Spring 5-15-2019

Topics in Complex and Large-scale Data Analysis

Guanshengrui Hao
Washington University in St. Louis

WASHINGTON UNIVERSITY IN ST. LOUIS

Department of Mathematics

Dissertation Examination Committee:

Nan Lin, Chair

Likai Chen

Jimin Ding

Jose Figueroa-Lopez

Jingqin Luo

Topics in Complex and Large-scale Data Analysis

by

Guanshengrui Hao

A dissertation presented to the
Graduate School of Arts & Sciences
of Washington University in
partial fulfillment of the
requirements for the degree
of Doctor of Philosophy

May 2019

St. Louis, Missouri

Table of Contents

Page

List of Figures

List of Tables

Acknowledgments

Firstly and foremost, I would like to express my sincere gratitude to my advisor Professor Nan Lin, for his patience, motivation, and encouragement, to guide me through all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study.

Secondly, I would also like to thank the rest of my dissertation committee: Professor Likai Chen, Professor Jimin Ding, Professor Jose Figueroa-Lopez, and Professor Jingqin Luo, for their insightful comments and suggestions. I would like to thank Professor Xiang Tang, Professor Ari Stern, Professor Todd Kuffner, Professor Siddhartha Chib and Professor Kilian Weinberger, for teaching me statistics, mathematics, and machine learning, which laid a solid foundation for my further research and career.

Last but not least, I would like to thank my beloved wife, Wensi Zhang, for her persistent understanding, support, and for always being with me.

Guanshengrui Hao

Washington University in St. Louis

May 2019

Dedicated to My unborn Baby.

ABSTRACT OF THE DISSERTATION

Topics in Complex and Large-scale Data Analysis

by

Hao, Guanshengrui

Doctor of Philosophy in Mathematics,

Washington University in St. Louis, 2019.

Professor Nan Lin, Chair

Past few decades have witnessed skyrocketed development of modern technologies. As a result, data collected from modern technologies are evolving towards a direction with more complicated structure and larger scale, driving the traditional data analysis methods to develop and adapt. In this dissertation, we study three statistical issues rising in data with complicated structure and/or in large scale. In Chapter 2, we propose a Bayesian framework via exponential random graph models (ERGM) to estimate the model parameters and network structures for networks with measurement errors; In Chapter 3, we design a novel network sampling algorithm for large-scale networks with community structure; In Chapter 4, we introduce a proper framework to conduct discrete large-scale hypothesis testing procedure based on local false discovery rate (FDR). The performances of our procedures are evaluated through various simulations and real applications, while necessary theoretical properties are carefully studied as well.

1. Introduction

In this chapter, we introduce the background of three statistical issues rising in data with complicated structure and/or in large scale. Challenges brought by the issues are described, which serve as motivation of this disseration.

1.1 Network with measurement errors

During past few decades, network data have emerged explosively in many scientific fields such as biology, computer science, physics, sociology, economics, etc [1, 2, 3, 4, 5]. In such settings, the network structural relationships among the data instances are either themselves important or must be accounted for in an integrated analysis.

Many networks contain erroneous links due to measurement error. For instance, a gene regulatory network constructed from testing certain associations based on expression levels will include erroneous links due to type-I and type-II errors of the tests [6]. Such impacts have been explored in many theoretical studies [7, 8] as well as by simulation [9, 10].

While the impact of measurement errors is widely recognized [7, 8, 9, 10], accommodating it in real-data analysis is still challenging, partly due to the fact that relatively few formal probabilistic analyses exist for characterizing the propagation of errors [11]. [8] suggests to develop robust data analytic techniques to minimize the effects of measurement errors in social networks. A few works address the problem from different aspects. [12] focuses on stochastic networks that are evolving over time and propose a model-based approach to infer latent

time-specific topologies of evolving networks from observations. [13] proposes a probabilistic framework to recover the latent social network structure based on observational conversational data. [14] targets on the quantity/quality trade-off for the inference on erroneously observed graphs. Recently, [7] proposes a general nonparametric denoising approach using spectral decomposition to correct the impact caused by measurement errors to the summary statistics. [11] further shows that under certain assumptions, the distribution of discrepancy in summary statistics for networks with and without measurement errors can be approximated by a Skellam distribution.

Unlike previous nonparametric approaches, we consider a parametric setup and aim for network inference, with details discussed in Chapter 2. We model the network by the exponential random graph model (ERGM), which has been widely used in recent years and shown to be a good choice for network description and statistical inference [15, 16, 17, 18, 19]. A Gibbs sampler is constructed, which allows us to draw samples of "true" networks and model parameters, thus obtain the estimates of both summary statistics and model parameters. Simulation results show that through our approach, we can not only correct the impact caused by the measurement error effectively, but also obtain a good estimate of the model parameters.

1.2 Sampling large-scale networks with community structure

Sampling, as a fundamental statistical technique, aims at extracting a representative subset of the entire population, so that the characteristics of interest can be accurately estimated using the subset. It is applied when the cost of analyzing entire population is high and the accessibility is limited. As a result, when it comes to analyze large-scale networks, e.g., Facebook, Twitter, etc., where billions of users are actively interacted but only limited access can be granted due to

all kinds of policies and restrictions, sampling is inevitable. By the nature of network sampling procedures, in which nodes of networks are visited one-by-one via links in between, network samplers are often called crawlers.

Roughly, crawling techniques can be divided into two categories, (i) graph traversal techniques and (ii) random walks (RWs) [20]. Graph traversal techniques, including Breath-First Search (BFS), Depth-First Search (DFS), Forest Fire (FF), etc., visit each node only once, and they differ with each other only by the order they visit the nodes. Though extensively used [21, 22, 23], it has been shown that samples crawled from graph traversal techniques in general are biased and therefore cannot represent the entire network [24]. On the other hand, RWs allow node re-visiting. [25] provides a thorough survey. In particular, the probability for a node to be visited by a RW crawler is proportional to its degree. It means that the RWs are still biased, but the bias is statistically tractable. To extract bias-free samples so that each node is sampled with equal probability, [20] proposes to modify the RW crawler by a Metropolis filter to create a Metropolis-Hastings random walk (MHRW) crawler.

This dissertation particularly focuses on networks with community structure. Community structure is ubiquitous among networks [26], and one fundamental but important signature of community structure is that nodes within each community is more densely connected than those between different communities. The communities could be some virtual groups, like *Linkedin Groups*, *LEGO IDEAS*, etc., or a groups of nodes sharing the same nodal attribute which implicitly fosters them to connect densely together, or even groups detected by certain clustering algorithms. In this dissertation, we focus on the community structures in which communities are mutually exclusive with each other.

To analyze large-scale networks with community structure, sampling techniques are inevitable, as the large sizes and access limitations make it difficult or even impossible to load the whole networks and analyze them [27]. And for many scenarios, bias-free samples are desired, meaning that nodes from each community should be sampled with equal probability. Directly applying the RW crawler will result in sampling bias towards communities with large volumes (total number of links with at least one end belonging to the community). The MHRW crawler could provide bias-free-sampled nodes, but when it comes to communities, the probability for each community to be sampled will be proportional to its size (total number of nodes within a community). Since communities may vary a lot in scale [26], samples obtained from the MHRW crawler are still biased.

As we can see, the RW crawler and its remedy, the MHRW crawler, cannot provide bias-free samples of communities. In Chapter 3, we design a community-volume-adjusted random walk (CRW) crawler that can fulfill the task under the condition that the volume of each community is known. In real applications where such condition does not hold, an adaptive version of the CRW crawler is introduced so that when crawled long enough, it will converge to its un-adaptive counterpart. We theoretically prove that for certain types of networks and community structures, the CRW crawler can traverse across different communities faster than the RW crawler. Simulation studies are conducted to compare the performances of the CRW crawler and the RW crawler on synthetic networks.

1.3 Discrete large-scale hypothesis testing based on local FDR

Driven by the rapid development of high-throughput technologies, large-scale hypothesis testing, where thousands or even millions of tests are conducted simultaneously, has become

one of common statistical practice [28, 29]. First introduced in [30] and later formally conceptualized in [31] and [32], the false discovery rate (FDR) is shown to be less conservative when compared to the traditional family-wise error rate (FWER), and is thus widely used in large-scale hypothesis testing problems.

In particular, there are two types of FDR, tail area-based FDR (Fdr) and local fdr (fdr). A simple but general Bayesian model [32] would help us clarify the difference. Suppose we conduct m hypotheses $H_1, \ldots, H_m$ simultaneously, with their corresponding test statistics being $Z_1, \ldots, Z_m$. Assume that the m hypotheses are divided into two classes, null or non-null, with prior probabilities π_0 and $\pi_1 = 1 - \pi_0$, respectively. The density and cdf of a test statistic depend on its class, with density being f_0 and cdf F_0 if null, while density being f_1 and cdf F_1 if non-null. Without loss of generality, suppose small values of test statistics provide evidence against the null. Under the above setup, Fdr is given by

$$Fdr(z) = \Pr(\text{null}|Z \leq z) = F_0^+(z)/F(z), \tag{1.1}$$

where $F(z) = \pi_0 F_0(z) + \pi_1 F_1(z)$ and $F_0^+(z) = \pi_0 F_0(z)$. Both the Benjamini-Hochberg FDR procedures [31, 33, 34, 35] and the Storey's q-value methods [36, 37] handle large-scale hypothesis testing problems based on Fdr. On the other hand, fdr, proposed by [32] is defined as

$$fdr(z) = \Pr(\text{null}|Z = z) = f_0^+(z)/f(z), \tag{1.2}$$

where $f(z) = \pi_0 f_0(z) + \pi_1 f_1(z)$ and $f_0^+(z) = \pi_0 f_0(z)$. The density $f(z)$ is called the mixture density, while $f_0^+(z)$ is called the null sub-density. Efron's method [32, 38, 39, 40] estimates $fdr(z)$ by estimating $f_0^+(z)$ and $f(z)$, i.e.

$$\hat{fdr}(z) = \hat{f_0^+}(z)/\hat{f}(z), \tag{1.3}$$

and make rejections based on the estimated local FDRs $\hat{fdr}(z)$. The Fdr and fdr are analytically related by

$$Fdr(z) = \frac{\int_{-\infty}^{z} fdr(Z)f(Z)\mathrm{d}Z}{\int_{-\infty}^{z} f(Z)\mathrm{d}Z} = \mathbb{E}_f\{fdr(Z)|Z \leq z\}. \tag{1.4}$$

Most early Fdr and fdr estimation or control procedures implicitly assume that the test statistics of the large-scale hypothesis testing problem are continuous [31, 32, 36]. The continuity assumption is natural and suitable for data obtained from high-throughput technologies like gene expression microarrays. However, recent skyrocketed development of next-generation sequencing (NGS) technology has revolutionized the genomic research. Presented in the form of discrete read counts at different levels of coverages, NGS data differ from previous data type. Tests needed for such data such as Fisher's exact test (FET) and the Binomial test [41, 42] will produce discrete test statistics and p-values, which violate the continuity assumption. It has been shown that FDR control or estimation procedures without properly addressing the discreteness issue would lead to over-conservative performance [43, 44]. As a result, discrete large-scale hypothesis testing problem is invoked.

Quite a few recent studies [43, 45, 46, 47, 48, 49, 50] have been conducted to adjust the tail area-based FDR control procedures for the discrete large-scale hypothesis testing problems. [51] provides a thorough review and comparison. However, few studies have been done for the local FDR procedures.

Unlike Efron's method, [52] revisits the definition of local fdr in (1.2)

$$fdr(z) = \frac{\pi_0 f_0(z)}{\pi_0 f_0(z) + \pi_1 f_1(z)} = \frac{1}{1 + \frac{\pi_1}{\pi_0}\frac{f_1(z)}{f_0(z)}} \tag{1.5}$$

and proposes the LR method. The LR method first obtains a "rough" local FDR $\tilde{fdr}(z_i)$ for each test H_i,

$$\tilde{fdr}(z_i) = \frac{1}{1 + \frac{\hat{\pi}_1}{\hat{\pi}_0} \frac{\hat{f}_{i1}}{f_{i0}}(z_i)}, \tag{1.6}$$

where $\frac{\hat{f}_{i1}}{f_{i0}}(z_i)$ estimates $\frac{f_{i1}(z_i)}{f_{i0}(z_i)}$ by

$$\frac{\hat{f}_{i1}}{f_{i0}}(z_i) = \frac{\hat{L}(z_i|H_i \text{ is non-null})}{\hat{L}(z_i|H_i \text{ is null})}. \tag{1.7}$$

It then regresses the "rough" local FDRs $\{\tilde{fdr}(z_i)\}_{i=1}^m$ on the $\{z_i\}_{i=1}^m$ by the least trimmed-squares regression [53] to obtain smoothed estimates of local FDRs $\{\hat{fdr}(z_i)\}_{i=1}^m$. The LR method can proceed no matter the test statistics z_i's are continuous or discrete. However, it requires a separate step to estimate the null proportion π_0, as that in (1.5), both $\frac{\pi_1}{\pi_0}$ and $\frac{f_1(z)}{f_0(z)}$ need to be estimated. Moreover, the smoothing procedure using least trimmed-squares regression without any theoretical guidance seems somewhat ad-hoc, and the results in [52] show that when the null proportion π_0 is close to 1, the false discovery rate is not controlled.

[54] proposes a randomized p-value method to convert the discrete p-values to continuous p-values using auxiliary random variables. Such a conversion bridges the discrete and continuous paradigms, so that methods used within the continuous paradigm can be applied to the discrete paradigm under proper adjustment. However, directly applying Fdr and fdr estimation or control procedures to the randomized p-values, like those done in [54] and [55], are incomplete and unstable [46, 48]. Based on the randomized p-value method, [50] has properly adjusted the tail-based FDR control method to discrete large-scale hypothesis testing problems. We will on the other hand provide a formal local FDR estimation procedure in Chapter 4. Section 4.1 and Section 4.2 briefly review Efron's method to estimate local FDR and Habiger's randomized p-value method, respectively. We introduce our method in Section 4.3 to properly

perform discrete large-scale hypothesis testing procedure based on local FDR. Simulation studies are conducted in Section 4.4 to evaluate the performance of our method, compare and make suggestion between Efron's method using the empirical null and the theoretical null.

2. Network with Measurement Errors

2.1 Setup

A network consists of a set of nodes and a set of edges representing the relationship between the nodes. For example, in a scientific co-authorship network, the set of nodes represents the scientists, and two scientists are connected by an edge if they have coauthored a paper [5]. Here we only consider networks that are undirected, unweighted and have no self loops, i.e. an edge connected at both ends to the same node.

We denote a network with n nodes by an $n \times n$ adjacency matrix W, where the (i, j)-th entry $W_{ij} = 1$ if the dyad (i, j) is connected by an edge, and $W_{ij} = 0$ otherwise. Since no self loop is allowed, $W_{ii} = 0$ for all $i = 1, \ldots, n$. Let $\mathscr{W}$ be the space of all possible networks on n nodes. It is easy to see that the size of W is $|\mathscr{W}| = 2^{\binom{n}{2}}$.

When measurement errors are considered, we follow the assumption in [7] that

$$W^{obs} = W^{true} + W^{noise}, \tag{2.1}$$

where W^{obs} denotes the network which is contaminated by measurement errors, W^{true} denotes the "true" realization of some random graph W, and W^{noise} denotes the noise matrix. We assume that W^{true} is from an ERGM with likelihood, for given parameters $\boldsymbol{\theta} = \{\theta_1, \cdots, \theta_K\}^\top$,

$$P(W^{true}|\boldsymbol{\theta}) = \frac{\exp\left[\boldsymbol{\theta}^\top \boldsymbol{s}(W^{true})\right]}{z(\boldsymbol{\theta})}, \tag{2.2}$$

where $\boldsymbol{s}$ are summary statistics describing certain network characteristics, e.g. number of edges, number of triangles, graph diameter, etc., and $z(\boldsymbol{\theta})$ is a constant which only depends on $\boldsymbol{\theta}$, i.e.

$$z(\boldsymbol{\theta}) = \sum_{W \in \mathscr{W}} \exp\left[\boldsymbol{\theta}^\top \boldsymbol{s}(W)\right]. \tag{2.3}$$

For the noise matrix W_{noise}, we assume

$$-W_{ij}^{noise} \sim \text{Bernoulli}(p), \text{ if } W^{true} = 1, \tag{2.4}$$

$$W_{ij}^{noise} \sim \text{Bernoulli}(q), \text{ if } W^{true} = 0, \tag{2.5}$$

$$W_{ij}^{noise} \perp W_{kl}^{noise} \text{ if } (i,j) \neq (k,l) \text{ or } (l,k), \tag{2.6}$$

where p and q are some constants in $(0,1)$. For example, if the network is constructed from hypothesis testing on each dyad (i,j) and the probability to form an edge is assumed to be a constant, p is then the probability of Type-II error (false negative) and q the probability of Type-I error (false positive). In this paper, we assume that p and q are known constants, but in reality, it is possible that they may not be known or not even constants.

To get a glance of the impact of measurement errors under such setups, we consider networks with 100 nodes and choose the summary statistics incorporated in ERGM to be the number of edges, the number of nodes with degree no less than 5 and the geometrically weighted degree (GWD) [56], i.e.

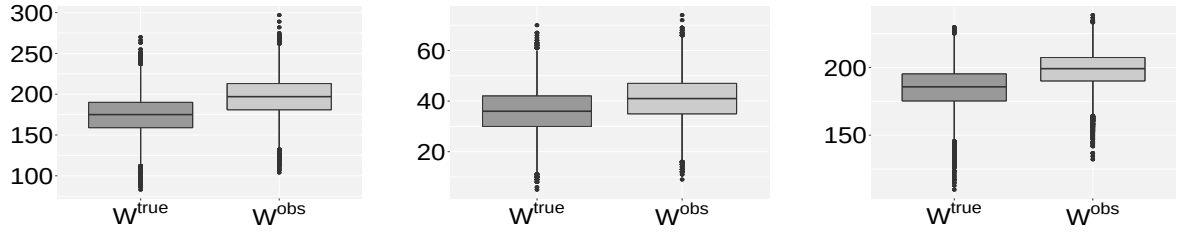
$$s_1(W) = \frac{1}{2}\sum_{i=1}^{n-1} D_i(W), \tag{2.7}$$

$$s_2(W) = \sum_{i=5}^{n-1} D_i(W). \tag{2.8}$$

$$s_3(W) = e^{\theta_s}\sum_{i=1}^{n-1}\left\{1-(1-e^{-\theta_s})^i\right\} D_i(W) \tag{2.9}$$

where $D_i(W)$ denotes the number of nodes in W that have exactly i edges linked to them and θ_s denotes the decay parameter for GWD. [13] suggests to incorporate GWD into the model to

avoid model degeneracy problem. Setting $p = 0.01$, $q = 0.005$ and $\theta_s = (-3, 0.75, -1)^\top$, we can draw multiple networks from (2.2), add noise to each network we drew, and compare the summary statistics before / after adding noise, i.e. $\boldsymbol{s}(W^{true})$ and $\boldsymbol{s}(W^{obs})$. Figure 2.1 shows a big difference in the summary statistics between two groups of networks.



(a) Number of edges: $s_1(W)$ (b) Number of nodes with degree no less than 5: $s_2(W)$ (c) Value of GWD: $s_3(W)$

Figure 2.1.: Boxplots for the summary statistics before / after adding noise.

2.2 Bayesian inference

This section contains the details for the implementation of our Gibbs sampler. Assume a proper prior distribution $\pi(\boldsymbol{\theta})$ on $\boldsymbol{\theta}$, the posterior distribution $f(\boldsymbol{\theta}|W^{obs})$ is then

$$f(\boldsymbol{\theta}|W^{obs}) \propto f(W^{obs}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta}) \int f(W^{obs}|W^{true}, \boldsymbol{\theta}) P(W^{true}|\boldsymbol{\theta}) \mathrm{d}W^{true}. \tag{2.10}$$

Solving the integration in (2.10) involves enumerating all possible $2^{\binom{n}{2}}$ configurations of $W^{true} \in \mathscr{W}$, and becomes intractable even for a moderate value of n.

Instead, the augmented posterior distribution $f(\boldsymbol{\theta}, W^{true}|W^{obs})$ allows derivation of a Gibbs sampler, which samples iteratively from the full conditional distributions, $f(W^{true}|W^{obs}, \boldsymbol{\theta})$ and $f(\boldsymbol{\theta}|W^{obs}, W^{true})$. Throughout this paper, we use $s_1(W)$ to denote the number of edges in W,

which is commonly used as a summary statistic in ERGMs, and θ_1 the corresponding coefficient. Theorem 2.2.1 gives the form of the first full condition distribution $f(W^{true}|W^{obs},\boldsymbol{\theta})$.

Theorem 2.2.1 *Let p and q be the noise constants introduced in (2.4) and (2.5). Denote $\boldsymbol{s}_{-1}(W^{true})$ and $\boldsymbol{\theta}_{-1}$ as the summary statistics for W^{true} and corresponding coefficients excluding $s_1(W^{true})$ and θ_1, respectively. The full condition distribution $f(W^{true}|W^{obs},\boldsymbol{\theta})$ has the following form*

$$f(W^{true}|W^{obs},\boldsymbol{\theta}) \propto \exp\left[\left(\theta_1+\log\frac{1-p}{q}\right)\sum_{W_{ij}^{obs}=1} W_{ij}^{true} + \left(\theta_1+\log\frac{p}{1-q}\right)\sum_{W_{ij}^{obs}=0} W_{ij}^{true}\right]\exp\left[\boldsymbol{\theta}_{-1}^{\top}\boldsymbol{s}_{-1}(W^{true})\right]. \quad (2.11)$$

Proof Recall in Section 2.1, we assume that

$$P(W_{ij}^{obs}=0|W_{ij}^{true}=1) = p \quad (2.12)$$

$$P(W_{ij}^{obs}=1|W_{ij}^{true}=0) = q \quad (2.13)$$

Therefore, the conditional distribution $f(W^{obs}|W^{true})$ can be expressed as

$$\begin{aligned} f(W^{obs}|W^{true}) &= q^{N^+}(1-q)^{M_0-N^+}p^{N^-}(1-p)^{M_1-N^-} \\ &= \exp\left[N^+\log q+(M_0-N^+)\log(1-q)\right. \\ &\quad \left.+N^-\log p+(M_1-N^-)\log(1-p)\right] \end{aligned} \quad (2.14)$$

where M_0 and M_1 denote the number of non-edges and edges in W^{true} respectively, and N^+ and N^- denote the number of $+1$ and -1's in $W^{noise}=W^{obs}-W^{true}$, respectively. In other words, $W_{ij}^{noise}=+1$ means the dyad (i,j) is non-edge in W^{true} but edge in W^{obs}, while $W_{ij}^{noise}=-1$ means it is edge in W^{true} but non-edge in W^{obs}. Meanwhile, we can interpret M_0-N^+ as the

number of dyads which are non-edges in both W^{true} and W^{obs}, while on the other hand $M_1 - N^-$ as the number of dyads which are edges in both. In this way, we can reformulate (2.14) as

$$\begin{aligned} f(W^{obs}|W^{true}) &= \exp\left[\sum_{W_{ij}^{obs}=0} W_{ij}^{true}\log p + \sum_{W_{ij}^{obs}=0}\left(1-W_{ij}^{true}\right)\log(1-q)\right] \\ &\quad \exp\left[\sum_{W_{ij}^{obs}=1}\left(1-W_{ij}^{true}\right)\log q + \sum_{W_{ij}^{obs}=1} W_{ij}^{true}\log(1-p)\right] \\ &= \exp\left[\sum_{W_{ij}^{obs}=0}\left(W_{ij}^{true}\log\frac{p}{1-q} + \log(1-q)\right)\right] \\ &\quad \exp\left[\sum_{W_{ij}^{obs}=1}\left(W_{ij}^{true}\log\frac{1-p}{q} + \log q\right)\right] \end{aligned} \tag{2.15}$$

For simplicity, assume the ERGM only contains one summary statistic, the number of edges $s(W) = \sum_{i<j} W_{ij}$. Then the likelihood function $f(W^{true}|\theta)$ can be written as

$$f(W^{true}|\theta) \propto \exp\left(\theta \sum_{i<j} W_{ij}^{true}\right) \tag{2.16}$$

The dyad set can be divided into two subsets based on W^{obs}: $E_{obs} = \left\{(i,j):\ W_{ij}^{obs} = 1\right\}$ and $E_{obs}^c = \left\{(i,j):\ W_{ij}^{obs} = 0\right\}$, and (2.16) can be rewritten based on this division

$$f(W^{true}|\theta) \propto \exp\left(\theta \sum_{W_{ij}^{obs}=0} W_{ij}^{true} + \theta \sum_{W_{ij}^{obs}=1} W_{ij}^{true}\right). \tag{2.17}$$

Utilizing the Bayes formula, together with (2.15) and (2.17), we have the full conditional distribution $f(W^{true}|W^{obs},\theta)$ expressed by

$$\begin{aligned}
f(W^{true}|W^{obs},\theta) &\propto f(W^{obs}|W^{true},\theta)f(W^{true}|\theta) \\
&= f(W^{obs}|W^{true})f(W^{true}|\theta) \qquad (2.18) \\
&\propto \exp\left[\sum_{W_{ij}^{obs}=0}\left(W_{ij}^{true}\log\frac{p}{1-q}+\log(1-q)\right)\right] \\
&\quad \exp\left[\sum_{W_{ij}^{obs}=1}\left(W_{ij}^{true}\log\frac{1-p}{q}+\log q\right)\right] \\
&\quad \exp\left(\theta\sum_{W_{ij}^{obs}=0}W_{ij}^{true}+\theta\sum_{W_{ij}^{obs}=1}W_{ij}^{true}\right) \qquad (2.19) \\
&\propto \exp\left[\left(\theta+\log\frac{p}{1-q}\right)\sum_{W_{ij}^{obs}=0}W_{ij}^{true}\right. \\
&\quad \left.+\left(\theta+\log\frac{1-p}{q}\right)\sum_{W_{ij}^{obs}=0}W_{ij}^{true}\right] \qquad (2.20)
\end{aligned}$$

which is just (2.11) when $s(W)$ is the number of edges. For models with other summary statistics, we can see through the derivation above, other summary statistics will not be affected. Therefore, we obtain (2.11). ■

If the number of edges is not considered as a summary statistic in the ERGM, we can set $\theta_1 = 0$ as a constant and still represent $f(W^{true}|W^{obs},\boldsymbol{\theta})$ by (2.11). Notice that (2.11) is in the form of an ERGM, so we can sample from (2.11) using the existing sampling methods for ERGMs, e.g. the TNT (tie / no tie) sampler introduced in [57].

The other full conditional distribution $f(\boldsymbol{\theta}|W^{obs}, W^{true})$ can be simplified as $f(\boldsymbol{\theta}|W^{true})$, because the model parameter $\boldsymbol{\theta}$ is independent of W^{obs} given W^{true}. With the prior distribution $\pi(\boldsymbol{\theta})$, we have

$$f(\boldsymbol{\theta}|W^{true}) \propto P(W^{true}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}). \tag{2.21}$$

To sample $\boldsymbol{\theta}$ from $f(\boldsymbol{\theta}|W^{obs}, W^{true})$ is then equivalent to fitting W^{true} into the ERGM in a Bayesian framework [58].

Our Gibbs sampler contains the following steps.

Algorithm *I*

1. Initialize $W^{true,0}$ and $\boldsymbol{\theta}^0$;
2. For $t = 1, 2, \ldots$, in the t-th step,
   a. Draw $W^{true,t+1}$ from $f(\cdot|W^{obs}, \boldsymbol{\theta}^t)$,
   b. Draw $\boldsymbol{\theta}^{t+1}$ from $f(\cdot|W^{true,t+1})$;
3. Stop when the chain converges.

We next discuss two issues: (1) sampling a network from distribution (2.11) in Step 2*a* of Algorithm *I*, and (2) sampling model parameters $\boldsymbol{\theta}$ in an ERGM in Step 2*b* of Algorithm *I*.

2.2.1 Updating W^{true}

To draw $W^{true,t+1}$ from the full conditional distribution $f(\cdot|W^{obs}, \boldsymbol{\theta})$, we use the TNT sampler suggested by [57]. Our algorithm works through the following steps, as in Algorithm *I*-1.

Algorithm I-1

1. Start with $W^0 = W^t$, where W^t is the network in the t-th iteration in step 2 Algorithm I;

2. Iteratively, in the k-th step with the network W^k,

   a. With an equal probability, do one of the two followings,

      i. Randomly pick a dyad $(i^k, j^k) \in E_k^c$, where E_k^c is the set of non-edges for W^k,

      ii. Randomly pick a dyad $(i^k, j^k) \in E_k$, where E_k is the set of edges for W^k.

   b. Propose a new network W^* constructed by

$$W_{ij}^* = \begin{cases} 1 - W_{ij}^k, & \text{if } (i,j) = (i^k, j^k) \text{ or } (j^k, i^k), \\ W_{ij}^k, & \text{otherwise.} \end{cases}$$

   c. Calculate the acceptance ratio

$$r(W^k, W^*) = \frac{f(W^*|W^{obs}, \boldsymbol{\theta}) q(W^k|W^*)}{f(W^k|W^{obs}, \boldsymbol{\theta}) q(W^*|W^k)},$$

   where $q(W^*|W^k)$ is the probability to draw W^* based on W^k and $q(W^k|W^*)$ is the probability to draw W^k based on W^*,

   d. Accept the proposed move to W^* with probability

$$a(W^k, W^*) = \min\left(1, r(W^k, W^*)\right);$$

3. Stop when the chain converges.

2.2.2 Updating θ

The procedure to draw the model parameters $\boldsymbol{\theta}^{t+1}$ from the full conditional distribution $f(\cdot|W^{true,t+1})$ is equivalent to fitting $W^{true,t+1}$ into the ERGM in a Bayesian framework [58, 59]. It utilizes the exchange algorithm [60], which samples from the augmented distribution

$$P(\boldsymbol{\theta}^*, W^*, \boldsymbol{\theta}^t | W^{true,t+1}) \propto P(W^{true,t+1}|\boldsymbol{\theta}^t)\pi(\boldsymbol{\theta}^t)P(W^*|\boldsymbol{\theta}^*)q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^t), \tag{2.22}$$

where $W^{true,t+1}$ and $\boldsymbol{\theta}^t$ are from the t-th iteration in step 2 of Algorithm I, $P(W^*|\boldsymbol{\theta}^*)$ follows the same distribution as $P(W^{true,t+1}|\boldsymbol{\theta}^t)$, $\pi(\boldsymbol{\theta}^t)$ is the prior distribution for parameter $\boldsymbol{\theta}^t$ and $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^t)$ is the proposal distribution. Appropriately choosing the proposal distribution, e.g. a random walk centered at $\boldsymbol{\theta}^t$, to draw $\boldsymbol{\theta}^*$ based on $\boldsymbol{\theta}^t$, the algorithm can be written in the following steps.

Algorithm I-2

1. Draw $\boldsymbol{\theta}^*$ from $q(\cdot|\boldsymbol{\theta}^k)$;
2. Draw W^* from $P(\cdot|\boldsymbol{\theta}^*)$;
3. Accept the proposed move from $\boldsymbol{\theta}^t$ to $\boldsymbol{\theta}^*$ with probability

$$a(\boldsymbol{\theta}^t, \boldsymbol{\theta}^*) = \min\left(1, \frac{P(W^*|\boldsymbol{\theta}^k)\pi(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}^k|\boldsymbol{\theta}^*)P(W^{true,t+1}|\boldsymbol{\theta}^*)}{P(W^{true,t+1}|\boldsymbol{\theta}^k)\pi(\boldsymbol{\theta}^k)q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^k)P(W^*|\boldsymbol{\theta}^*)}\right). \tag{2.23}$$

Notice that in (2.23), two normalizing constants $z(\boldsymbol{\theta}^*)$ and $z(\boldsymbol{\theta}^t)$ are involved in both the numerator and denominator, hence cancel out. Through Algorithm I-2, we can draw samples from the augmented distribution $P(\boldsymbol{\theta}^*, W^*, \boldsymbol{\theta}^t | W^{true,t+1})$, thus obtain the marginalized estimate of parameters $\boldsymbol{\theta}$. In order to improve mixing, [58] also proposes to use a parallel adaptive direction sampler (ADS) [61, 62], which consists of a collection of chains interacting with one another. The algorithm is implemented in an R package called **Bergm** [63], which contains more details for the implementation of the exchange algorithm and parallel ADS.

2.3 Numerical results

2.3.1 Simulation

In this section, we apply our Gibbs sampler to simulated networks and show that it can correct the impact caused by the measurement errors. In our simulation, we set the number of nodes in the network $n = 100$, and choose the summary statistics in the ERGM to be the number of edges, the number of nodes with degree no less than 5 and the GWD, which have been defined in (2.7), (2.8) and (2.9), respectively. The ERGM in (2.2) is thus

$$P(W|\boldsymbol{\theta}) \propto \exp\left(\theta_1 \frac{1}{2}\sum_{i=1}^{n-1} D_i(W) + \theta_2 \sum_{i=5}^{n-1} D_i(W) + \theta_3 e^{\theta_s} \sum_{1}^{n-1} \left\{1-(1-e^{-\theta_s})^i\right\} D_i(W)\right). \tag{2.24}$$

Set the model parameters as $\boldsymbol{\theta}^0 = (\theta_1^0, \theta_2^0, \theta_3^0)^\top = (-3, 0.75, -1)^\top$, the decay parameter θ_s as 1 fixed, and the noise constants p and q as $p = 0.01$ and $q = 0.005$. We draw 100 networks from (2.24) with $(\theta_1, \theta_2, \theta_3) = (\theta_1^0, \theta_2^0, \theta_3^0)^\top$ and treat them as W^{true}'s. We then obtain a network by adding noise onto each W^{true} and treat it as the network contaminated with measurement errors W^{obs}. We apply our Gibbs sampler to each W^{obs} and run 100 parallel simulations. To implement the Gibbs sampler in Algorithm *I*, we place a vague multivariate normal prior to the model parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)^\top$,

$$\pi(\boldsymbol{\theta}) \sim \mathcal{N}(\mathbf{0}, 15^2 \boldsymbol{I}_3), \tag{2.25}$$

where $\boldsymbol{I}_3$ is the identity matrix with dimension 3. Some pilot explorations suggest us to use three independent random walks for θ_1^k, θ_2^k and θ_3^k separately in step 1 of Algorithm I-2, i.e.

$$q(\cdot|\theta_1^k) \sim \mathcal{N}(\theta_1^k, \sigma_1^2), \tag{2.26}$$

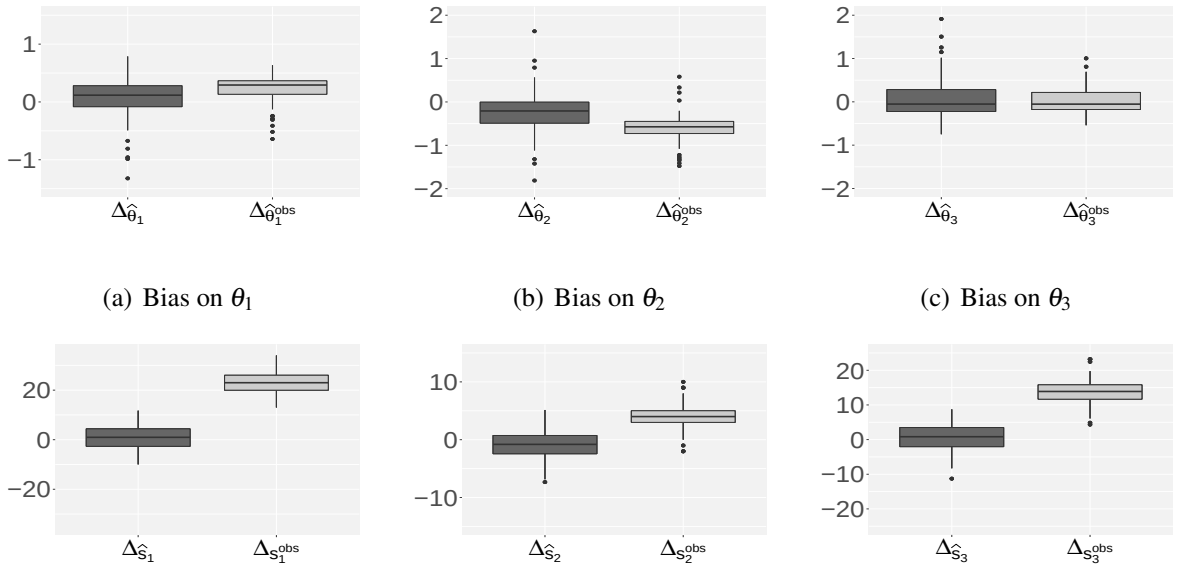
$$q(\cdot|\theta_2^k) \sim \mathcal{N}(\theta_2^k, \sigma_2^2), \tag{2.27}$$

$$q(\cdot|\theta_3^k) \sim \mathcal{N}(\theta_3^k, \sigma_3^2), \tag{2.28}$$

with σ_1, σ_2 and σ_3 all equal to 0.25, which can improve the mixing of the algorithm. We choose the number of iterations to draw $W^{true,t+1}$ in Step 2a of Algorithm I to be 10, which gives an adequate acceptance rate and avoids bringing in more computation burden.

We run $25,000$ MCMC iterations in each simulation, burn in the first $5,000$ and obtain the posterior means for the model parameters $\hat{\boldsymbol{\theta}} = (\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3)^\top$ and the summary statistics $\hat{\boldsymbol{s}} = (\hat{s}_1, \hat{s}_2, \hat{s}_3)^\top$ using the remaining $20,000$. Meanwhile, we also estimate the model parameters and calculate the summary statistics directly based on W^{obs}, denoted by $\hat{\boldsymbol{\theta}}^{obs} = (\hat{\theta}_1^{obs}, \hat{\theta}_2^{obs}, \hat{\theta}_3^{obs})^\top$ and $\boldsymbol{s}^{obs} = (s_1^{obs}, s_2^{obs}, s_3^{obs})^\top$, respectively. We obtain the bias of the model parameters $\Delta_{\hat{\boldsymbol{\theta}}} = \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^0$ and $\Delta_{\hat{\boldsymbol{\theta}}^{obs}} = \hat{\boldsymbol{\theta}}^{obs} - \boldsymbol{\theta}^0$ and obtain the bias of the summary statistics $\Delta_{\hat{\boldsymbol{s}}} = \hat{\boldsymbol{s}} - \boldsymbol{s}^0$ and $\Delta_{\boldsymbol{s}^{obs}} = \boldsymbol{s}^{obs} - \boldsymbol{s}^0$, where $\boldsymbol{s}^0$ is the summary statistics calculated based on W^{true}'s. We compare the difference between $\Delta_{\hat{\boldsymbol{\theta}}}$ and $\Delta_{\hat{\boldsymbol{\theta}}^{obs}}$, $\Delta_{\hat{\boldsymbol{s}}}$ and $\Delta_{\boldsymbol{s}^{obs}}$. Figure 2.2 shows that the biases of the posterior means through our Gibbs sampler are all centered around 0, which implies that our Gibbs sample has the ability to correct the impact caused by the measurement errors. Two sample t-tests on two groups of model parameters and summary statistics are also in favor of that the two groups are significantly different. The results are summarized in Table 2.1.

Out of 100 simulations, we randomly pick one and analyze the performance of our Gibbs sampler. The summary statistics s_1, s_2, s_3 for W^{true} and W^{obs} are summarized in Table 2.2.



(a) Bias on θ_1 (b) Bias on θ_2 (c) Bias on θ_3

(d) Bias on s_1 (e) Bias on s_2 (f) Bias on s_3

Figure 2.2.: Boxplots for the bias of the model parameters and the summary statistics.

Model parameters	p-value	Summary statistics	p-value
θ_1	9.34e-05	s_1	1.07e-84
θ_2	1.10e-07	s_2	4.23e-31
θ_3	5.68e-01	s_3	4.96e-60

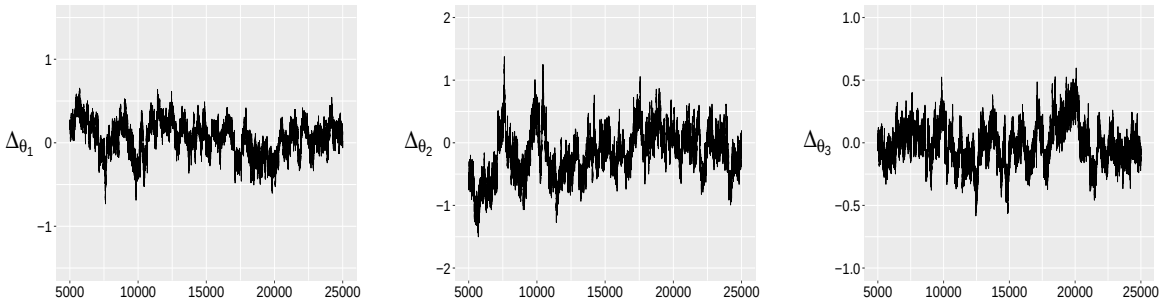
Table 2.1: *P-values of two sample t-tests for comparing the model parameters and summary statistics.*

The traceplots of the bias of the model parameters $\Delta_{\boldsymbol{\theta}^t} = \boldsymbol{\theta}^t - \boldsymbol{\theta}^0$ are shown in Figure 2.3. And Tables 2.3 and 2.4 give the posterior summary for the model parameters and network summary statistics. We also compare three estimates of $\boldsymbol{\theta}$, the estimates based on W^{true}, the estimates based on W^{obs} and the estimates through our Gibbs sampler. The comparison is summarized in

Table 2.5. The estimates based on W^{true} and W^{obs} are obtained using the method in [58]. Tables 2.3, 2.4 and 2.5 show that the impact of the measurement errors have been well corrected by our Gibbs sampler.

Summary statistics	W^{true}	W^{obs}
s_1	176	198
s_2	35	40
s_3	184.91	198.48

Table 2.2: *Summary statistics of* W^{true} *and* W^{obs}.



(a) Bias on θ_1 (b) Bias on θ_2 (c) Bias on θ_3

Figure 2.3.: Traceplots for the bias of the model parameters.

Parameters	Mean	SD	Naive SE	Time-series SE
θ_1	-2.94	0.20	1.42e-03	2.84e-02
θ_2	0.57	0.38	2.70e-03	5.85e-02
θ_3	-1.01	0.16	1.14e-03	2.37e-02

Table 2.3: *Summary for the posterior samples of the model parameters.*

Summary statistics	Mean	SD	Naive SE	Time-series SE	W^{true}	W^{obs}
s_1	177.63	5.13	3.62e-02	0.85	176	198
s_2	35.79	2.75	1.95e-02	0.42	35	40
s_3	186.94	3.37	2.38e-02	0.49	184.91	198.48

Table 2.4: *Summary for the posterior samples of the summary statistics, compared with those calculated based on W^{true} and W^{obs}.*

Parameters	W^{true}	W^{obs}	Posterior Mean
θ_1	-2.83	-2.71	-2.94
θ_2	0.50	0.17	0.57
θ_3	-1.12	-1.09	-1.01

Table 2.5: *Comparison of the posterior mean and model parameters estimated based on W^{true} and W^{obs}.*

2.3.2 Comparison with nonparametric network denoising

Our Bayesian method can also be used for denoising the summary statistics. And we compare with the nonparametric method proposed by [7] through spectral decomposition. First, they construct a näive unbiased estimator of W^{true},

$$\widetilde{W}_{obs} = \frac{W^{obs} - qW_{K_n}}{1-(p+q)}, \tag{2.29}$$

where W_{K_n} is a matrix of ones with zero diagonals. And the nonparametric denoising estimator is then

$$\widehat{W}^r = \sum_{i=1}^{r} \left\langle \phi_i, \widetilde{W}^{obs}\phi_i \right\rangle \phi_i \phi^\top, \tag{2.30}$$

where $\langle \cdot, \cdot \rangle$ is the inner product operator, $\{\mu_i\}_{i=1}^n$ are the eigenvalues of $\widetilde{W}^{obs}$ and $\{\phi_i\}_{i=1}^n$ are the corresponding eigenvectors. $\{\phi_i, \mu_i\}_{i=1}^n$ are ordered decreasingly according to the magnitude of squared eigenvalues $\{\mu_i^2\}_{i=1}^n$. They proved that as the number of nodes $n \to \infty$, the optimal choice for r is $r = 1$.

While nonparametric method requires asymptotic conditions, Bayesian method can do exact inference for finite-sample cases. Therefore, it is worth comparing the performance of the nonparametric estimator with our Bayesian approach for networks with a moderate number of nodes. Notice that through nonparametric denoising, the estimator $\widehat{W}^r$ is no longer 0/1 valued. As a result, we need to extend the definition of the summary statistics we use for $\widehat{W}^r$. According to an alternative definition of s_1

$$s_1(W) = \sum_{i<j} W_{ij},$$

we can analogously define

$$s_1(\widehat{W}^r) = \sum_{i<j} \widehat{W}^r_{ij},. \tag{2.31}$$

for $\widehat{W}^r$, where $\widehat{W}^r_{ij}$ represents the (i,j)-th entry of $\widehat{W}^r$. For s_2, we can extend it for $\widehat{W}^r$ based on its graphical meaning that it represents the number of nodes with degree no less than 5, i.e.

$$s_2(\widehat{W}^r) = \sum_{i=1}^{n} I(d_i(\widehat{W}^r) \geq 5), \tag{2.32}$$

where $d_i(\widehat{W}^r)$ represents the degree of the i-th node of $\widehat{W}^r$ and $I(\cdot)$ is the indicator function. But since the entries of $\widehat{W}^r$ are not 0/1 valued, extending s_2 for $\widehat{W}^r$ in the fashion of (2.32) would tend to underestimate the quantity. Instead, we define

$$s_2(\widehat{W}^r) = (\sum_{i=1}^{n} I(d_i(\widehat{W}^r) \geq 5) + \sum_{i=1}^{n} I(d_i(\widehat{W}^r) > 4))/2, \tag{2.33}$$

which can be treated as a balance between underestimation and overestimation. We will omit the comparison for s_3, since the definition of s_3 in (2.9) requires the degree of the nodes to be integers. That cannot be satisfied for $\widehat{W}^r$ in general when its entries are not 0/1 valued.

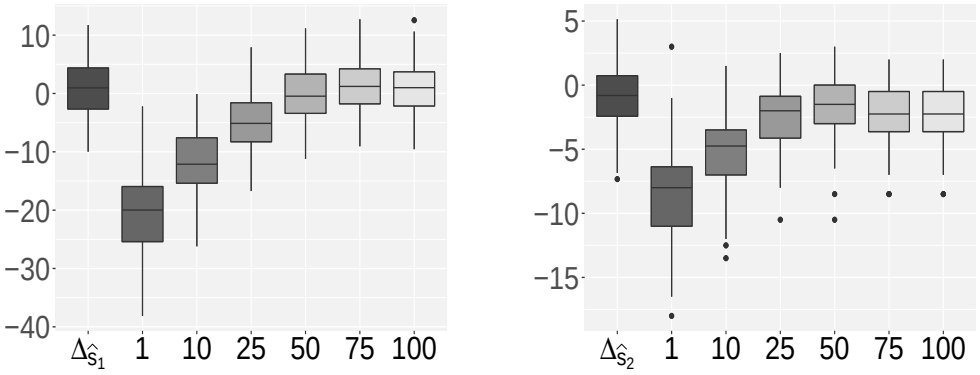
Now consider the 100 simulations we did in Section 2.3.1 again. Within each simulation, we construct the nonparametric estimators of W^{true} by (2.30) with $r = 1, 10, 25, 50, 75, 100$, and then estimate the two summary statistics using the extended definition (2.31) and (2.33). Denote the nonparametric estimates of the summary statistics by $\tilde{\boldsymbol{s}}^r = (\tilde{s}^r_1, \tilde{s}^r_2)^\top$. Similarly, we obtain the bias of the summary statistics $\Delta_{\tilde{\boldsymbol{s}}^r} = \tilde{\boldsymbol{s}}^r - \boldsymbol{s}^0$ for each simulation, where $\boldsymbol{s}^0$ is the summary statistics calculated based on W^{true}. In Figure 2.4, we compare $\Delta_{\tilde{\boldsymbol{s}}^r}$ with $\Delta_{\hat{\boldsymbol{s}}}$ obtained through the Bayesian approach in Section 2.3.1. We also perform two sample t-tests between $\Delta_{\hat{\boldsymbol{s}}}$ and $\Delta_{\tilde{\boldsymbol{s}}^r}$, which is summarized in Table 2.6.

The figures and table show that the asymptotic optimal choice $r = 1$ performs inadequately in the simulation when the number of nodes in the network is only 100. The nonparametric approach achieves comparable results to the Bayesian approach for s_1 only when $r \geq 75$, but always underestimate s_2 even when we take the treatment in (2.33). In real situations when the

number of nodes is not large enough as the asymptotic optimal choice requires, and the truth for the summary statistics are not known, it is not easy to find a proper r. Another restriction for the nonparametric approach, as stated in [7], is that it only works for Lipschitz continuous summary statistics, i.e.

$$|\boldsymbol{s}(W_1) - \boldsymbol{s}(W_2)| \leq C\|W_1 - W_2\|_1.$$

Even for some summary statistics that are indeed Lipschitz continuous, e.g., s_2 in our case, the nonparametric approach may not perform very well. That is mainly caused by the issue that the entries of nonparametric estimator $\widehat{W}_r$ are no longer 0 or 1, which makes it difficult to extend the definition of those summary statistics to $\widehat{W}^r$. As a contrast, our Bayesian approach is applicable for any summary statistics.



(a) Bias on s_1

(b) Bias on s_2

Figure 2.4.: Boxplots for the bias of the summary statistics.

Summary statistics	$r=1$	$r=10$	$r=25$	$r=50$	$r=75$	$r=100$
s_1	6.93e-53	3.71e-38	8.05e-14	0.33	0.41	0.73
s_2	8.45e-40	2.68e-22	8.62e-06	8.03e-03	4.25e-05	4.25e-05

Table 2.6: *P-values of two sample t-tests for comparing summary statistics estimates.*

2.3.3 Sensitivity analysis

In this section we analyze the sensitivity of our Bayesian method to the noise constant p, the probability to erroneously remove an edge from W^{true}. The motivation to perform this analysis is from two aspects. In reality, most networks tend to be sparse, i.e. the number of edges m scales slower than quadratic in the number of nodes n [11]. Formally, [11] assumes that sparse networks follow

$$m = O(n \log n). \tag{2.34}$$

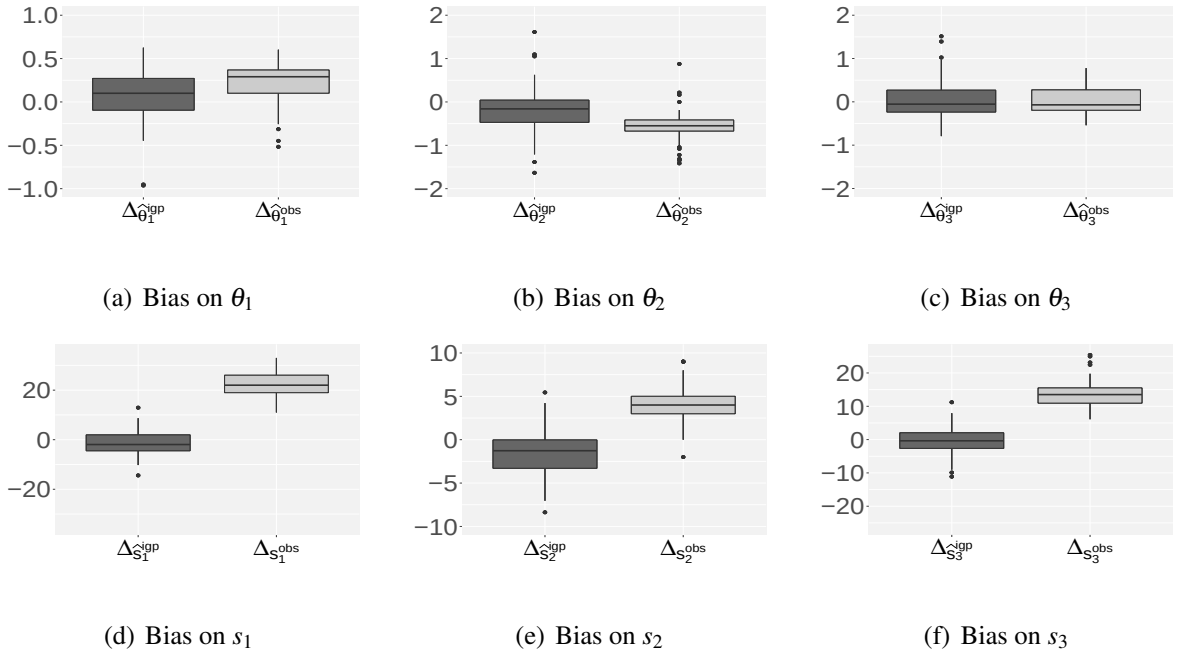
When W^{true} is sparse, and the noise constants p and q are comparable in magnitude, the noise introduced by p, the probability to randomly remove an edge from W^{true}, is more likely to be negligible. On the other hand, real world networks are often constructed from hypothesis testing on each dyad with certain significance level. In such cases, q can be evaluated based on the given significance level, while p is the probability of Type-II error which is often not known. If our Bayesian method is insensitive to the value of p, it will be easier to apply onto the real world sparse networks where p is not known.

To analyze the sensitivity of our Bayesian method in p based on simulation, we consider the same setups as in Section 2.3.1. Figure 2.1 shows that when the true model parameter $\boldsymbol{\theta}^0 = (-3, 0.75, -1)^\top$, the number of edges for the networks drawn from the corresponding

ERGM (2.24) will mostly lie below 300, which satisfies the sparsity assumption (2.34). Therefore, we still draw 100 networks from (2.24) with $\boldsymbol{\theta}^0 = (-3, 0.75, -1)^\top$, treat them as the true underlying networks W^{true}, then add noise onto each of them with same $p = 0.01$, $q = 0.005$ to obtain the observed network with measurement errors W^{obs}. To mimic that we do not know the value of p, we plug 1×10^{-8} instead of the true value 0.01 for p into Algorithm *I*. All the other setups remain the same as in Section 2.3.1. We burn in the first $5,000$ iterations and use the remaining $20,000$ to obtain the posterior mean of model parameters $\hat{\boldsymbol{\theta}}^{igp} = (\hat{\theta}_1^{igp}, \hat{\theta}_2^{igp}, \hat{\theta}_3^{igp})$ and summary statistics $\hat{\boldsymbol{s}}^{igp} = (\hat{s}_1^{igp}, \hat{s}_2^{igp}, \hat{s}_3^{igp})$. Similar as in Section 2.3.1, we compare the bias of the model parameters $\Delta_{\hat{\boldsymbol{\theta}}^{igp}} = \hat{\boldsymbol{\theta}}^{igp} - \boldsymbol{\theta}^0$ and the bias of the summary statistics $\Delta_{\hat{\boldsymbol{s}}^{igp}} = \hat{\boldsymbol{s}}^{igp} - \boldsymbol{s}^0$ with $\Delta_{\hat{\boldsymbol{\theta}}^{obs}} = \hat{\boldsymbol{\theta}}^{obs} - \boldsymbol{\theta}^0$ and $\Delta_{\boldsymbol{s}^{obs}} = \boldsymbol{s}^{obs} - \boldsymbol{s}^0$ respectively, where $\boldsymbol{s}^0$ are the summary statistics of the true underlying network W^{true}, which are shown in Figure 2.5. Compare with Figure 2.3, we can see the performance when treating p close to 0 is comparable to the performance with exact value of p, which means our method is not sensitive to the value of p under such setups.

2.3.4 Empirical results

In this section, we apply our Bayesian method to a real world network with reported measurement error. Consider the regulator-regulator interaction network in [64], which has been fitted into ERGM by [16]. The network consists of 106 nodes and 108 directed links, with each node representing a transcriptional regulator and each directed link representing that the expression of the transcriptional regulator it starts from regulates the expression of the one it points to. Similar to the treatment in [16], we convert the original network into an undirected one by eliminating the direction of the links and removing the self loops . For those pairs



(a) Bias on θ_1 (b) Bias on θ_2 (c) Bias on θ_3

(d) Bias on s_1 (e) Bias on s_2 (f) Bias on s_3

Figure 2.5.: Boxplots for the bias of the model parameters and the summary statistics.

of nodes with links pointing to each other, we eliminate the direction of both links and treat it as only one edge. After the conversion, there are 96 undirected edges left. The converted undirected network is shown in Figure 2.6.

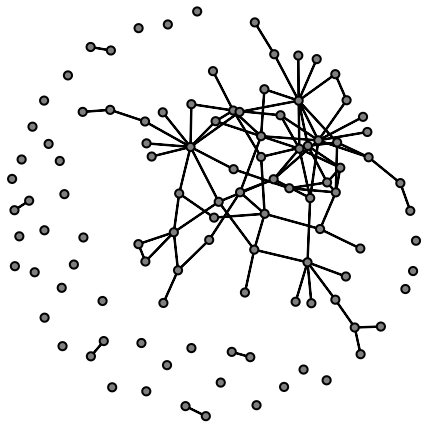


Figure 2.6.: The undirected regulator-regulator interation network.

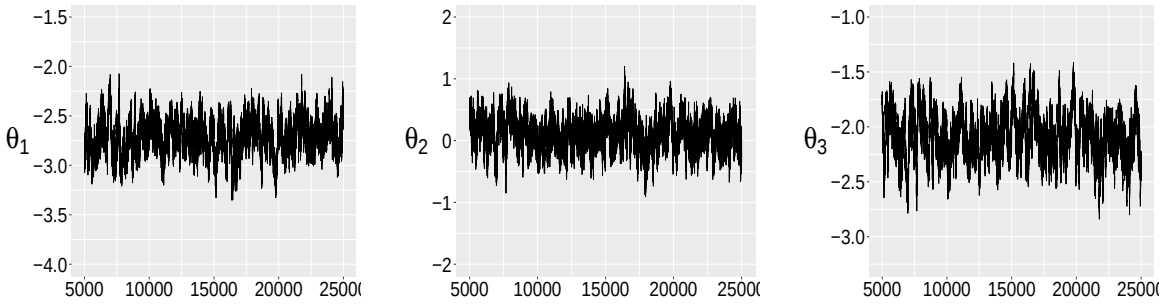
The false positive rate to form a link reported in [64] is 0.001, which corresponds to the value of q in our context. The value of p is not reported, but based on the discussion in Section 2.3.3, we can just set $p = 1 \times 10^{-8}$ and our framework will still apply.

For the summary statistics, [16] suggests to include the GWD and number of edges or the number of 2-stars in the ERGM. Therefore, we can still use the ERGM model in 2.24. To avoid the estimation of decay parameter θ_s in the GWD, we perform a pilot search and find it lying around 0.5. Therefore, we set $\theta_s = 0.5$ fixed. Based on the same implementation setups of the Gibbs sampler in Section 2.3.1, we run $25,000$ iterations in total, burning in first $5,000$ and using the remaining to make inference. Figure 2.7 shows the traceplots of the model parameters. The estimation of the model parameters and the summary statistics are summarized in Tables 2.7 and 2.8, respectively. The results show that the estimated summary statistics are smaller than those from W^{obs}, which indicates effectiveness of our Bayesian treatment. Figure 2.8 compares the heatmaps of adjacency matrix for W^{obs} and $\widehat{W}$, the average of posterior samples of $W^{true,t}$ for $t = 5001, 5002, \ldots, 25000$, i.e.

$$\widehat{W} = \frac{1}{20000} \sum_{t=5001}^{25000} W^{true,t}.$$

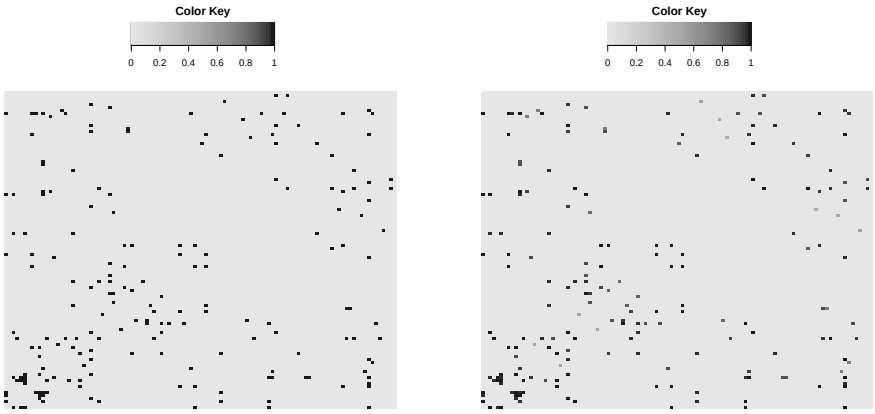
Parameters	Mean	SD	Naive SE	Time-series SE
θ_1	-2.72	0.18	1.26e-03	1.52e-02
θ_2	0.11	0.26	1.82e-03	1.97e-02
θ_3	-2.09	0.21	1.47e-03	1.81e-02

Table 2.7: *Summary for the posterior samples of the model parameters.*



(a) θ_1 (b) θ_2 (c) θ_3

Figure 2.7.: Traceplots for the model parameters.



(a) W^{obs} (b) $\widehat{W}$

Figure 2.8.: Heatmaps of adjacency matrix for W^{obs} and $\widehat{W}$.

Summary statistics	Mean	SD	Naive SE	Time-series SE	W^{obs}
s_1	89.28	2.37	1.68e-02	0.22	96
s_2	11.54	0.64	4.53e-03	5.48e-02	12
s_3	86.22	3.36	2.37e-02	0.31	94.78

Table 2.8: *Summary for the posterior samples of the summary statistics, compared with those calculated based on* W^{obs}.

3. Sampling Large-scale Networks with Community Structure

In this chapter, we propose a novel network crawler for large-scale networks with community structure. We theoretically prove that our crawler can avoid sampling bias for communities so that uniform samples of communities are generated. We also show that under mild assumptions, our crawler can traverse through different communities faster than benchmark random walk crawler. For the purpose of simplicity, the networks we consider in this chapter are all undirected and connected networks.

3.1 Setup

Let W denote a network, with $\mathscr{V}$ being the set of all nodes of W and $\mathscr{E}$ the set of all edges of W. Denote N as the total number of nodes and E the total number of edges in W. For each node $u \in \mathscr{V}$, define the set of all its neighbors by

$$\mathscr{N}(u) = \{v : (u,v) \in \mathscr{E}\}. \tag{3.1}$$

Let the number of neighbors

$$d_u = |\mathscr{N}(u)| \tag{3.2}$$

be the degree of the node u. Suppose there are K mutually exclusive communities on the network W, and each node belongs to one and only on community. Let $g(\cdot)$ be the function that

collects the community index a node belongs to. Define the size of the k-th community as the total number of nodes within the community, i.e.,

$$N_k = |\{u : g(u) = k\}|, \tag{3.3}$$

and define the volume of the k-th community as the number of links with at least one end belonging to the community, which, equivalently, equals to the summation of degrees of the nodes within the community, i.e.

$$V_k = \sum_{g(u)=k} d_u \tag{3.4}$$

3.2 Random walk crawlers and sampling bias

Random walk crawler (RW) is one of the most widely used large-scale network sampling techniques [65, 66, 67, 68, 69] that preserves good statistical properties. From a start node, for each step, the random walk crawler explores all its neighbors and randomly moves to one of its neighbors with equal probability. Formally, an RW crawler can be described in Algorithm *RW*.

Algorithm *RW*

Start from a randomly selected node u_1. Suppose the crawler reaches node u_t at the tth step,

a. Find the set of neighbors $\mathcal{N}(u_t)$ for u_t;

b. With equal probability, randomly select one node $v \in \mathcal{N}(u_t)$, and let

$$u_{t+1} = v.$$

Random walks crawlers are closely related to Markov chains. [25] points out that random walks on connected undirected networks are equivalent to time-reversible Markov chains, and proves that the distribution

$$q_u^{rw} = \frac{d_u}{2E} \tag{3.5}$$

is stationary and unique. Here the distribution q_u for a crawler is stationary is defined as that for each crawling step, the marginal sampling probability for each node u is q_u [25].

Though widely used in practice, one major drawback of the RW crawler is that it is biased towards sampling nodes with large degrees. That is caused by the fact that as the sampling step goes large enough, the probability for each node u to be sampled is proportional to its degree d_u, as the sampling distribution converges to the stationary distribution q_u^{rw}.

One method to correct the sampling bias is to construct a Metropolis-Hastings random walk (MHRW) crawler by applying a Metropolis filter. The crawler is described in Algorithm *MHRW*.

Algorithm *MHRW*

Start from a randomly selected node u_1. Suppose the crawler reaches node u_t at the tth step,

a. Find the set of neighbors $\mathscr{N}(u_t)$ for u_t;

b. With equal probability, randomly select one node $v \in \mathscr{N}(u_t)$;

c. Move the crawler to

$$u_{t+1} = \begin{cases} v, & \text{with probability } \min\left(\frac{d_{u_t}}{d_v}, 1\right), \\ u_t, & \text{with probability } 1 - \min\left(\frac{d_{u_t}}{d_v}, 1\right). \end{cases} \tag{3.6}$$

It follows directly from [70, 71] that the stationary distribution is uniform, i.e.

$$q_u^{mhrw} = \frac{1}{N}, \tag{3.7}$$

so that each node has equal probability to be sampled.

Suppose we use the RW crawler to sample a network with K mutually exclusive communities. Then, marginally, the probability for the t-th sampled node u_t to be from the k-th community is

$$P(g(u_t) = k) = \sum_{u:\ g(u)=k} q_u^{rw} = \frac{\sum_{u:\ g(u)=k} d_u}{2E} = \frac{V_k}{2E}, \tag{3.8}$$

which is proportional to the volume of the k-th community. Similarly, if we use MHRW crawler, then marginally, the probability for the t-th sampled node u_t to be from the k-th community is

$$P(g(u_t) = k) = \sum_{u:\ g(u)=k} q_u^{mhrw} = \frac{\sum_{u:\ g(u)=k} 1}{N} = \frac{N_k}{N}, \tag{3.9}$$

which is proportional to the size of the k-th community. Neither the RW nor the MHRW crawler provides an equal probability for each community to be sampled.

3.3 Cummunity-volume-adjusted random walk crawler

3.3.1 Algorithm

Inspired by the MHRW crawler, where transition probability is adjusted based on the nodal degrees, we can similarly adjust the transition probability based on the volume of communities, so that each community has an equal probability to be sampled. Suppose that we know the volume of each community on the network in advance, then the community-volume-adjusted random walk (CRW) crawler can be described in Algorithm *CRW*.

Algorithm *CRW*

1. Start from a randomly selected node u_1. Suppose the crawler reaches node u_t at the tth step,

   a. Find the set of neighbors $\mathscr{N}(u_t)$ for u_t;

   b. With equal probability, randomly select one node $v \in \mathscr{N}(u_t)$;

   c. Move the crawler to

$$u_{t+1} = \begin{cases} v, & \text{with probability } \min\left(\frac{V_{g(u_t)}}{V_{g(v)}}, 1\right), \\ u_t, & \text{with probability } 1 - \min\left(\frac{V_{g(u_t)}}{V_{g(v)}}, 1\right). \end{cases} \tag{3.10}$$

For the CRW crawler, we have the following theorem.

Theorem 3.3.1 *Given the description of Algorithm CRW, the transition probability to move from a node u to one of its neighbors v is*

$$P(u_{t+1} = v | u_t = u) \stackrel{def}{=} p_{uv} = \frac{1}{d_u} \min\left(\frac{V_{g(u)}}{V_{g(v)}}, 1\right), \tag{3.11}$$

and the stationary distribution for node u is

$$q_u = \frac{1}{K}\frac{d_u}{V_{g(u)}} \tag{3.12}$$

Proof The transition probability

$$\begin{aligned} p_{uv} &= P(\text{choose } v \text{ from } \mathscr{N}(u))P(\text{move to } v) \\ &= \frac{1}{d_u}\min\left(\frac{V_{g(u)}}{V_{g(v)}}, 1\right). \end{aligned}$$

Let $p_u = \frac{1}{K}\frac{d_u}{V_{g(u)}}$, and let $p_v = \frac{1}{K}\frac{d_v}{V_{g(v)}}$, we have

$$p_u p_{uv} = \frac{1}{K}\frac{d_u}{V_{g(u)}} \times \frac{1}{d_u}\min(\frac{V_{g(u)}}{V_{g(v)}}, 1) = \frac{1}{K}\min(\frac{1}{V_{g(v)}}, \frac{1}{V_{g(u)}}),$$

$$p_v p_{vu} = \frac{1}{K}\frac{d_v}{V_{g(v)}} \times \frac{1}{d_v}\min(\frac{V_{g(v)}}{V_{g(u)}}, 1) = \frac{1}{K}\min(\frac{1}{V_{g(u)}}, \frac{1}{V_{g(v)}}).$$

Since $p_u p_{uv} = p_v p_{vu}$, so the stationary distribution of u is $q_u = p_u = \frac{1}{K}\frac{d_u}{V_{g(u)}}$. ■

An instant result follows Theorem 3.3.1 is that the sampling probability for the k-th community is

$$q_k = \sum_{g(u)=k} q_u = \sum_{g(u)=k} \frac{1}{K}\frac{d_u}{V_{g(u)}} = \frac{1}{K}, \tag{3.13}$$

which indicates that the CRW crawler samples each community with equal probability.

3.3.2 Comparison with the RW crawler

In general, it is hard to compare the performances between two different crawlers, as each crawler may suit some types of networks better under certain scenario. We can prove that for certain types of networks, under mild conditions, the CRW crawler performs better than the RW crawler, in the way that on average the CRW crawler traverses through different communities faster than the RW crawler. It can be formulated as a theorem as follows.

Theorem 3.3.2 *Suppose that the large-scale network with exclusive community structure satisfies the following condition,*

$$\sum_{u} \sum_{\substack{v \in \mathcal{N}(u) \\ g(u) \neq g(v)}} \frac{1}{V/K} < \sum_{u} \sum_{\substack{v \in \mathcal{N}(u) \\ g(u) \neq g(v)}} \min\left(\frac{1}{V_{g(v)}}, \frac{1}{V_{g(u)}}\right). \tag{3.14}$$

Then, we claim that on average, for a fixed number of crawling steps, the number of communities visited by the CRW crawler is larger than that of the RW crawler.

Proof It is equivalent to prove that for each crawling step, the probability to move to a different community for the CRW crawler is higher than the RW crawler.

On one hand, for a RW crawler, the probability to move from one community "this" to another community "other" in a single crawling step is

$$\begin{aligned} P(\text{other}|\text{this}, RW) &= \sum_{u} \sum_{v \in \mathcal{N}(u), g(u) \neq g(v)} p_{uv}^{rw} q_u^{rw} \\ &= \sum_{u} \sum_{v \in \mathcal{N}(u), g(u) \neq g(v)} \frac{1}{d_u} \frac{d_u}{V} \\ &= \sum_{u} \sum_{v \in \mathcal{N}(u), g(u) \neq g(v)} \frac{1}{K} \frac{1}{V/K} \end{aligned}$$

On the other hand, for a CRW crawler, the probability to move from one community "this" to another community "other" in a single crawling step is

$$\begin{aligned} P(\text{other}|\text{this}, CRW) &= \sum_{u} \sum_{v \in \mathcal{N}(u), g(u) \neq g(v)} p_{uv}^{crw} q_u^{crw} \\ &= \sum_{u} \sum_{v \in \mathcal{N}(u), g(u) \neq g(v)} \frac{1}{K} \min\left(\frac{1}{V_{g(v)}}, \frac{1}{V_{g(u)}}\right) \end{aligned}$$

Therefore,

$$P(\text{other}|\text{this}, RW) < P(\text{other}|\text{this}, CRW)$$

is equivalent to

$$\sum_{u} \sum_{\substack{v \in \mathcal{N}(u) \\ g(u) \neq g(v)}} \frac{1}{V/K} < \sum_{u} \sum_{\substack{v \in \mathcal{N}(u) \\ g(u) \neq g(v)}} \min\left(\frac{1}{V_{g(v)}}, \frac{1}{V_{g(u)}}\right).$$

■

The intuition behind this theorem is straightforward. Given that nodes are much more densely connected within communities than across communities, when proposed to traverse to a node belonging to a larger community, where it is easier to get trapped inside, the CRW crawler has the ability to reject the proposal, while the RW crawler can do nothing but move forward. As for the cases when proposed to traverse to a node belonging to a smaller community or one at the same scale, the CRW crawler performs similarly as the RW crawler.

3.3.3 Comparison on synthetic networks

To show that the CRW crawler indeed performs better than the RW crawler as shown in Theorem 3.3.2, we will generate synthetic networks, apply the two crawlers on them and compare their performance. The synthetic networks are generated as follows.

Algorithm *Synthetic Network Generator*

1. Set the number of distinct communities as $K = 25$;

2. Simulate the size of each community by the power law

$$P(N_k) \propto N_k^{-\gamma},\ k = 1, \ldots, K, \tag{3.15}$$

with the minimum and maximum possible sizes set as 50 and 5000, respectively.

3. Simulate links between each pair of nodes (u, v) under the following rule.

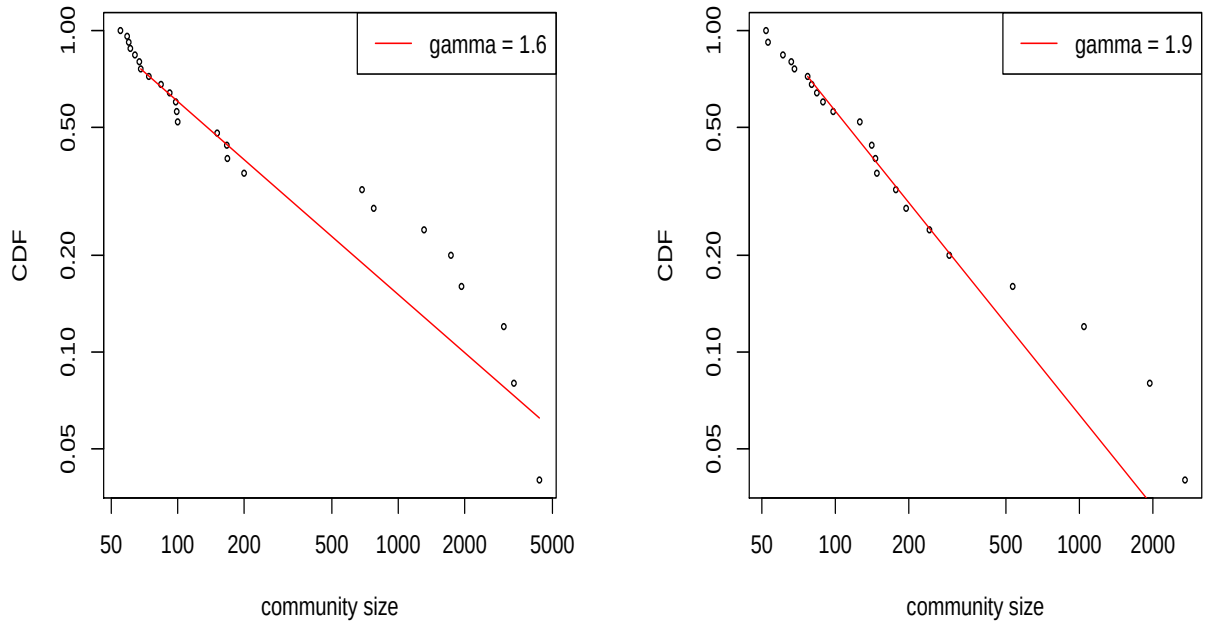
   - If $g(u) \neq g(v)$, then generate a link between (u, v) with probability $\frac{2}{N}$, where N is the total number of nodes in the network;

   - Otherwise, generate a link between (u, v) with probability

$$p_k \propto \frac{1}{N_k^{1/2}} \tag{3.16}$$

4. Check for isolated nodes. If there is any node with no neighbor, randomly choose one node from the same community of the isolated node and form a link between them.

We consider the power law distribution in (3.15) because many real-world large-scale networks with community structure have community size distributions following power laws with different values of γ [26]. The intuition behind Step 3 is bi-fold. On one hand, for each node u, the expected number of across-community neighbors is 2, while that of within-community neighbors is $N_{g(u)}^{1/2}$. For a community with minimum community size 50, the ratio between across-community neighbors and within-community neighbors is roughly $2/7$. The ratio is much smaller for larger communities. This result aligns with the nature of community that nodes within are more densely connected than across. On the other hand, although the expected number of neighbors grows when the community becomes large, the rate to form links



(a) Nodal degree distribution when $\gamma = 1.5$

(b) Nodal degree distribution when $\gamma = 2$

Figure 3.1.: Power law distributions of nodal degrees in generated synthetic networks. Values of γ in the legends are estimated using the **poweRlaw** package in R.

decreases. It reflects the phenomenon that in the real-world where a node often represents a person or an agent, cognitive constraints and time costs limit the total number of links one node can maintain [72, 73].

In our simulation, we choose two values of γ in (3.15), $\gamma = 1.5$ and $\gamma = 2$, and simulate two synthetic networks. Figure 3.1 shows that the nodal degree distributions of the simulated synthetic networks follow the desired power law as (3.15) properly.

With synthetic networks generated, we repeatedly crawl them using both the RW crawler and the CRW crawler 100 times. Each repetition, the two crawlers start from the same randomly

selected node and crawl for 2500 steps. To avoid the dependence between the performances and the starting node, across different repetitions the starting nodes are randomly selected instead of fixed. Every 50 steps, the number of visited communities for each crawler is calculated. Figure 3.2 plots the average number of communities visited imposed with the 95% confidence interval for each crawler over 100 repetitions.

We can see from the plots that after a small number of steps, the CRW crawler tends to visit more communities than the RW crawler, which empirically aligns with the claim of Theorem 3.3.2.

3.4 Practical concern and the adaptive version

Although the CRW crawler enjoys good theoretical property and performs well on the synthetic networks, in practice, when the volume of each community is unknown, the CRW crawler cannot be applied directly. However, we can make certain adjustment to adapt to the real scenarios.

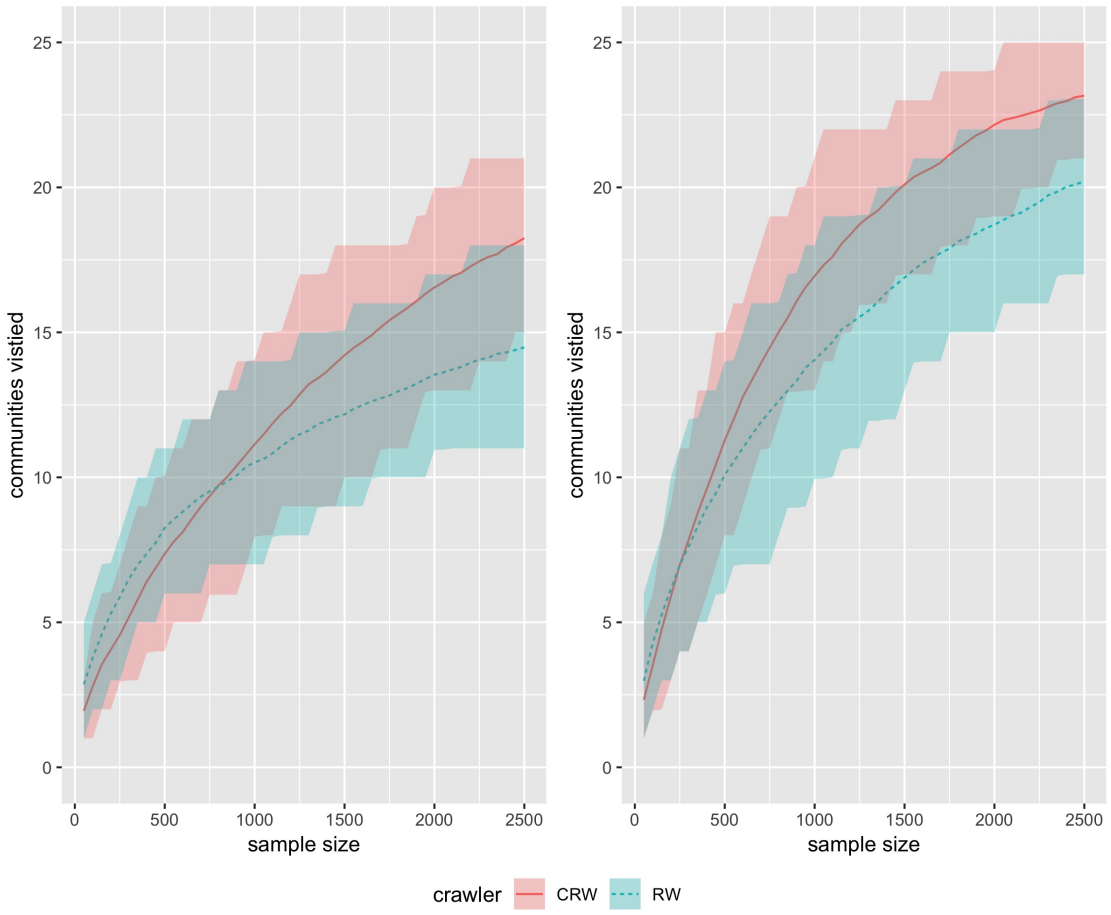


Figure 3.2.: Comparison between the RW crawler and the RW crawler. The plot on the left corresponds to the synthetic network generated with $\gamma = 1.5$, while the plot on the right corresponds to the synthetic network generated with $\gamma = 2$.

3.4.1 Algorithm

For each crawling step in a CRW crawler, when a new node that has never been visited is proposed, before deciding to move or stay, the crawler can update the volume of the community that the proposed node belongs to. If a node from a new community that has never been visited is proposed, initiate the volume of that community by the degree of the proposed node and move the crawler to the proposed node with probability 1. If a node which has been visited or proposed before is proposed again, then leave the volume of the community that the proposed node belongs to unchanged. In such a way, the crawler can estimate the volume of each community based on both visited nodes and proposed but not visited nodes, while avoid over-estimating the volume of each community from the re-visited or re-proposed nodes. The adaptive community-volume-adjusted random walk (ACRW) crawler works in the following way as described in Algorithm *ACRW*.

Algorithm *ACRW*

1. Start from a randomly selected node u_1. Suppose the crawler reaches node u_t at the t-th step;

2. Check the community $g(u_t)$ that node u_t belongs to.

   - If $g(u_t) = k$ for some community k that has been previously visited,

     * If u_t has never been visited or proposed, update the volume of community k by

     $$V_k = V_k + d_{u_t};$$

     * Otherwise, do not update the volume of community k.

   - Otherwise, add a new community index k^* with its initial community volume as

   $$V_{k^*} = d_{u_t};$$

3. Perform a CRW crawler step, which

   a. finds the set of neighbors $\mathcal{N}(u_t)$ for u_t;

   b. with equal probability, randomly selects one node $v \in \mathcal{N}(u_t)$;

   c. checks the community $g(v)$ that node v belongs to.

   - If $g(v) = k'$ for some community k' that has been previously visited,

     * If v has never been visited or proposed, update the volume of community k' by

     $$V_{k'} = V_{k'} + d_v;$$

     move the crawler to

     $$u_{t+1} = \begin{cases} v, & \text{with probability } \min\left(\frac{V_{g(u_t)}}{V_{g(v)}}, 1\right), \\ u_t, & \text{with probability } 1 - \min\left(\frac{V_{g(u_t)}}{V_{g(v)}}, 1\right). \end{cases}$$

   - Otherwise, move the crawler to v.

3.4.2 Comparison on synthetic networks

To compare the performances of the ACRW crawler, the CRW crawler and the RW crawler, we similarly generate synthetic networks following the Algorithm *Synthetic Network Generator* and apply the three crawlers on them. Still choosing the value of γ in (3.15) to be $\gamma = 1.5$ and $\gamma = 2$, we generate two synthetic networks just like what we did in Section 3.3.3.

Similarly, we repeatedly crawl on the two synthetic networks using all three crawlers 100 times. Each repetition, the three crawlers start from the same randomly selected node and crawl for 2500 steps. To avoid the dependence between the performances and the starting node, across different repetitions the starting nodes are randomly selected instead of fixed. Every 50 steps, the number of visited communities for each crawler is calculated. Figure 3.3 plots the average number of communities visited imposed with the 95% confidence interval for each crawler over 100 repetitions.

We can see from Figure 3.3 that at the beginning, the ACRW crawler and the RW crawler perform quite similarly. This is because that when only a small group of nodes have been visited, most of the proposed moves will be accepted to foster exploration. Therefore, the ACRW crawler essentially works just like a RW crawler, which always accepts the proposed moves. After a sufficient number of nodes have been visited and the volumes of visited communities are also well estimated, the proposed moves will be decided by the estimated volumes of the communities. Reflected on Figure 3.3, it corresponds to the phenomenon that gradually the average number of communities visited by the ACRW crawler coincides with that of the CRW crawler.

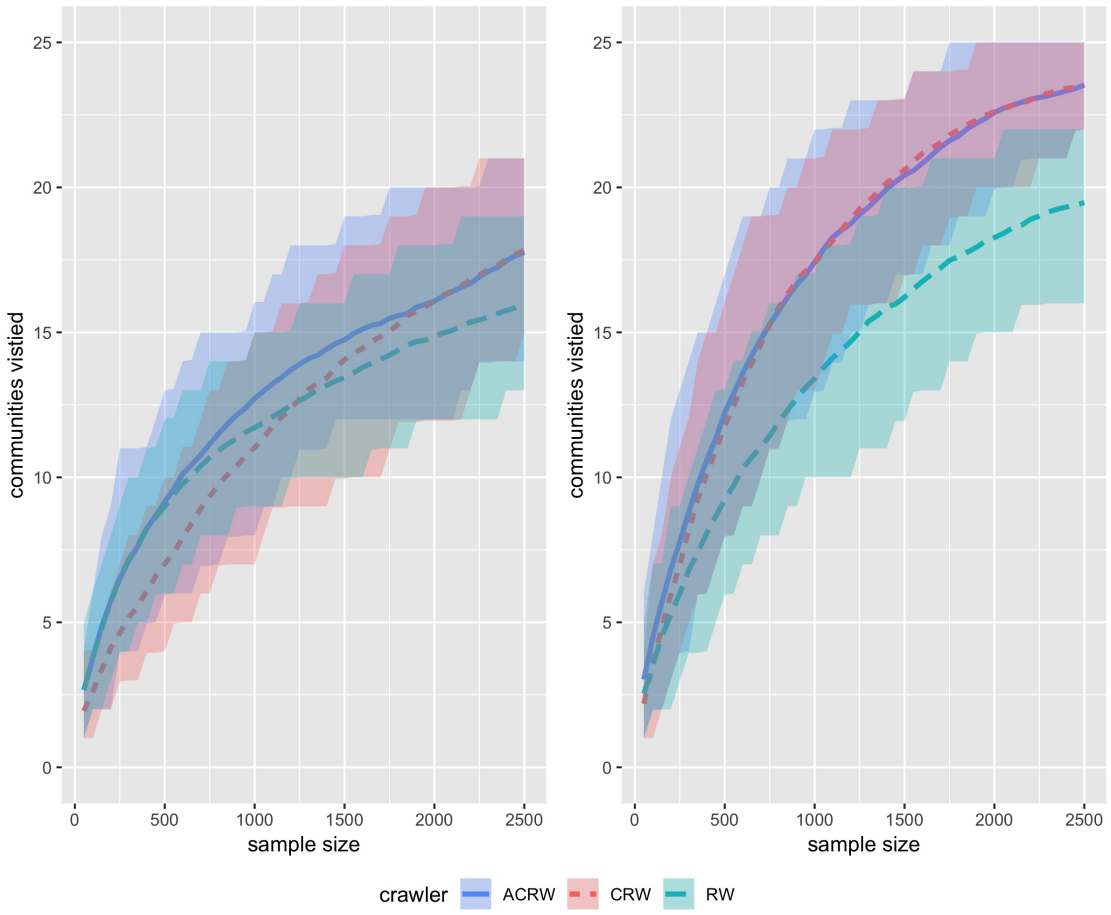


Figure 3.3.: Comparison among the RW crawler, the CRW crawler and the ACRW crawler. The plot on the left corresponds to the synthetic network generated with $\gamma = 1.5$, while the plot on the right corresponds to the synthetic network generated with $\gamma = 2$.

4. Discrete Large-scale Hypothesis Testing based on Local FDR

In this Chapter, we propose a formal procedure to conduct discrete large-scale hypothesis testing based on local FDR. We first introduce Efron's method in Section 4.1, which is used to conduct large-scale hypothesis testing based on local FDR when each test is continuous. We then introduce Habiger's randomized p-value method in Section 4.2, which provides a way to convert discrete p-values continuous. The formal procedure we propose is described in Section 4.3, together with a power diagnostic statistic used to assess statistical power. Simulation studies are conducted in Section 4.4 to compare the theoretical null and the empirical null in Efron's method, and also evaluate the performance of the power diagnostic statistic.

4.1 Efron's method

Following the same setup as in Section 1.3, Efron's method estimates $fdr(z_i)$ (1.2) for each test *i* by estimating the *mixture density* f and the *null sub-density* f_0^+ separately. The estimation procedures are introduced in detail in [29] and made available through R package **locfdr** [74]. We will go through the estimation procedures briefly in this section.

4.1.1 Estimating the mixture density f

The mixture density f is estimated through a standard Poisson generalized linear regression (GLM) procedure. Suppose that the m z-values $z_1, \ldots z_m$ are binned into K bins with bin

counts $y_1, \ldots, y_K$ summing to m and equal bin width δ. Efron's method assumes that the y_k's are independent Poisson counts,

$$y_k \overset{\text{ind}}{\sim} \text{Pois}(\nu_k),\ k = 1, 2, \ldots, K, \tag{4.1}$$

with ν_k proportional to the mixture density f evaluated at the midpoint of the kth bin z_k^{mid}, i.e., approximately,

$$\nu_k = m\delta f(z_k^{\text{mid}}). \tag{4.2}$$

By modeling $\log(\nu_k)$ as a Dth degree polynomial function of z_k^{mid}, (4.1) and (4.2) lead to a Poisson generalized linear model.

4.1.2 Estimating the null sub-density f_0^+

Two situations are considered to estimate

$$f_0^+ = \pi_0 f_0 \tag{4.3}$$

in Efron's method. The *theoretical null* $f_0 \sim N(0,1)$, which would be used for each individual hypothesis testing problem, may or may not be satisfactory for testing m hypotheses simultaneously. In practice, many factors, e.g., failed distributional assumptions on the data, unobserved covariates, correlation across different tests, correlation between samples, etc., could render the theoretical null to fail [38, 39]. When the theoretical fails, Efron's method would fit an *empirical null* instead [38, 39].

Assume the empirical null is still normal but not necessarily mean 0 and variance 1, say,

$$f_0 \sim N(\mu_0, \sigma_0^2). \tag{4.4}$$

One method that [39] provides to estimate μ_0, σ_0 as well as π_0 so as to fit the empirical null is called "central matching". Plugging into $\log(f_0^+(z))$ with (4.3) and (4.4) gives

$$\log(f_0^+(z)) = \log \pi_0 - \frac{1}{2}\left\{\frac{\mu_0^2}{\sigma_0^2} + \log(2\pi\sigma_0^2)\right\} + \frac{\mu_0}{\sigma_0^2}z - \frac{1}{2\sigma_0^2}z^2, \tag{4.5}$$

where π_0 is the null proportion and π is the mathematical constant. "Central matching" method uses $\log(\hat{f_0^+}(z))$ to quadratically approximate $\log(\hat{f(z)})$ near $z = 0$, so the estimated values $\hat{\pi_0}, \hat{\mu_0}, \hat{\sigma_0}$ are obtained from

$$\hat{\beta_0} = \log \pi_0 - \frac{1}{2}\left\{\frac{\mu_0^2}{\sigma_0^2} + \log(2\pi\sigma_0^2)\right\} \tag{4.6}$$

$$\hat{\beta_1} = \frac{\mu_0}{\sigma_0^2} \tag{4.7}$$

$$\hat{\beta_2} = \frac{1}{2\sigma_0^2} \tag{4.8}$$

where $\hat{\beta_0}$, $\hat{\beta_1}$, $\hat{\beta_2}$ are the estimated coefficients of the constant term, first order term and second order term from the Poisson generalized linear model for $\log(\hat{f}(z))$ in Section 4.1.1, respectively. The rationale of this method is the "zero assumption" [39], that the z-values close to zero are all realized from null cases.

The "central matching" method to estimate the empirical null, together with the option of using the theoretical null, are available in the R package **locfdr**. In practice, estimating the empirical null instead of directly using the theoretical null is recommended.

4.1.3 Efron's method fails for discrete large-scale hypothesis testing

As we can see from Section 4.1.2, Efron's method assumes normality and hence continuity for the null density $f_0(z)$, no matter it uses the theoretical null or the empirical null. Such continuity assumption is violated when each test is discrete, which causes Efron's method to fail when directly applied to discrete large-scale hypothesis testing problems.

Figure 4.1 shows the performance of Efron's method directly applied to a simulated discrete large-scale hypothesis testing problem. The simulated problem contains $m = 10,000$ hypotheses, each of which is a FET to discern if the success probabilities between two groups are the same. Each FET is built on a 2×2 contingency table, shown in Table 4.1. Detail of the simulation procedure is described in Scenario A in Section 4.4.2. The histogram of raw z-values in Figure 4.1 shows a huge peak centered at $z = 0$, and it drops significantly to both side of the huge peak. The green solid curve and the blue dashed curve in Figure 4.1 represent fitted mixture density f and null sub-density f_0^+, respectively [74]. We can see that Efron's method fails to capture the discretely supported z-values.

4.2 The randomized p-value method

The randomized p-value method [54] introduces an independent uniformly distributed random variable to convert the discretely supported p-values continuous and hence achieves exact control of type I error rate.. Consider a single test $H_0 : \theta = \theta_0$ versus $H_1 : \theta \neq \theta_0$ based on the realization x of a random X, whose distribution G is assumed to belong to a known class of distribution $\mathscr{G} = \{G(\cdot, \theta) : \theta \in \Theta\}$. Let $T(X)$ be a one-dimensional discretely distributed test statistic, and Q be the cdf of $T(X)|X \sim G(\cdot; \theta_0) = G_0$. Define

$$q(t) = \Delta Q(t) = Q(t) - Q(t-) \tag{4.9}$$

and the quantile function

$$Q^{-1}(u) = \inf\{t : Q(t) \geq u\}. \tag{4.10}$$

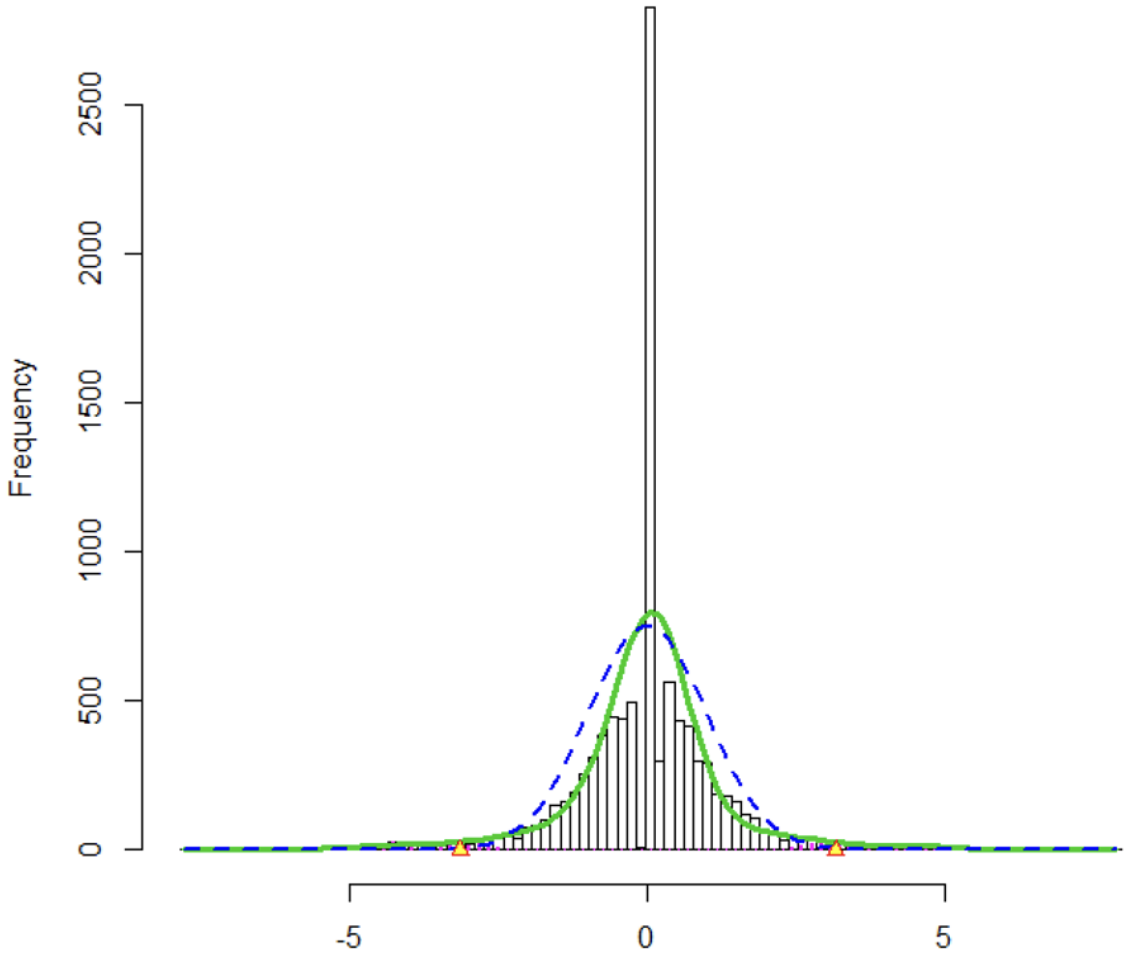


Figure 4.1.: Performance of Efron's method for discrete large-scale hypothesis testing. Plots obtained by directly applying Efron's method to a discrete large-scale hypothesis testing problem using the R package **locfdr**.

Without loss of generality, suppose the test rejects H_0 for small values of $T(X)$, the p-value is then

$$P(X) = Q(T(X)),$$

which is also discretely supported on $[0,1]$. [54] introduces an independent random variable U which is uniformly distributed on $[0,1]$, and defined the induced randomized p-value as

$$P_{\Delta^-}(X,U) = Q(T(X)-) + U \cdot q(T(X)). \tag{4.11}$$

The induced randomized p-value $P_{\Delta^-}(X,U)$ is then continuously supported on $[0,1]$. Furthermore, [54] proves that $P_{\Delta^-}(X,U)$ is G_0-*uniform*, i.e.

$$\mathbb{P}_{G_0}\{P_{\Delta^-}(X,U) \leq p\} = p \text{ for all } p \text{ in } [0,1].$$

From a Bayesian point of view, we can treat this randomized p-value method as a data augmentation procedure. Although the random variable X itself is discrete in nature, we can manually augment it by another independent random variable U, leading the p-value for the augmented data (X,U) continuous and G_0-uniform on $[0,1]$.

4.3 Methods

4.3.1 Local FDR estimation procedure

As discussed in Section 4.1.2, the estimation of the null sub-density in Efron's method assumes normality of the null z-values, which implicitly assumes continuity. When the continuity assumption is violated, Efron's method will fail to give reasonable local FDR estimates. The randomized p-value method proposed by [54] as discussed in Section 4.2 provides a bridge to apply Efron's method for discrete large-scale hypothesis testing problems.

Under the same setup as in Section 1.3, for each hypothesis H_i, $i = 1, \ldots, m$, following the randomized p-value method, we can introduce an independent random variable U_i and form an induced randomized p-value as in (4.11),

$$P_{\Delta^-}(X_i, U_i) = Q_i(T(X_i)-) + U_i \cdot q_i(T(X_i)), \tag{4.12}$$

where X_i is a random observable data for the ith test, Q_i and q_i are defined likewise as in (4.9) and (4.10). Now as the induced randomized p-values $P_{\Delta^-}(X_i, U_i)$ are continuously distributed on $[0,1]$, we can compute the induced Z-values, which is also continuously distributed, via

$$Z(X_i, U_i) = \Phi^{-1}(P_{\Delta^-}(X_i, U_i)), \tag{4.13}$$

where $\Phi^{-1}(\cdot)$ is the quantile function of the standard normal distribution. Based on the realized induced z-values for the ith test

$$z(x_i, u_i) = Z(X_i = x_i, U_i = u_i), \tag{4.14}$$

Efron's method can therefore be applied to estimate the local FDR,

$$fdr(z(x_i, u_i)) = P(H_{i0} \text{ is true} | Z(X_i, U_i) = z(x_i, u_i)), \tag{4.15}$$

which can be equivalently denoted as

$$fdr(x_i, u_i) = P(H_{i0} \text{ is true} | X_i = x_i, U_i = u_i)). \tag{4.16}$$

However, just obtaining the estimation of $fdr(x_i, u_i)$, as done in the application section of [54], is incomplete. Our target is not to test based on the joint data $\{(X_i, U_i)\}_{i=1}^m$, but the original data $\{X_i\}_{i=1}^m$. In other words, the target is to estimate the local FDR

$$fdr(x_i) = P(H_{i0} \text{ is true} | X_i = x_i), \tag{4.17}$$

instead of $fdr(x_i, u_i)$ as in (4.16). To achieve this target, we can simply marginalize out U_i from $fdr(x_i, U_i)$, i.e.

$$fdr(x_i) = \int fdr(x_i, U_i) \mathrm{d}U_i. \tag{4.18}$$

And an empirical way to do so is to simulate $u_i^1, \ldots u_i^J \sim U_i$, estimate $fdr(x_i, u_i^j)$ using Efron's method by $\hat{fdr}(x_i, u_i^j)$ and estimate $fdr(X_i)$ by

$$\hat{fdr}(X_i) = \frac{1}{J} \sum_{j=1}^{J} \hat{fdr}(X_i, u_i^j). \tag{4.19}$$

4.3.2 Discrete large-scale hypothesis testing procedure

With the local FDR for each discrete test estimated properly, a discrete large-scale hypothesis testing procedure could be conducted following Algorithm 1.

Algorithm 1

1. For $j = 1, \ldots, J$,

    1.Draw $u_i^j \sim \text{unif}(0,1)$ for each $i = 1, \ldots, m$;

    2.Calculate the induced randomized p-values $P_{\Delta^-}(x_i, u_i^j)$ for each $i = 1, \ldots, m$, as in (4.27);

    3.Transform the induced randomized p-values into induced z-values $z(x_i, u_i^j)$ for each $i = 1, \ldots, m$, as in (4.28);

    4.Obtain the estimate of the local FDR value $\hat{fdr}(X_i, u_i^j)$ for each $i = 1, \ldots, m$, using Efron's method;

2. For each $i = 1, \ldots, m$, obtain the estimate of the local FDR value $\hat{fdr}(X_i)$ as in (4.19);

3. Reject the ith test if $\hat{fdr}(X_j) < \eta$.

The cutoff value η to determine rejection is suggested to take 0.2 [39]. As for Efron's method in our algorithm, we directly use the R package **locfdr**. Furthermore, we suggest to estimate the empirical null instead of directly using the theoretical null for Efron's method, as discussed in Section 4.1.2. We will use a simulation study similar to the example used in [38] to show that the theoretical null fails in certain scenario while the empirical null remains reasonably good performance.

4.3.3 Power diagnostic

[39] provides a power diagnostic statistic to help assess power, the probability of rejecting genuinely non-null cases. The diagnostic statistic is an estimator of the *expected non-null false discovery rate*, i.e.

$$\text{Efdr} = \mathbb{E}_{F_1} fdr(z) = \int fdr(z) f_1(z) \mathrm{d}z, \tag{4.20}$$

where f_1 is the non-null density aligned with what we defined in Section 1.3. As we discussed in Section 4.1, Efron's method bins the z-values and provides an estimated mixture density $\hat{f}(z_k^{\text{mid}})$ and an estimated local FDR $\hat{fdr}(z_k^{\text{mid}})$ at each bin center z_k^{mid}. For simplicity, denote $\hat{f}(z_k^{\text{mid}})$ by $\hat{f}_k$ and $\hat{fdr}(z_k^{\text{mid}})$ by $\hat{fdr}_k$. Based on the following equation

$$f_1(z) = (1 - fdr(z)) f(z) / \int (1 - fdr(z')) f(z') \mathrm{d}z', \tag{4.21}$$

Efron's method estimates the non-null density f_1 at each bin center by

$$\hat{f}_{1k} = \hat{f}_1(z_k^{\text{mid}}) = (1 - \hat{fdr}_k)\hat{f}_k / \sum_{k=1}^{K} (1 - \hat{fdr}_k)\hat{f}_k. \tag{4.22}$$

With the estimated local FDR and non-null density, an estimator of Efdr is given by

$$\widehat{\text{Efdr}} = \sum_{k=1}^{K} \hat{fdr}_k \hat{f}_{1k} = \frac{\sum_{k=1}^{K} \hat{fdr}_k (1 - \hat{fdr}_k)\hat{f}_k}{\sum_{k=1}^{K} (1 - \hat{fdr}_k)\hat{f}_k}. \tag{4.23}$$

A small value of Efdr would suggest good power.

We can similarly bring the power diagnostic statistic into our framework. For each iteration j in Step 1 of Algorithm 1, we introduce m independent variables $u_1^j, \ldots, u_m^j$ to convert discrete p-values continuous, the power diagnostic statistic for the jth step $\widehat{\text{Efdr}}_j$ is actually

$$\widehat{\text{Efdr}}_j = \widehat{\text{Efdr}}(u_1^j, \ldots, u_m^j). \tag{4.24}$$

	C	$N-C$	Total
Binomial(n_{1i}, q_{1i})	c_{1i}	$n_{1i}-c_{1i}$	n_{1i}
Binomial(n_{2i}, q_{2i})	c_{2i}	$n_{2i}-c_{2i}$	n_{2i}

Table 4.1: *Contingency table for a FET*

Therefore, we can simply marginalize out the independent variables and give the power diagnostic statistic by

$$\widehat{Efdr} = \frac{1}{J}\sum_{j=1}^{J}\widehat{Efdr}_j. \tag{4.25}$$

4.4 Simulation Study

In this section, we evaluate the performance of our method through simulation studies. Consider testing $m = 10,000$ hypotheses between the two groups, and let $j = 1,2$ denote the two groups under comparison. Set the null proportion $\pi_0 = 0.9$. Each hypothesis i is to discern if the success probabilities q_{ji} between two groups are the same. We conduct a FET for each test i. The FET is built on a 2×2 contingency table, which consists of the counts $(C, N-C)$ from two independent Binomial distributions, Binomial(n_{1i}, q_{1i}) and Binomial(n_{2i}, q_{2i}). Table 4.1 shows the contingency table.

4.4.1 Evaluate the performance

One principle for hypothesis testing problem is the conservativeness. For a single hypothesis testing problem, conservativeness is reflected as that the false positive rate (type I error)

should be no higher than a given nominal significance level; For a large-scale hypothesis testing problem relying on tail area-based FDR, conservativeness is that the tail area-based FDR should not exceed a given nominal level [75]; As for a large-scale hypothesis testing problem relying on local FDR, conservativeness is that the estimated local FDR should not be smaller than the actual local FDR. The rationale behind the last case, which is what this dissertation concerns, is that: If the estimated local FDR turns to be smaller than the actual local FDR, then more tests are likely to be rejected than it is supposed to, which is anti-conservative.

To evaluate the performance and check conservativeness of our algorithm for the simulation study, we could compare the estimations with the actual local FDR values. Like Efron's method discussed in Section 4.1.1, we can bin the raw z-values

$$z(x_i) = \Phi^{-1}(P(x_i)), \text{where } P(X_i) \text{ is the raw p-value of the } i\text{th test}, \tag{4.26}$$

into L bins with equal width ε. So the actual local FDR value in the lth bin is

$$fdr(z^l) = \frac{\sum_{i=1}^{m} \mathbb{1}(H_{i0} \text{ is true}) \mathbb{1}(z^l - \varepsilon/2 < z(x_i) < z^l + \varepsilon/2)}{\max\{\sum_{i=1}^{m} \mathbb{1}(z^l - \varepsilon/2 < z(x_i) < z^l + \varepsilon/2), 1\}}, \tag{4.27}$$

and the mean estimated local FDR is

$$\bar{fdr}(z^l) = \frac{\sum_{i=1}^{m} \hat{fdr}(x_i) \mathbb{1}(z^l - \varepsilon/2 < z(x_i) < z^l + \varepsilon/2)}{\max\{\sum_{i=1}^{m} \mathbb{1}(z^l - \varepsilon/2 < z(x_i) < z^l + \varepsilon/2), 1\}}, \tag{4.28}$$

where z^l is the center of the lth bin. On one hand, considering that in (4.2), Efron's method bins the z-values into intervals with equal bin width and sets the default bin width as 0.1, the bin width ε for performance evaluation procedure should be no less than 0.1. On the other hand, in the performance evaluation, if there is no raw p-values in a certain bin, then (4.27) and (4.28) for this bin will both be zero. To avoid such empty bin, we choose the bin width ε for performance evaluation as 0.4.

4.4.2 Comparison between using the theoretical null and the empirical null

Two scenarios are considered to reflect that the theoretical null may fail in practice. Data are generated differently corresponding to each scenario.

A the theoretical null is satisfactory:

(a) Randomly choose $\pi_0 \cdot m$ tests as true nulls. For each test i from this set, generate the common success probability $q_{1i} = q_{2i}$ from unif$(0,1)$;

(b) For each test i from the other $(1-\pi_0)\cdot m$ tests, generate $q_{1i} \sim$ unif$(0,1)$, and let $q_{2i} = q_{1i} - d_i \cdot \text{sign}(q_{1i} - 0.5)$, where $d_i \sim \text{unif}(r, 0.5)$, r is the minimum effect size. Here set $r = 0.2$;

(c) Randomly sample m indices from a WGBS data. We use the total count of the ith brain sample as the number of trials n_{i0} for the control group, and the total count of the ith es sample as n_{i1} for the treatment group;

(d) Draw the count $c_{ji} = \text{Binomial}(n_{ji}, q_{ji})$;

B the theoretical null fails:

(a) Randomly choose $\pi_0 \cdot m$ tests as true nulls. For each test i from this set, generate $q_i \sim$ unif$(0,1)$, and let $q_{1i} = q_i - \gamma_i$ while $q_{2i} = q_i + \gamma_i$, where $\gamma_i \sim \text{unif}(-b, b)$ and $b = \min\{0.1, 1-q_i, q_i\}$;

(b) For each test i from the other $(1-\pi_0)\cdot m$ tests, generate $q_{1i} \sim$ unif$(0,1)$, and let $q_{2i} = q_{1i} - d_i \cdot \text{sign}(q_{1i} - 0.5)$, where $d_i \sim \text{unif}(r, 0.5)$, r is the minimum effect size. Here set $r = 0.2$;

(c) Randomly sample m indices from a WGBS data. We use the total count of the ith brain sample as the number of trials n_{i0} for the control group, and the total count of the ith es sample as n_{i1} for the treatment group;

(d) Draw the count $c_{ji} = \text{Binomial}(n_{ji}, q_{ji})$;

Four possible factors causing the theoretical null to fail are listed by [38, 39], as mentioned in Section 4.1.2. The data generation procedure for Scenario *B* is analogous to the example [38] provides, where the existence of unobservable covariates, γ_i's, renders failure of the theoretical null: γ_i's introduce extra variation among the z-values of null cases, which results in a heavier-tail density compared to the theoretical null; However, γ_i's are unobservable with mean 0, leaving each test genuine null.

4.4.3 Performance of the power diagnostic statistic

To illustrate that the power diagnostic statistic $\widehat{Efdr}$ we proposed in Section 4.3.3 is a good indicator of statistical power, we consider a sequence of simulation studies with different minimum effect size r. Follow the same setup in Scenario A in Section 4.4.2, except that in (b), set the minimum effect size r to be 0.2, 0.3 and 0.4. We generate data and follow Algorithm 1 to conduct discrete large-scale hypothesis testing respectively for each value of r. The power diagnostic statistic $\widehat{Efdr}$ is calculated for each testing procedure, together with the *realized power* defined as

$$\text{realized power} = \frac{\text{number of true non-null hypotheses rejected}}{\text{total number of true non-null hypotheses}} \tag{4.29}$$

and the *realized tail area-based FDR* defined as

$$\text{realized tail area-based FDR} = \frac{\text{number of true null hypotheses rejected}}{\text{total number of rejections}}. \tag{4.30}$$

Intuitively, larger minimum effect size will deliver bigger differences in success probabilities for the non-null cases between the control group and treatment group, which makes the non-null cases easier to be detected by the testing procedure. As a result, it is expected to see higher realized power for simulation study with larger minimum effect size, given the realized tail area-based FDR controlled under the same level. If the power diagnostic statistic $\widehat{Efdr}$ is indeed a good indicator of statistical power, it should decrease along the way the minimum effect size increases.

4.4.4 Simulation results

For each scenario in Section 4.4.2, Efron's method in Algorithm 1 using both the theoretical null and the empirical null are conducted and compared. Figure 4.2 and Table 4.2 summarize the results for Scenario A, while Figure 4.3 and Table 4.2 summarize for Scenario B. We can see that in the two scenarios, the results obtained through empirical null always yield conservativeness. To compare with, although the result obtained through theoretical null yields closer estimated local FDR values to the actual local FDR and higher realized power for Scenario A, it turns anti-conservative for Scenario B. Such anti-conservativeness, or equivalently, over-optimistic result, will result in un-controlled number of false positive findings in real applications. Therefore, we suggest using empirical null in real applications instead of theoretical null.

Algorithm 1 is conducted with Efron's method using the empirical null for each simulation study in Section 4.4.3. Table 4.4 summarizes the power diagnostic statistics $\widehat{Efdr}$, realized powers, and realized tail area-based FDRs across different minimum effect sizes r. We can see that as the minimum effect size r increases from 0.2 to 0.4, it is as expected as mentioned

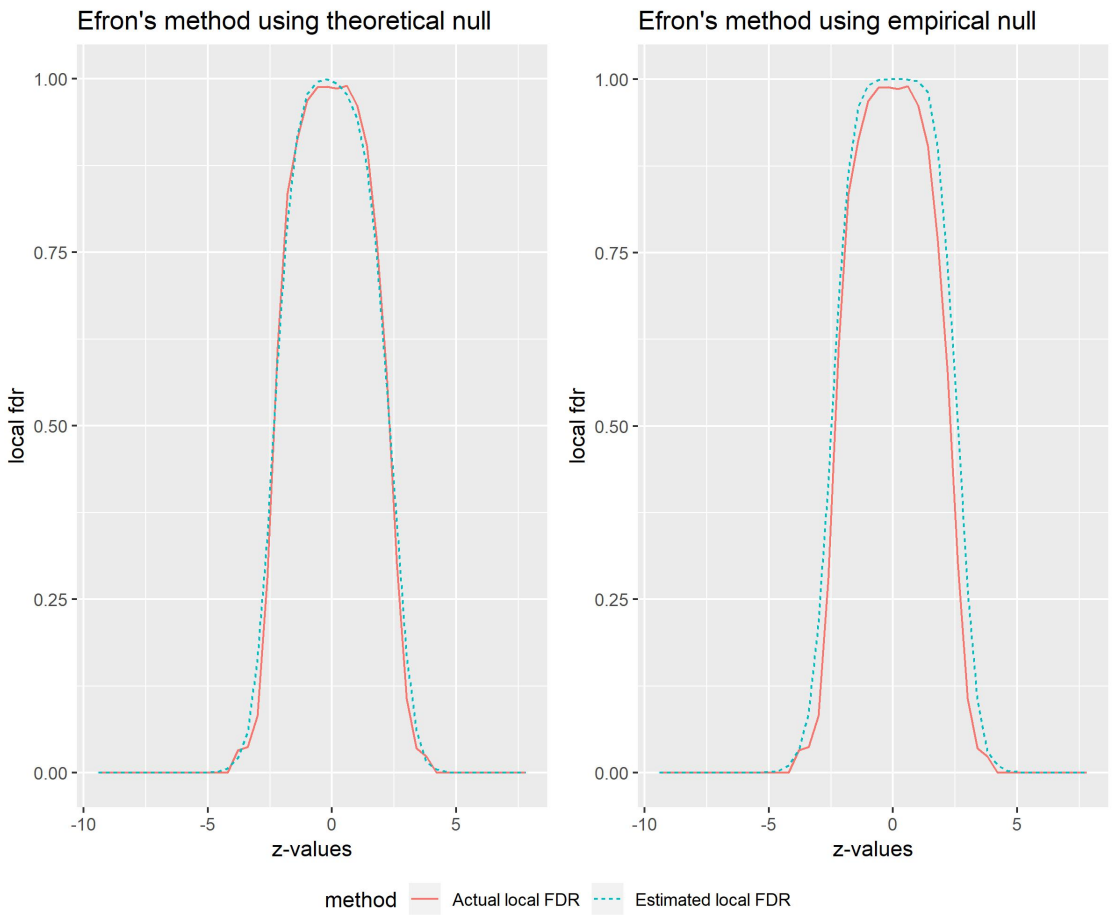


Figure 4.2.: The estimated local FDR versus the actual local FDR for Scenario A, where the theoretical null is satisfactory. The estimated local FDR in the plot to the left is obtained using theoretical null, while the one in the plot to the right is obtained using empirical null.

in Section 4.4.3 that the realized power increases and the power diagnostic statistic $\widehat{Efdr}$ decreases, while the realized tail area-based FDRs are controlled under the same level. Such result shows that $\widehat{Efdr}$ is indeed a good statistic to assess power.

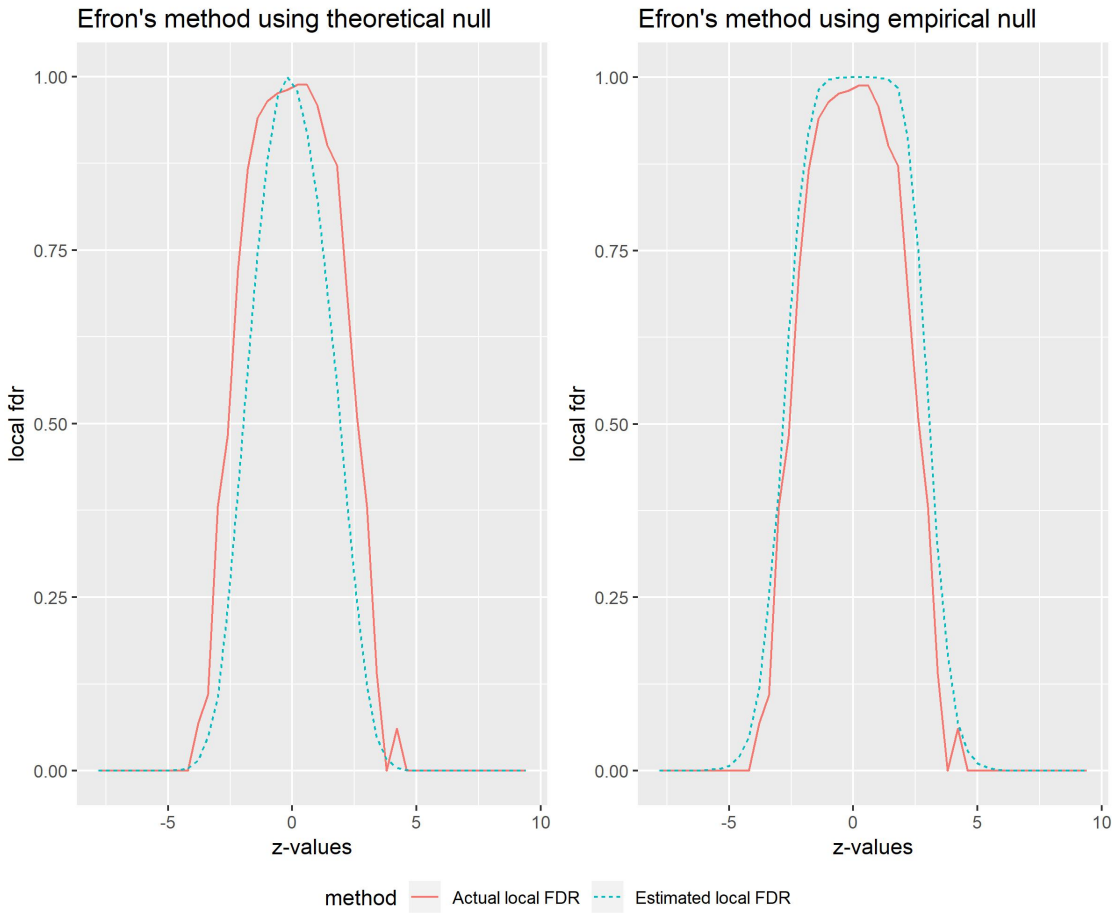


Figure 4.3.: The estimated local FDR versus the actual local FDR for Scenario *B*, where the theoretical null fails. The estimated local FDR in the plot to the left is obtained using theoretical null, while the one in the plot to the right is obtained using empirical null.

Table 4.2: *Summary statistics for testing result of Scenario A*

	$\widehat{Efdr}$	Realized power	Realized tail area-based FDR
Theoretical null	0.303	0.456	0.032
Empirical null	0.296	0.397	0.017

Table 4.3: *Summary statistics for testing result of Scenario B*

	$\widehat{Efdr}$	Realized power	Realized tail area-based FDR
Theoretical null	0.341	0.504	0.180
Empirical null	0.341	0.269	0.023

Minimum effect size	$r = 0.2$	$r = 0.3$	$r = 0.4$
$\widehat{Efdr}$	0.296	0.238	0.175
Realized power	0.397	0.529	0.728
Realized tail area-based FDR	0.017	0.033	0.038

Table 4.4: *Summary statistics for testing results across different minimum effect size r.*

5. Conclusion and Future Work

In this dissertation, we addressed three statistical issues that rise in data with complicated structure and/or in large scale.

Firstly, we proposed a novel Bayesian approach for inferences of the summary statistics and model parameters in ERGMs under measurement errors. We provided a Gibbs sampler to iteratively draw "true" networks and the model parameters. Simulation results show that our Bayesian treatment effectively correct the impact of measurement errors. Comparison with previous nonparametric approaches shows that our method is more adequate for the inference of networks with moderate number of nodes. We also show that our method is insensitive to the noise constant p when W^{true} is sparse and p, q are comparable in magnitude, and apply our method to perform inference for real world networks if only the value of q can be obtained.

Secondly, we introducd the CRW crawler and the ACRW crawler for sampling large-scale networks with exclusive communities. We proved that the probability for each community to be sampled using the CRW crawler is uniform, and that under certain condition the CRW crawler traverses across different communities faster than the widely used RW crawler. The ACRW crawler is proposed to handle the situation that in real applications where communities volumes are not known.

Lastly, we brought a solution for discrete large-scale hypothesis testing problems using local FDR. We handled the discreteness by applying Habiger's randomized p-value method to convert the discrete p-values continuous, so that Efron's method can be applied to estimate the

local FDR for the augmented test data. Proper treatments are conducted to marginalize out the auxiliary variable so as to deliver estimated local FDR for the original tests. We also provided power diagnostic statistics to assess the statistical power of the testing procedure. Simulation studies show the adequacy of using the empirical null in Efron's method, as well as using the power diagnostic statistics we proposed to assess the power.

Some directions of future work should be considered:

1. In Chapter 2, the parameters p and q describing the measurement errors are assumed to be known, independent and additive. In practice, the values of p and q may be unknown. Moreover, p and q on different dyads may be random and could possibly be correlated. We acknowledge the limitation that we have to assume additive error terms so that to apply ERGM models. And we understand that if the error terms are assumed additive but unknown random variables, it is possible to bring them into the Bayesian inference framework as well. However, as the computational cost for the current framework is already very high, adding more steps into the framework will make the cost even higher.

2. In Chapter 3, we proposed two crawlers for sampling large-scale networks with exclusive community structures. They both start from one randomly selected node. It is worth considering parallelizing the crawlers so that they can start from multiple different nodes in order to make the sampling procedure faster. Meanwhile, we only considered exclusive community structures in this dissertation. However, there are many real-world network structures that are not mutually exclusive, meaning that one node can simultaneously belong to multiple communities. One potential work in the future is how to properly sample large-scale networks with overlapping community structures.

3. In Chapter 4, how to conduct discrete large-scale hypothesis testing using local FDR is answered as a seemingly separate topic. Yet, large-scale hypothesis testing problem is closely related to networks. For many real-world networks, especially those emerging from genomics like gene regulatory networks, links or edges are not directly observed. Whether or not there exists a link is answered through hypothesis testing. When the number of nodes is large and consequently the number of dyads is even larger, it becomes a large-scale hypothesis testing problem. One potential work in the future is to construct a framework to bridge those two topics.

Bibliography

[1] Uri Alon. *An Introduction to Systems Biology: Design Principles of Biological Circuits*. CRC Press, 2006.

[2] Anna Goldenberg, Alice X Zheng, Stephen E Fienberg, and Edoardo M Airoldi. A survey of statistical network models. *Foundations and Trends in Machine Learning*, 2(2):129–233, 2010.

[3] Matthew O Jackson. *Social and Economic Networks*, volume 3. Princeton University Press Princeton, 2008.

[4] Eric D Kolaczyk. *Statistical Analysis of Network Data: Methods and Models*. Springer Science & Business Media, 2009.

[5] Mark EJ Newman. Coauthorship networks and patters of scientific collaboration. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5200–5205, 2004.

[6] Nan Lin and Hongyu Zhao. Are scale-free networks robust to measurement errors? *BMC Bioinformatics*, 6(1):119, 2005.

[7] Prakash Balachandran, Edoardo Airoldi, and Eric Kolaczyk. Inference of network summary statistics through network denoising. *arXiv:1310.0423*, 2013.

[8] Paul W. Holland and Samuel Leinhardt. The structural implications of measurement error in sociometry. *The Journal of Mathematical Sociology*, 3(1):85–111, 1973.

[9] André Fujita, Alexandre G Patriota, João R Sato, and Satoru Miyano. The impact of measurement error in the identification of regulatory networks. *BMC Bioinformatics*, 10(1):412, 2009.

[10] Dan J Wang, Xiaolin Shi, Daniel A McFarland, and Jure Leskovec. Measurement error in network data: a re-classification. *Social Networks*, 34(4):396–409, 2012.

[11] Prakash Balachandran, Eric D Kolaczyk, and Weston Viles. On the propagation of low-rate measurement error to subgraph counts in large, sparse networks. *arXiv:1409.5640*, 2014.

[12] Fan Guo, Steve Hanneke, Wenjie Fu, and Eric P Xing. Recovering temporally rewiring networks: A model-based approach. In *Proceedings of the 24th International Conference on Machine Learning*, pages 321–328, 2007.

[13] Danny Wyatt, Tanzeem Choudhury, and Jeff A Bilmes. Learning hidden curved exponential family models to infer face-to-face interaction networks from situated speech data. In *AAAI*, pages 732–738, 2008.

[14] Carey E Priebe, Daniel L Sussman, Minh Tang, and Joshua T Vogelstein. Statistical inference on errorfully observed graphs. *Journal of Computational and Graphical Statistics*, 24(4):930–953, 2015.

[15] Stanley Wasserman and Philippa Pattison. Logit models and logistic regressions for social networks: I. an introduction to markov graphs and p^*. *Psychometrika*, 61(3):401–425, 1996.

[16] Zachary M Saul and Vladimir Filkov. Exploring biological network structure using exponential random graph models. *Bioinformatics*, 23(19):2604–2611, 2007.

[17] Garry Robins, Pip Pattison, Yuval Kalish, and Dean Lusher. An introduction to exponential random graph (p^*) models for social networks. *Social Networks*, 29(2):173–191, 2007.

[18] Garry Robins, Tom Snijders, Peng Wang, Mark Handcock, and Philippa Pattison. Recent developments in exponential random graph (p^*) models for social networks. *Social Networks*, 29(2):192–215, 2007.

[19] Skyler J Cranmer and Bruce A Desmarais. Inferential network analysis with exponential random graph models. *Political Analysis*, 19(1):66–86, 2011.

[20] Minas Gjoka, Maciej Kurant, Carter T Butts, and Athina Markopoulou. Walking in Facebook: A case study of unbiased sampling of OSNs. In *2010 Proceedings IEEE INFOCOM*, pages 1–9, 2010.

[21] Alan Mislove, Massimiliano Marcon, Krishna P Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *IMC*, pages 29–42, 2007.

[22] Yong-Yeol Ahn, Seungyeop Han, Haewoon Kwak, Sue Moon, and Hawoong Jeong. Analysis of topological characteristics of huge online social networking services. In *Proceedings of the 16th International Conference on World Wide Web*, pages 835–844, 2007.

[23] Christo Wilson, Bryce Boe, Alessandra Sala, Krishna PN Puttaswamy, and Ben Y Zhao. User interactions in social networks and their implications. In *Proceedings of the 4th ACM European Conference on Computer Systems*, pages 205–218, 2009.

[24] Sang Hoon Lee, Pan-Jun Kim, and Hawoong Jeong. Statistical properties of sampled networks. *Physical Review E*, 73(1):016102, 2006.

[25] Lovász. Random walks on graphs: A survey. *Combinatorics*, 2(1):1–46, 1993.

[26] Alex Arenas, Leon Danon, Albert Diaz-Guilera, Pablo M Gleiser, and Roger Guimera. Community analysis in social networks. *The European Physical Journal B*, 38(2):373–380, 2004.

[27] Minas Gjoka, Maciej Kurant, Carter T Butts, and Athina Markopoulou. Practical recommendations on crawling online social networks. *IEEE Journal on Selected Areas in Communications*, 29(9):1872–1892, 2011.

[28] Adib Shafi, Cristina Mitrea, Tin Nguyen, and Sorin Draghici. A survey of the approaches for identifying differential methylation using bisulfite sequencing data. *Briefings in Bioinformatics*, 19:737–753, 2017.

[29] Bradley Efron. Local false discovery rates. *Technical Report*, 2005.

[30] Tore Schweder and Eil Spjøtvoll. Plots of p-values to evaluate many tests simultaneously. *Biometrika*, 69(3):493–502, 1982.

[31] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57:289–300, 1995.

[32] Bradley Efron, Robert Tibshirani, John D Storey, and Virginia Tusher. Empirical Bayes analysis of a microarray experiment. *Journal of the American Statistical Association*, 96(456):1151–1160, 2001.

[33] Yoav Benjamini and Yosef Hochberg. On the adaptive control of the false discovery rate in multiple testing with independent statistics. *Journal of Educational and Behavioral Statistics*, 25:60–83, 2000.

[34] Yoav Benjamini and Wei Liu. A step-down multiple hypotheses testing procedure that controls the false discovery rate under independence. *Journal of Statistical Planning and Inference*, 82:163–170, 1999.

[35] Yoav Benjamini and Daniel Yekutieli. The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, 29:1165–1188, 2001.

[36] John D Storey. A direct approach to false discovery rates. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(3):479–498, 2002.

[37] John D Storey, Jonathan E Taylor, and David Siegmund. Strong control, conservative point estimation and simultaneous conservative consistency of false discovery rates: a unified approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(1):187–205, 2004.

[38] Bradley Efron. Large-scale simultaneous hypothesis testing: the choice of a null hypothesis. *Journal of the American Statistical Association*, 99(465):96–104, 2004.

[39] Bradley Efron. Size, power and false discovery rates. *The Annals of Statistics*, 35(4):1351–1377, 2007.

[40] Bradley Efron. Microarrays, empirical Bayes and the two-groups model. *Statistical Science*, 23(1):1–22, 2008.

[41] Ryan Lister, Mattia Pelizzola, Robert H Dowen, R David Hawkins, Gary Hon, Julian Tonti-Filippini, Joseph R Nery, Leonard Lee, Zhen Ye, Que-Minh Ngo, et al. Human dna methylomes at base resolution show widespread epigenomic differences. *Nature*, 462(7271):315, 2009.

[42] Christian Perez-Llamas and Nuria Lopez-Bigas. Gitools: analysis and visualisation of genomic data using interactive heat-maps. *PloS ONE*, 6(5):e19541, 2011.

[43] Peter B Gilbert. A modified false discovery rate multiple-comparisons procedure for discrete data, applied to human immunodeficiency virus genetics. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 54(1):143–158, 2005.

[44] Stan Pounds and Cheng Cheng. Robust estimation of the false discovery rate. *Bioinformatics*, 22(16):1979–1987, 2006.

[45] Joseph F Heyse. A false discovery rate procedure for categorical data. In *Recent advances in biostatistics: False discovery rates, survival analysis, and related topics*, pages 43–58. World Scientific, 2011.

[46] Xiongzhi Chen and Rebecca W Doerge. A weighted FDR procedure under discrete and heterogeneous null distributions. *arXiv:1502.00973*, 2015.

[47] Kun Liang. False discovery rate estimation for large-scale homogeneous discrete p-values. *Biometrics*, 72(2):639–648, 2016.

[48] Xiongzhi Chen, Rebecca W Doerge, and Joseph F Heyse. Multiple testing with discrete data: Proportion of true null hypotheses and two adaptive FDR procedures. *Biometrical Journal*, 60(4):761–779, 2018.

[49] Sebastian Döhler, Guillermo Durand, and Etienne Roquain. New FDR bounds for discrete and heterogeneous tests. *Electronic Journal of Statistics*, 12(1):1867–1900, 2018.

[50] Xiaoyu Dai, Nan Lin, Daofeng Li, and Ting Wang. A non-randomized procedure for large-scale heterogeneous multiple discrete testing based on randomized tests. *Biometrics*, 2018.

[51] Guanshengrui Hao and Nan Lin. Discrete multiple testing in detecting differential methylation using sequencing data. In *Springer Book on Biostatistics and Bioinformatics*. 2019.

[52] Isaac Dialsingh. False discovery rates when the statistics are discrete. *PhD Thesis, The Pennsylvania State University*, 2011.

[53] Peter J Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880, 1984.

[54] Joshua D Habiger and Edsel A Pena. Randomised P-values and nonparametric procedures in multiple testing. *Journal of Nonparametric Statistics*, 23(3):583–604, 2011.

[55] Joshua D Habiger. Multiple test functions and adjusted p-values for test statistics with discrete distributions. *Journal of Statistical Planning and Inference*, 167:1–13, 2015.

[56] David R Hunter. Curved exponential family models for social networks. *Social Networks*, 29(2):216–230, 2007.

[57] Martina Morris, Mark S Handcock, and David R Hunter. Specification of exponential-family random graph models: terms and computational aspects. *Journal of Statistical Software*, 24(4):1548, 2008.

[58] Alberto Caimo and Nial Friel. Bayesian inference for exponential random graph models. *Social Networks*, 33(1):41–55, 2011.

[59] Alberto Caimo and Antonietta Mira. Efficient computational strategies for doubly intractable problems with applications to Bayesian social networks. *Statistics and Computing*, 25(1):113–125, 2015.

[60] Iain Murray, Zoubin Ghahramani, and David MacKay. MCMC for doubly-intractable distributions. *arXiv:1206.6848*, 2012.

[61] Walter R Gilks, Gareth O Roberts, and Edward I George. Adaptive direction sampling. *The Statistician*, pages 179–189, 1994.

[62] Gareth O Roberts and Walter R Gilks. Convergence of adaptive direction sampling. *Journal of Multivariate Analysis*, 49(2):287–298, 1994.

[63] Alberto Caimo and Nial Friel. Bergm: Bayesian exponential random graphs in R. *Journal of Statistical Software*, 61(2):1–25, 2014.

[64] Tong Ihn Lee, Nicola J Rinaldi, François Robert, Duncan T Odom, Ziv Bar-Joseph, Georg K Gerber, Nancy M Hannett, Christopher T Harbison, Craig M Thompson, and Itamar Simon. Transcriptional regulatory networks in Saccharomyces Cerevisiae. *Science*, 298(5594):799–804, 2002.

[65] Monika R Henzinger, Allan Heydon, Michael Mitzenmacher, and Marc Najork. On near-uniform URL sampling. *Computer Networks*, 33(1-6):295–308, 2000.

[66] Daniel Stutzbach, Reza Rejaie, Nick Duffield, Subhabrata Sen, and Walter Willinger. On unbiased sampling for unstructured peer-to-peer networks. *IEEE/ACM Transactions on Networking (TON)*, 17(2):377–390, 2009.

[67] Amir Hassan Rasti, Mojtaba Torkjazi, Reza Rejaie, Nick Duffield, Walter Willinger, and Daniel Stutzbach. Respondent-driven sampling for characterizing unstructured overlays. In *IEEE INFOCOM 2009*, pages 2701–2705, 2009.

[68] Christos Gkantsidis, Milena Mihail, and Amin Saberi. Random walks in peer-to-peer networks. In *IEEE INFOCOM 2004*, volume 1, 2004.

[69] Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *KDD*, pages 631–636, 2006.

[70] W Keith Hastings. Monte Carlo sampling methods using markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[71] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[72] Robin IM Dunbar. Do online social media cut through the constraints that limit the size of offline social networks? *Royal Society Open Science*, 3(1):150292, 2016.

[73] Emilio Ferrara. A large-scale community structure analysis in Facebook. *EPJ Data Science*, 1(1):9, 2012.

[74] Bradley Efron, Brit Turnbull, and Balasubramanian Narasimhan. locfdr: Computes local false discovery rates. *R package version*, 1:1–7, 2011.

[75] LJ Wei. Asymptotic conservativeness and efficiency of kruskal-wallis test for k dependent samples. *Journal of the American Statistical Association*, 76(376):1006–1009, 1981.