 8m\tau^2\log(\alpha/2)}}{-2m}. \tag{A.3}$$

We can add ϵ from (A.3) to $L_m(\theta)$ for a bias correction.

□

A.1.1 Proof of Corollary 3.3.3.1

Proof. From proof A.1, by rearranging to solve for m instead of ϵ we have,

$$2\exp\left\{\frac{-m\epsilon^2}{2\tau^2 + 2M\epsilon/3}\right\} = \alpha \tag{A.4}$$

$$\iff -m\epsilon^2 = \log(\alpha/2)(2\tau^2 + 2M\epsilon/3) \tag{A.5}$$

$$\iff m = -\epsilon^{-2}\log(\alpha/2)(2\tau^2 + 2M\epsilon/3). \tag{A.6}$$

□

A.1.2 Proof of Corollary 3.3.3.2

Proof. Notice,

$$\begin{aligned} Pr(L(\theta) - L_m(\theta) > \epsilon) &= Pr(L(\theta) - L_m(\theta) + 1 > \epsilon + 1) \\ &= Pr(\log[L(\theta) - L_m(\theta) + 1] > \log[\epsilon + 1]). \end{aligned}$$

Now we want a bound on the $Pr(\ell(\theta) - \ell_m(\theta) > \log(\epsilon + 1))$. So we need to relate $\ell(\theta) - \ell_m(\theta)$ to $\log[L(\theta) - L_m(\theta) + 1]$. The only assumption we make is that $L_m(\theta) \leq L(\theta)$ for all θ. This assumption is not overly constrictive since we mentioned after Corollary 3.3.1.1, that we are only focusing on when $L_m(\theta)$ underestimates. Also recall by definition $L(\theta) > 0$ and $L_m(\theta) > 0$ for all θ.

Let $\alpha = 2\exp\left\{\frac{-m\epsilon^2}{2\tau^2 + 2M\epsilon/3}\right\}$. We break the proof into two cases: when 1) $L_m(\theta) \geq 1$ and 2) $L_m(\theta) \in (0, 1)$ and

Case 1 ($L_m(\theta) \geq 1$): We have that,

$$\begin{aligned} & L_m(\theta) \leq L(\theta) \quad \text{and} \quad L_m(\theta) \geq 1 \\ & \Leftrightarrow (L_m(\theta) - L(\theta))(L_m(\theta) - 1) \leq 0 \\ & \Leftrightarrow L_m(\theta)^2 - L_m(\theta) - L(\theta)L_m(\theta) - L(\theta) \leq 0 \\ & \Leftrightarrow L_m(\theta) - L(\theta) - 1 \leq -\frac{L(\theta)}{L_m(\theta)} \\ & \Leftrightarrow L(\theta) - L_m(\theta) + 1 \geq \frac{L(\theta)}{L_m(\theta)} \\ & \Leftrightarrow \log(L(\theta) - L_m(\theta) + 1) \geq \log\left(\frac{L(\theta)}{L_m(\theta)}\right) = \ell(\theta) - \ell_m(\theta) \end{aligned}$$

Which implies,

$$Pr(\ell(\theta) - \ell_m(\theta) > \log[\epsilon + 1]) \leq Pr(\log[L(\theta) - L_m(\theta) + 1] > \log[\epsilon + 1]) \leq \alpha,$$

which is what we wanted.

Case 2 ($L_m(\theta) \in (0, 1)$):

First notice by the complement rule since $Pr(\log[L(\theta) - L_m(\theta) + 1] > \log[\epsilon + 1]) \leq \alpha$,

$$1 \geq Pr(\log[L(\theta) - L_m(\theta) + 1] \leq \log[\epsilon + 1]) \geq 1 - \alpha.$$

Then,

$$\begin{aligned}
&L_m(\theta) \leq L(\theta) \quad \text{and} \quad L_m(\theta) \in (0, 1) \\
&\Leftrightarrow (L_m(\theta) - L(\theta))(L_m(\theta) - 1) \geq 0 \\
&\Leftrightarrow L_m(\theta) - L(\theta) - 1 \geq -\frac{L(\theta)}{L_m(\theta)} \\
&\Leftrightarrow L(\theta) - L_m(\theta) + 1 \leq \frac{L(\theta)}{L_m(\theta)} \\
&\Leftrightarrow \log(L(\theta) - L_m(\theta) + 1) \leq \log\left(\frac{L(\theta)}{L_m(\theta)}\right) = \ell(\theta) - \ell_m(\theta).
\end{aligned}$$

So now we have that,

$$Pr(\ell(\theta) - \ell_m(\theta) \leq \log(\epsilon + 1)) \leq Pr(\log(L(\theta) - L_m(\theta) + 1) \leq \log(\epsilon + 1)).$$

The lower bound probability of $(1 - \alpha)$ can then occur in one of two places. Case i,

$$1 - \alpha \leq Pr(\ell(\theta) - \ell_m(\theta) \leq \log[\epsilon + 1]) \leq Pr(\log[L(\theta) - L_m(\theta) + 1] \leq \log[\epsilon + 1]),$$

or Case ii,

$$Pr(\ell(\theta) - \ell_m(\theta) \leq \log[\epsilon + 1]) \leq 1 - \alpha \leq Pr(\log[L(\theta) - L_m(\theta) + 1] \leq \log[\epsilon + 1]).$$

If Case i is true, then we again have the result of Case 1. If Case ii is true, then we have that,

$$Pr(\ell(\theta) - \ell_m(\theta) > \log[\epsilon + 1]) \geq \alpha.$$

That says, at minimum, the probability that this difference in the log functions exceeds $\log(\epsilon + 1)$ is α. Therefore the minimum correction necessary is when ϵ is a function of

α. The largest correction would be when $\alpha = 1$. Case ii occurs when the scale of $L(\theta)$ and $L_m(\theta)$ is quite small, in which $L(\theta)/L_m(\theta) >> L(\theta) - L_m(\theta)$. This is more likely in situations when the dimension of the random effects is large.

Across Case 1 and Case 2 (with sub-cases i and ii), using the ϵ we found in (A.3), we can shift the log likelihood estimate up by $\log(\epsilon + 1)$. □

A.2 Proof of Theorem 3.3.4

Proof. We use Taylor's theorem.

Define $L_{j,m} = \bar{G}_j = \frac{1}{m}\sum_{t=1}^{m} \frac{f_{\theta_j}(y,u_t)}{\tilde{f}(u_t)}$. Then, $\ell_{j,m} = \log(\bar{G}_j)$, or all together,

$$g\begin{pmatrix}\bar{G}_1\\ \bar{G}_2\\ \vdots\\ \bar{G}_k\end{pmatrix} = \begin{pmatrix}\log(\bar{G}_1)\\ \log(\bar{G}_2)\\ \vdots\\ \log(\bar{G}_k)\end{pmatrix}_{k\times 1}. \tag{A.7}$$

For a general $k-$dimensional vector this implies,

$$g\begin{pmatrix}a_1\\ a_2\\ \vdots\\ a_k\end{pmatrix} = \begin{pmatrix}\log(a_1)\\ \log(a_2)\\ \vdots\\ \log(a_k)\end{pmatrix} \Rightarrow \nabla g = \begin{pmatrix}1/a_1 & 0 & \cdots & 0\\ 0 & 1/a_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & 1/a_k\end{pmatrix}_{k\times k}.$$

The second derivative, H, is a $k \times k \times k$ array, where if $g_1 = \log(a_1), \ldots, g_k = \log(a_k)$ then $H = (H_1, H_2, \ldots, H_k)^T$ where,

$$H_1 = H(g_1) = \begin{pmatrix}\frac{\partial^2 g_1}{\partial a_1^2} & \frac{\partial^2 g_1}{\partial a_1 a_2} & \cdots & \frac{\partial^2 g_1}{\partial a_1 a_k}\\ \frac{\partial^2 g_1}{\partial a_2 a_1} & \frac{\partial^2 g_1}{\partial a_2^2} & \cdots & \frac{\partial^2 g_1}{\partial a_2 a_k}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{\partial^2 g_1}{\partial a_k a_1} & \frac{\partial^2 g_1}{\partial a_k a_2} & \cdots & \frac{\partial^2 g_1}{\partial a_k^2}\end{pmatrix} = \begin{pmatrix}-1/a_1^2 & 0 & \cdots & 0\\ 0 & 0 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & 0\end{pmatrix}_{k\times k}.$$

Let $\bar{G} = (\bar{G}_1, \ldots, \bar{G}_k)^T, L = (L_1, \ldots, L_k)^T, \ell_m = (\log(\bar{G}_1), \ldots, \log(\bar{G}_k))^T$, and $\ell = (\ell_1, \ldots, \ell_k)^T$ where all are k-dimensional vectors. Then,

$$\ell_m = g(\bar{G}) = \underbrace{g(L)}_{k\times 1} + \underbrace{\nabla g(L)}_{k\times k}\underbrace{(\bar{G}-L)}_{k\times 1} + \frac{1}{2}\underbrace{\left\{\underbrace{I_k}_{k\times k} \otimes \underbrace{(\bar{G}-L)^T}_{1\times k}\right\}}_{k\times k^2}\underbrace{H}_{k^2\times k}\underbrace{(\bar{G}-L)}_{k\times 1} + O(m^{-2}). \tag{A.8}$$

Then taking the expectation with respect to $\tilde{F}$,

$$\begin{aligned}\mathbb{E}_{\tilde{F}}[\ell_m] &= \ell + \nabla g(L)\underbrace{\mathbb{E}_{\tilde{F}}[\bar{G}-L]}_{=0} + \frac{1}{2}\mathbb{E}_{\tilde{F}}[\left\{I_k \otimes (\bar{G}-L)^T\right\} H(\bar{G}-L)] + O(m^{-2}) \\ &= \ell + \frac{1}{2}\mathbb{E}_{\tilde{F}}[\left\{I_k \otimes (\bar{G}-L)^T\right\} H(\bar{G}-L)] + O(m^{-2}).\end{aligned}$$

Let $0_k = (0, \ldots, 0)^T \in \mathbb{R}^k$, then

$$\left\{I_k \otimes (\bar{G}-L)^T\right\} H(\bar{G}-L) = \begin{pmatrix} (\bar{G}-L)^T & 0_k & \cdots & 0_k \\ 0_k & (\bar{G}-L)^T & \cdots & 0_k \\ \vdots & \vdots & \ddots & \vdots \\ 0_k & 0_k & \cdots & (\bar{G}-L)^T \end{pmatrix} \times$$
$$\begin{pmatrix} -1/L_1^2 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & -1/L_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \end{pmatrix} \begin{pmatrix} \bar{G}_1 - L_1 \\ \bar{G}_2 - L_2 \\ \vdots \\ \bar{G}_k - L_k \end{pmatrix}$$

$$= \begin{pmatrix} -\frac{(\bar{G}_1 - L_1)}{L_1^2} & 0 & \cdots & 0 \\ 0 & -\frac{(\bar{G}_2 - L_2)}{L_2^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -\frac{(\bar{G}_k - L_k)}{L_k^2} \end{pmatrix} \begin{pmatrix} \bar{G}_1 - L_1 \\ \bar{G}_2 - L_2 \\ \vdots \\ \bar{G}_k - L_k \end{pmatrix}$$

$$= \begin{pmatrix} -\frac{(\bar{G}_1 - L_1)^2}{L_1^2} \\ -\frac{(\bar{G}_2 - L_2)^2}{L_2^2} \\ \vdots \\ -\frac{(\bar{G}_k - L_k)^2}{L_k^2} \end{pmatrix}.$$

By row we have,

$$\begin{aligned} \mathbb{E}_{\tilde{F}}[\ell_{j,m}] &= \ell_j + \frac{1}{2}\mathbb{E}_{\tilde{F}}\left[-\frac{(\bar{G}_j - L_j)^2}{L_j^2}\right] + O(m^{-2}) \\ &= \ell_j - \frac{1}{2L_j^2}\left\{\mathbb{E}_{\tilde{F}}[\bar{G}_j^2] - 2L_j\mathbb{E}_{\tilde{F}}[\bar{G}_j] + L_j^2\right\} + O(m^{-2}) \\ &= \ell_j - \frac{1}{2L_j^2}\left\{Var_{\tilde{F}}(\bar{G}_j) + L_j^2 - 2L_j^2 + L_j^2\right\} + O(m^{-2}) \\ &= \ell_j - \frac{Var_{\tilde{F}}(\bar{G}_j)}{2L_j^2} + O(m^{-2}) \end{aligned}$$

where $\frac{Var_{\tilde{F}}(\bar{G}_j)}{2L_j^2} > 0$ which shows we are indeed underestimating the true log likelihood, but this bias diminishes as the Monte Carlo sample size goes to infinity since $Var_{\tilde{F}}(\bar{G}_j) \to 0$ as $m \to \infty$. □

A.3 Proof of Theorem 3.4.1

Proof.

$$\hat{L}_m(\theta|y) = \frac{\frac{1}{m}\sum_{t=1}^{m}\frac{f_\theta(y|U_t)f_\theta(U_t)}{\tilde{f}(U_t)}}{\frac{1}{m}\sum_{t=1}^{m}\frac{f_\theta(U_t)}{\tilde{f}(U_t)}} = \frac{\frac{1}{m}\sum_{t=1}^{m}f_\theta(y|U_t)w_\theta(U_t)}{\frac{1}{m}\sum_{t=1}^{m}w_\theta(U_t)} = \frac{1}{m}\sum_{t=1}^{m}f_\theta(y|U_t)\frac{w_\theta(U_t)}{\bar{w}_\theta}.$$

Again we have that for all y, θ, $f_\theta(y|u) \in [0,1]$ so this implies,

$$\frac{1}{m}\sum_{t=1}^{m} f_\theta(y|U_t)\frac{w_\theta(U_t)}{\bar{w}_\theta} \leq \frac{1}{m}\sum_{t=1}^{m}\frac{w_\theta(U_t)}{\bar{w}_\theta} = \frac{1}{m}\sum_{t=1}^{m}\tilde{w}_\theta(U_t).$$

Where $\tilde{w}_\theta(U_t) \leq \sup_\theta \tilde{w}_\theta(U) = \tilde{M} < \infty$. And,

$$Var_{\tilde{F}}(f_\theta(y|u)\tilde{w}_\theta(u)) \leq Var_{\tilde{F}}(\tilde{w}_\theta(u)) \leq \sup_{\theta\in\Theta} Var_{\tilde{F}}(\tilde{w}_\theta(u)) = \tilde{\tau}^2 < \infty.$$

Again we can use Bernstein's inequality. The remainder of the proof follows identically to the the proof of Theorem 3.3.3 except using $\tilde{M}$ and $\tilde{\tau}^2$ instead. □

A.3.1 Proof of Corollary 3.4.1.1

Proof. From proof A.3, by rearranging to solve for m instead of $\tilde{\epsilon}$ we have,

$$2\exp\left\{\frac{-m\tilde{\epsilon}^2}{2\tilde{\tau}^2 + 2\tilde{M}\tilde{\epsilon}/3}\right\} = \alpha \tag{A.9}$$

$$\iff -m\tilde{\epsilon}^2 = \log(\alpha/2)(2\tilde{\tau}^2 + 2\tilde{M}\tilde{\epsilon}/3) \tag{A.10}$$

$$\iff m = -\tilde{\epsilon}^{-2}\log(\alpha/2)(2\tilde{\tau}^2 + 2\tilde{M}\tilde{\epsilon}/3). \tag{A.11}$$

□

A.3.2 Proof of Corollary 3.4.1.2

Proof. Again, using the same procedure as in Proof A.1.2, except replacing M and τ^2 with $\tilde{M}$ and $\tilde{\tau}^2$ respectively, we have that if $\hat{L}_m(\theta) \geq 1$ (Case 1) that,

$$Pr(\ell(\theta) - \hat{\ell}_m(\theta) > \log[\tilde{\epsilon}+1]) \leq 2\exp\left\{\frac{-m\tilde{\epsilon}^2}{2\tilde{\tau}^2 + 2\tilde{M}\tilde{\epsilon}/3}\right\}. \tag{A.12}$$

Otherwise if (Case 2.i.) $\hat{L}_m(\theta) \in (0,1)$ and

$$1-\alpha \leq Pr(\ell(\theta) - \hat{\ell}_m(\theta) \leq \log[\tilde{\epsilon}+1]) \leq Pr(\log[L(\theta) - \hat{L}_m(\theta) + 1] \leq \log[\tilde{\epsilon}+1]),$$

then the results of Case 1 still hold. Otherwise if (Case 2.ii.) $\hat{L}_m(\theta) \in (0,1)$ and

$$Pr(\ell(\theta) - \hat{\ell}_m(\theta) \leq \log[\tilde{\epsilon}+1]) \leq 1-\alpha \leq Pr(\log[L(\theta) - \hat{L}_m(\theta)+1] \leq \log[\tilde{\epsilon}+1]),$$

then,

$$Pr(\ell(\theta) - \hat{\ell}_m(\theta) > \log[\tilde{\epsilon}+1]) \geq 2\exp\left\{\frac{-m\tilde{\epsilon}^2}{2\tilde{\tau}^2 + 2\tilde{M}\tilde{\epsilon}/3}\right\}.$$

□

A.4 Proof of Theorem 3.4.2

Proof. As described in the WIS bands Section 3.4.1,

$$\sqrt{m}\left(\begin{pmatrix}\bar{G}_1/\bar{W}_1\\ \vdots \\ \bar{G}_k/\bar{W}_k\end{pmatrix} - \begin{pmatrix}L_1\\ \vdots \\ L_k\end{pmatrix}\right) \xrightarrow{d} N_k\left(\begin{pmatrix}0\\ \vdots \\ 0\end{pmatrix}, \nabla g^T \Sigma^* \nabla g\right).$$

Define ∇g and H as in proof A.2. If we take a Taylor approximation where $\hat{L}_{j,m} = \bar{G}_j/\bar{W}_j$ and $\hat{L}_m = (\hat{L}_{1,m}, \ldots, \hat{L}_{k,m})^T$,

$$\hat{\ell}_m = g_{DM}(\hat{L}_m) = g_{DM}(L) + \nabla_{DM}^T(\hat{L}_m - L) + \frac{1}{2}\left\{I_k \otimes (\hat{L}_m - L)^T\right\} H(\hat{L}_m - L) + O(m^{-2}).$$

Taking the expectation with respect to $\tilde{F}$,

$$\mathbb{E}_{\tilde{F}}[\hat{\ell}_m] = \ell + \nabla_{DM}^T \mathbb{E}_{\tilde{F}}(\hat{L}_m - L) + \frac{1}{2}\mathbb{E}_{\tilde{F}}[\left\{I_k \otimes (\hat{L}_m - L)^T\right\} H(\hat{L}_m - L)] + O(m^{-2}).$$

Recall the WIS estimator is biased for L_j. Using Proposition 1.5.3 this means,

$$\mathbb{E}_{\tilde{F}}(\hat{L}_{j,m} - L_j) = m^{-1}(L_j\tau_j^2 - \rho_{jj}\sigma_j\tau_j).$$

In the second order term,

$$\left\{I_k \otimes (\hat{L}_m - L)^T\right\} H(\hat{L}_m - L) = \begin{pmatrix} -\frac{(\hat{L}_{1,m}-L_1)^2}{L_1^2} \\ -\frac{(\hat{L}_{2,m}-L_2)^2}{L_2^2} \\ \vdots \\ -\frac{(\hat{L}_{k,m}-L_k)^2}{L_k^2} \end{pmatrix}.$$

then the row-wise expectation is,

$$\mathbb{E}_{\tilde{F}}\left[-\frac{(\hat{L}_{j,m} - L_j)^2}{L_j^2}\right] = -\frac{1}{L_j^2}\mathbb{E}_{\tilde{F}}[(\hat{L}_{j,m} - L_j)^2] = -\frac{1}{L_j^2} Var_{\tilde{F}}(\hat{L}_{j,m}).$$

Putting this together we have,

$$\begin{aligned}
\mathbb{E}_{\tilde{F}}[\hat{\ell}_{j,m}] &= \ell_j + \frac{1}{mL_j}(L_j\tau_j^2 - \rho_{jj}\sigma_j\tau_j) - \frac{1}{2L_j^2} Var_{\tilde{F}}(\hat{L}_{j,m}) + O(m^{-2}) \\
&= \ell_j + \frac{1}{mL_j}(L_j\tau_j^2 - \rho_{jj}\sigma_j\tau_j) - \frac{1}{2mL_j^2}(\sigma_j^2 - 2L_j\rho_{jj}\sigma_j\tau_j + L_j^2\tau_j^2) + O(m^{-2}) \\
&= \ell_j + \frac{\tau_j^2}{m} - \frac{\rho_{jj}\sigma_j\tau_j}{mL_j} - \frac{\sigma_j^2}{2mL_j^2} + \frac{\rho_{jj}\sigma_j\tau_j}{mL_j} - \frac{\tau_j^2}{2m} + O(m^{-2}) \\
&= \ell_j + \frac{1}{2m}\left(\tau_j^2 - \frac{\sigma_j^2}{L_j^2}\right) + O(m^{-2}).
\end{aligned}$$

□

Appendix B

Acronyms

Acronyms in this thesis

B.1 Acronyms

Table B.1: Acronyms

Acronym	Meaning
CLT	Central Limit Theorem
IS	Importance Sampling
MCMC	Markov Chain Monte Carlo
MCLA	Monte Carlo Likelhood Approximation
OSN	Online Social Network
SIS	Simple Importance Sampling
WIS	Weighted Importance Sampling