2019

Statistical Designs for Network A/B Testing

Victoria V. Pokhilko
Virginia Commonwealth University

Statistical Designs for Network A/B Testing

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University

by
Victoria V. Pokhilko
MBA, VCU, 2010
MS in Statistics, VCU, 2015

Directors: Dr. D'Arcy Mays, Associate Professor, Department of Statistical Sciences and Operation Research, VCU
Dr. Qiong Zhang, Assistant Professor, School of Mathematical and Statistical Sciences, Clemson University

Department of Statistical Sciences and Operation Research
Richmond, VA, USA
November 2019

Acknowledgements

I would like to thank my advisors, Dr. Zhang and Dr. Mays, for their time, wisdom and encouragement. I would also like to thank my committee members for providing a positive influence on the work presented in this dissertation and to thank my friends and colleagues in the SSOR department for their support. Lastly, I would like to thank my parents, Valentina and Valery, for their continuous support, love and faith in me.

Contents

List of Algorithms

List of Figures

List of Tables

Abstract

STATISTICAL DESIGNS FOR NETWORK A/B TESTING

by Victoria V. Pokhilko

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University

Directors: Dr. D'Arcy Mays, Associate Professor, Department of Statistical Sciences and Operation Research

Dr. Qiong Zhang, Assistant Professor, School of Mathematical and Statistical Sciences, Clemson University

A/B testing refers to the statistical procedure of conducting an experiment to compare two treatments, A and B, applied to different testing subjects. It is widely used by technology companies such as Facebook, LinkedIn, and Netflix, to compare different algorithms, web-designs, and other online products and services. The subjects participating in these online A/B testing experiments are users who are connected at different scales on social networks. Two connected subjects are similar in terms of their social behaviors, education and financial background, and other demographic aspects. Hence, it is only natural to assume that their reactions to online products and services are related to their network adjacency.

In Chapter 2, we propose to use the conditional auto-regressive model to present the network structure and include the network effects in the estimation and inference of the treatment effect. A D-optimal design criterion is developed based on the proposed model. Mixed integer programming formulations are developed to obtain the D-optimal designs. The effectiveness of the proposed method is shown through numerical results with synthetic networks and real social networks.

In Chapter 3 we propose a re-randomization approach to constructing an experimental design for A/B testing procedure, where stopping criterion for re-randomization are formulated based on D-optimal design objectives and CAR model assumptions. We test the performance of our method on synthetic networks and compare the results with exact D-optimal and random designs.

Finally in Chapter 4 we develop a Bayesian sequential experimental design for A/B testing that incorporates the network information and structure into network covariates that are consequently used in a Bayesian model. We propose several algorithms that describe the procedure, test the performance of our covariate assisted Bayesian model on synthetic and real-world networks and compare the results to a Bayesian sequential model that does not use network covariates in its posterior updates.

Chapter 1

Introduction

A/B testing has been widely used online to test which of two alternatives, A or B, is better. Those alternatives could be options of an online commercial, web page alternatives, or any new online feature that needs to be tested for an overall success of implementation. The measures of success can be numerical values of profits, sales, return on investment, number of clicks, etc. Usually A/B testing is run by selecting a group of subjects upon which the experiment is being conducted, and by randomly assigning those subjects to either treatment or control groups. This procedure works well when the subjects or users are not related to each other, i.e., independent. However, often with online experiments on social networks, the users are connected to their friends and may exert influence on each other when making decisions about offered alternatives. From a statistical perspective, the challenging aspect of these applications is how to account for the presence of connections, or network data, observed before experimentation, possibly with uncertainty. While there is a well-developed literature on several aspects of the statistical analysis of network data (Goldenberg et al., 2010; Kolaczyk and Csárdi, 2014), the literature on methods for experimentation and causal analyses that use observed connections is still growing (Rosenbaum, 2007; Hudgens and Halloran, 2008; Toulis and Kao, 2013; Ogburn and VanderWeele, 2017).

The need to account for network connections in A/B testing has led scholars to focus on two specific problem settings: network interference, where the potential outcomes of unit i are functions of the treatment assigned to unit i and of the treatments assigned to other units that are related to unit i through the network (Toulis and Kao, 2013; Ugander et al., 2013; Aronow et al., 2017; Eckles et al., 2017); and network-correlated outcomes, where the network informs the correlation among the potential outcomes because the potential outcomes of unit i depend on its covariates, and the covariates of units that are connected are more similar than the covariates of those that are not (McPherson et al., 2001; Shalizi and Thomas, 2011; Manski, 2013). In this work, we focus on the second setting: network correlated outcomes. Our research is targeted at creating designs of online A/B testing experiments that incorporate network information either in terms of network structure or network covariates, or both.

This dissertation is organized as three different projects presented in Chapters 2, 3 and 4. In Chapter 2 we focus on the construction of A/B testing experimental designs for network-correlated outcomes when users who are connected in a network share some common social and demographic backgrounds. We propose a spatial network model for A/B testing, called conditional auto-regressive model or CAR (Schmidt and Nobre, 2014) to incorporate the correlated network structure in the analysis. To conduct the experiment on the test subjects that form the network, we choose the D-optimal design based on the proposed model. The D-optimal design is the optimal solution of minimizing the determinant of the generalized variance matrix of the parameter estimates for the pre-specified model (Fedorov, 2010; Wu and Hamada, 2011) with respect to the experimental design setting. In the A/B testing experiment, the key parameter is the treatment effect, and thus the D-optimal criterion measures the variance of the estimated treatment effect. Mixed integer programming formulations are developed to optimize the D-optimal utility function with respect to the treatment assignments of the test subjects, i.e., the experimental design in our context. Finally, we conduct simulation studies on synthetic and real social networks to demonstrate the performances of the proposed methods compared to the random designs, which do not consider the network structure.

In Chapter 3 we continue working with the assumptions of Chapter 2. We focus on the construction of A/B testing experimental designs on social networks, assuming that users connected on the network are similar to a certain degree and that the relationship among those users can be described by CAR model. To conduct the experiment on the test subjects that form the network, we propose an algorithm for re-randomization procedure and formulate stopping criteria that are based on the D-optimal design objectives of the CAR model (Pokhilko et al., 2019). Finally, we conduct simulation studies on synthetic networks to demonstrate the performances of the proposed method compared to the D-optimal designs: original-MIP and modified-MIP (Pokhilko et al., 2019); and random designs, which do not consider the network structure.

In Chapter 4 we develop a Bayesian sequential experimental design for A/B testing on social networks. We consider the case when network members' responses are continuous and we use normal conjugate priors to sequentially update a Bayesian model. Since the subjects participating in these online A/B testing experiments are users who are connected in different scales of social networks, two connected subjects are similar in terms of their social behaviors, education and financial background, and other demographic aspects. Hence, it is only natural to assume that their reactions to online products and services are related to their network adjacency. Therefore, we summarize information about the network into covariates, such as neighborhood size, average A and/or B response in user's first tier neighborhood, and add those covariates into the Bayesian model. We propose an algorithm to sequentially update our model parameters based on new experimental outcomes and use a variance reduction term to choose treatment allocation at each step of the procedure. Finally, we test the performance of our covariate-assisted Bayesian model on synthetic and real-world networks and compare the results to a Bayesian sequential model that does not use network covariates in its posterior updates.

Chapter 2

D-optimal Design for Network A/B Testing

2.1 Introduction

The theory of A/B testing dates back to Ronald Fisher's experiments at the Rothamsted Agricultural Experimental Station in England in the 1920s (Yates, 1964). It deals with the allocation of two treatment options A and B to experimental units, and the analysis of outcome data. A standard statistical A/B testing framework is the Rubin causal model (Rubin, 1974). A key assumption made in the Rubin causal model is the Stable Unit Treatment Value Assumption (SUTVA), which states that the behavior of each test subject in the experiment depends only on the individual treatment and not on the treatments of others. Under this assumption, there are many literature studies on the experimental designs for A/B testing with a given set of covariates associated with each experimental unit, for example, Morgan and Rubin (2012), Hore et al. (2014), and Bertsimas et al. (2015). Also, if a certain model assumption is given, this experimental allocation problem can be done by constructing an optimal design, such as Basse and Airoldi (2015) and Bhat et al. (2017).

Recently, A/B testing has been widely used online to test which alternative or treatment out of the two, A or B, leads to better outcomes. For example, in Xu et al. (2015), the authors described the situations where A/B testing experiments are used by social network sites such as Facebook, LinkedIn, and Twitter to evaluate user engagement or satisfaction from a new service, feature, or product, so as to make data-driven decisions. The treatments could be options of an online commercial, web page designs, different recommendation algorithms, or any new online features that need to be evaluated so that the companies can make informed decisions. The response measures of the experiments can be numerical values of profits, sales, return on investment, click through rate, etc. In these situations, although the covariate information of an individual user is likely to be known to the experimenter, due to the privacy concern of the users or the privacy policy of the companies, the covariate information may not be completely available. On the other hand, the network structure of the users is more likely to be available to the experimenter. Then the problem is to assign those subjects to either the treatment or control group by incorporating the network structure. Many other existing works have also assumed various scenarios where the network structure is known by the experimenter, such as Gui et al. (2015), Eckles et al. (2017), Ogburn et al. (2017), Basse and Airoldi (2018a), Chen et al. (2018), and Nandy et al. (2019).

Much literature on social network focuses on the network interference, which assumes that the potential outcome for a unit is associated with the treatments of its neighboring units. Under this assumption, cluster-based randomized treatment allocation has been developed to tackle this problem. Such examples can be found in Xu et al. (2015), Saveski et al. (2017) and Pouget-Abadie et al. (2017). In this work, we assume that the outcomes of two connected nodes are positively correlated because the features of these two nodes are more similar than those of the unconnected nodes. This assumption rules out the presence of network interference and is referred to as the network-correlated outcomes in Basse and Airoldi (2018b). Basse and Airoldi (2018b) proposed a general framework to deal with design allocation under network-correlated outcomes. The randomized experimental approach is used to minimize the mean squared error of the estimated treatment effect under a class of models. If a reasonable model can be assumed for the effects of the treatment and network (Chen et al., 2018), the classic model-based optimal design (Atkinson et al., 2007; Wu and Hamada, 2011) can also be used for A/B testing experiments. This type of approach also requires the accompanying of an optimization strategy to find such optimal designs. Unfortunately, there has not been much development in this direction and we decide to fill the gap.

In this work, we focus on the construction of A/B testing experimental designs for network-correlated outcomes when users who are connected in a network share some common social and demographic backgrounds. We propose a spatial network model for A/B testing, called conditional auto-regressive model or CAR (Schmidt and Nobre, 2014) to incorporate the correlated network structure in the analysis. To conduct the experiment on the test subjects that form the network, we choose the D-optimal design based on the proposed model. The D-optimal design is the optimal solution of minimizing the determinant of the generalized variance matrix of the parameter estimates for the pre-specified model (Fedorov, 2010; Wu and Hamada, 2011) with respect to the experimental design setting. In the A/B testing experiment, the key parameter is the treatment effect, and thus the D-optimal criterion measures the variance of the estimated treatment effect. Mixed integer programming formulations are developed to optimize the D-optimal utility function with respect to the treatment assignments of the test subjects, i.e., the experimental design in our context. Finally, we conduct simulation studies on synthetic and real social networks to demonstrate the performances of the proposed method compared to the random designs that do not consider the network structure.

2.2 D-Optimal Design for CAR Model

2.2.1 Network A/B Testing with CAR Model

We consider an A/B testing experiment conducted on a social network with n nodes. The social network is considered to be an undirected graph in the context of this paper. The edges of this network are recorded by an $n \times n$ adjacency matrix $W = \{w_{ij}\}$ whose (i,j)-th entry is w_{ij}. The diagonal entries w_{ii}'s of this matrix are zeros, whereas the off-diagonal entries are

$$w_{ij} = \begin{cases} 1, & \text{if node } i \text{ and node } j \text{ are adjacent} \\ 0, & \text{otherwise.} \end{cases}$$

We call that two nodes are adjacent if they are connected by an edge. We denote $m_i = \sum_{j=1}^{n} w_{ij}$ as the degree of the ith vertex, and $N = \sum\limits_{i} m_i = \sum\limits_{i} \sum\limits_{j} w_{ij}$ as twice of the total number of edges of the network. The experimental design is the plan to allocate A or B treatment to each node. Let $\boldsymbol{x} = (x_1, \ldots, x_n)^\top$ with $x_i \in \{1, -1\}$ for $i = 1, \ldots, n$ be the design of the i-th node and the two settings $\{1, -1\}$ represent A and B treatments. We denote the scalar response observation of the i-th node by y_i. In this work, we focus on the case where the response is continuous and assume a linear regression model for the response as follows.

$$y_i = \beta_0 + x_i\beta + \delta_i, \tag{2.1}$$

where β_0 is the intercept, β represents the treatment effect, and δ_i is a zero mean random variable. For the experiments on networks, two connected users share similarities in their social behaviors and other backgrounds, and thus their responses are often correlated. To incorporate this social correlation, we model δ_i in (2.1) by the conditional auto-regressive (CAR) model (Besag, 1974)

$$\delta_i | \boldsymbol{\delta}_{-i} \sim N\left(\rho \sum_{j \neq i} \frac{w_{ij}\delta_j}{m_i}, \frac{\sigma^2}{m_i}\right), \tag{2.2}$$

where $\boldsymbol{\delta}_{-i} = \{\delta_1, \ldots, \delta_{i-1}, \delta_{i+1}, \ldots, \delta_n\}$, and σ^2 is the variance parameter that is assumed to be a constant in our scope, and $|\rho| < 1$ is the correlation parameter of the CAR model. If $\rho = 0$, δ_i's are independent with each other, which corresponds to the extreme case when the network only has n nodes but without any edges. As noted in Section 2.1, the connected users tend to have similar reactions to the same treatment. Hence, without loss of generality, we restrict the correlation parameter to be non-negative, i.e., $0 \leq \rho < 1$.

The covariance model in (2.2) can be expressed in a multivariate normal form given by the following proposition:

Proposition 2.2.1 *By the Brook's Lemma (Brook, 1964),* $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_n)^\top$ *in* (2.2) *follows a multivariate normal distribution:*

$$\boldsymbol{\delta} \sim \mathcal{MVN}_n(0, \sigma^2(D - \rho W)^{-1}) \tag{2.3}$$

where $D = \mathrm{diag}(m_1, \ldots, m_n)$ *with* m_i*'s denoting the degree of the* i*th node.*

The proof of Proposition 2.2.1 is deferred to Appendix 2.7.1. The maximum likelihood method can be used to fit the model (2.1) with δ_i from (2.2) and estimate the model parameters. The goal of A/B testing is to accurately assess the treatment effect β. Next, we determine values of x_i's using D-optimal design to improve the accuracy of the estimate $\hat{\beta}$. D-optimal design has been one of the classical design of experiments methods (Kiefer et al., 1959; Atwood, 1969; Pukelsheim, 1993; Atkinson et al., 2007; Fedorov, 2010; Wu and Hamada, 2011). Based on various model assumption, D-optimal design can be applied to linear regression models (Woods, 2005), generalized linear models and nonlinear models (Atkinson and Woods, 2015; Yang et al., 2013). Here we apply the D-optimal design in order to minimize the variance of the estimated treatment effect β, under the strong model assumption (2.1).

2.2.2 D-optimal Design for CAR Model

In this work, we assume that the network structure is known, i.e., the network adjacency matrix W is known. Given the network structure, the covariance matrix in (2.3) is available, but it still contains unknown parameters σ^2 and ρ. The goal is to develop the D-optimal design under this CAR model assumption. Using a well-known weighted least squares result (Draper and Smith, 1966), we can estimate the parameters β_0 and β in (2.1) by

$$(\hat{\beta}_0, \hat{\beta})^\top = (X^\top V^{-1} X)^{-1} X V^{-1} \boldsymbol{y},$$

where $X = (\mathbf{1}, \boldsymbol{x})$ is an $n \times 2$ design matrix with i-th row $(1, x_i)$, $\boldsymbol{y} = (y_1, \ldots, y_n)^\top$, and $V = (D - \rho W)^{-1}$, and the variance matrix of $(\hat{\beta}_0, \hat{\beta})^\top$ is

$$\mathrm{Var}\left\{(\hat{\beta}_0, \hat{\beta})^\top\right\} = \sigma^2 (X^\top V^{-1} X)^{-1} = \sigma^2 (X^\top (D - \rho W) X)^{-1}. \tag{2.4}$$

Under our model assumption in (2.1), the D-optimal design is determined by maximizing the determinant of the matrix $X^\top(D - \rho W)X$ in (2.4) with respect to $\boldsymbol{x} = (x_1, \ldots, x_n)^\top \in \{-1, 1\}^n$, i.e.,

$$\mathrm{argmax}_{\boldsymbol{x} \in \{-1,1\}^n} \{D(\boldsymbol{x}) := |X^\top (D - \rho W) X|\}. \tag{2.5}$$

Notice that,

$$X^\top (D - \rho W) X = \begin{bmatrix} (1-\rho)N & \sum\limits_i m_i x_i - \rho \sum\limits_i \sum\limits_{j>i} w_{ij}(x_i + x_j) \\ \sum\limits_i m_i x_i - \rho \sum\limits_i \sum\limits_{j>i} w_{ij}(x_i + x_j) & N - \rho \sum\limits_i \sum\limits_j w_{ij} x_i x_j \end{bmatrix},$$

where $N = \sum\limits_i m_i = \sum\limits_i \sum\limits_j w_{ij}$. Then

$$D(\boldsymbol{x}) = (1-\rho)N(N - \rho \sum_i \sum_j w_{ij} x_i x_j) - (1-\rho)^2 (\sum_i m_i x_i)^2. \tag{2.6}$$

Furthermore, since

$$\left\{X^\top(D-\rho W)X\right\}^{-1} = \frac{1}{D(\boldsymbol{x})}\begin{bmatrix} N-\rho\sum\limits_i\sum\limits_j w_{ij}x_ix_j & -(1-\rho)\sum\limits_i m_ix_i \\ -(1-\rho)\sum\limits_i m_ix_i & (1-\rho)N \end{bmatrix},$$

the variance of $\hat{\beta}$ can be expressed by

$$\mathrm{Var}(\hat{\beta}) = \frac{\sigma^2(1-\rho)N}{D(\boldsymbol{x})}.$$

Given a network, $\sigma^2(1-\rho)N$ is a constant. Therefore, the D-optimal design also minimizes the variance of the treatment effect. Based on (2.6), Proposition 2.2.2 gives a simplified objective function to obtain the D-optimal design.

Proposition 2.2.2 *Under model assumptions* (2.1) *and* (2.2)*, the D-optimal design* $\boldsymbol{x}$ *is the solution of the following optimization problem.*

$$\mathrm{argmax}_{\boldsymbol{x}\in\{-1,1\}^n} D(\boldsymbol{x}) = \mathrm{argmin}_{\boldsymbol{x}\in\{-1,1\}^n}\left\{a\sum_i\sum_j w_{ij}x_ix_j + \left(\sum_i m_ix_i\right)^2\right\}, \tag{2.7}$$

where $a = \frac{\rho}{1-\rho}N$.

The proof of this proposition is provided in Appendix A2. We now provide some insights into the desired optimal design. Since the objective function in (2.7) can be decomposed to a weighted sum of $\sum_{ij} w_{ij}x_ix_j$ and $\left(\sum\limits_i m_ix_i\right)^2$, the desired optimal design should achieve smaller values on both of terms. This observation provides insights of the features of the desired optimal design: (1) the first term $\sum_{ij} w_{ij}x_ix_j$ suggests that the two connected nodes should be assigned to different treatments; (2) the second term indicates that if we treat the size of neighborhood as a feature associated with each node, then the design allocation is desired to be "orthogonal" with this feature. Based on (2.7), Proposition 2.2.3 gives an upper bound on D-optimality measure.

Proposition 2.2.3 *Under model assumptions* (2.1) *and* (2.2) *and D-optimal design obtained in Proposition 2.2.2, we have the following upper bound for the D-optimality measure.*

$$D(\boldsymbol{x}) \leq (1-\rho^2)N^2 \tag{2.8}$$

The proof of this proposition is provided in the Appendix A3. Using the upper bound obtained in (2.8), we have the following definition.

Definition 2.2.1 (D-efficiency) *Under model assumptions* (2.1) *and* (2.2) *and upper bound obtained in Proposition 2.2.3, we define D-efficiency measure by:*

$$\frac{D(\boldsymbol{x})}{(1-\rho^2)N^2} \tag{2.9}$$

This D-efficiency measure ranges from 0 to 1, which evaluates the quality of the design without concerning the scale of the $D(\boldsymbol{x})$. A larger value of D-efficiency corresponds to a better design. The definition of the D-efficiency is consistent with the conventional version in literature (Atkinson et al., 2007), which would be the $1/p$th root of (2.9) for p experimental factors. Here $p = 1$ as there is only one experimental factor involved and no other covariates appear in the CAR model nor in the D-optimality.

2.3 Mixed Integer Programming Formulations for D-optimal Design

Since the decision space in (2.7) is $\{-1, 1\}^n$, the optimization problem is an integer programming problem (Wolsey and Nemhauser, 2014). To solve this problem, this section formulates the D-optimal design problem into a mixed integer programming problem with the original objective function in (2.7) and a modified objective function. The two formulations are described in Section 2.3.1 and 2.3.2, respectively.

2.3.1 A Mixed Integer Programming Formulation for D-optimal Design

By observing that $(\sum_i m_i x_i)^2 = \sum_i m_i^2 + \sum_i \sum_{j\neq i} m_i m_j x_i x_j$, we express the objective function in (2.7) by

$$a \sum_i \sum_j w_{ij} x_i x_j + \sum_i \sum_{j\neq i} m_i m_j x_i x_j = \sum_i \sum_{j\neq i} b_{ij} x_i x_j,$$

where $b_{ij} = aw_{ij} + m_i m_j$. By introducing new variables $v_i = (x_i + 1)/2$ and $u_{ij} = v_i v_j$, we can formulate the original optimization problem (2.7) into the following mixed integer linear program (MIP) problem.

$$\min \left[\sum_i \sum_{j\neq i} b_{ij} u_{ij} - \sum_i \sum_{j\neq i} b_{ij} v_i \right] \tag{2.10}$$

$$\begin{aligned} \text{subject to } & u_{ij} \leq v_i, \quad u_{ij} \leq v_j \\ & u_{ij} \geq v_i + v_j - 1, \\ & u_{ij} \in R, \quad u_{ij} \geq 0, \\ & v_i \in \{0, 1\}, \text{ for } i = 1, \ldots, n, j = 1, \ldots, n. \end{aligned}$$

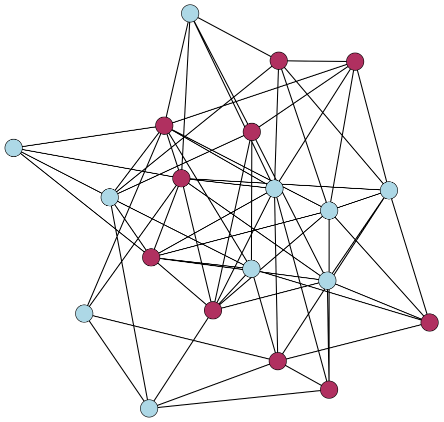

Figure 2.1: D-optimal Design obtained from the mixed integer programming formulation in Section 2.3.1. Two colors represents -1 or 1 allocation.

Figure 2.1 depicts an example of solving this MIP problem to construct a D-optimal design to a network of size 20. Notice that b_{ij} depends on the value of a, which is a function of the correlation parameter ρ in (2.2). It is impractical to assume that ρ can be accurately estimated before data collection. Using the Bayesian framework, we can derive the Bayesian D-optimal design criterion which is the expectation of D-optimality measure $D(\boldsymbol{x})$ with respect to a user-specified prior distribution of ρ. From (2.7), we can see that the expected D-optimality measure only depends on the expected value of the ratio $\rho/(1-\rho)$, which essentially is a tuning parameter. Instead of the Bayesian approach, we decide to take an equivalent route by modifying the objective function and moving this tuning parameter into the constraint.

2.3.2 A Mixed Integer Programming Formulation for a Modified D-optimality Criterion

As we point out above, reducing values of $\sum\limits_i \sum\limits_j w_{ij}x_ix_j$ or $(\sum\limits_i m_ix_i)^2$ would improve D-efficiency defined in (2.9). To remove the parameter a from the objective function in (2.7), an alternative solution is to use $\sum\limits_i \sum\limits_j w_{ij}x_ix_j$ as the objective function, and bound the value of $(\sum\limits_i m_ix_i)^2$ by a constraint. We modify the optimization problem in (2.7) to be

$$\min\nolimits_{\boldsymbol{x}\in\{-1,1\}^n} \sum_i \sum_j w_{ij}x_ix_j \tag{2.11}$$

$$\text{s.t.} \quad -\phi \le \sum_i m_ix_i \le \phi, \text{ and } x_i \in \{-1,1\} \text{ for } i = 1,\dots,n,$$

where $\phi > 0$ is a prespecified parameter.

Now we discuss how to specify the tuning parameter ϕ. For different networks, the ranges of $\sum\limits_i m_ix_i$ can be different. So we need to transfer the value of ϕ by normalizing it to a unified range for different networks. We derive the suitable range of ϕ by considering the theoretical quantiles of $\sum\limits_i m_ix_i$ when x_i's are randomly generated. Assume that x_i is a random variable taking value from $\{-1,1\}$ with equal weights. Hence,

$$\mathrm{E}\left(\sum_{i=1}^{n} m_i x_i\right) = \sum_{i=1}^{n} m_i \mathrm{E} x_i = 0 \text{ and } \mathrm{Var}\left(\sum_{i=1}^{n} m_i x_i\right) = \sum_{i=1}^{n} m_i^2 \mathrm{Var}(x_i) = \sum_{i=1}^{n} m_i^2.$$

If $n^{-2}\sum_{i=1}^{n} m_i^2 < \infty$ and $n^{-1}m_i \to 0$ for $i = 1, \dots, n$, we have that $\sum_{i=1}^{n} m_i x_i / \sqrt{\sum_{i=1}^{n} m_i^2}$ asymptotically follows a standard normal distribution by the Lindeberg's Central Limit Theorem. Therefore, as $n \to \infty$,

$$\mathrm{P}\left(-\Phi^{-1}(\alpha)\sqrt{\sum_i m_i^2} < \sum m_i x_i < \Phi^{-1}(\alpha)\sqrt{\sum_i m_i^2}\right) = 2\alpha - 1,$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution, and $\alpha \in (0.5, 1)$. By

$$\phi = \Phi^{-1}(\alpha)\sqrt{\sum_i m_i^2},$$

ϕ is increasing with α. After this transformation, the parameter ϕ has been re-scaled to α, whose range is not associated with the network structure.

Define v_i and u_i for $i = 1, \dots, n$ in the same way as in Section 2.3.1, the problem in (2.11) becomes

$$\begin{aligned} &\min \quad \left[\sum_i \sum_{j \neq i} w_{ij} u_{ij} - \sum_i \sum_{j \neq i} w_{ij} v_i\right] \qquad (2.12)\\ &\text{subject to } \sum_i m_i v_i \leq \frac{1}{2}\left(N + \Phi^{-1}(\alpha)\sqrt{\sum_i m_i^2}\right)\\ &\sum_i m_i v_i \geq \frac{1}{2}\left(N - \Phi^{-1}(\alpha)\sqrt{\sum_i m_i^2}\right)\\ &u_{ij} \leq v_i, \quad u_{ij} \leq v_j\\ &u_{ij} \geq v_i + v_j - 1,\\ &u_{ij} \in R, \quad u_{ij} \geq 0,\\ &v_i \in \{0, 1\}, \text{ for } i = 1, \dots, n, j = 1, \dots, n. \end{aligned}$$

The solution of (2.12) gives an approximate D-optimal design under a prespecified α value. In practice, we may obtain designs from multiple α values and the corresponding D-efficiencies defined in (2.9). We can choose the design that returns the highest D-efficiency.

As a summary, this section proposed two MIP-based approach, and we provide their definition as follows:

Definition 2.3.1 *Original-MIP design is an exact D-optimal design generated by solving reformulated mixed-integer optimization problem in* (2.10)*.*

Definition 2.3.2 *Modified-MIP design is an approximate D-optimal design generated by solving reformulated mixed-integer optimization problem in* (2.12)*.*

Here are some remarks on the differences between those two reformulated optimization problems. First, solving the original-MIP in Section 2.3.1 gives the exact D-optimal design, whereas solving the modified-MIP in Section 2.3.2 does not guarantee that the exact D-optimal design can be found. Second, the original-MIP requires that the correlation parameter ρ to be known, whereas the modified-MIP does not. Third, the original numbers of decision variables in both formation are $n(n-1)/2+n$. However, u_{ij} can be removed if the corresponding coefficient (b_{ij} in original-MIP or w_{ij} in modified-MIP) is 0. Since w_{ij} is more likely to be zero than b_{ij}, we expect that the number of decision variables of the formulation for the modified-MIP is often much smaller than that of the original-MIP. Although both programming formulations are NP-hard, our observation based on the simulation study in Section 2.4 is that the reduction of the number of decision variables often leads to less computation in solving MIP.

2.4 Numerical Study on Synthetic Networks

This section compares three methods: original-MIP in Definition 2.3.1, modified-MIP in Definition 2.3.2, and random design. In this work, the random design is referred to as the procedure that randomly allocates -1 or 1 with equal probability to each node.

We generate random networks to compare the three methods. For a network with n nodes, the $n \times n$ adjacency matrix records the edges of this network. We randomly assign 0 and 1 to the upper or lower off-diagonal entries of this matrix. The proportion of ones is specified to be p. As defined in Section 2.2.1, the zero entry means that the two corresponding nodes are not adjacent, whereas one entry means the opposite. The proportion p is referred to as the density of this network. Once we construct the designs using the three methods for a given network, we compare the designs on three aspects, computational efficiency, D-efficiency, and the empirical variance of the estimated β in (2.1).

2.4.1 Computational Efficiency

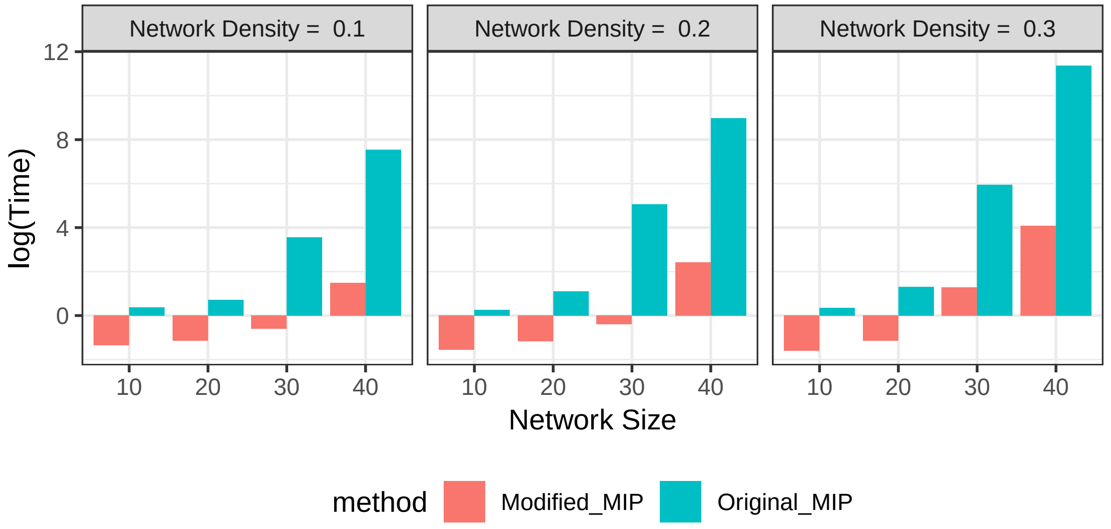

Figure 2.2: Logarithm of running times in seconds of original-MIP and modified-MIP. For original-MIP, $\rho = 0.2$; for modified MIP, $\alpha = 0.6$.

In terms of computational efficiency, we compare the computational time of solving the objective functions of original-MIP and modified-MIP with GUROBI solver (`http://www.gurobi.com/`). We set the network correlation coefficient to be $\rho = 0.2$ for the original-MIP and the tuning parameter to be $\alpha = 0.6$ for the modified-MIP. For original-MIP, the longest allowable running time is 24 hours, whereas for modified-MIP, the longest allowable running time is 4 hours. Exceeding that limit, the solver is terminated whether it reaches the optimal solution or not, and we report the optimality gap:

$$\text{optimality gap} = \frac{ub - lb}{ub},$$

where ub and lb are estimated upper and lower bounds of the objective function. A smaller optimality gap indicates that the objective value corresponding to the current solution is closer to the true optimal objective value. As pointed out earlier, the number of decision variables of the modified-MIP is likely much smaller than that of original-MIP, thus the modified-MIP would take less time to run than original-MIP. To confirm this, the running time (in seconds) of these two methods with networks of different sizes are given in Figure 2.2. It shows that the running time required to solve original-MIP increases dramatically as both network size and density increase. For the small networks in this simulation, the original-MIP is already time-consuming thus it is not practical to be applied to the real-world social networks whose size is usually of thousands. Since the tuning parameter α determines the feasible region of the modified-MIP, we conduct an additional simulation whose results are shown in Table 2.4 in Appendix 2.7.4. It implies that the value of α does not affect the computational time of modified-MIP significantly in our numerical experiments.

2.4.2 D-efficiency

The section compares the D-efficiency of the three design methods, whose value in (2.9) depends on the true correlation parameter ρ of the CAR model. We randomly generate a network of size $n = 50$ and density $p = 0.1$. For the modified-MIP, we vary the value of α from 0.6, 0.7, to 0.8, and the running time of each case is less than one minute in GUROBI. For the random design, because $\mathrm{P}(x_i = 1) = \mathrm{P}(x_i = -1) = 1/2$, the expected D-efficiency is given by

$$\frac{N^2 - (1-\rho)\sum_{i=1}^{n} m_i^2}{(1+\rho)N^2},$$

which only depends on the network. The optimal criterion of the original-MIP contains an unknown parameter ρ. We specify $\rho = 0.2$ to generate the design, whereas in computing the D-efficiency, we vary the true value of ρ from 0 to 0.9. The optimality gap of the original-MIP after 24 hours is 10.0863%.

Table 2.1 gives the D-efficiency values. Both MIP based methods are better than the random design, especially when the correlation parameter is not zero. Also, the D-efficiency of modified-MIP under different α values is comparable with the original-MIP method.

Table 2.1: The D-Efficiency measures of different methods for a network of size $n = 50$ with density $p = 0.1$

Method	$\rho = 0$	$\rho = 0.1$	$\rho = 0.2$	$\rho = 0.3$	$\rho = 0.6$	$\rho = 0.8$	$\rho = 0.9$
Random	0.98	0.90	0.82	0.76	0.62	0.55	0.53
Original-MIP	1.00	0.96	0.93	0.90	0.84	0.80	0.79
Modified-MIP ($\alpha = 0.6$)	1.00	0.96	0.93	0.90	0.84	0.80	0.79
Modified-MIP ($\alpha = 0.7$)	1.00	0.96	0.92	0.90	0.83	0.80	0.79
Modified-MIP ($\alpha = 0.8$)	1.00	0.96	0.92	0.90	0.83	0.80	0.79

2.4.3 Empirical Variance

We now compare the designs in terms of the empirical variance of the estimates $\hat{\beta}$. Using the random network we investigated in Table 2.1, and a pre-specified correlation coefficient ρ, we generate $\boldsymbol{\delta}$ in (2.2) from the multivariate normal distribution as expressed in (2.3). Set the values of the parameters as $\beta_0 = 0$, $\beta = 2$ and $\sigma^2 = 1$ in the model (2.1). Given the design of x_i, we generate the response y_i for each node as in (2.1) and (2.2). Recall that, our response are generated from CAR model. We consider $\hat{\beta}$ from the CAR model as the estimates for all three methods. For random design, we also include a modeling alternative linear regression model (LM), which is not the true model in our comparison. For the CAR model, the parameters β_0,

β and ρ are estimated by the maximum likelihood method using `R` package `spdep` (Bivand et al., 2005). By repeating this procedure for $R = 500$ times, we measure the accuracy of an estimate by calculating its sample variance :

$$\hat{\mathrm{V}}(\hat{\beta}) = \frac{1}{R-1}\sum_{l=1}^{R}\left(\hat{\beta}_l - \frac{1}{R}\sum_{k=1}^{R}\hat{\beta}_k\right)^2 ,$$

where $\hat{\beta}_1, \ldots, \hat{\beta}_R$ are the $R = 500$ copies of the estimates for β. Under our response generation scheme, the bias of the estimate $\hat{\beta}$ is tiny for all the methods. Hence, the mean squared error of $\hat{\beta}$ is dominated by the variance of $\hat{\beta}$, and the bias is negligible. So we ignore the bias and report empirical variance only. Notice that the modified-MIP and the original-MIP generate deterministic designs for a given network, to the contrary of the random design. To make the three approaches comparable, 100 random designs are generated and we report the average value of $\hat{\mathrm{V}}(\hat{\beta})$ over the 100 random designs. The empirical variance values of $\hat{\beta}$ for each method are given in Table 2.2. As in Table 2.1, we also vary the value of the tuning parameter α from 0.6, 0.7, to 0.8 for the modified-MIP, and fix $\rho = 0.2$ for the original-MIP to generate design. The value of ρ used in design generation is not equal to its real value for generating responses when $\rho = 0$, 0.1, 0.2, 0.3, 0.6, 0.8 and 0.9 in Table 2.2. For the random design, we consider the β estimated from both CAR and LM, although the data are generated from the CAR model. We summarize our main observations from Table 2.2 as follows:

- For a non-zero correlation coefficient (i.e.,$\rho > 0$), the variances of MIP based methods are smaller than the random design.
- The variances resulted from modified-MIP are comparable with those from original-MIP.

As shown in Table 2.2, the performance of modified-MIP is comparable with that of original-MIP for small networks. But it is much more practical because it takes less time to generate a design using GUROBI and does not require the correlation coefficient value ρ. Even though it does involve the prespecified parameter α, its

Table 2.2: Empirical variances of $\hat{\beta}$ based on designs generated from different methods for a network of size $n = 50$ and density $p = 0.1$

Method	$\rho = 0$	$\rho = 0.1$	$\rho = 0.2$	$\rho = 0.3$	$\rho = 0.6$	$\rho = 0.8$	$\rho = 0.9$
Original-MIP	0.0046	0.0043	0.0041	0.0038	0.0033	0.0030	0.0029
Modified-MIP ($\alpha = 0.6$)	0.0044	0.0043	0.0040	0.0037	0.0033	0.0031	0.0030
Modified-MIP ($\alpha = 0.7$)	0.0049	0.0043	0.0043	0.0038	0.0034	0.0030	0.0030
Modified-MIP ($\alpha = 0.8$)	0.0047	0.0042	0.0042	0.0038	0.0034	0.0030	0.0029
Random (CAR)	0.0047	0.0047	0.0046	0.0046	0.0046	0.0048	0.0047
Random (LM)	0.0045	0.0045	0.0044	0.0045	0.0047	0.0051	0.0052

performance is robust to the value of α. So we choose $\alpha = 0.6$ in remaining numerical examples and only compare the random design and modified-MIP for larger networks.

Finally we test our method on larger networks. Since the original-MIP was not computationally efficient on networks with more than 50 nodes, we compare the modified-MIP design with random designs for large networks. We calculate the variance of $\hat{\beta}$ over 500 replications and use different values of ρ in the generation of response vectors. For modified-MIP, the optimality gap achieves 38.52% after four hours. From Table 2.3, the variance of $\hat{\beta}$ was lower for modified-MIP design in comparison to random designs. As ρ increased, the results show improved advantages of modified-MIP compared to random designs.

Table 2.3: Empirical variances of $\hat{\beta}$ based on designs generated from different methods for a network of size $n = 500$ and density $p = 0.01$

Method	$\rho = 0$	$\rho = 0.1$	$\rho = 0.2$	$\rho = 0.3$	$\rho = 0.6$	$\rho = 0.8$	$\rho = 0.9$
Modif-MIP (CAR)	0.00042	0.00035	0.00039	0.00031	0.00030	0.00029	0.00028
Random (CAR)	0.00041	0.00041	0.00041	0.00041	0.00041	0.00041	0.00042
Random (LM)	0.00041	0.00041	0.00041	0.00041	0.00043	0.00046	0.00048

2.5 Numerical Study on Real-world Networks

This section studies two examples of real-world networks and compares the performance of our proposed modified-MIP design in Definition 2.3.2 to the performance of random designs. Section 2.5.1 considers examples of ego networks from Facebook. Section 2.5.2 considers examples of large networks containing multiple clusters.

2.5.1 Ego Networks

We consider seven ego networks extracted from Facebook (Leskovec and Mcauley, 2012). These networks are collected from survey participants who used the Facebook App. Ego networks consist of a focal node ("ego") and the nodes to whom ego is directly connected to (these are called "alters") plus the ties, if any, among the alters. Each alter in an ego network has his/her own ego network, and all ego networks interlock to form the human social network. The focal node represents the main character or person in the ego network and the alters are the "friends" in his circle or people he is connected with. Those friends can also be connected among themselves, forming their own ego networks, and that is represented by the ties or edges among alters. The sizes of these seven ego networks are $n = 52$, 61, 168, 224, 333, 534 and 1034 respectively. The densities of them range from 0.03 to 0.15. We use the modified-MIP with $\alpha = 0.6$ to construct the designs. As in Section 2.4, the maximum running time is set to be 4 hours in GUROBI. The optimality gaps of these networks are 0%, 0%, 33.6%, 54.67%, 48.56%, 58.99% and 90.67%. Figure 2.3 depicts the designs generated from the modified-MIP for the ego networks of size 52 and 61.

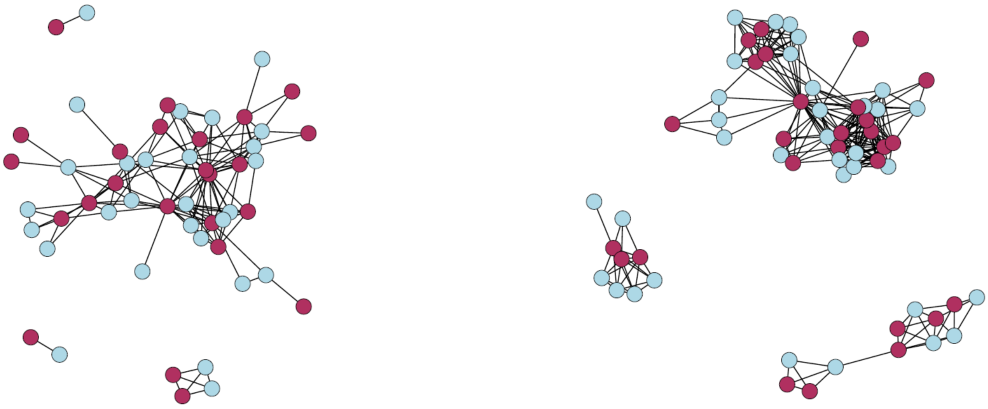

Figure 2.3: Left: ego network of size 52; right: ego network of size 61. The two different colors represent design allocation to -1 or 1 using modified-MIP

After obtaining designs for each ego network, we evaluate the empirical variances of $\hat{\beta}$, which is calculated the same as in Section 2.4. The true correlation parameter

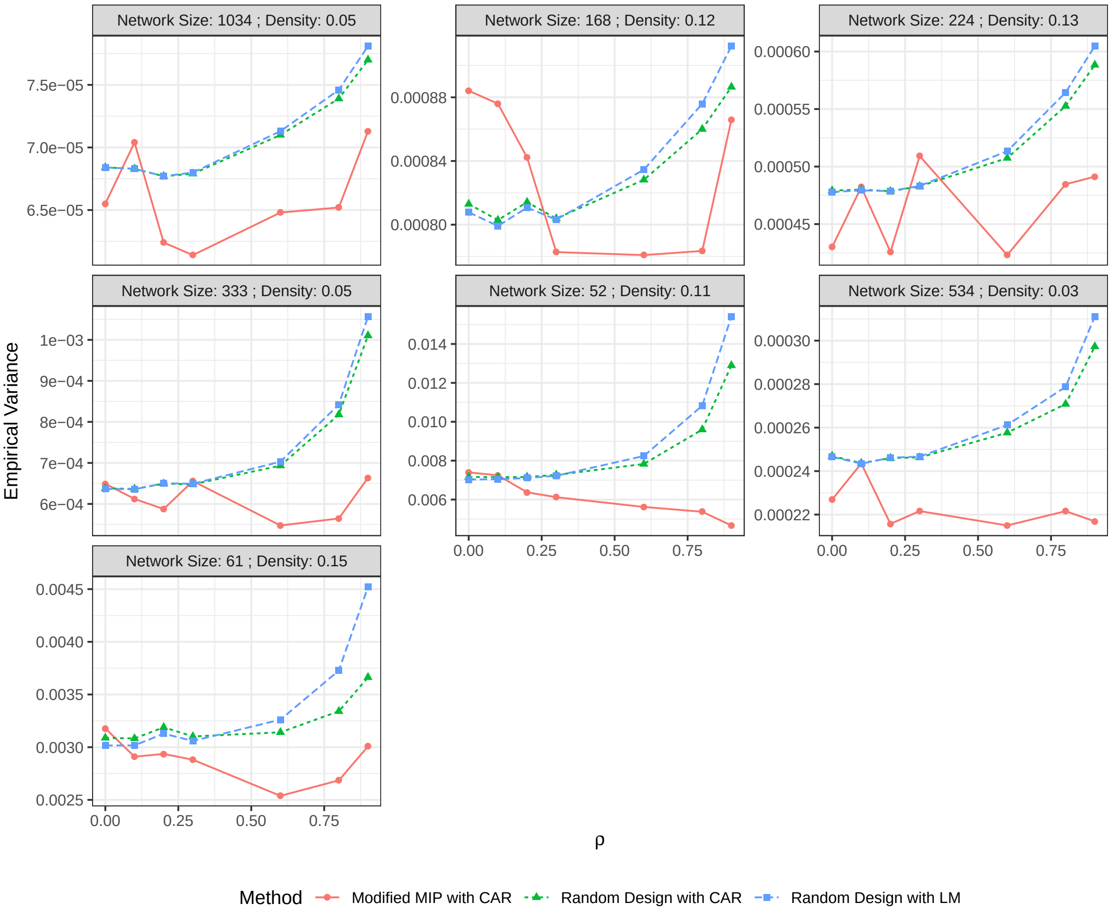

Figure 2.4: Empirical variances of $\hat{\beta}$ of seven ego networks. MIP running time 4 hours for all ego-networks.

ρ is specified to be 0, 0.1, 0.2, 0.3, 0.6, 0.8 and 0.9. Three methods are compared: 1) the modified-MIP with $\hat{\beta}$ estimated using the CAR model; 2) the random design with $\hat{\beta}$ estimated using the CAR model; 3) the random design with $\hat{\beta}$ estimated using a linear regression model (LM). The results are provided in Figure 2.4. We can see that in most cases the modified-MIP gives smaller variances than the random design when ρ grows large. However, the advantage of modified-MIP is not significant when the correlation parameter ρ is as small as 0.1. We also recorded additional simulation results on the performance of the modified-MIP designs with $\hat{\beta}$ estimated using a linear regression model (LM) in Figure 2.7 of Appendix 2.7.4.

2.5.2 Large Networks with Disjoint Clusters

For extremely large networks (i.e., more than 10,000 nodes), it is impossible to generate exact optimal design based on MIP. An alternative is to separate those large networks into relatively small networks, compute the optimal design for each small networks, and then combine the allocation results from all the small networks to conduct analysis. We consider a sample from Facebook network available at `https://snap.stanford.edu/data/`. To sample the network, we randomly selected K disjoint clusters or sub-networks, each cluster with approximately 50 nodes. Each sub-network is disjoint, it consists of separate groups of individuals, who may be related to each other within each sub-network, but have no relations to each other between different sub-networks. The K clusters form a network on which we apply our design. We pick $K = 20$, 30, and 40 and the corresponding networks consist of approximately 1000 to 2000 nodes. Then, we simulate the responses for each cluster of the large network. For each cluster, the correlation parameter ρ is randomly generated from a uniform distribution. We consider two scenarios: $\rho \sim U(0.15, 0.25)$ and $\rho \sim U(0.7, 0.9)$. Other parameters are specified the same as in Section 2.4. The empirical variances are computed the same way as in Section 2.4. The estimates of β can be obtained by fitting the CAR model or a linear regression model. Repeating this procedure 100 times, we assess the accuracy of the estimates by their sample variances.

We now discuss how to generate the MIP based designs for such large networks with disjoint clusters. For a network of size over 500, it is nearly impossible to directly obtain the MIP based designs with optimality gap of zero. Since the large network is constructed by disjoint clusters, we can alternatively generate the MIP based design for each smaller cluster, and then combine them to obtain the design for the entire large network. Let $W_1, \ldots, W_K$ be the adjacency matrices of K disjoint clusters. For each of them, we generate the two-level design using the modified-MIP described in Section 3.2. We denote the resulted design as x_{ik} for $k = 1, \ldots, K$ and $i = 1, \ldots, n_k$, where $x_{ik} \in \{-1, 1\}$ is the treatment assignment for the i-th node

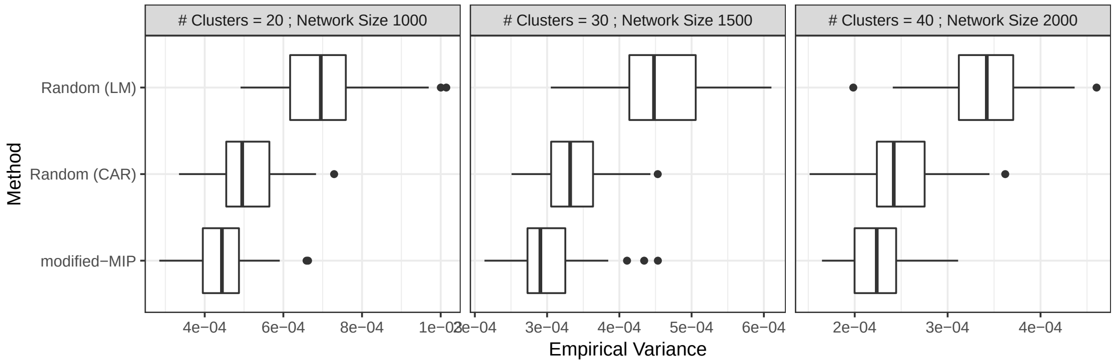

Figure 2.5: Boxplots of Empirical variances of $\hat{\beta}$ for large networks with clusters. In response generation we used $\rho \sim U(0.15, 0.25)$.

from the k-th network. Therefore, a combined design for the large network can be $\{c_k x_{ik} : i = 1, \dots, n_k, k = 1, \dots K\}$, where $c_k \in \{-1, 1\}$. By varying the choices of c_k for $k = 1, \dots, K$, the total number of combined designs for the large network is 2^K. We take an additional step to choose the optimal value of c_k's by minimizing

$$\text{argmin}_{\boldsymbol{c} \in \{-1,1\}^K} \left(\sum_{k=1}^{K} c_k \sum_{i=1}^{n_k} m_{ik} x_{ik} \right)^2,$$

where $\boldsymbol{c} = (c_1, \dots, c_K)$, and m_{ik} is the total number of neighbors for the i-th node from the k-th network, which leads that $\sum_{i=1}^{n_k} m_{ik} x_{ik}$ is a known constant. The optimal solution of $\boldsymbol{c}$ gives the smallest value of $(\sum_i m_i x_i)^2$ in (2.7) for the large network. This optimization problem is the same as original-MIP in Section 2.3.1. For $K = 20$, 30 or 40, this problem can be solved by GUROBI efficiently.

For $K = 20$, 30 or 40, we generate 100 large networks with disjoint clusters as described above and compare the empirical variances of $\hat{\beta}$ of the three methods as in Section 2.5.1. The empirical variances of the 100 large networks for each K and different ρ used in response generation are depicted in Figures 2.5 and 2.6. In all the cases we see that the median of empirical variances of modified-MIP is the smallest. According to the results from the Wilcox's rank sum test, the median of modified-MIP is significantly smaller than that of the random design for each case. The improvement of using modified-MIP is more significant in 2.6 with $\rho \sim U(0.7, 0.9)$ than 2.5 with $\rho \sim U(0.15, 0.25)$.

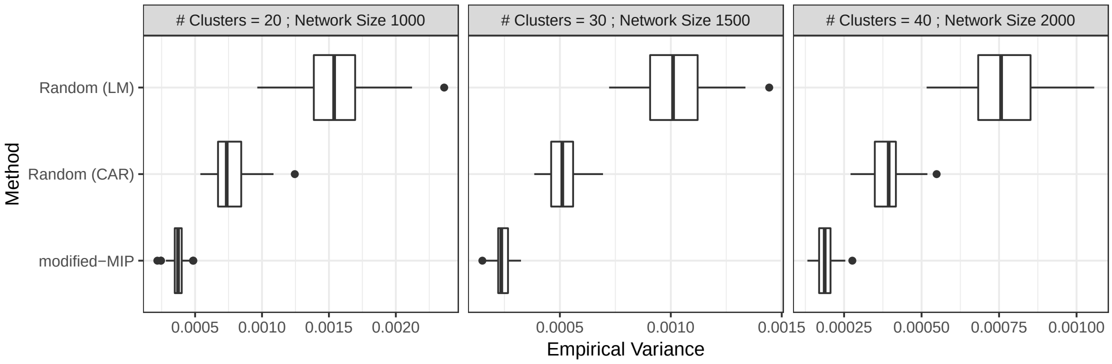

Figure 2.6: Boxplots of Empirical variances of $\hat{\beta}$ for large networks with clusters. In response generation we used $\rho \sim U(0.7, 0.9)$.

2.6 Discussion

This chapter proposes using the D-optimal design for A/B testing conducted on a social network. We use the CAR model to characterize the dependence of the responses from the adjacent nodes in a network. Mixed integer programming formulations are proposed to solve the D-optimal objectives and construct the designs. Numerical examples are provided to show the effectiveness of the proposed method. Notice that our D-optimal design is proposed under the CAR model assumption with the given network structure. Readers should be cautious to examine the CAR model assumption when the data are collected, which can be done through procedures such as the likelihood ratio test. When a network model assumption is not available, a randomized design is a robust strategy. If the network structure is unknown, we can impose a Gaussian graphical prior to the network structure, and then develop the optimal design to incorporate the uncertainty of network structure.

In addition to CAR, spatial autoregressive model (SAR, Wall (2004)) is another popular alternative to model the network-based outcome correlation. The major difference between SAR and CAR lies in their the covariance matrices. As noted earlier, the covariance matrix of $\boldsymbol{\delta}$ under CAR can be expressed in (2.3). If we simply assume that the error term $\boldsymbol{\delta}$ in (2.3) has a covariance matrix as in SAR, Wall (2004) indicates that the covariance matrix becomes $\sigma^2(I_n - \rho W)^\top(I_n - \rho W)$, where I_n is the $n \times n$ identity matrix. Therefore, its corresponding formulation of the

D-optimality measure $D(\boldsymbol{x})$ will still be a quadratic function of $\boldsymbol{x}$. The parameters of this quadratic function is different from the objective function we have obtained in (2.6). Under this situation, the MIP formulation developed in Section 2.3.1 can still be used to solve the optimal design problem for SAR. However, the computational efficient modified-MIP formulation in Section 2.3.2 will not hold for SAR any more.

2.7 Appendix

2.7.1 The proof of Proposition 2.2.1

We first state the Brook's Lemma (Brook, 1964).

Lemma 2.7.1 (Brook's Lemma) *Let $\boldsymbol{Y} \in R^n$ be a vector of n random variables, $\boldsymbol{Y} = (Y_1, Y_2, \ldots, Y_n)^\top$. Define $\boldsymbol{Y}_0 = (Y_{01}, Y_{02}, \ldots, Y_{0n})^\top \in R^n$, where $Y_{01}, Y_{02}, \ldots, Y_{0n}$ are copies of realizations of $Y_1, Y_2, \ldots, Y_n$. The following result holds.*

$$\frac{P(\boldsymbol{Y})}{P(\boldsymbol{Y}_0)} \propto \prod_{i=1}^{n} \frac{P(Y_i | Y_{01}, \ldots, Y_{0i-1}, Y_{i+1}, \ldots, Y_n)}{P(Y_{0i} | Y_{01}, \ldots, Y_{0i-1}, Y_{i+1}, \ldots, Y_n)},$$

where $P(\cdot)$ denotes the probability density function, and $P(\cdot|\cdot)$ denotes the conditional probability density function.

Let $\tilde{w}_{ij} = w_{ij}/m_i$, $\sigma_i^2 = \sigma^2/m_i$ and $\boldsymbol{\delta_0} = (\delta_{01}, \delta_{02}, \ldots, \delta_{0n})^\top$ is a realization of $\boldsymbol{\delta}$ with probability density function $P(\boldsymbol{\delta})$. Using Brook's Lemma and model assumptions (2.1) and (2.2), we have that

$$\frac{P(\boldsymbol{\delta})}{P(\boldsymbol{\delta}_0)} \propto \prod_{i=1}^{n} \frac{P(\delta_i | \delta_{01}, \ldots, \delta_{0i-1}, \delta_{i+1}, \ldots, \delta_n)}{P(\delta_{0i} | \delta_{01}, \ldots, \delta_{0i-1}, \delta_{i+1}, \ldots, \delta_n)} \propto \prod_{i=1}^{n} \frac{\exp\{-\frac{1}{2\sigma_i^2}(\delta_i - \rho\sum_{j>i} \tilde{w}_{ij}\delta_j)^2\}}{\exp\{-\frac{1}{2\sigma_{0i}^2}(0 - \rho\sum_{j>i} \tilde{w}_{ij}\delta_j)^2\}}$$
$$\propto \exp\{\sum_{i=1}^{n}[-\frac{1}{2\sigma_i^2}(\delta_i - \rho\sum_{j>i}\tilde{w}_{ij}\delta_j)^2 + \frac{1}{2\sigma_i^2}(\rho\sum_{j>i}\tilde{w}_{ij}\delta_j)^2]\}$$
$$\propto \exp\{\sum_{i=1}^{n}[-\frac{1}{2\sigma_i^2}(\delta_i^2 - 2\rho\delta_i\sum_{j>i}\tilde{w}_{ij}\delta_j + (\rho\sum_{j>i}\tilde{w}_{ij}\delta_j)^2 - (\rho\sum_{j>i}\tilde{w}_{ij}\delta_j)^2)]\}$$
$$\propto \exp\{\sum_{i=1}^{n}[-\frac{1}{2\sigma_i^2}(\delta_i^2 - 2\rho\delta_i\sum_{j>i}\tilde{w}_{ij}\delta_j)]\} \propto \exp\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n} m_i(\delta_i^2 - 2\rho\delta_i\sum_{j>i}\tilde{w}_{ij}\delta_j)\}$$
$$\propto \exp\{-\frac{1}{2\sigma^2}(\sum_{i=1}^{n} m_i\delta_i^2 - 2\rho\sum_{i=1}^{n}\sum_{j>i}\delta_i m_i\tilde{w}_{ij}\delta_j)\}$$
$$\propto \exp\{-\frac{1}{2\sigma^2}(\sum_{i=1}^{n} m_i\delta_i^2 - 2\rho\sum_{i=1}^{n}\sum_{j>i}\delta_i\frac{m_i w_{ij}}{m_i}\delta_j)\}$$
$$\propto \exp\{-\frac{1}{2\sigma^2}(\sum_{i=1}^{n} m_i\delta_i^2 - 2\rho\sum_{i=1}^{n}\sum_{j>i}\delta_i w_{ij}\delta_j)\} \propto \exp\{-\frac{1}{2\sigma^2}\boldsymbol{\delta}^T(D - \rho W)\boldsymbol{\delta}, \}$$

which is proportional to the probability density function of the multivariate normal distribution $\mathcal{MVN}_n(0, \sigma^2(D - \rho W)^{-1})$. Therefore,

$$\boldsymbol{\delta} = (\delta_1, \ldots, \delta_n)^\top \sim \mathcal{MVN}_n(0, \sigma^2(D - \rho W)^{-1}).$$

2.7.2 Proof of Proposition 2.2.2

Following the expression in (2.5), we have that

$$\begin{aligned}
&\max\nolimits_{\boldsymbol{x} \in \{-1,1\}^n}[D(\boldsymbol{x})] \\
&= \max\nolimits_{\boldsymbol{x} \in \{-1,1\}^n}[(1-\rho)N(\sum_i m_i x_i^2 - \rho \sum_i \sum_j w_{ij} x_i x_j) - (1-\rho)^2(\sum_i m_i x_i)^2] \\
&= \max\nolimits_{\boldsymbol{x} \in \{-1,1\}^n}[N^2 - \rho N \sum_i \sum_j w_{ij} x_i x_j - (1-\rho)(\sum_i m_i x_i)^2],
\end{aligned}$$

which is equivalent to

$$\begin{aligned}
&= \min\nolimits_{\boldsymbol{x} \in \{-1,1\}^n} \left[\rho N \sum_i \sum_j w_{ij} x_i x_j + (1-\rho)(\sum_i m_i x_i)^2 \right] \\
&= \min\nolimits_{\boldsymbol{x} \in \{-1,1\}^n} \left[a \sum_i \sum_j w_{ij} x_i x_j + (\sum_i m_i x_i)^2 \right],
\end{aligned}$$

where $\boldsymbol{x} = (x_1, \ldots, x_n)$, $x_i \in \{-1, 1\}$, and $a = \frac{\rho}{1-\rho} N$.

2.7.3 Proof of Proposition 2.2.3

Since a is non-negative, a lower bound of the objective function in (2.7),

$$\min_{\boldsymbol{x} \in \{-1,1\}^n} \left\{ a \sum_i \sum_j w_{ij} x_i x_j + (\sum_i m_i x_i)^2 \right\},$$

can be attained by minimizing $\sum\limits_i \sum\limits_j w_{ij} x_i x_j$ and $\left(\sum\limits_i m_i x_i \right)^2$, separately. The lower bound of $\sum\limits_i \sum\limits_j w_{ij} x_i x_j$ is $-\sum\limits_i \sum\limits_j w_{ij}$ if $x_i \neq x_j$ for all (i, j) with $w_{ij} = 1$. The lower bound of $\left(\sum\limits_i m_i x_i \right)^2$ is zero if $\sum\limits_i m_i x_i = 0$, which represents that the nodes allocated with -1 and the nodes allocated with 1 have equal number of first order neighborhoods. Equivalently, we can obtain an upper bound for the D-optimality measure

$$D(\boldsymbol{x}) \leq (1-\rho)(\sum_{i=1}^{n} m_i)^2 + (1-\rho)\rho \sum_{i=1}^{n} m_i \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} = (1-\rho^2)N^2.$$

2.7.4 Additional Numerical Results on the Computational Times of Modified-MIP

Table 2.4 gives the time (in seconds) took for GUROBI to solve modified-MIP for different network sizes and different choices of α. Similar to the original-MIP, the running times of modified-MIP increase for larger networks. According to the formulation in (2.12), smaller α values correspond to tighter constraints. From the instances in Table 2.4, the computational times of modified-MIP tend to be robust to the value of α.

Table 2.4: Running times (in sec) of modified-MIP for different values of α averaged across 20 network sizes n keeping network density $p = 0.1$.

α	$n = 20$	$n = 30$	$n = 40$	$n = 50$
0.55	0.47	1.63	3.61	13.26
0.6	0.58	1.31	3.95	24.80
0.7	0.57	1.40	3.67	13.65
0.8	0.60	1.43	2.12	21.65
0.9	0.58	1.47	4.36	17.08

2.7.5 Numerical Results under Model Misspecification

This section provides some numerical results under model misspecification. In Section 2.7.5.1 we evaluate the robustness of our approach when the true correlation parameters are misspecified under a Gaussian distribution. Section 2.7.5.2 provides a case with non-Gaussian outcomes.

2.7.5.1 Model Misspecification under Gaussian Distribution

Following the same numerical setting in Table 2.3, Table 2.5 provides simulation results conducted on the modified-MIP with $\hat{\beta}$ estimated using a linear regression model (LM). In this Table, the column $\rho = 0$ corresponds to the case that the responses are generated from a linear model without correlation structure.

When the outcomes are generated from a linear model (i.e., $\rho = 0$), LM is the true model, whereas CAR is the misspecified model. However, the empirical variances of CAR and LM are the same under modified-MIP or Random design. Since there is no network correlation under $\rho = 0$, modified-MIP can not outperform random design. When the outcomes are generated from the CAR model with $\rho > 0$, the CAR model always leads smaller empirical variances than LM when ρ is relatively large under both modified-MIP and Random design. Also, modified-MIP has better performances compared to the random design for both models.

Table 2.5: Empirical variances of $\hat{\beta}$ based on Linear model-misspecification for a network of size $n = 500$ and density $p = 0.01$

Method	$\rho = 0$	$\rho = 0.1$	$\rho = 0.2$	$\rho = 0.3$	$\rho = 0.6$	$\rho = 0.8$	$\rho = 0.9$
Modif-MIP (CAR)	0.00042	0.00035	0.00039	0.00031	0.00030	0.00029	0.00028
Modif-MIP (LM)	0.00042	0.00036	0.00040	0.00032	0.00032	0.00033	0.00031
Random (CAR)	0.00041	0.00041	0.00041	0.00041	0.00041	0.00041	0.00042
Random (LM)	0.00041	0.00041	0.00041	0.00041	0.00043	0.00046	0.00048

Figure 2.7 compares empirical variances of $\hat{\beta}$ simulated from seven ego networks described in Section 2.5.1. In this figure, we include the results from LM under the design of Modified-MIP. As in Table 2.5, $\rho = 0$ is associated with the situation that the outcomes are generated from a linear model. The figure shows similar information with Table 2.5: 1) when the outcomes are generated without network correlation, the design approach which includes network correlation (Modified-MIP) may not significantly worsen the performance compared to the random design for most networks; 2) when the outcomes are generated with a relatively large network correlation (say, $\rho > 0.5$), Modified-MIP always improves the performances of random design.

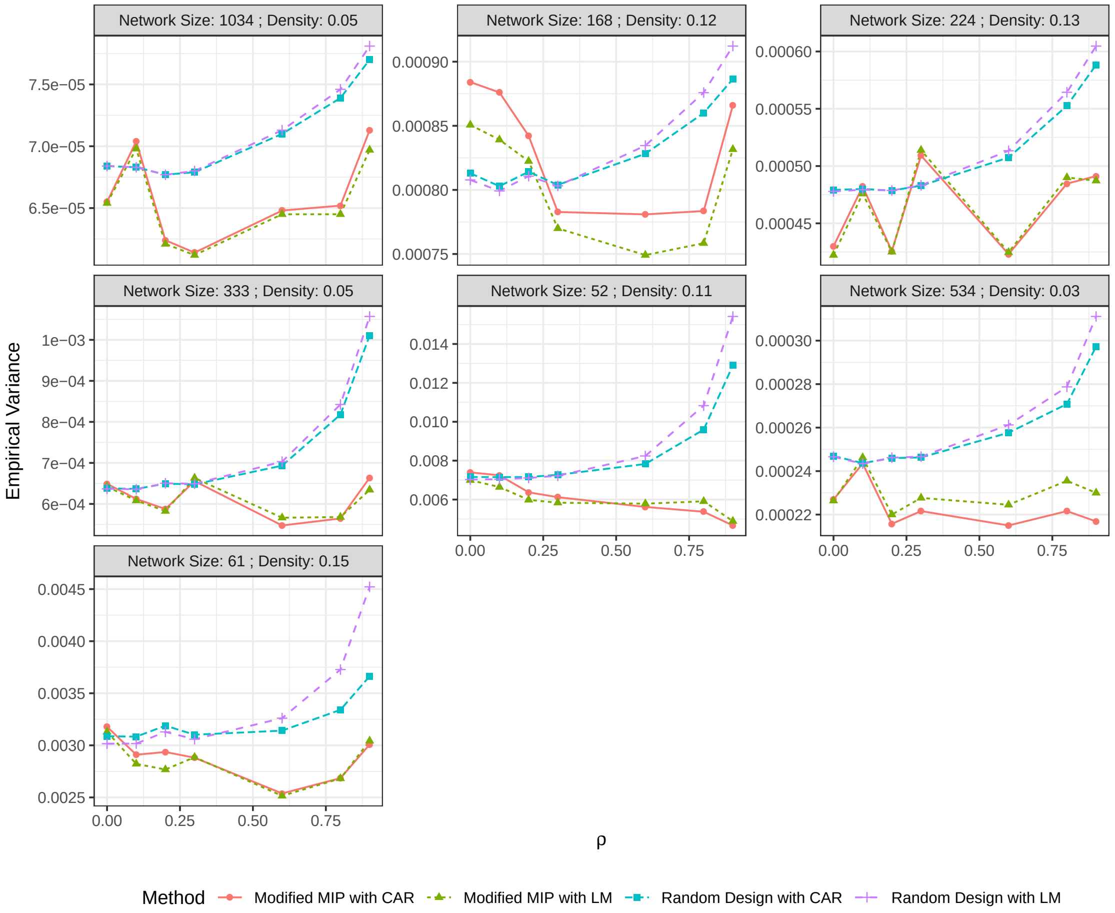

Figure 2.7: Empirical variances of $\hat{\beta}$ of seven ego networks. MIP running time 4 hours for all ego-networks.

2.7.5.2 Model Misspecification under Gamma Distribution

This section evaluates the performances of modified-MIP when the outcomes are not generated from a Gaussian distribution. We use the same network as in Table 2.5. The network outcomes are generated as follows:

$$y_i = \beta_0 + x_i\beta + \delta_i,$$

where $\delta_i \sim N(\rho \sum_{i \neq j} w_{ij} z_j / m_i, \sigma^2/m_i)$ and $z_j \sim Gamma(1,1)$. Therefore, the outcomes are not from a Gaussian distribution. The design approach Modified-MIP is still based on the CAR model, and we evaluate its performance under model misspecification. As we can observe from the results of Table 2.6, random designs have a better performance (smaller variance of $\hat{\beta}$) than the design generated using our proposed method. The results show that, as many other model-based optimal design approaches, our proposed method has some limitations. It can be critically important to examine the model assumption before determine the design approaches.

Table 2.6: Empirical variances of $\hat{\beta}$ based on Gamma model-misspecification case for a network of size $n = 500$ and density $p = 0.01$

Method	$\rho = 0$	$\rho = 0.1$	$\rho = 0.2$	$\rho = 0.3$	$\rho = 0.6$	$\rho = 0.8$	$\rho = 0.9$
Modif-MIP (CAR)	0.00011	0.00012	0.00014	0.00018	0.00037	0.00058	0.00070
Modif-MIP (LM)	0.00011	0.00012	0.00014	0.00018	0.00038	0.00058	0.00070
Random (CAR)	0.00011	0.00011	0.00013	0.00015	0.00026	0.00037	0.00044
Random (LM)	0.00011	0.00011	0.00013	0.00015	0.00026	0.00037	0.00044

Chapter 3

A Re-randomization Experimental Design Approach for Network A/B Testing

3.1 Introduction

A/B testing is a statistical experiment that aims to estimate the effects of two treatments: A and B. It has been used on unprecedented scale by many online companies, such as Amazon, LinkedIn, Facebook and other companies that have at their disposal the information about the structure of their social network users. For example, if the goal is to test which web page design, A or B, produces a higher click-through rate or results in more revenue, we can select 2 groups of users on the network and conduct the experiment to estimate the treatment effect. Randomization is considered to be the "gold standard" for estimating causal effects, but we want to assign those subjects to either treatment or control group by incorporating the network structure. We believe that a network structure contains information about users connected on a network and we assume that the outcomes of two connected nodes are positively correlated because the features of these two nodes are more similar than those of the unconnected nodes. Pokhilko et al. (2019) propose original-MIP and modified-MIP designs based on D-optimality criteria, making the assumption of network correlated outcomes that are modeled using conditional auto-regressive model or CAR (Schmidt and Nobre, 2014). In this chapter we continue working with D-optimal design objectives and network correlated outcomes and propose a re-randomization approach to constructing experimental designs for A/B testing on social networks.

Re-randomization has been widely used in practice mainly as a remedy when a random design creates groups that are notably unbalanced in terms of subjects' covariate information. Re-randomization ensures the balance, while also retaining the benefits of randomization. Checking the design for balance and re-randomizing when needed has been advocated repeatedly. Sprott and Farewell (1993) and Worrall (2010) recommend re-randomization, when an "obvious" confounding is noticed under current randomization allocation. Rubin (2008) advises to re-randomize, if important imbalances occur and continue to do so until balance is achieved, recording the reasons for discarding randomizations. Cox (2009) and Krause and Howard (2003) advocate to create multiple randomizations and select the one with the most balance. Soares and Wu (1985) suggest to specify a bound for the difference in treatment and control covariate means for each covariate and re-randomize until all differences are within these bounds. Some scientific literature even discuss the danger of relying on pure randomization to balance the experimental design (Rubin, 2008; Urbach, 1985; Krause and Howard, 2003; Rosenberger et al., 2008; Worrall, 2010). Also, much work has been done historically on whether purposefully balanced designs should be preferred over randomization (Gosset, 1938; Yates, 1939; Greenberg, 1951; Harville, 1975; Arnold, 1986). The implications of re-randomization have also been theoretically explored. Morgan and Rubin (2012) define the sufficient conditions under which re-randomization is valid, provide corresponding theoretical results and describe in detail the procedure for implementing re-randomization.

Despite the fact that re-randomization is widely used in practice (Holschuh, 1980; Urbach, 1985; Bailey and Rowley, 1987; Imai et al., 2008; Bruhn and McKenzie, 2009), some criticism of it exists in the scientific literature (Fisher, 1992; Anscombe, 1948; Grundy and Healy, 1950; Holschuh, 1980; Bailey, 1983; Urbach, 1985; Bailey and Rowley, 1987). Main critiques of re-randomization are related to not specifying the rejection criteria in advance and that forms of analysis using Gaussian distribution theory are no longer valid. Morgan and Rubin (2012) state that "re-randomization changes the distribution of the test statistic ... by decreasing the true standard error, thus traditional methods of analysis that do not take this into account will result in overly conservative inferences in the sense that tests will reject true null hypotheses less often than the nominal level and confidence intervals will cover the true value more often than the nominal level". However, those criticisms of re-randomizing can be addressed by specifying an objective re-randomization rule before the procedure and then analysing results using randomization-based inference, since re-randomization can be accounted for during analysis (Anscombe, 1948; Kempthorne, 1955; Tukey, 1993; Moulton, 2004; Rosenberger and Lachin, 2015).

Although a few re-randomization methods have been proposed (Moulton, 2004; Maclure et al., 2006; Bruhn and McKenzie, 2009; Cox, 2009), none of them explore re-randomization experimental design on a social network, and we decide to fill the gap. In this chapter, we focus on the construction of A/B testing experimental designs for network-correlated outcomes when users are connected on a network and share some common demographic and social features. We use CAR (Schmidt and Nobre, 2014) spatial network model to incorporate the correlated network structure in the analysis. To conduct the experiment on the test subjects that form the network, we propose an algorithm for re-randomization procedure and formulate stopping criteria that are based on the D-optimal design objectives of the CAR model (Pokhilko et al., 2019). The main goal of the proposed stopping criteria is to find a random design that can approximate the exact D-optimal design in its objectives. In the A/B testing experiment, the key parameter is the treatment effect, and thus the D-optimal criterion measures the variance of the estimated treatment effect. Finally, we conduct simulation studies on synthetic networks to demonstrate the performances of the proposed re-randomization method compared to the D-optimal designs: original-MIP and modified-MIP (Pokhilko et al., 2019); and random designs, which do not consider the network structure.

3.2 D-optimal Design for CAR Model

In this section we provide theoretical details for the model that we use, outline the assumptions for the model and list all the definitions. In this Chapter we use the same model set up as in Pokhilko et al. (2019). We consider A/B testing experiments on undirected social networks. The connections or edges of the network are recorded by an $n \times n$ adjacency matrix $W = \{w_{ij}\}$ whose (i, j)-th entry is w_{ij}. The diagonal entries w_{ii}'s of this matrix are zeros, whereas the off-diagonal entries are

$$w_{ij} = \begin{cases} 1, & \text{if node } i \text{ and node } j \text{ are adjacent} \\ 0, & \text{otherwise.} \end{cases}$$

We call that two nodes are adjacent if they are connected by an edge. We denote $m_i = \sum_{j=1}^{n} w_{ij}$ as the degree of the ith vertex, and $N = \sum\limits_i m_i = \sum\limits_i \sum\limits_j w_{ij}$ as twice of the total number of edges of the network. Our goal is to create an experimental design to allocate A or B treatment to each node. Let $\boldsymbol{x} = (x_1, \ldots, x_n)^\top$, with $x_i \in \{1, -1\}$ for $i = 1, \ldots, n$, be the design of the i-th node and the two settings $\{1, -1\}$ represent A and B treatments. We denote the scalar response observation of the i-th node by y_i. In this paper, we focus on the case where the response is continuous and assume a linear regression model for the response as follows.

$$y_i = \beta_0 + x_i\beta + \delta_i, \tag{3.1}$$

where β_0 is the intercept, β represents the treatment effect, and δ_i is a zero mean random variable. For the experiments on social networks, two connected users share similarities in their social behaviors and other backgrounds, and thus their responses are often correlated. To incorporate this social correlation, we model δ_i in (3.1) by the conditional auto-regressive (CAR) model (Besag, 1974)

$$\delta_i|\boldsymbol{\delta}_{-i} \sim N\left(\rho\sum_{j\neq i}\frac{w_{ij}\delta_j}{m_i}, \frac{\sigma^2}{m_i}\right), \tag{3.2}$$

where $\boldsymbol{\delta}_{-i} = \{\delta_1, \dots, \delta_{i-1}, \delta_{i+1}, \dots, \delta_n\}$, and σ^2 is the variance parameter that is assumed to be known in our scope, and ρ is the correlation parameter of the CAR model. We restrict the correlation parameter ρ to be non-negative, i.e., $0 \leq \rho < 1$, since connected users tend to have similar reactions to the same treatment. If $\rho = 0$, it corresponds to the extreme case when users of the network are independent of each other, i. e., have no edges or connection.

According to Proposition 2.1 in Pokhilko et al. (2019) the covariance model in (3.2) can be expressed in the following multivariate normal form:

$$\boldsymbol{\delta} \sim \mathcal{MVN}_n(0, \sigma^2(D - \rho W)^{-1})$$

where $D = \text{diag}(m_1, \dots, m_n)$ with m_i's denoting the degree of the ith node.

The maximum likelihood method can be used to fit the model (3.1) with δ_i from (3.2) and estimate the model parameters. The goal of A/B testing is to accurately assess the treatment effect β. Next, we use D-optimal design (Kiefer et al., 1959; Atwood, 1969; Pukelsheim, 1993; Atkinson et al., 2007; Woods, 2005; Atkinson and Woods, 2015) to improve the accuracy of the estimated treatment effect β under the strong model assumption (3.1). According to Proposition 2.2 in Pokhilko et al. (2019), such D-optimal design $\boldsymbol{x}$ is the solution to the following optimization problem:

$$\text{argmax}_{\boldsymbol{x}\in\{-1,1\}^n} D(\boldsymbol{x}) = \text{argmin}_{\boldsymbol{x}\in\{-1,1\}^n}\left\{a\sum_i\sum_j w_{ij}x_ix_j + \left(\sum_i m_ix_i\right)^2\right\}, \tag{3.3}$$

where $a = \frac{\rho}{1-\rho}N$.

Since the objective function in (3.3) can be decomposed to a weighted sum of $\sum_{ij} w_{ij}x_ix_j$ and $\left(\sum\limits_i m_ix_i\right)^2$, the desired optimal design should achieve smaller values on both terms. This observation provides insights of the features of the desired optimal design: (1) the first term $\sum_{ij} w_{ij}x_ix_j$ suggests that the two connected nodes should be assigned to different treatments; (2) the second term indicates that if we treat the size of neighborhood as a feature associated with each node, then the design allocation is desired to be "orthogonal" with this feature.

3.3 Re-randomization Formulation for D-optimal Design

To develop a re-randomization design we focus on optimization problem (3.3). Due to computational constraints it is quite difficult, if not impossible, to obtain the D-optimal design by solving this optimization program directly. Pokhilko et al. (2019) proposed a modified mixed integer program that produced an approximation to a D-optimal design (3.3) that had significant savings in terms of computational time. In this paper we propose a re-randomization design as an alternative to the D-optimal design formulation (3.3).

If we assume that x_i is a random variable taking value from $\{-1, 1\}$ with equal weights and let

$$T_1(\boldsymbol{x}) = \boldsymbol{x}^\top W\boldsymbol{x} = \sum_{ij} w_{ij}x_ix_j,$$

then

$$\mathrm{E}(T_1(\boldsymbol{x})) = \mathrm{E}\left(\sum_i\sum_j w_{ij}x_ix_j\right) = \sum_i\sum_j w_{ij}\mathrm{E}[x_ix_j] = 0 \text{ and}$$

$$\mathrm{Var}(T_1(\boldsymbol{x})) = \mathrm{Var}\left(\sum_i\sum_j w_{ij}x_ix_j\right) = \sum_i\sum_j w_{ij}^2\mathrm{Var}(x_ix_j) = \sum_i\sum_j w_{ij}.$$

and we have the following theorem.

Theorem 3.3.1 *Consider that $x_1, \ldots, x_n$ in $\boldsymbol{x}$ are independent and identically distributed random variables from the discrete distribution with* $\mathrm{P}(x_i = 1) = \mathrm{P}(x_i =$

$-1) = 0.5$. *If the network structure is fixed, as $n \to \infty$,*

$$\frac{T_1(\boldsymbol{x})}{\sqrt{\sum w_{ij}}} \xrightarrow{d} N(0,1)$$

The proof of this theorem is to be published in the work of Zhang and Kang early 2020. Similarly, if we let

$$T_2(\boldsymbol{x}) = (\sum_i^n m_i x_i)^2,$$

then

$$\begin{aligned} \mathrm{E}(T_2(\boldsymbol{x})) &= \mathrm{E}\left(\sum_{i=1}^n m_i x_i\right) = \sum_{i=1}^n m_i \mathrm{E} x_i = 0 \text{ and} \\ \mathrm{Var}(T_2(\boldsymbol{x})) &= \mathrm{Var}\left(\sum_{i=1}^n m_i x_i\right) = \sum_{i=1}^n m_i^2 \mathrm{Var}(x_i) = \sum_{i=1}^n m_i^2. \end{aligned}$$

and the following theorem follows.

Theorem 3.3.2 *Consider that $x_1, \ldots, x_n$ in $\boldsymbol{x}$ are independent and identically distributed random variables from the discrete distribution with* $\mathrm{P}(x_i = 1) = \mathrm{P}(x_i = -1) = 0.5$. *If the network structure is fixed, and as $n \to \infty$*

$$n^{-2} \sum_{i=1}^n m_i^2 < \infty \quad \text{and} \quad n^{-1} m_i \to 0$$

then

$$\frac{T_2(\boldsymbol{x})}{\sum_{i=1}^n m_i^2} \xrightarrow{d} \chi_1^2, \quad \text{as} \quad n \to \infty.$$

The proof of the Theorem 3.3.2 is provided in Appendix 3.6.1.

Since our goal is to minimize $T_1(\boldsymbol{x})$ and $T_2(\boldsymbol{x})$, we keep generating random designs $\boldsymbol{x} \in \{-1, 1\}^n$ until we find one $\boldsymbol{x}$ that satisfies the following conditions:

$$\frac{T_1(\boldsymbol{x})}{\sqrt{\sum w_{ij}}} \leq Z_{\alpha_1} \text{ and } \frac{T_2(\boldsymbol{x})}{\sum_{i=1}^n m_i^2} \leq \chi^2_{1,\alpha_2},$$

where Z_{α_1} and χ^2_{1,α_2} are values of standard normal and chi-squared with 1 degree of freedom distributions corresponding to their α_1 and α_2 percentiles and α_1 and α_2 are chosen to be arbitrarily small numbers. Algorithm 1 describes the procedure for estimating treatment effect and calculating its variance using re-randomization design. Since for the same network multiple re-randomization designs can be generated, in this paper we will average the results from 100 re-randomization designs to get the final estimates.

Algorithm 1: Procedure for Calculating the Empirical Variance of the Treatment Effect Using Re-randomization Design

```
Input: Adjacency Matrix, W; percentiles, $\alpha_1$ and $\alpha_2$; number of
       re-randomization designs, $I = 100$; sample size or number of
       replications for treatment effect estimation, $J = 500$
for $i \leftarrow 1$ to $I$ do
    Construct re-randomization design, $\boldsymbol{x}$:
    repeat
        Create a random design $\boldsymbol{x}$
    until $\boldsymbol{x}$, s.t., $\frac{T_1(\boldsymbol{x})}{\sqrt{\sum w_{ij}}} \leq Z_{\alpha_1}$ and $\frac{T_2(\boldsymbol{x})}{\sum_{i=1}^{n} m_i^2} \leq \chi^2_{1,\alpha_2}$;
    Calculate $\hat{\text{V}}(\hat{\beta}_i)$:
    for $j \leftarrow 1$ to $J$ do
        Conduct the experiment using the design $\boldsymbol{x}$ and observe user
         responses $\boldsymbol{y}$
        Estimate the treatment effect $\beta_j$
    end
    $\hat{\text{V}}(\hat{\beta}_i) = \frac{1}{J-1}\sum_{l=1}^{J}\left(\hat{\beta}_l - \frac{1}{J}\sum_{k=1}^{J}\hat{\beta}_k\right)^2$
end
Calculate $\hat{\text{V}}(\hat{\beta}) = \sum_{i=1}^{I}\hat{\text{V}}(\hat{\beta}_i)/I$
```

3.4 Numerical Study on Synthetic Networks

In this chapter we compare the performances of the four designs: original-MIP and modified-MIP designs defined in Pokhilko et al. (2019), re-randomization designs generated using the proposed method and completely random designs. We generate those designs on synthetic networks as described in Pokhilko et al. (2019). We conduct the comparison in terms of the empirical variance of the β estimate and D-efficiency. Random designs are produced by randomly allocating 1 or -1 with equal probability to each node or experimental unit.

3.4.1 D-efficiency

Here we compare the D-efficiencies of the four designs. We took the results from Table 1 of Pokhilko et al. (2019) and added D-efficiency calculated on re-randomization designs. We set $\alpha_1 = \alpha_2$ and vary their product to be either 0.05 or 0.001. We want to see if re-randomization designs produced with those two scenarios have different D-efficiencies. For this purpose, we calculate empirical D-efficiency (as defined in Pokhilko et al. (2019)) averaged over 100 re-randomization designs. The results are depicted in Table 3.1. We summarize the results from Tables 3.1 as follows:

- Both MIP based designs produce higher D-efficiencies than completely random and re-randomization design (when correlation parameter $\rho \neq 0$), and those differences get larger as ρ increases.
- D-efficiency of re-randomization design is higher than D-efficiency of completely random designs for all values of correlation parameter ρ.
- Re-randomization designs generated using smaller value of $\alpha_1 \cdot \alpha_2$ have larger D-efficiency than those produced using larger value of $\alpha_1 \cdot \alpha_2$.

Table 3.1: The D-Efficiency measures of designs generated using 4 different methods for a fixed network of size $n = 50$ and density $p = 0.1$

Method	$\rho = 0$	$\rho = 0.1$	$\rho = 0.2$	$\rho = 0.3$	$\rho = 0.6$	$\rho = 0.8$	$\rho = 0.9$
Random	0.98	0.90	0.82	0.76	0.62	0.55	0.53
Rerand ($\alpha_1 \cdot \alpha_2 = 0.05$)	1.00	0.92	0.85	0.79	0.67	0.60	0.58
Rerand ($\alpha_1 \cdot \alpha_2 = 0.001$)	1.00	0.92	0.86	0.80	0.68	0.62	0.60
Original-MIP	1.00	0.96	0.93	0.90	0.84	0.80	0.79
Modified-MIP ($\alpha = 0.6$)	1.00	0.96	0.93	0.90	0.84	0.80	0.79
Modified-MIP ($\alpha = 0.7$)	1.00	0.96	0.92	0.90	0.83	0.80	0.79
Modified-MIP ($\alpha = 0.8$)	1.00	0.96	0.92	0.90	0.83	0.80	0.79

3.4.2 Empirical Variance

In this section we compare the empirical variances of $\hat{\beta}$ that were estimated using four different designs: original-MIP, modified-MIP, re-randomization and completely random designs. We use the results from Table 2 of Pokhilko et al. (2019) and the same small network of size 50 and density $p = 0.1$ to add the empirical variances calculated based on re-randomization designs using $\alpha_1 \cdot \alpha_2 = 0.05$ and $\alpha_1 \cdot \alpha_2 = 0.001$, where $\alpha_1 = \alpha_2$ in both cases. To generate empirical variances we use the same procedure outlined in Pokhilko et al. (2019). To make the four approaches comparable, 100 re-randomization designs are produced and for each design the procedure is repeated $R = 500$ times to measure the accuracy of β estimate by calculating its sample variance:

$$\hat{\mathrm{V}}(\hat{\beta}) = \frac{1}{R-1} \sum_{l=1}^{R} \left(\hat{\beta}_l - \frac{1}{R} \sum_{k=1}^{R} \hat{\beta}_k \right)^2 ,$$

where $\hat{\beta}_1, \ldots, \hat{\beta}_R$ are the $R = 500$ copies of the estimates for β. After that we average the value of $\hat{V}(\hat{\beta})$ over the 100 re-randomization designs and report it in Table 3.2. We summarize the results from Table 3.2 as follows:

- When correlation coefficient ρ is small ($\rho < 0.3$), re-randomization designs produce the smallest variance.

- When correlation coefficient ρ increases ($\rho \geq 0.3$), MIP based designs produce the smallest variance.

- For all ρ values re-randomization designs produce smaller variance than random designs.

- The empirical variances of re-randomization designs with different values of $\alpha_1 \cdot \alpha_2$ are comparable; designs produced with $\alpha_1 \cdot \alpha_2 = 0.001$ have slightly smaller variances than designs produced with $\alpha_1 \cdot \alpha_2 = 0.05$, where $\alpha_1 = \alpha_2$.

Next we tested the performance of the four methods on a larger network of size 500 with network density $p = 0.01$. We used the results from Table 3 of Pokhilko et al. (2019) as a base and added the results of using re-randomization designs on the same network for comparison. Since previous results showed that the performance of re-randomization designs was robust to the choice of $\alpha_1 \cdot \alpha_2$, going forward we will

Table 3.2: Empirical variances of $\hat{\beta}$ based on designs generated from different methods for a network of size $n = 50$ and density $p = 0.1$

Method	$\rho = 0$	$\rho = 0.1$	$\rho = 0.2$	$\rho = 0.3$	$\rho = 0.6$	$\rho = 0.8$	$\rho = 0.9$
Original-MIP	0.0046	0.0043	0.0041	0.0038	0.0033	0.0030	0.0029
Modified-MIP ($\alpha = 0.6$)	0.0044	0.0043	0.0040	0.0037	0.0033	0.0031	0.0030
Modified-MIP ($\alpha = 0.7$)	0.0049	0.0043	0.0043	0.0038	0.0034	0.0030	0.0030
Modified-MIP ($\alpha = 0.8$)	0.0047	0.0042	0.0042	0.0038	0.0034	0.0030	0.0029
Rerand ($\alpha_1 \cdot \alpha_2 = 0.05$)	0.0040	0.0040	0.0040	0.0039	0.0038	0.0038	0.0038
Rerand ($\alpha_1 \cdot \alpha_2 = 0.001$)	0.0040	0.0039	0.0039	0.0038	0.0037	0.0036	0.0036
Random (CAR)	0.0047	0.0047	0.0046	0.0046	0.0046	0.0048	0.0047
Random (LM)	0.0045	0.0045	0.0044	0.0045	0.0047	0.0051	0.0052

only test re-randomization designs produced using $\alpha_1 \cdot \alpha_2 = 0.001$, where $\alpha_1 = \alpha_2$. For all the designs we also consider the β estimated from both CAR and linear regression model (LM), although the data are generated from the CAR model. We summarize our main observations from Table 3.3 as follows:

- Re-randomization and random designs are comparable to each other: as correlation coefficient ρ grows large, re-randomization designs have slightly lower variance in comparison to completely random designs.

- The variances produced using modified-MIP designs tend to decrease as correlation coefficient ρ grows large.

- The difference in variances between modified-MIP and all other designs increases as ρ increases.

- Variances calculated under model misspecification (LM) are comparable to the true case when ρ is small and tend to increase as ρ increases.

- When we compare the Tables 3.2 and 3.3 we observe that the differences in variance between re-randomization and random designs decrease as network size grows large.

Table 3.3: Empirical variances of $\hat{\beta}$ based on designs generated from different methods for a network of size $n = 500$ and density $p = 0.01$

Method	$\rho = 0$	$\rho = 0.1$	$\rho = 0.2$	$\rho = 0.3$	$\rho = 0.6$	$\rho = 0.8$	$\rho = 0.9$
Modif-MIP (CAR)	0.00042	0.00035	0.00039	0.00031	0.00030	0.00029	0.00028
Re-random (CAR)	0.00041	0.00041	0.00040	0.00040	0.00040	0.00040	0.00040
Random (CAR)	0.00041	0.00041	0.00041	0.00041	0.00041	0.00041	0.00042
Modif-MIP (LM)	0.00042	0.00036	0.00040	0.00032	0.00032	0.00033	0.00031
Re-random (LM)	0.00041	0.00041	0.00041	0.00040	0.00042	0.00044	0.00046
Random (LM)	0.00041	0.00041	0.00041	0.00041	0.00043	0.00046	0.00048

3.4.3 Empirical Variance under Gamma Distribution Model Misspecification

In this section we evaluate the performances of designs generated using the four different methods when the outcomes are not generated from a Gaussian distribution. We test this on small and large networks that we used in Tables 3.2 and 3.3. The network outcomes are generated as follows:

$$y_i = \beta_0 + x_i\beta + \delta_i,$$

where $\delta_i \sim N(\rho \sum_{i \neq j} w_{ij} z_j / m_i, \sigma^2/m_i)$ and $z_j \sim Gamma(1, 1)$. Therefore, the outcomes are not from a Gaussian distribution. The modified-MIP and re-randomization designs are still based on the CAR model, and we evaluate their performance under this model misspecification. The results are depicted in Tables 3.4 and 3.5 and are summarized as follows:

- All designs have similar variance for small values of correlation coefficient $\rho < 0.3$.

- As correlation coefficient grows large ($\rho \geq 0.3$), so does the difference in variance between modified-MIP and randomization-based designs. The variance of modified-MIP designs becomes larger than randomization-based designs as ρ increases.

- For all values of ρ, re-randomization and random designs have approximately the same variance.

- The network size does not appear to have an effect on the performance of all the designs other than relative size of the variance.

Table 3.4: Empirical variances of $\hat{\beta}$ based on Gamma model-misspecification case for a network of size $n = 50$ and density $p = 0.1$

Method	$\rho = 0$	$\rho = 0.1$	$\rho = 0.2$	$\rho = 0.3$	$\rho = 0.6$	$\rho = 0.8$	$\rho = 0.9$
Modif-MIP (CAR)	0.0010	0.0010	0.0012	0.0016	0.0035	0.0055	0.0068
Re-random (CAR)	0.0011	0.0011	0.0013	0.0015	0.0026	0.0037	0.0044
Random (CAR)	0.0011	0.0011	0.0013	0.0015	0.0026	0.0037	0.0044
Modif-MIP (LM)	0.0009	0.0010	0.0012	0.0015	0.0035	0.0055	0.0068
Re-random (LM)	0.0010	0.0011	0.0012	0.0014	0.0025	0.0036	0.0043
Random (LM)	0.0011	0.0011	0.0012	0.0014	0.0025	0.0036	0.0043

Table 3.5: Empirical variances of $\hat{\beta}$ based on Gamma model-misspecification case for a network of size $n = 500$ and density $p = 0.01$

Method	$\rho = 0$	$\rho = 0.1$	$\rho = 0.2$	$\rho = 0.3$	$\rho = 0.6$	$\rho = 0.8$	$\rho = 0.9$
Modif-MIP (CAR)	0.00011	0.00012	0.00014	0.00018	0.00037	0.00058	0.00070
Re-random (CAR)	0.00011	0.00012	0.00013	0.00015	0.00026	0.00037	0.00044
Random (CAR)	0.00011	0.00011	0.00013	0.00015	0.00026	0.00037	0.00044
Modif-MIP (LM)	0.00011	0.00012	0.00014	0.00018	0.00038	0.00058	0.00070
Re-random (LM)	0.00011	0.00012	0.00013	0.00015	0.00026	0.00037	0.00044
Random (LM)	0.00011	0.00011	0.00013	0.00015	0.00026	0.00037	0.00044

3.5 Discussion

In this paper we introduce a re-randomization strategy to producing designs for A/B testing on networks with known structures. Numerical examples on synthetic networks are provided to show the effectiveness of the proposed method. Our approach outperforms random designs in terms of decreasing variance of treatment effect estimate, but cannot beat the performance of optimal designs, both original and modified-MIP. Our approach works especially well on smaller networks, but as a network size increases, the performance of re-randomization designs becomes comparable to random designs. In case of violation of the model assumptions, our proposed method outperforms optimal designs and is comparable to random designs.

So the benefits of our method is that it is more robust to model misspecification than modified-MIP designs.

3.6 Appendix

3.6.1 The proof of Theorem 3.3.2

Let $Y_i = m_i x_i$. Then Y_is are independent random variables with a finite variance. If $n^{-2}\sum_{i=1}^n m_i^2 < \infty$ and $n^{-1}m_i \to 0$ as $n \to \infty$, we have that

$$\frac{m_i}{\sqrt{\sum_{i=1}^n m_i^2}} \to 0.$$

Therefore, the Lindeberg's condition

$$\sum_{i=1}^n \mathrm{E}\left[(Y_i - \mathrm{E}Y_i)^2 I(|Y_i - \mathrm{E}Y_i|^2 > \varepsilon \mathrm{Var}(\sum_{i=1}^n Y_i))\right] = o\left(\mathrm{Var}(\sum_{i=1}^n Y_i)\right)$$

holds for any $\varepsilon > 0$ as $n \to \infty$. According to the Lindeberg's central limit theorem,

$$\frac{\sum_{i=1}^n Y_i}{\sqrt{\sum_{i=1}^n m_i^2}} \xrightarrow{d} N(0,1)$$

as $n \to \infty$. Then the conclusion holds.

Chapter 4

Covariate-Assisted Bayesian Sequential Design for Network A/B Testing

4.1 Introduction

Though the idea of A/B testing dates back to Sir Ronald A. Fisher's experiments at the Rothamsted Agricultural Experimental Station in England in the 1920s, the advent of the internet and smartphones has given web and app developers a new edge for implementing these tests at large scales. In its basic form, consider two possible alternatives, denoted A and B. The technique consists of directing small portions of user traffic to experimental designs of an app, an online commercial, a game, or a website, and the designers can employ noisy feedback from users to optimize any observable metric with respect to the product's configuration. Based on the particular phase of a product's life, the click-through rate might be more relevant objective to optimize for an ad, whereas for a game it may be some measure of user engagement; new subscriptions may be more valuable than user retention or revenue, or vice versa.

The important question is how to optimally sample these subsets of users in order to find the best product with high probability within a predetermined sample budget, or how to direct traffic sequentially in order to optimize a cumulative metric while incurring the least opportunity cost (Chapelle and Li, 2011; Kohavi et al., 2009; Scott, 2010). Eckles et al. (2017) worked on design of treatment and control assignment that considers the ability to perform random assignment to treatments that is correlated in the network, such as through graph cluster randomization. Gui et al. (2015) proposed an efficient and effective estimator for Average Treatment Effect (ATE) that considers the interference between users in real online experiments. It is defined as the difference of the average outcomes between applying treatment to the *entire* user population and applying control to the *entire* user population. In reality, however, a user can only receive one treatment at a time. In our research we focus on dynamic design, where we utilize sequential Bayesian assignment of alternatives based on user network information.

Much research has been done and recorded in scientific literature on Bayesian optimization. In academia, it is impacting a wide range of areas, including robotics (Lizotte et al., 2007; Martinez-Cantin et al., 2007), environmental monitoring (Marchant and Ramos, 2012), information extraction (Wang et al., 2014), algorithm configuration (Hutter et al., 2011; Wang et al., 2016), automatic machine learning (Bergstra et al., 2011; Xia et al., 2017; Hoffman et al., 2014; Wu, 2017; Snoek et al., 2012; Swersky et al., 2013; Thornton et al., 2013), sensor networks (Srinivas et al., 2009), adaptive Monte Carlo (Mahendran et al., 2012) and experimental design (Azimi et al., 2012). Fundamentally, Bayesian optimization is a sequential model-based approach to finding a global maximizer (or minimizer) of an unknown objective function f:

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x})$$

where $\mathcal{X}$ is a design space of interest. In this setting, at step t, a sequential search algorithm selects a location x_{t+1}, at which to query f and observe y_{t+1}. After T queries, the algorithm makes a final recommendation $\hat{x}_N$, which represents the algorithm's best estimate of the optimizer.

Shahriari et al. (2016) identifies two key ingredients of Bayesian optimization. The first ingredient has a surrogate model with a prior distribution that captures our beliefs about the behavior of an unknown objective function and an observation model that describes how the data is generated. The second ingredient is an acquisition or loss function that optimizes the sequence of queries. When we take a closer look at the first ingredient, based on a specific problem, we can employ Bayesian optimization with parametric or nonparametric models. A good example of Bayesian optimization using parametric modeling would be Thompson sampling (Thompson, 1933) in the Beta-Bernoulli bandit model (Agrawal and Goyal, 2012) with 2 arms. The objective is to identify which arm of the bandit to pull, e.g., which drug to administer, which movie to recommend, or which advertisement to display.

In this chapter we develop a Bayesian sequential experimental design for A/B testing on social networks. We consider the case when responses of network members are continuous and we use normal conjugate priors to sequentially update a Bayesian model. We summarize information about network into covariates, such as neighborhood size, average A and/or B response in user's first tier neighborhood and others, and add those covariates into the Bayesian model. We propose an algorithm to sequentially update our model parameters based on new experimental outcomes and use variance reduction term to choose treatment allocation at each step of the procedure. Finally, we test the performance of our covariate-assisted Bayesian model on synthetic and real-world networks and compare the results to a Bayesian sequential model that does not use network covariates in its posterior updates.

4.2 Covariate-Assisted Bayesian D-Optimal Design for Network AB Testing

Assume that a marketing manager at a company wants to test two alternative designs of an advertisement on a certain network community, for example, on Facebook members. Each Facebook member has its own network of friends who might have previously been selected for participation in A/B testing procedure. We want to incorporate that information in the Bayesian model and test the hypothesis that a model that takes into consideration the information about the network performs better than a model that does not consider such information. We can build a model that summarizes the information about the network into covariates. Possible covariates can be a number of friends that a users has or a number of members in the first tier neighborhood, a number of members in the first tier neighborhood who responded favorably to an alternative A, an average value of A responses in a user's first tier neighborhood, an average value of B responses in a user's first tier neighborhood, and others. We can extend the number of covariates to represent members of second tier neighborhood or friends of the friends, etc. Let's assume that we have d network covariates and treatment options A and B are represented by a vector $x \in \{-1, 1\}$. Then we consider the following model with normal conjugate priors:

$$y = \boldsymbol{G}^T\boldsymbol{\phi} + \epsilon = x\beta + \mathbf{z}^T\boldsymbol{\gamma} + \epsilon \tag{4.1}$$

where

- $\boldsymbol{G} = \begin{bmatrix} x \\ \mathbf{z} \end{bmatrix}$, $x \in \{-1, 1\}$ is a treatment assignment and $\mathbf{z} = [1, z_1, \ldots, z_d]^T \in R^{d+1}$ are intercept and network covariates,
- $\epsilon \sim N(0, \eta^2)$ is an error term, where η^2 is known or estimated from historical data,
- $\boldsymbol{\phi} = \begin{bmatrix} \beta \\ \boldsymbol{\gamma} \end{bmatrix} \sim MVN_{d+2}(\boldsymbol{\theta}, V)$ is a prior distribution, where $\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \boldsymbol{\theta_{-1}} \end{bmatrix}$ and $V = \begin{bmatrix} \sigma^2 & \boldsymbol{r^T} \\ \boldsymbol{r} & \Sigma \end{bmatrix}$
- θ_1 and σ^2 are scalars, $\boldsymbol{\theta_{-1}}$ and $\boldsymbol{r}$ are $(d+1)\times 1$ vectors and Σ is a $(d+1)\times(d+1)$ matrix.

- $y|\phi \sim N(G\phi, \eta^2)$

We plan to conduct the experiment sequentially, one at a time. New experiment is chosen by minimizing VAR($\beta|y_{t+1}$), with regards to both x and $\boldsymbol{z}$. In order for us to know what is VAR($\beta|y_{t+1}$), we need to know the distribution of $\beta|y_{t+1}$. For that we first need to find the posterior distribution of $\phi|y_{t+1}$. It can be expressed in a multivariate normal form given by the following proposition:

Proposition 4.2.1 *Using the Bayesian paradigm and model assumptions of* (4.1)*, the posterior distribution of $\phi|y_{t+1}$ follows a multivariate normal distribution:*

$$\phi|y_{t+1} \sim MVN_{p+1}(\theta_t + \frac{y_{t+1} - G_{t+1}^T\theta_t}{\eta^2 + G_{t+1}^T V_t G_{t+1}} V_t G_{t+1}, V_t - \frac{V_t G_{t+1} G_{t+1}^T V_t}{\eta^2 + G_{t+1}^T V_t G_{t+1}}),$$

where y_{t+1} is a new response observed after applying treatment x_{t+1} and G_{t+1} contains a vector of updated network covariates, z_{t+1}, and the treatment allocation, x_{t+1}, at step $t+1$.

The proof of this proposition is provided in Appendix 4.7.1. Therefore at step $t+1$ after observing the response y_{t+1} we update the parameters of posterior distribution $\phi|y_{t+1}$ as follows:

$$\theta_{t+1} = \theta_t + \frac{y_{t+1} - G_{t+1}^T\theta_t}{\eta^2 + G_{t+1}^T V_t G_{t+1}} V_t G_{t+1}$$

$$V_{t+1} = V_t - \frac{V_t G_{t+1} G_{t+1}^T V_t}{\eta^2 + G_{t+1}^T V_t G_{t+1}}$$

We found the distribution of $\phi|y_{t+1}$, now we can provide the information about the distribution of $\beta|y_{t+1}$, given by the following Proposition.

Proposition 4.2.2 *Under model assumptions* (4.1) *and Proposition 4.2.1, the distribution of $\beta|y_{t+1} \sim N(\theta_{1,t+1}, \sigma_{t+1}^2)$, where*

$$E[\beta|y_{t+1}] = \theta_{1,t+1} = \theta_{1,t} + \frac{(y_{t+1} - x_{t+1}\theta_{1,t} - \boldsymbol{z}_{t+1}^T\boldsymbol{\theta}_{-1,t})(\sigma_t^2 x_{t+1} + \boldsymbol{r}_t^T\boldsymbol{z}_{t+1})}{\eta^2 + x_{t+1}^2\sigma_t^2 + 2x_{t+1}\boldsymbol{z}_{t+1}^T\boldsymbol{r}_t + \boldsymbol{z}_{t+1}^T\Sigma_t\boldsymbol{z}_{t+1}}$$

$$VAR[\beta|y_{t+1}] = \sigma_{t+1}^2 = \sigma_t^2 - \frac{(\sigma_t^2 x_{t+1} + \boldsymbol{r}_t^T\boldsymbol{z}_{t+1})^2}{\eta^2 + x_{t+1}^2\sigma_t^2 + 2x_{t+1}z_{t+1}^T\boldsymbol{r}_t + \boldsymbol{z}_{t+1}^T\Sigma_t\boldsymbol{z}_{t+1}} \quad (4.2)$$

The proof of Proposition 4.2.2 is provided in Appendix 4.7.2. The second term in (4.2) is a variance reduction term, which is the amount by which we expect the variance to decrease after observing the next response. We can calculate the variance reduction term for both treatment allocation $x \in \{-1, 1\}$ and for the next step, allocate x for which the variance reduction term is the largest.

For comparison, we consider the simpler model without network covariates, which is the original model (4.1) when z = 1. Then the model (4.1) simplifies as follows:

$$y = \boldsymbol{G}^T \boldsymbol{\phi} + \epsilon = x\beta + \gamma + \epsilon \tag{4.3}$$

where

- $\boldsymbol{G} = \begin{bmatrix} x \\ 1 \end{bmatrix}$, $x \in \{-1, 1\}$ is a treatment assignment
- $\epsilon \sim N(0, \eta^2)$ is an error term, where η^2 is known or estimated from historical data
- $\boldsymbol{\phi} = \begin{bmatrix} \beta \\ \gamma \end{bmatrix} \sim MVN_2(\theta, V)$ is a prior distribution, where $\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$ and $V = \begin{bmatrix} \sigma^2 & r \\ r & \omega \end{bmatrix}$
- $y|\phi \sim N(G\phi, \eta^2)$

Then under model assumptions (4.3) and Propositions 4.2.1 and 4.2.2, the distribution of $\beta | y_{t+1} \sim N(\theta_{1,t+1}, \sigma^2_{t+1})$, where

$$\theta_{1,t+1} = \theta_{1,t} + \frac{(y_{t+1} - x_{t+1}\theta_{1,t} - \theta_{2,t})(\sigma_t^2 x_{t+1} + r_t)}{\eta^2 + x_{t+1}^2 \sigma_t^2 + 2x_{t+1} r_t + \omega_t^2}$$

$$\sigma^2_{t+1} = \sigma_t^2 - \frac{(\sigma_t^2 x_{t+1} + r_t)^2}{\eta^2 + x_{t+1}^2 \sigma_t^2 + 2x_{t+1} r_t + \omega_t^2}$$

Again, we can calculate a variance reduction term for both alternatives $x \in \{-1, 1\}$ and at step $t+1$ allocate design point x for which the variance reduction term is maximum.

4.3 Proposed Algorithms

We want to compare the performance of the model with network covariates (4.1) and the model without network covariates (4.3) on simulated data. We have two alternatives, A and B, that we want to test within a network. As users come in sequentially, we want to use their network information to make a decision about what design to offer. For the simulation study we propose an Algorithm 2 that involves a model with network covariates (4.1) and Algorithm 3 for a model without network covariates (4.3). In both algorithms, when a new user arrives for testing we calculate a variance reduction term - we call it ν, for both alternatives of the design vector $\boldsymbol{x} \in \{-1, 1\}$ and assign the user a treatment for which ν is the largest. After that we observe the user's response, update networks covariates and/or parameters of a prior distribution. Then a new user comes in and we repeat the procedure until we reach a stopping criteria. In our study we use a maximum number of steps as a stopping criteria.

For the numerical simulation that follows we consider 4 network covariates:

- z_{1i} is a number of friends that user i has in his network or the number of members in his first tier neighborhood.

- z_{2i} is a number of members in the first tier neighborhood of user i, who responded favorably to treatment A or $x = 1$.

- z_{3i} is an average value of responses to treatment A or $x = 1$ in a user i first tier neighborhood.

- z_{4i} is an average value of responses to treatment B or $x = -1$ in a user i first tier neighborhood.

It is worth to point out that we use the network covariates from the previous step t to calculate the variance reduction term ν, unlike proposed in the equation (4.2). Since covariates z_{3i} and z_{4i} are dependent of the response y, in order to update them, we need to observe the response y_{t+1}, which is not possible at that point in the process.

Algorithm 2: Bayesian Sequential Design for A/B Testing Using a Model With Network Covariates

```
Input: $W$: network structure; $\eta^2$: error term variance
Estimate $\theta, V$: initial hyperparameters of the MVN prior;
$t = 0$
repeat
    Observe network information $z$ of a respondent $t + 1$
    Calculate variance reduction term $\nu$ for both alternatives $x \in (-1, 1)$:
    $\nu_x = \frac{(\sigma_t^2 x + r_t^T z_t)^2}{\eta^2 + x^2\sigma_t^2 + 2x z_t^T r_t + z_t^T \Sigma_t z_t}$
    if $\nu_1 > \nu_{-1}$ then
        $x_{t+1} = 1$
    else
        $x_{t+1} = -1$
    end
    Observe response $y_{t+1}$ based on selected alternative $x_{t+1}$
    Update network information $z_{t+1}$
    Update hyperparameters $\theta$ and $V$, where $G_{t+1} = [x_{t+1}, z_{t+1}]^T$:
    $\theta_{t+1} = \theta_t + \frac{Y_{t+1} - G_{t+1}^T \theta_t}{\eta^2 + G_{t+1}^T V_t G_{t+1}} V_t G_{t+1}$
    $V_{t+1} = V_t - \frac{V_t G_{t+1} G_{t+1}^T V_t}{\eta^2 + G_{t+1}^T V_t G_{t+1}}$
    Record the value of $\theta_{1,t+1}$
    $t = t + 1$
until Stopping Criteria reached;
```

We test the performance of our proposed Bayesian sequential design, that uses a model with network covariates against a design that does not use network covariates, using synthetic and real-world networks. We define the social network to be an undirected graph in the context of this work. The edges of this network are recorded by an $n \times n$ adjacency matrix $W = \{w_{ij}\}$ whose (i, j)-th entry is w_{ij}. The diagonal entries w_{ii}'s of this matrix are zeros, whereas the off-diagonal entries are

$$w_{ij} = \begin{cases} 1, & \text{if node } i \text{ and node } j \text{ are adjacent} \\ 0, & \text{otherwise.} \end{cases}$$

We call that two nodes are adjacent if they are connected by an edge. To generate a response y_{t+1} during the simulation we use Conditional Autoregressive (CAR) model

Algorithm 3: Bayesian Sequential Design for A/B Testing Using a Model Without Network Covariates

```
Input: $W$: network structure; $\eta^2$: error term variance
Estimate $\theta, V$: initial hyperparameters of the MVN prior;
$t = 0$
repeat
    Calculate variance reduction term $\nu$ for both alternatives $x \in (-1, 1)$:
    $\nu_x = \frac{(\sigma_t^2 x + r_t)^2}{\eta^2 + x^2\sigma_t^2 + 2xr_t + \omega_t^2}$
    if $\nu_1 > \nu_{-1}$ then
        $x_{t+1} = 1$
    else
        $x_{t+1} = -1$
    end
    Observe response $y_{t+1}$ based on selected alternative $x_{t+1}$
    Update hyperparameters $\theta$ and $V$, where $G_{t+1} = [x_{t+1}, 1]^T$:
    $\theta_{t+1} = \theta_t + \frac{Y_{t+1} - G_{t+1}^T\theta_t}{\eta^2 + G_{t+1}^T V_t G_{t+1}} V_t G_{t+1}$
    $V_{t+1} = V_t - \frac{V_t G_{t+1} G_{t+1}^T V_t}{\eta^2 + G_{t+1}^T V_t G_{t+1}}$
    Record the value of $\theta_{1,t+1}$
    $t = t + 1$
until Stopping Criteria reached;
```

(Besag, 1974):

$$y_i = \alpha + x_i\beta + \delta_i, \tag{4.4}$$

where

$$\delta_i|\boldsymbol{\delta}_{-i} \sim N\left(\rho\sum_{j\neq i}\tilde{w}_{ij}\delta_j, \tau_i^2\right), \tag{4.5}$$

$\boldsymbol{\delta}_{-i} = \{\delta_1, \ldots, \delta_{i-1}, \delta_{i+1}, \ldots, \delta_n\}$, $\tau_i^2 = \tau^2/z_{1i}$, with z_{1i} as described above, τ^2 is assumed to be known, $\tilde{w}_{ij} = w_{ij}/z_{1i}$ (row-standardized neighborhood matrix) and $x_i \in \{-1, 1\}$ is a treatment assignment. Since we make an assumption that users connected on a network tend to have similar reactions to the same treatment, we restrict the correlation parameter of the CAR model, ρ to be non-negative, i.e., $0 \leq \rho < 1$. Under the CAR model assumptions (4.4) and (4.5), Brook's Lemma (Brook, 1964) and Proposition 2.1 in Pokhilko et al. (2019), the covariance model in (4.5) can be expressed in a multivariate normal form:

$$\boldsymbol{\delta} = (\delta_1, \ldots, \delta_n)^\top \sim \mathcal{MVN}_n(0, \tau^2(D - \rho W)^{-1}),$$

where $D = \text{diag}(z_{11}, \ldots, z_{1n})$ and W is a neighborhood or adjacency matrix.

Now a few words about estimating initial hyperparameters θ and V of the MVN prior in model (4.1). We estimate them in batch from historical data that are simulated on a small subset of the network. Approximately 10% of total number of users on the network is randomly selected for those purposes. Then we create a random treatment allocation vector $x \in \{-1, 1\}$. Based on the design x we generate a vector of responses using CAR model (4.4). Without loss of generality we assume that $\alpha = 0$. Using the design x and response y, we create the network covariates. After that we estimate the hyperparameters θ and V for the model with covariates (4.1) by fitting the Multiple Linear Regression model:

$$y = \theta_1 x + \boldsymbol{\theta}_{-1}\boldsymbol{z} + \epsilon$$

and for the model without covariates (4.3) by fitting the Simple Linear Regression

$$y = \theta_1 x + \theta_2 + \epsilon$$

Finally, we use the initial parameters of the prior distribution of ϕ as an input for the main simulation.

4.4 Numerical Study on Synthetic Networks

This section compares two approaches: network method, that uses network covariates as described in Algorithm 2 and independent approach, that does not use the network covariates using Algorithm 3. The comparison will be done on random networks. For a network with n nodes, the $n \times n$ adjacency matrix records the edges of this network. We randomly assign 0 and 1 to the upper or lower off-diagonal entries of this matrix. The proportion of ones is specified to be p. The zero entry means that the two corresponding nodes are not adjacent, whereas the entry of one means the opposite. The proportion p is referred to as the density of this network.

We proceed by first generating 1000 random networks of size 1000 and density $p = 0.05$. We assumed that true value of $\beta = 0.05$. For each network we generated one realization of the response based on either a covariate-assisted model or a model without covariates. As a result we had 1000 updated parameters, θ and σ^2 at each step t and for each model. Next we evaluated the performance of each simulation scenario described by Algorithms 2 and 3 by first plotting at each step $t = 1 \ldots T$ of the procedure the probability of correct selection, $P_{cs} = \frac{\sum_{k=1}^{K} I_{\{\theta_k > 0\}}}{K}$, where $K = 1000$ is a total number of parameter estimates we have at each step t and $T = 800$ is a total number of steps. We also plotted sum of squared errors or $SSE = \sum_{k=1}^{K}(\theta_k - \beta)^2$, calculated at each step t of the procedure. The complete procedure is described in Algorithm 4.

Algorithm 4: Comparison of Bayesian Sequential Designs for A/B Testing Procedure With and Without Network Covariates on Multiple Networks

```
Input: $\rho$: network correlation parameter; $\tau/\beta$: noise-to-signal ratio;
       $K$: number of random networks; $T$: total number of steps
for $k \leftarrow 1$ to $K$ do
    Create a random network
    Algorithm 2
    Algorithm 3
end
Calculate SSE for both models at each step $t$:  $SSE = \sum_{k=1}^{K}(\theta_k - \beta)^2$
Calculate probability of correct selection, $P_{cs}$, for both models at each step
  $t$:  $P_{cs} = \frac{\sum_{k=1}^{K} I_{\{\theta_k > 0\}}}{K}$
Plot $P_{cs}$ and $SSE$ for both models at each step $t$
```

We repeated the procedure in Algorithm 4 for different values of $\rho \in \{0, 0.1, 0.3, 0.5, 0.7, 0.9\}$ and noise-to-signal ratios, $\tau/\beta \in \{10, 100\}$, that were used in response generation. The results are depicted in Figures 4.1, 4.2, 4.3 and 4.4. Figure 4.1 shows P_{cs} for noise-to-signal ratio of 10. We observe that for all values of ρ, P_{cs} is 1 for both methods and all steps. In Figure 4.2 we increase noise-to-signal ratio to 100. We observe that P_{cs} is higher for a model without covariates for all values of the correlation parameter ρ.

Figures 4.3 and 4.4 show SSE at each step t for different values of ρ and noise-to-signal ratio. We observe that SSE is the smallest for a model without covariates

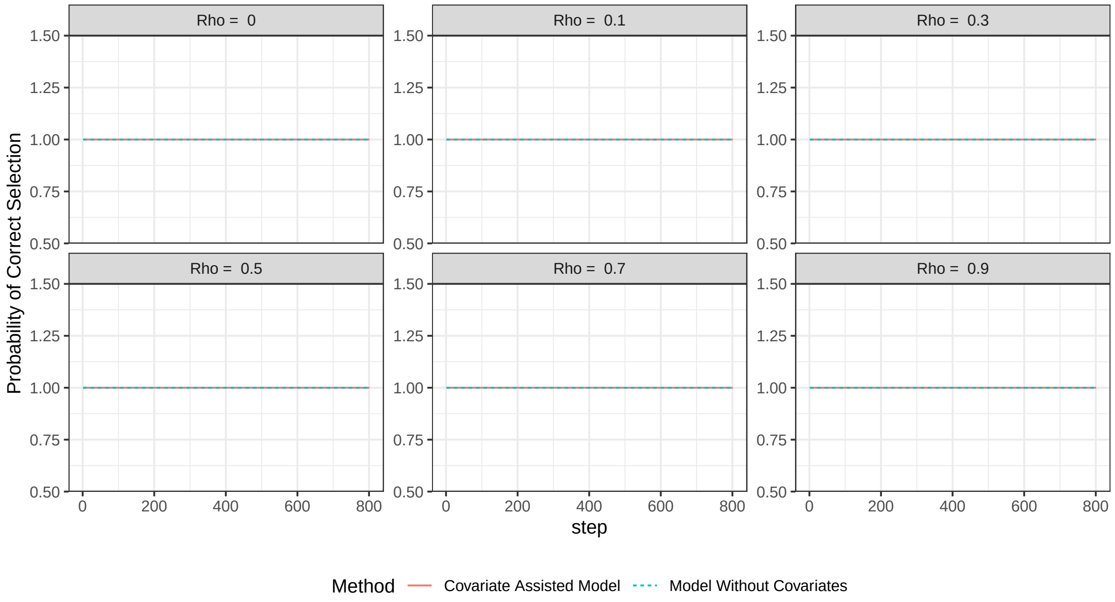

Figure 4.1: Probability of correct selection calculated at each step of the procedure, using noise-to-signal ratio of 10 over 1000 random networks of size 1000 and density 0.05

regardless the choice of ρ or noise-to-signal ratio. We can conclude that a covariate assisted model loses in terms of P_{cs} and SSE to a model without covariates as tested on synthetic random networks.

4.5 Numerical Study on Real-World Networks

This section studies an example of real-world network and compares the performance of our proposed covariate assisted model to the performance of a model without covariates. We consider one of the ego networks extracted from Facebook (Leskovec and Mcauley, 2012). These networks are collected from survey participants who used the Facebook App. Ego networks consist of a focal node ("ego") and the nodes to whom ego is directly connected to (these are called "alters") plus the ties, if any, among the alters. Each alter in an ego network has his/her own ego network, and all ego networks interlock to form the human social network. The focal node represents the main character or person in the ego network and the alters are the "friends" in her circle or people she is connected with. Those friends can also be connected among themselves, forming their own ego networks, and that is represented by the

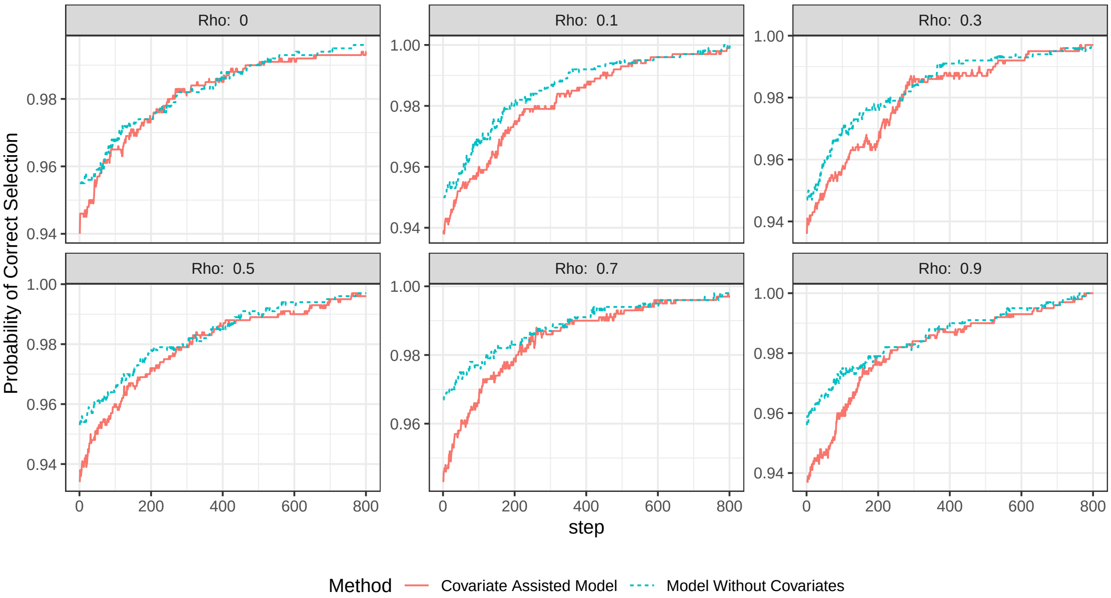

Figure 4.2: Probability of correct selection calculated at each step of the procedure, using noise-to-signal ratio of 100 over 1000 random networks of size 1000 and density 0.05

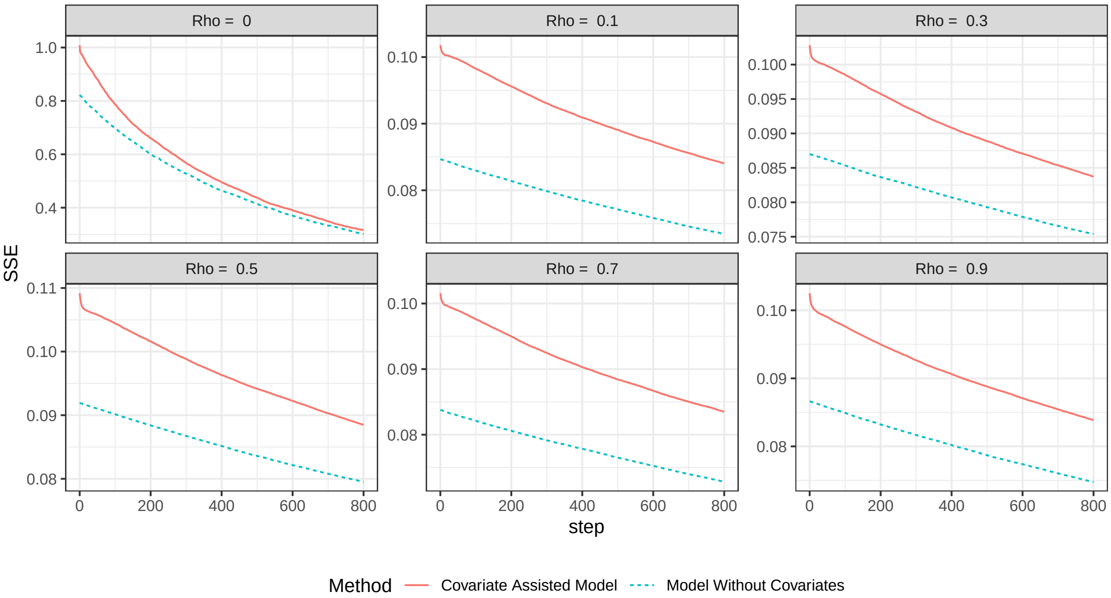

Figure 4.3: SSE calculated at each step of the procedure, using noise-to-signal ratio of 10 over 1000 random networks of size 1000 and density 0.05

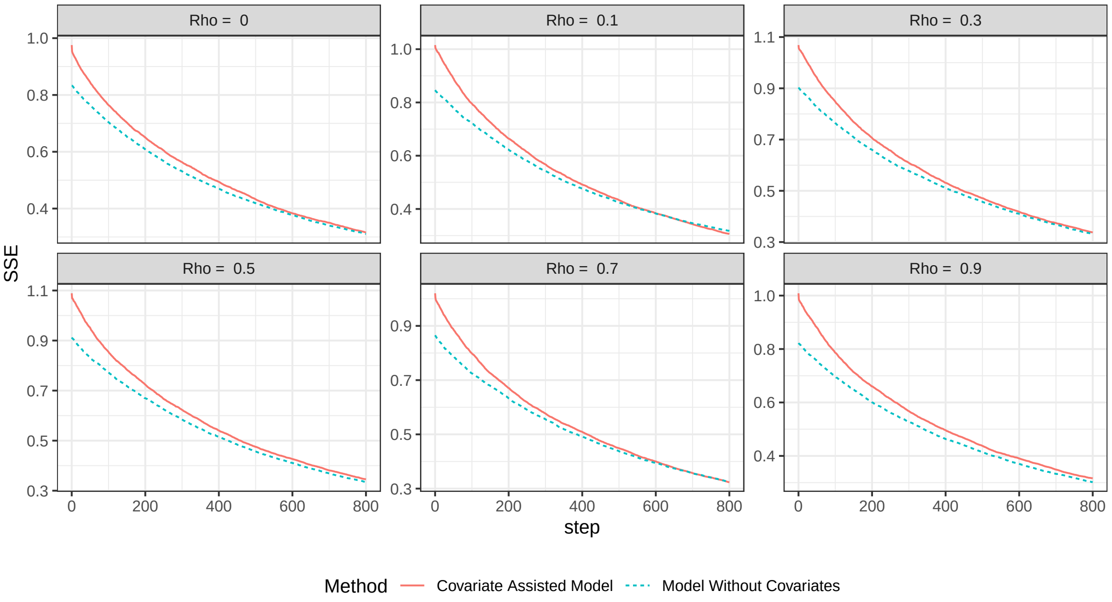

Figure 4.4: SSE calculated at each step of the procedure, using noise-to-signal ratio of 100 over 1000 random networks of size 1000 and density 0.05

ties or edges among alters.

For our simulation we select an ego network of size 1034 that has a density $p = 0.05$. Since we only have one network, in order for us to calculate probability of correct selection P_{cs} and SSE, we generate $K = 100$ iterations or realizations of response using CAR model assumptions (4.4) and (4.5) at each step of the procedure, t. As a result we have 100 parameter estimates, θ and σ^2, for each step t and each model. The updated version of the procedure is provided in Algorithm 5.

We repeat the process described in Algorithm 5 for different values of $\rho \in \{0, 0.1, 0.3, 0.5, 0.7, 0.9\}$ and noise-to-signal ratios, $\tau/\beta \in \{10, 100\}$. The results of the simulation are provided in Figures 4.5, 4.6, 4.7 and 4.8. Figure 4.5 shows P_{cs} for noise-to-signal ratio of 10. We observe that for all values of ρ, P_{cs} is 1, except for the model without covariates falling behind for a first few steps of the procedure for $\rho = 0$ and 0.7. In Figure 4.6 we increase noise-to-signal ratio to 100. We observe that P_{cs} is higher for a covariate assisted model, especially at the beginning of the learning procedure, for all values of the correlation parameter, ρ.

Figures 4.7 and 4.8 show SSE at each step t for different values of ρ and noise-to-signal ratio. We observe that SSE is the smallest for a covariate assisted model

Algorithm 5: Comparison of Bayesian Sequential Designs for A/B Testing Procedure With and Without Network Covariates on a Single Network

```
Input: $W$: network adjacency matrix; $\rho$: network correlation parameter;
       $\tau/\beta$: noise-to-signal ratio, $K$: total number of iterations; $T$: total
       number of steps
for $k \leftarrow 1$ to $K$ do
    Generate a vector of errors, $\boldsymbol{\delta}$
    Estimate hyperparameters, $\theta_0$ and $V_0$
    Algorithm 2
    Algorithm 3
end
Calculate SSE for both models at each step $t$:  $SSE = \sum_{k=1}^{K}(\theta_k - \beta)^2$
Calculate probability of correct selection, $P_{cs}$, for both models at each step
  $t$:  $P_{cs} = \frac{\sum_{k=1}^{K} I_{\{\theta_k > 0\}}}{K}$
Plot $P_{cs}$ and $SSE$ for both models at each step $t$
```

regardless the choice of ρ and noise-to-signal ratio. We can conclude that a covariate assisted model outperforms a model without covariates in terms of P_{cs} and SSE, as tested on the real-world network.

4.6 Discussion

In this paper we introduce a covariate assisted Bayesian sequential design for A/B testing on social networks. We use the CAR model to characterize the dependence of the responses from the adjacent nodes in a network. Numerical examples on synthetic and real-world networks are provided to show the effectiveness of the proposed method. Our approach outperforms a model without network covariates on Facebook ego networks in terms of decreasing SSE of treatment effect estimate, and increasing probability of correct selection, P_{cs}, but loses in performance to a model without covariates when tested on purely random synthetic networks. The choice of a network correlation parameter, ρ, used in response generation during the simulation study, shows no effect on the performance of both models. This means that the covariate assisted model is robust to the degree of network dependency assumed for user responses as long as the proposed method is applied on clustered social networks.

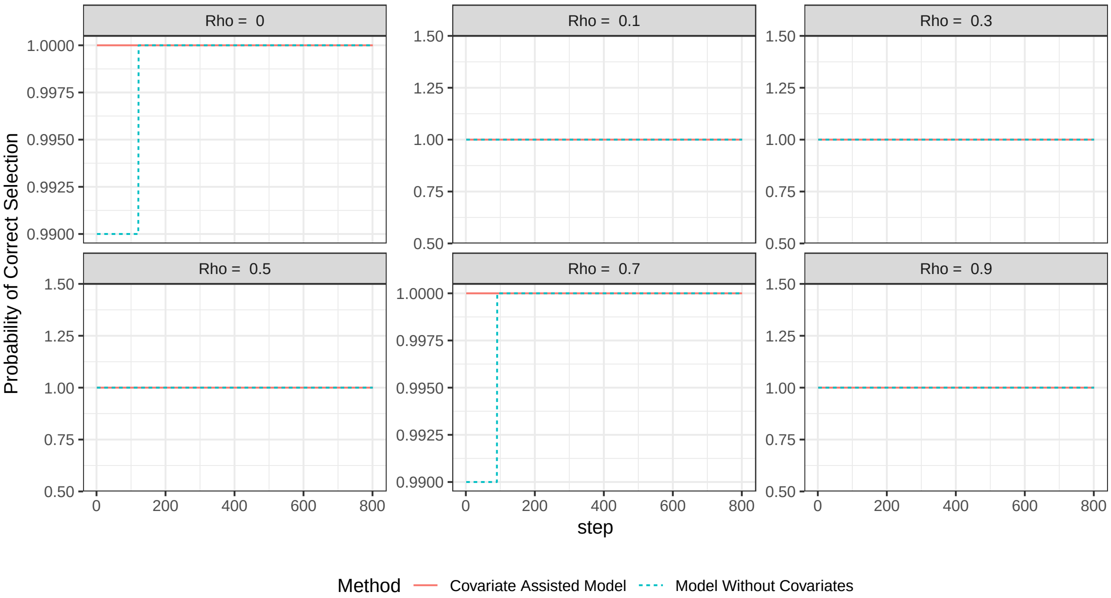

Figure 4.5: Probability of correct selection calculated at each step of the procedure, using noise-to-signal ratio of 10 over one Facebook ego network of size 1034 and density 0.05

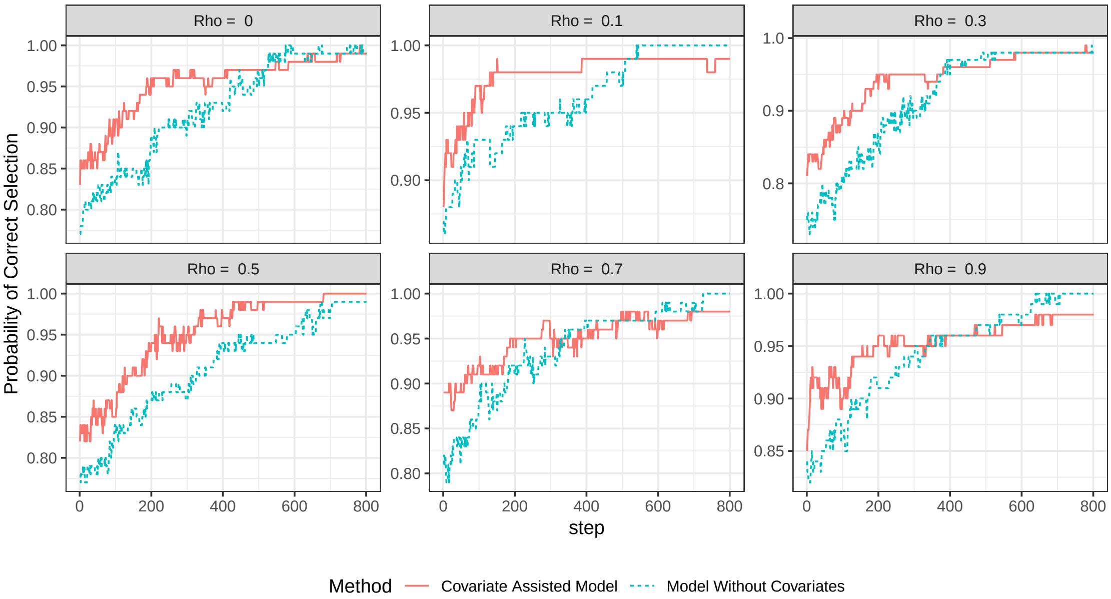

Figure 4.6: Probability of correct selection calculated at each step of the procedure, using noise-to-signal ratio of 100 over one Facebook ego network of size 1034 and density 0.05

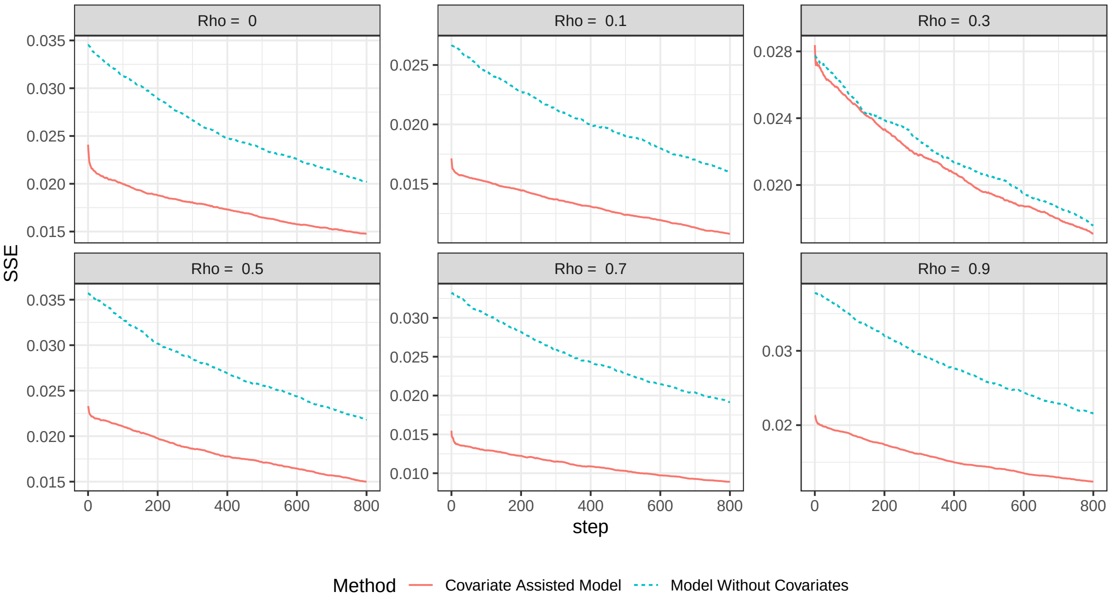

Figure 4.7: SSE calculated at each step of the procedure, using noise-to-signal ratio of 10 over one Facebook ego network of size 1034 and density 0.05

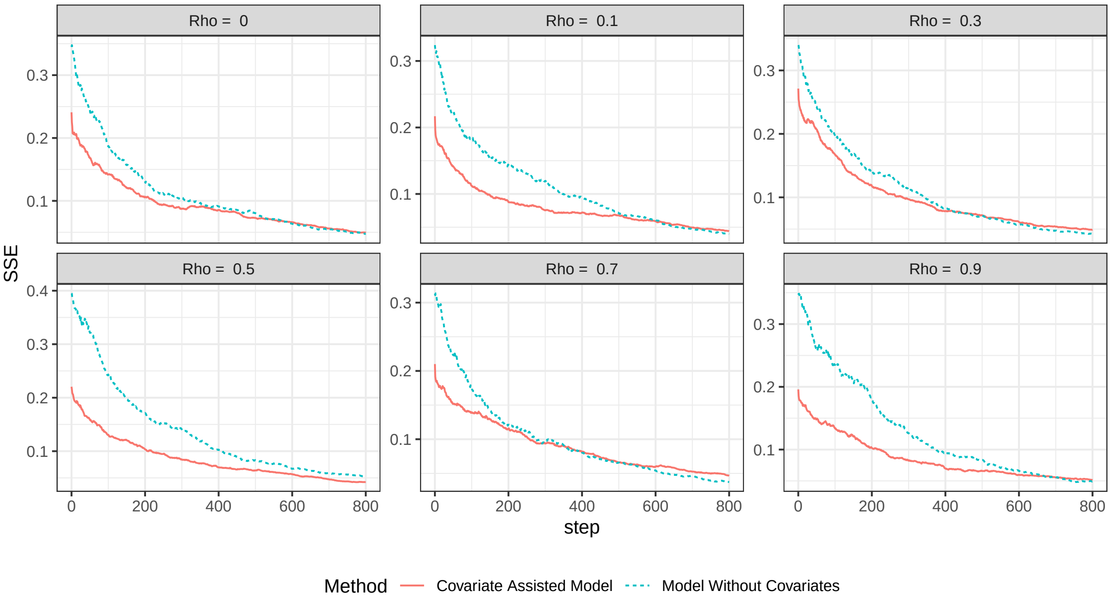

Figure 4.8: SSE calculated at each step of the procedure, using noise-to-signal ratio of 100 over one Facebook ego network of size 1034 and density 0.05

4.7 Appendix

4.7.1 The Proof of Proposition 4.2.1

We want to derive the posterior distribution of $\phi_{t+1}|y_{t+1}$ at step $t+1$. To make the description complete, we follow the standard development of Bayesian theory in scientific literature. Using the model (4.1), we get:

$$\begin{aligned}
f(\phi_{t+1}|y_{t+1}) &= \frac{f(\phi_{t+1}, y_{t+1})}{f(y_{t+1})} \propto f(y_{t+1}|\phi_{t+1}) f(\phi_{t+1}) \\
&\propto exp\{-\frac{1}{2\delta^2}(y_{t+1} - G_{t+1}^T\phi_{t+1})^2\} exp\{-\frac{1}{2}(\phi_{t+1} - \theta_t)^T \Sigma_t^{-1}(\phi_{t+1} - \theta_t)\} \\
&\propto exp\{-\frac{1}{2}[\frac{y_{t+1}^2}{\delta^2} - \frac{y_{t+1}G_{t+1}^T\phi_{t+1}}{\delta^2} - \frac{\phi_{t+1}^T G_{t+1} y_{t+1}}{\delta^2} + \phi_{t+1}^T \frac{G_{t+1}G_{t+1}^T}{\delta^2}\phi_{t+1} \\
&\quad + \phi_{t+1}^T \Sigma_t^{-1}\phi_{t+1} - \phi_{t+1}^T \Sigma_t^{-1}\theta_t - \theta_t^T \Sigma_t^{-1}\phi_{t+1} + \theta_t^T \Sigma_t^{-1}\theta_t]\}
\end{aligned}$$

After isolating the terms related to ϕ_{t+1} we get:

$$\begin{aligned}
f(\phi_{t+1}|y_{t+1}) \propto exp\{ &-\frac{1}{2}[\phi_{t+1}^T(\Sigma_t^{-1} + \frac{G_{t+1}G_{t+1}^T}{\delta^2})\phi_{t+1} - \phi_{t+1}^T(\Sigma_t^{-1}\theta_t + \frac{y_{t+1}}{\delta^2}G_{t+1}) \\
&- (\theta_t^T\Sigma_t^{-1} + \frac{y_{t+1}}{\delta^2}G_{t+1}^T)\phi_{t+1}]\}
\end{aligned}$$

Let $A = \Sigma_t^{-1} + \frac{G_{t+1}G_{t+1}^T}{\delta^2}$ and $b = \Sigma_t^{-1}\theta_t + \frac{y_{t+1}}{\delta^2}G_{t+1}$. Then

$$\begin{aligned}
f(\phi_{t+1}|y_{t+1}) &\propto exp\{-\frac{1}{2}[\phi_{t+1}^T A\phi_{t+1} - \phi_{t+1}^T b - b^T\phi_{t+1}]\} \\
&\propto exp\{-\frac{1}{2}[\phi_{t+1}^T A\phi_{t+1} - \phi_{t+1}^T b - b^T\phi_{t+1} + b^T A^{-1} b]\} \\
&\propto exp\{-\frac{1}{2}[\phi_{t+1}^T A\phi_{t+1} - \phi_{t+1}^T AA^{-1} b - b^T A^{-1}A\phi_{t+1} + b^T A^{-1}AA^{-1}b]\}
\end{aligned}$$

Let $\Sigma_{t+1} = A^{-1}$ and $\theta_{t+1} = A^{-1}b$, then

$$\begin{aligned}
f(\phi_{t+1}|y_{t+1}) &\propto exp\{-\frac{1}{2}[\phi_{t+1}^T\Sigma_{t+1}^{-1}\phi_{t+1} - \phi_{t+1}^T\Sigma_{t+1}^{-1}\theta_{t+1} - \theta_{t+1}^T\Sigma_{t+1}^{-1}\phi_{t+1} + \theta_{t+1}^T\Sigma_{t+1}^{-1}\theta_{t+1}]\} \\
&\propto exp\{-\frac{1}{2}(\phi_{t+1} - \theta_{t+1})^T\Sigma_{t+1}^{-1}(\phi_{t+1} - \theta_{t+1})\}
\end{aligned}$$

We end up with a kernel of a multivariate normal distribution with a mean vector

$\theta_{t+1} = A^{-1}b = (\Sigma_t^{-1} + \frac{G_{t+1}G_{t+1}^T}{\delta^2})^{-1}(\Sigma_t^{-1}\theta_t + \frac{y_{t+1}}{\delta^2}G_{t+1})$ and a covariance matrix $\Sigma_{t+1} = A^{-1} = (\Sigma_t^{-1} + \frac{G_{t+1}G_{t+1}^T}{\delta^2})^{-1}$. To further simplify the mean and covariance of the posterior function, we will use a formula from the work of Sherman and Morrison (1950) and Bartlett (1951):

$$(B + uv^T)^{-1} = B^{-1} - \frac{B^{-1}uv^TB^{-1}}{1 + v^TB^{-1}u},$$

where $B \in R^{n\times n}$ is an invertible square matrix and $u, v \in R^n$ are vectors. Applying this formula, we get the following results:

$$\begin{aligned}
\Sigma_{t+1} &= (\Sigma_t^{-1} + \frac{G_{t+1}G_{t+1}^T}{\delta^2})^{-1} = \Sigma_t - \frac{\Sigma_t G_{t+1}G_{t+1}^T\Sigma_t}{\delta^2} / (1 + \frac{G_{t+1}^T\Sigma_t G_{t+1}}{\delta^2}) \\
&= \Sigma_t - \frac{\Sigma_t G_{t+1}G_{t+1}^T\Sigma_t}{\delta^2}\frac{\delta^2}{\delta^2 + G_{t+1}^T\Sigma_t G_{t+1}} = \Sigma_t - \frac{\Sigma_t G_{t+1}G_{t+1}^T\Sigma_t}{\delta^2 + G_{t+1}^T\Sigma_t G_{t+1}}
\end{aligned}$$

$$\begin{aligned}
\theta_{t+1} &= (\Sigma_t - \frac{\Sigma_t G_{t+1}G_{t+1}^T\Sigma_t}{\delta^2 + G_{t+1}^T\Sigma_t^{-1}G_{t+1}})^{-1}(\Sigma_t^{-1}\theta_t + \frac{y_{t+1}}{\delta^2}G_{t+1}) \\
&= \theta_t + \frac{y_{t+1}}{\delta^2}\Sigma_t G_{t+1} - \frac{\Sigma_t G_{t+1}G_{t+1}^T\Sigma_t\Sigma_t^{-1}\theta_t}{\delta^2 + G_{t+1}^T\Sigma_t G_{t+1}} \quad - \frac{\Sigma_t G_{t+1}G_{t+1}^T\Sigma_t G_{t+1}y_{t+1}}{\delta^2(\delta^2 + G_{t+1}^T\Sigma_t G_{t+1})} \\
&= \theta_t + \frac{y_{t+1}\Sigma_t G_{t+1}(\delta^2 + G_{t+1}^T\Sigma_t G_{t+1}) - \Sigma_t G_{t+1}G_{t+1}^T\Sigma_t G_{t+1}y_{t+1}}{\delta^2(\delta^2 + G_{t+1}^T\Sigma_t G_{t+1})} - \frac{\Sigma_t G_{t+1}G_{t+1}^T\theta_t}{\delta^2 + G_{t+1}^T\Sigma_t G_{t+1}} \\
&= \frac{y_{t+1}\delta^2\Sigma_t G_{t+1} + y_{t+1}\Sigma_t G_{t+1}G_{t+1}^T\Sigma_t G_{t+1} - y_{t+1}\Sigma_t G_{t+1}G_{t+1}^T\Sigma_t G_{t+1}}{\delta^2(\delta^2 + G_{t+1}^T\Sigma_t G_{t+1})} + \theta_t - \frac{\Sigma_t G_{t+1}G_{t+1}^T\theta_t}{\delta^2 + G_{t+1}^T\Sigma_t G_{t+1}} \\
&= \theta_t + \frac{y_{t+1}\Sigma_t G_{t+1}}{\delta^2 + G_{t+1}^T\Sigma_t G_{t+1}} - \frac{G_{t+1}^T\theta_t\Sigma_t G_{t+1}}{\delta^2 + G_{t+1}^T\Sigma_t G_{t+1}} = \theta_t + \frac{y_{t+1} - G_{t+1}^T\theta_t}{\delta^2 + G_{t+1}^T\Sigma_t G_{t+1}}\Sigma_t G_{t+1}
\end{aligned}$$

Therefore after observing our first response y_{t+1}, the posterior distribution of $\phi_{t+1}|y_{t+1} \sim MVN_{d+1}(\theta_t + \frac{y_{t+1} - G_{t+1}^T\theta_t}{\delta^2 + G_{t+1}^T\Sigma_t G_{t+1}}\Sigma_t G_{t+1}, \Sigma_t - \frac{\Sigma_t G_{t+1}G_{t+1}^T\Sigma_t}{\delta^2 + G_{t+1}^T\Sigma_t G_{t+1}})$.

4.7.2 The proof of Proposition 4.2.2

Under model assumptions (4.1) and Proposition 4.2.1, we update the parameters of the prior distribution $\phi|y$ using:

$$\theta = \theta + \frac{y - G^T\theta}{\delta^2 + G^TVG}VG \tag{4.6}$$

$$V = V - \frac{VGG^TV}{\delta^2 + G^TVG} \tag{4.7}$$

where $\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \boldsymbol{\theta}_{-1} \end{bmatrix}$, $\boldsymbol{G} = \begin{bmatrix} x \\ \boldsymbol{z} \end{bmatrix}$ and $V = \begin{bmatrix} \sigma^2 & \boldsymbol{\rho}^T \\ \boldsymbol{\rho} & \Sigma \end{bmatrix}$

To ease the notation, we omit the step numbering in the subscripts of the above equations. Then $\beta|y$ is normally distributed with the mean θ_1 and variance σ^2 from parameters $\boldsymbol{\theta}$ and V of posterior distribution. Since we need to express θ_1 and σ^2 in terms of updated values, we focus on the following terms from equations (4.6) and (4.7):

$$\begin{aligned} VGG^TV &= \begin{bmatrix} \sigma^2 & \boldsymbol{\rho}^T \\ \boldsymbol{\rho} & \Sigma \end{bmatrix} \begin{bmatrix} x \\ \boldsymbol{z} \end{bmatrix} \begin{bmatrix} x & \boldsymbol{z} \end{bmatrix} \begin{bmatrix} \sigma^2 & \boldsymbol{\rho}^T \\ \boldsymbol{\rho} & \Sigma \end{bmatrix} \\ &= \begin{bmatrix} \sigma^2 x + \boldsymbol{\rho}^T \boldsymbol{z} \\ \boldsymbol{\rho} x + \Sigma \boldsymbol{z} \end{bmatrix} \begin{bmatrix} \sigma^2 x + \boldsymbol{z}^T \boldsymbol{\rho} & x\boldsymbol{\rho}^T + \boldsymbol{z}^T \Sigma \end{bmatrix} \\ &= \begin{bmatrix} (\sigma^2 x + \boldsymbol{\rho}^T \boldsymbol{z})^2 & (\sigma^2 x + \boldsymbol{\rho}^T \boldsymbol{z})(x\boldsymbol{\rho}^T + \boldsymbol{z}^T\Sigma) \\ (\boldsymbol{\rho} x + \Sigma\boldsymbol{z})(\sigma^2 x + \boldsymbol{\rho}^T \boldsymbol{z}) & (\boldsymbol{\rho} x + \Sigma\boldsymbol{z})(x\boldsymbol{\rho}^T + \boldsymbol{z}^T\Sigma) \end{bmatrix} \end{aligned}$$

$$\begin{aligned} G^TVG &= \begin{bmatrix} x & \boldsymbol{z} \end{bmatrix} \begin{bmatrix} \sigma^2 & \boldsymbol{\rho}^T \\ \boldsymbol{\rho} & \Sigma \end{bmatrix} \begin{bmatrix} x \\ \boldsymbol{z} \end{bmatrix} = (x\sigma^2 + \boldsymbol{z}^T\boldsymbol{\rho})x + (x\boldsymbol{\rho}^T + \boldsymbol{z}^T\Sigma)\boldsymbol{z} \\ &= x^2\sigma^2 + 2x\boldsymbol{z}^T\boldsymbol{\rho} + \boldsymbol{z}^T\Sigma\boldsymbol{z} \end{aligned}$$

$$G^T\theta = \begin{bmatrix} x & \boldsymbol{z} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \boldsymbol{\theta}_{-1} \end{bmatrix} = x\theta_1 + \boldsymbol{z}^T\boldsymbol{\theta}_{-1}$$

$$VG = \begin{bmatrix} \sigma^2 & \boldsymbol{\rho}^T \\ \boldsymbol{\rho} & \Sigma \end{bmatrix} \begin{bmatrix} x \\ \boldsymbol{z} \end{bmatrix} = \begin{bmatrix} \sigma^2 x + \boldsymbol{\rho}^T \boldsymbol{z} \\ x\boldsymbol{\rho} + \Sigma\boldsymbol{z} \end{bmatrix}$$

Putting the values G^TVG, $G^T\theta$, the element of the first row and first column of VGG^TV matrix and the first element of the vector VG into equations (4.6) and (4.7), we get the mean and variance of $\beta|y$. Then the conclusion holds.

4.7.3 Preliminary Study

Covariate Effect Estimation

First we want to estimate the effect of d network covariates by running multiple linear regression model

$$y = x\beta + \boldsymbol{z}^T\gamma + \epsilon, \tag{4.8}$$

where $x \in \{-1, 1\}$ is a treatment allocation, $\boldsymbol{z}^T \in R^{d+1}$ is a vector of network covariates and an intercept, described in Section 4.3, and $\epsilon \sim N(0, \eta^2)$ is an error term. We want to compare the average p-values of those covariate coefficient estimates for various values of $\rho \in \{0, 0.1, 0.3, 0.5, 0.7, 0.9\}$ and different noise-to-signal ratios $\tau/\beta \in \{10, 100\}$, used in simulation. We consider 3 scenarios:

1. We create a random network of size 1000 and density 0.05 and random treatment allocation vector $\boldsymbol{x} \in \{-1, 1\}$. Then we generate a response vector $\boldsymbol{y}$ using CAR model (4.4). After that we collect the network covariate matrix $\boldsymbol{z}$. Next we fit the multiple linear regression model 4.8 and record the p-values for covariate coefficient estimates. We repeat this procedure on 100 random networks, each time generating a new random treatment allocation vector and recording the p-values. Finally, we average those p-values to be displayed in Figure 4.9. This analysis is repeated for each combination of ρ and noise-to-signal ratio used in response generation.

2. We create a random network of size 1000 and density 0.05 and random treatment allocation vector $\boldsymbol{x} \in \{-1, 1\}$. Then we generate a response vector $\boldsymbol{y}$ using CAR model (4.4). After that we collect the network covariate matrix $\boldsymbol{z}$. Next we fit the multiple linear regression model 4.8 and record the p-values for covariate coefficient estimates. We repeat this procedure 100 times using the same network, but each time re-generating random treatment allocation $\boldsymbol{x}$ and

response $\boldsymbol{y}$. This analysis is done for each combination of ρ and noise-to-signal ratio used in response generation, and the results are depicted in Figure 4.10.

3. Finally we repeat the procedure described in the second scenario, but instead of a randomly generated network we use a Facebook ego network of size 1034 and density 0.05. The results of this analysis is depicted in Figure 4.11.

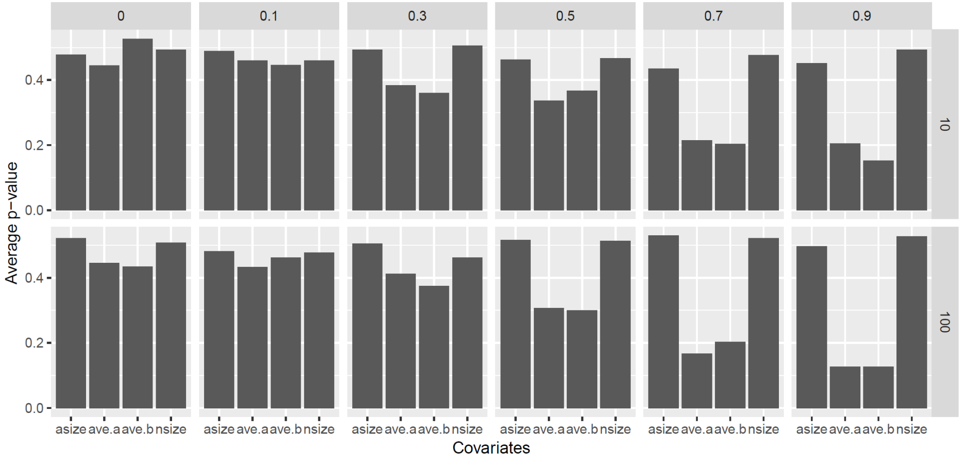

Figure 4.9: Significance of covariates for different values of ρ (top) and noise-to-signal ratios (right), evaluated over 100 random networks of size 1000 and density 0.05

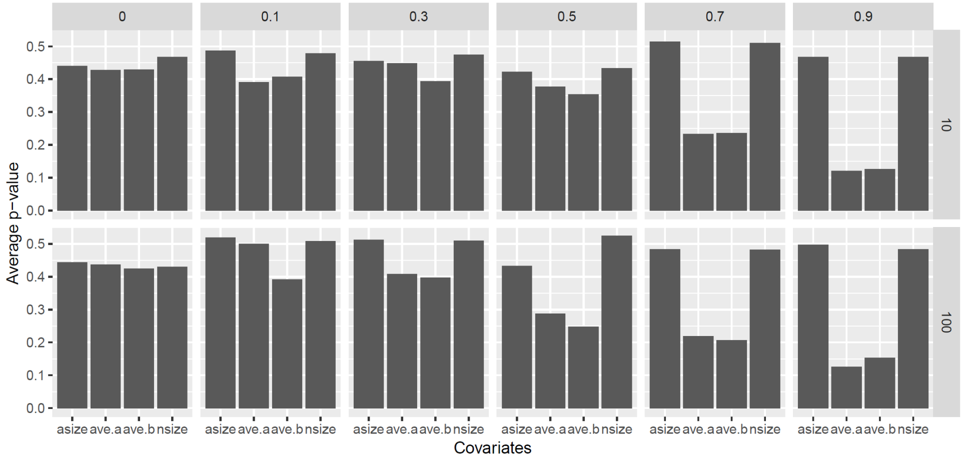

Figure 4.10: Significance of covariates for different values of ρ (top) and noise-to-signal ratios (right), evaluated over a fixed random network of size 1000 and density 0.05

The results are similar for the three scenarios and are as follows:

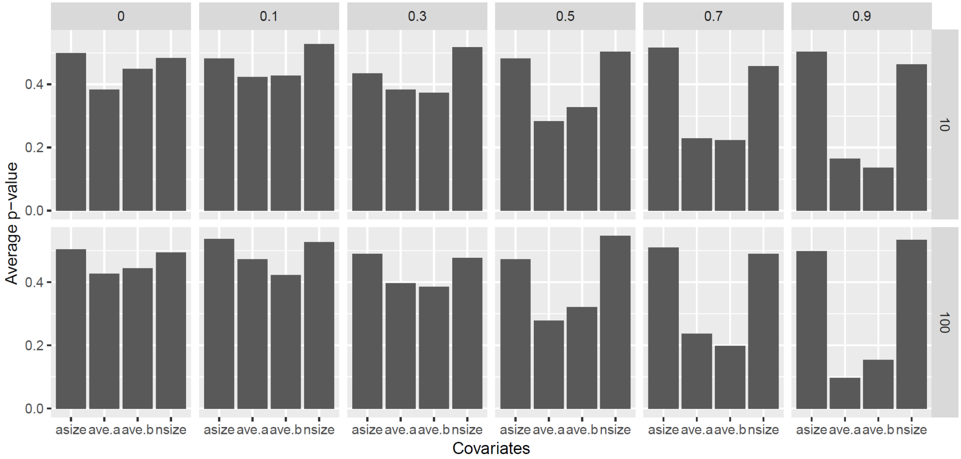

Figure 4.11: Significance of covariates for different values of ρ (top) and noise-to-signal ratios (right), evaluated over Facebook ego network of size 1034 and density 0.05

- p-values for the covariate estimates of z_1 and z_2 are not significant regardless of the parameter settings ρ and noise-to-signal ratio;
- the results are similar for different settings of noise-to-signal ratio;
- p-values for the covariate estimates of z_3 and z_4 tend to decrease as ρ increases.

Nested Models Comparison

Now we want to compare two nested models: the simple model without the network covariates $y = x\beta + \gamma + \epsilon$ and the model with covariates $y = x\beta + \boldsymbol{z}^T\gamma + \epsilon$. We need to decide if a more complex model is "better" at predicting the data than a simpler one. If the more complex model does not significantly contribute more information, then there's no reason to accept the parameters estimated there from. The simpler, "more parsimonious" model is preferred on all grounds. We set up our hypothesis as follows and use χ^2 test for nested models comparison: $\mathrm{H}_o : \gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 = 0$ versus H_α : At least one $\gamma_i \neq 0$.

We use similar scenarios as in Section 4.7.3:

1. We create a random network of size 1000 and density 0.05 and random treatment allocation vector $\boldsymbol{x} \in \{-1, 1\}$. Then we generate a response vector $\boldsymbol{y}$

using CAR model (4.4). After that we collect the network covariate matrix $\boldsymbol{z}$. Next we conduct χ^2 test for nested models (with and without covariates) and record the p-values from the test. We repeat this procedure on 100 random networks, each time generating a new random treatment allocation vector and recording the p-values from χ^2 test. Finally, we average those p-values to be displayed in Figure 4.12. This analysis is repeated for each combination of ρ and noise-to-signal ratio used in response generation.

2. We create a random network of size 1000 and density 0.05 and random treatment allocation vector $\boldsymbol{x} \in \{-1, 1\}$. Then we generate a response vector $\boldsymbol{y}$ using CAR model (4.4). After that we collect the network covariate matrix $\boldsymbol{z}$. Next we conduct χ^2 test for nested models (with and without covariates) and record the p-values from the test. We repeat this procedure 100 times using the same network, but each time re-generating random treatment allocation $\boldsymbol{x}$ and response vector $\boldsymbol{y}$. This analysis is done for each combination of ρ and noise-to-signal ratio used in response generation, and the results are depicted in Figure 4.10.

3. Finally we repeat the procedure described in the second scenario, but instead of a randomly generated network we use a Facebook ego network of size 1034 and density 0.05. The results of this analysis is depicted in Figure 4.11.

In all cases we observe that network covariates contribute significant information to the model when high values of correlation parameter ρ are used in simulation regardless of the choice of noise-to-signal ratio.

4.7.4 Additional Numerical Results on Synthetic Networks

Large Noise-to-Signal Ratio

Here we compare the performance of a covariate assisted model to a model without covariates using a large noise-to-signal ratio. We repeated the experiment described in Section 4.4, using $\tau/\beta = 1000$. The results are provided in Figures 4.15 and 4.16.

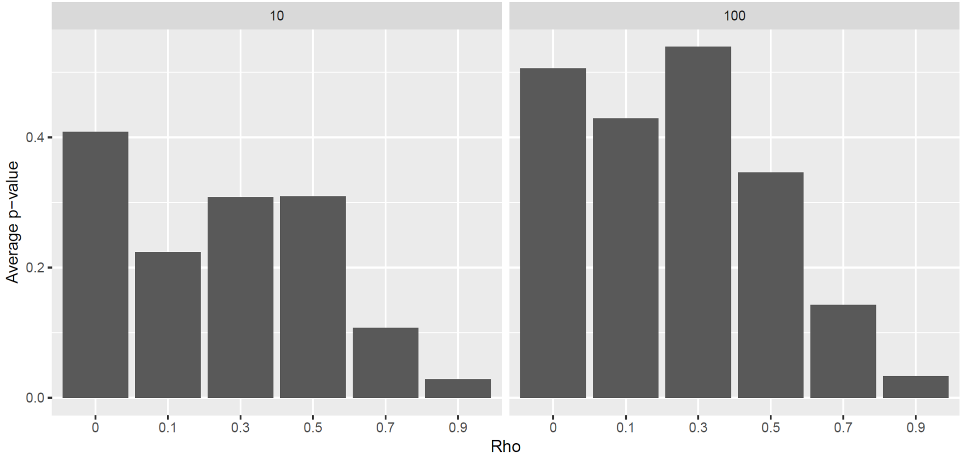

Figure 4.12: χ^2 test of nested models (with and without covariates) for different values of ρ and noise-to-signal ratios (top), evaluated over 100 random networks of size 1000 and density 0.05

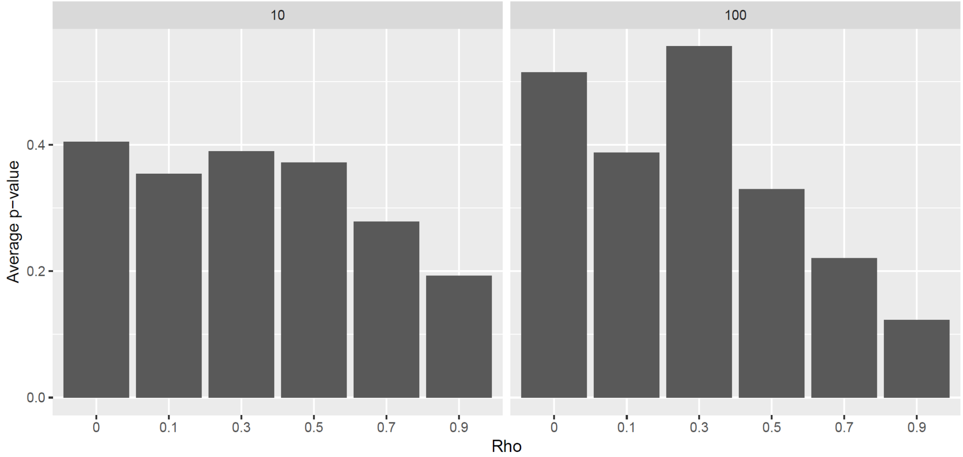

Figure 4.13: χ^2 test of nested models (with and without covariates) for different values of ρ and noise-to-signal ratios (top), evaluated over a fixed random network of size 1000 and density 0.05

From Figure 4.15 we observe that the covariate assisted model performs better when $\rho = 0.1$ and 0.9, a model without covariates outperforms the covariate-assisted model when $\rho = 0.3$, and for other values of ρ the results are inconclusive. The SSEs from Figure 4.16 are very similar for both models.

Testing a Subset of Covariates over a Fixed Random Network

According to the results of the preliminary study in Appendix 4.7.3, certain covariates tend to be more significant than others. In this section we compare the

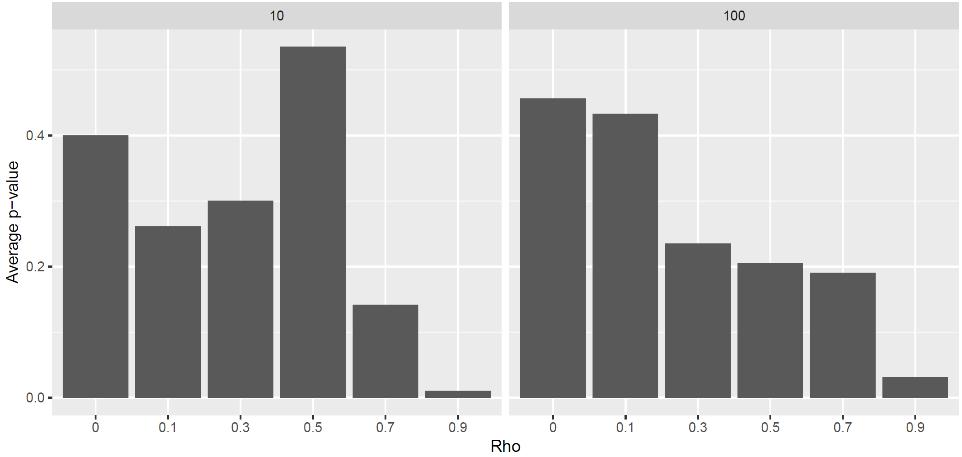

Figure 4.14: χ^2 test of nested models (with and without covariates) for different values of ρ and noise-to-signal ratios (top), evaluated over a Facebook ego network of size 1034 and density 0.05

performance of the model without network covariates to the model that uses only a subset of covariates considered in this paper, namely, z_3 and z_4, where z_3 is an average value of responses to treatment A or $x = 1$ in a user's first tier neighborhood and z_4 is an average value of responses to treatment B or $x = -1$ in a user's first tier neighborhood.

We follow the procedure outlined in Algorithm 5. For the simulation we use a single random network of size 1000 and density $p = 0.05$. Since we only have one network, in order for us to calculate probability of correct selection P_{cs} and SSE, we generate 100 realizations of response using CAR model assumption (4.4) and (4.5) at each step of the procedure, t. As a result we have 100 updates of parameters θ and σ^2 for each step t and each model.

The outcomes of the simulation are depicted in Figures 4.17, 4.18, 4.19 and 4.20. For noise-to-signal ratio of 10 both models correctly select the sign of the true parameter with probability of 1 for all values of network correlation parameter ρ, as shown in Figure 4.17. However, as we increase noise-to-signal ratio to 100, Figure 4.18 illustrates that a model without covariates performs better for all cases of ρ. This and the fact that SSEs for all ρ and noise-to-signal ratio appear to be smaller for a model without covariates (Figures 4.19 and 4.20) conform with the results of

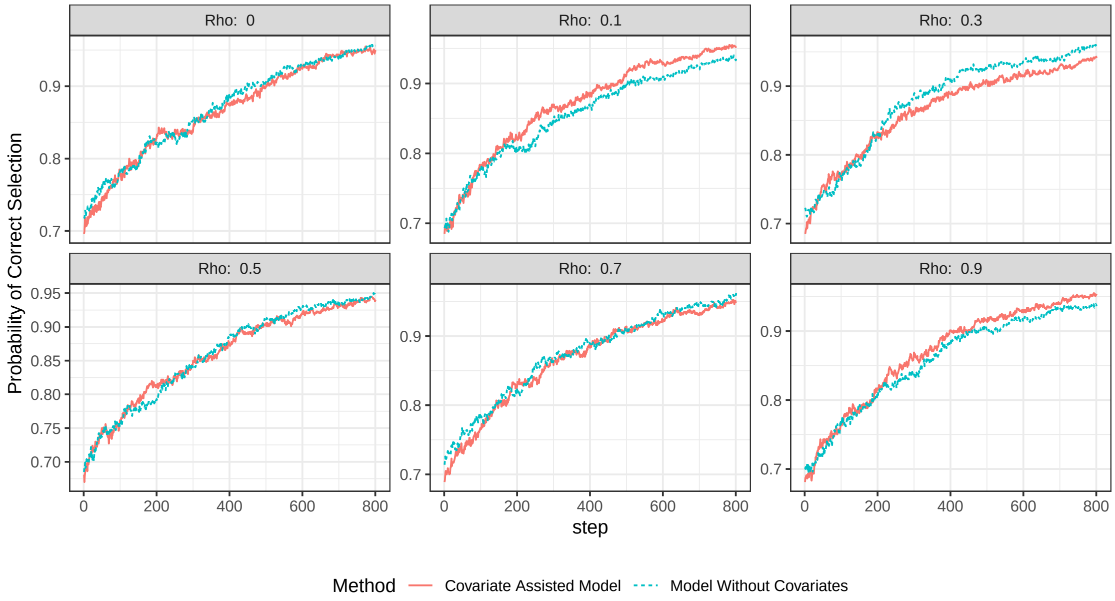

Figure 4.15: Probability of correct selection calculated at each step of the procedure, using noise-to-signal ratio of 1000 over 1000 random networks of size 1000 and density 0.05

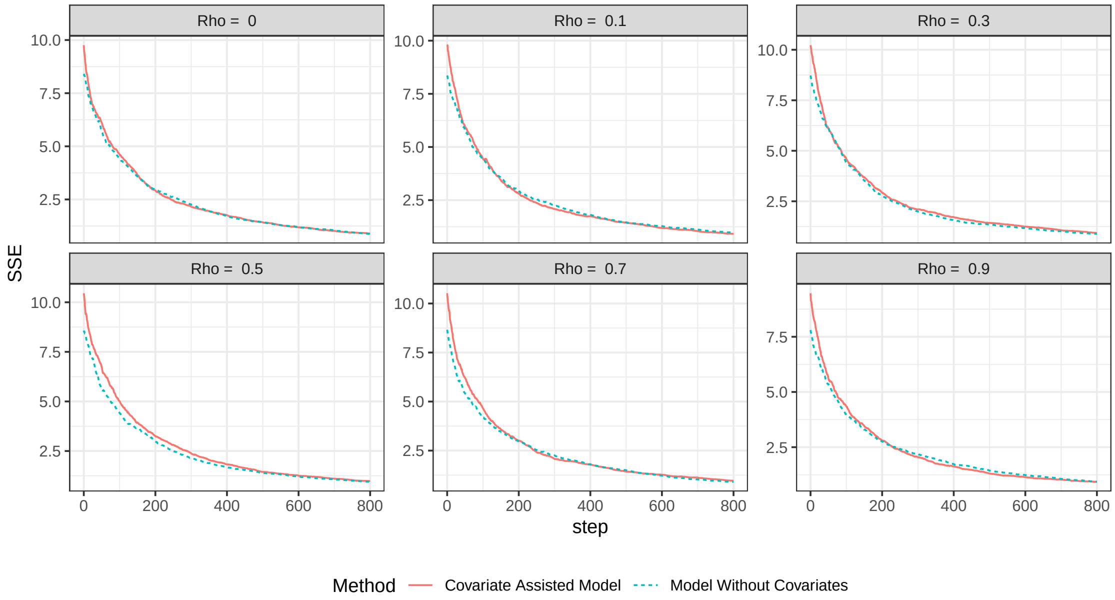

Figure 4.16: SSE calculated at each step of the procedure, using noise-to-signal ratio of 1000 over 1000 random networks of size 1000 and density 0.05

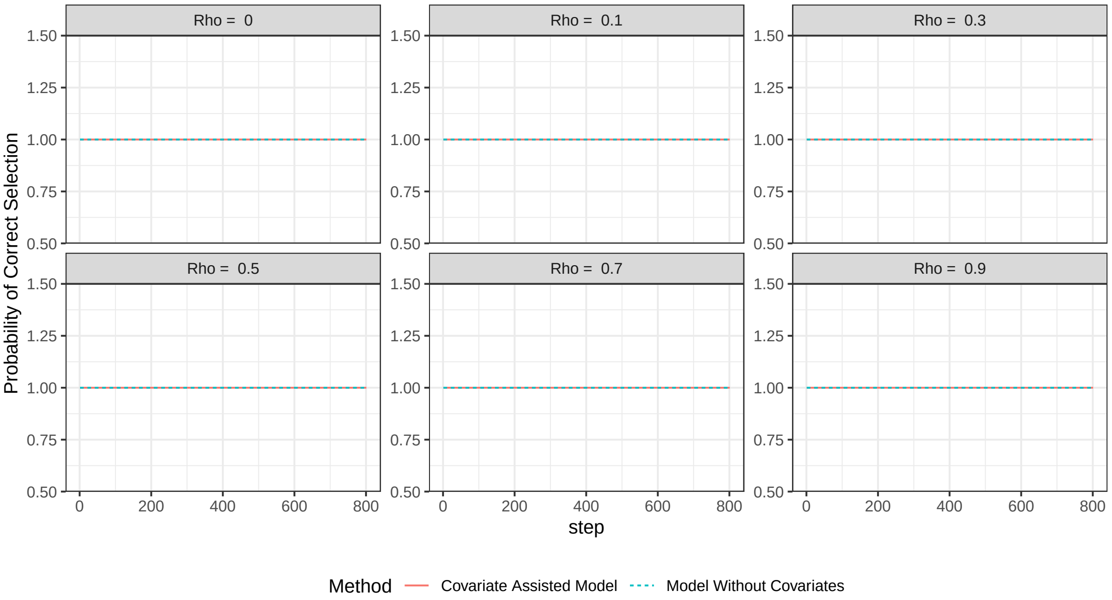

Figure 4.17: Probability of correct selection calculated at each step of the procedure, using noise-to-signal ratio of 10 and a subset of covariates over a fixed random network of size 1000 and density 0.05

section 4.4.

4.7.5 Additional Numerical Results on Real-World Networks

Large Noise-to-Signal Ratio

We want to compare the performance of a covariate assisted model to a model without covariates on a Facebook ego network of size 1034 using a large noise-to-signal ratio. We repeated the experiment described in Section 4.5, using a noise-to-signal ratio of 1000. The results are provided in Figures 4.21 and 4.22. From Figure 4.21 we observe a lot of variability in P_{cs} for both models. It looks like there is too much noise in the response and both models are struggling with learning the true parameter of β distribution. From Figure 4.22 the SSE of the covariate assisted model are smaller than SSE of the model without covariates for all values of network correlation parameter, ρ, especially at the first few hundrends of steps of the learning process.

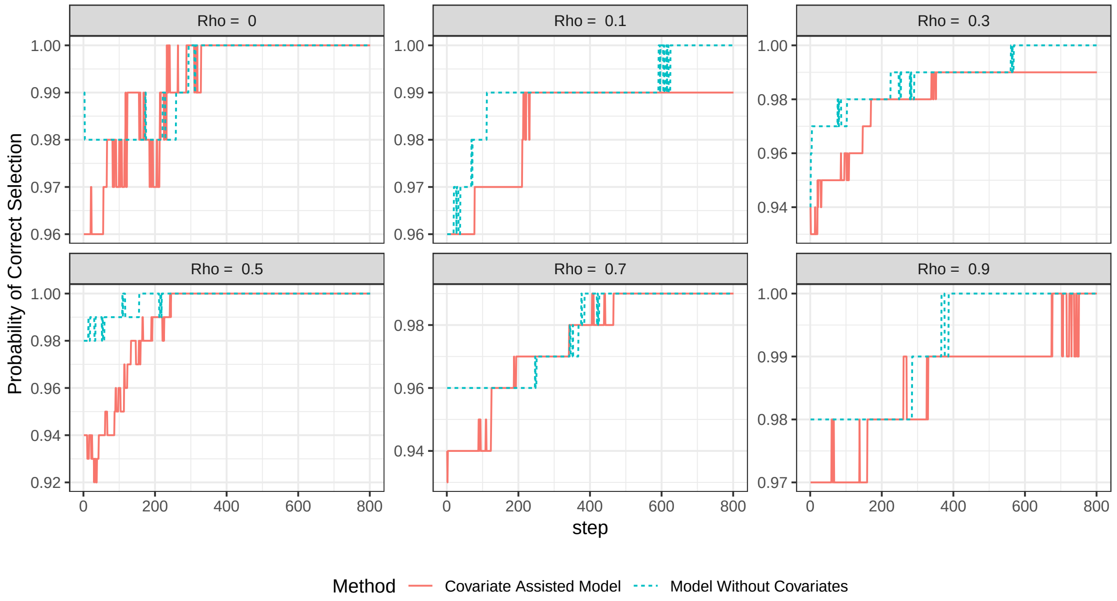

Figure 4.18: Probability of correct selection calculated at each step of the procedure, using noise-to-signal ratio of 100 and a subset of covariates over a fixed random network of size 1000 and density 0.05

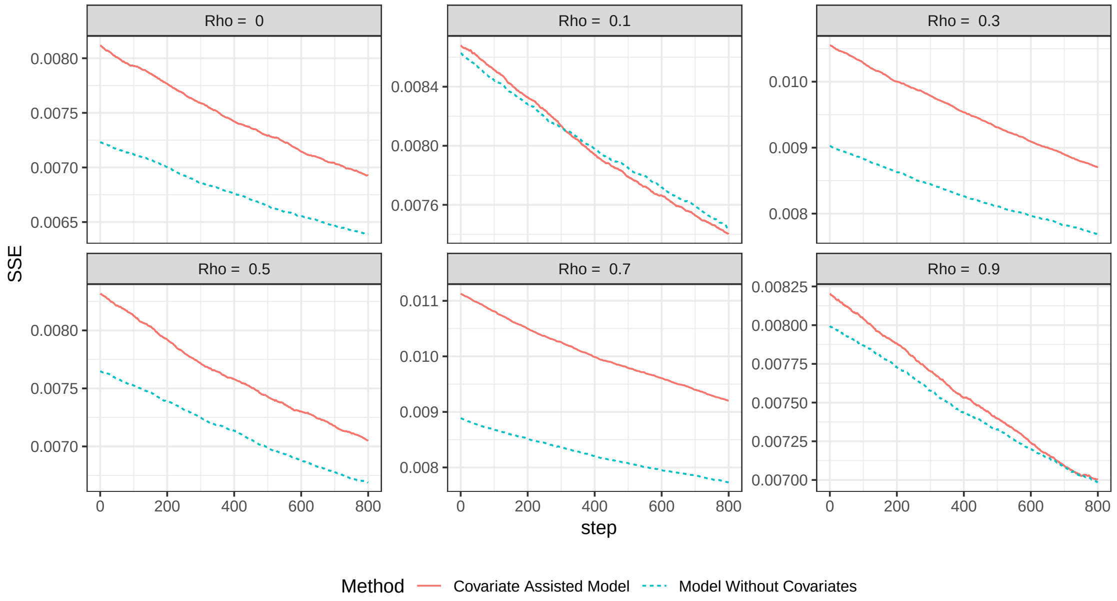

Figure 4.19: SSE calculated at each step of the procedure, using noise-to-signal ratio of 10 and a subset of covariates over a fixed random network of size 1000 and density 0.05

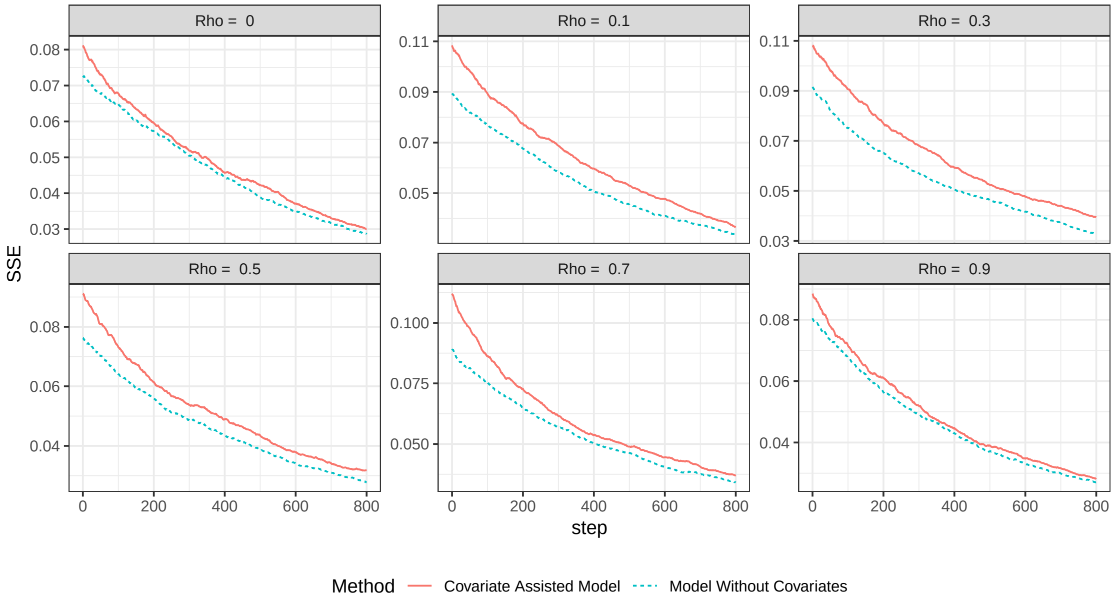

Figure 4.20: SSE calculated at each step of the procedure, using noise-to-signal ratio of 100 and a subset of covariates over a fixed random network of size 1000 and density 0.05

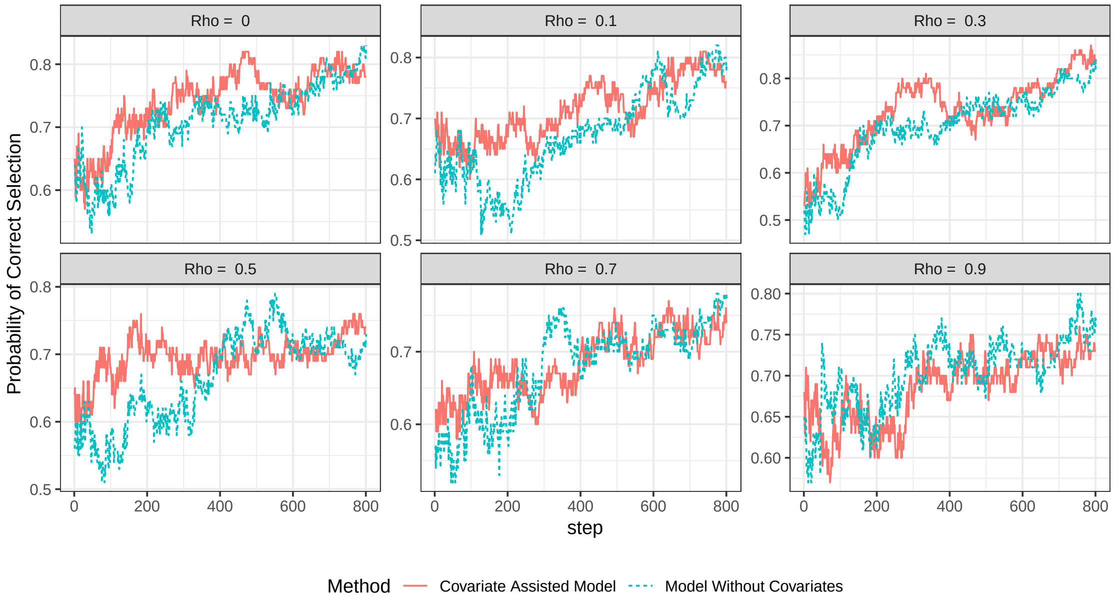

Figure 4.21: Probability of correct selection calculated at each step of the procedure, using noise-to-signal ratio of 1000 over a Facebook ego network of size 1034 and density 0.05

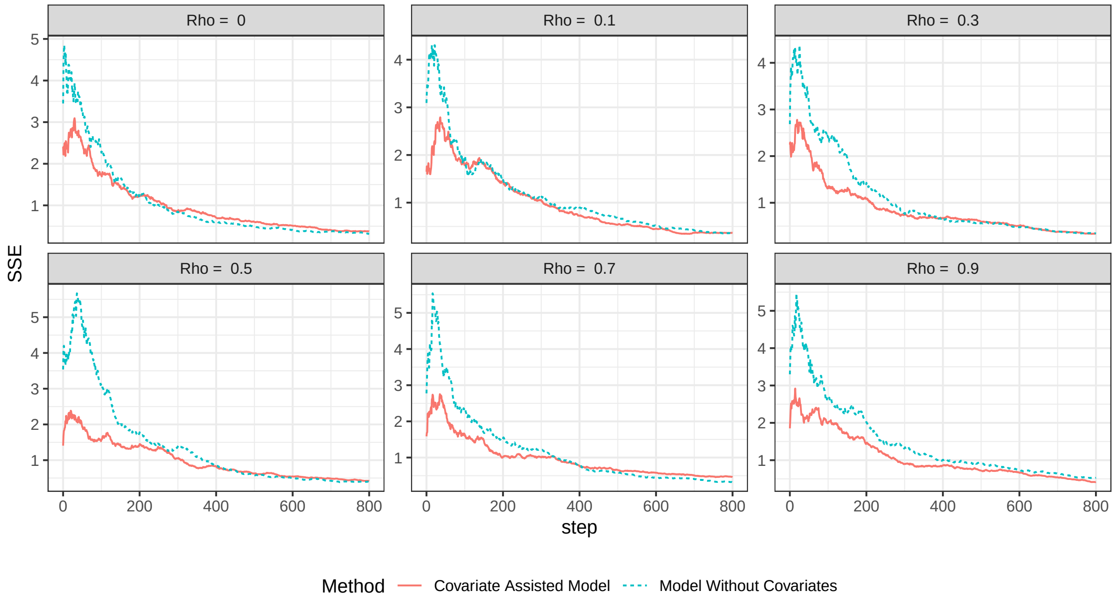

Figure 4.22: SSE calculated at each step of the procedure, using noise-to-signal ratio of 1000 over a Facebook ego network of size 1034 and density 0.05

Testing a Subset of Covariates

In this section we compare the performance of the model without network covariates to the model that uses only a subset of covariates considered in this paper, namely, z_3 and z_4, where z_3 is an average value of responses to treatment A or $x = 1$ in a user's first tier neighborhood and z_4 is an average value of responses to treatment B or $x = -1$ in a user's first tier neighborhood. Those two covariates proved to be more significant than others in the preliminary study that is described in Appendix 4.7.3.

We follow the procedure outlined in Algorithm 5. For the simulation we use a Facebook ego network of size 1034 and density $p = 0.05$. Since we only have one network, in order for us to calculate probability of correct selection P_{cs} and SSE, we generate 100 realizations of response using CAR model assumption (4.4) and (4.5) at each step of the procedure, t. As a result we have 100 updates of parameters θ and σ^2 for each step t and each model.

The outcomes of the simulation are depicted in Figures 4.23, 4.24, 4.25 and 4.26. For noise-to-signal ratio of 10 both models correctly select the sign of the true parameter with probability of 1 for most values of correlation parameter ρ,

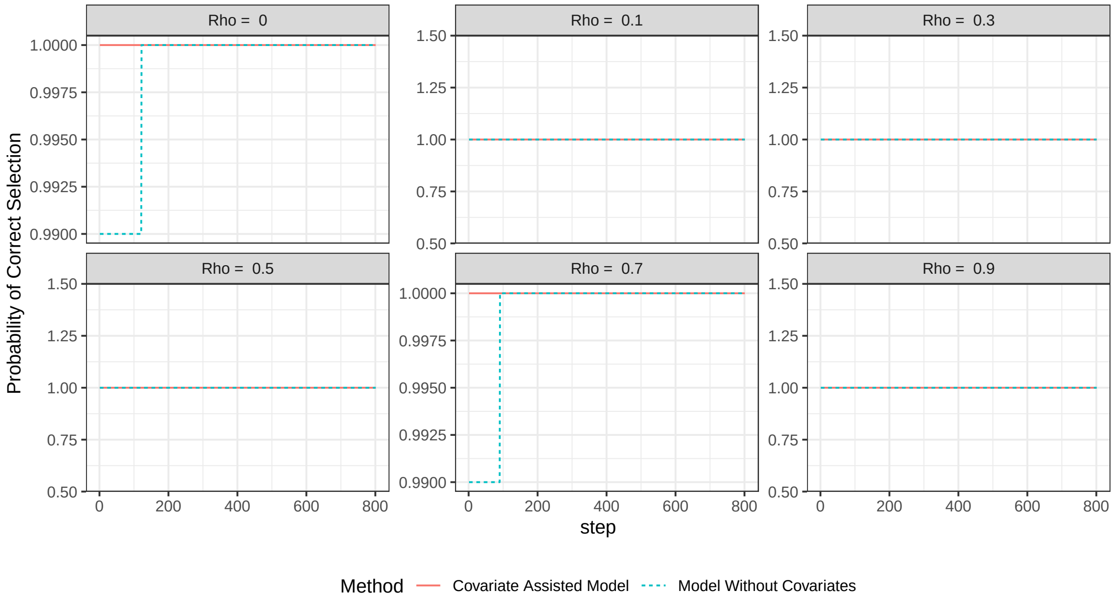

Figure 4.23: Probability of correct selection calculated at each step of the procedure, using noise-to-signal ratio of 10 and a subset of covariates over a Facebook ego network of size 1034 and density 0.05

except for a model without covariates falling slightly behind a covariate assisted model for $\rho = 0$ and 0.7, as shown in Figure 4.23. Figure 4.24 illustrates that as we increase noise-to-signal ration to 100, a covariate assisted model outperforms a model without covariates for all cases of ρ. This and the fact that SSEs for all ρ and noise-to-signal ratio appear to be smaller for a covariate assisted model (Figures 4.25 and 4.26) conform with the results of section 4.5.

Additional Results on Facebook ego network of size 747

Here we conduct a simulation study, as described in Section 4.5, but on a different Facebook ego network of size 747 and density 0.11. The results of the simulation are provided in Figures 4.27, 4.28, 4.29 and 4.30. Figure 4.27 shows P_{cs} for noise-to-signal ratio of 10. We observe that for all values of ρ, P_{cs} is 1, except for the model without covariates falling slightly behind for $\rho = 0$, 0.3 and 0.9. In Figure 4.28 we increase noise-to-signal ratio to 100. We observe that P_{cs} is higher for a covariate assisted model for all values of the correlation parameter, ρ.

Figures 4.29 and 4.30 show SSE at each step t for different values of ρ and noise-

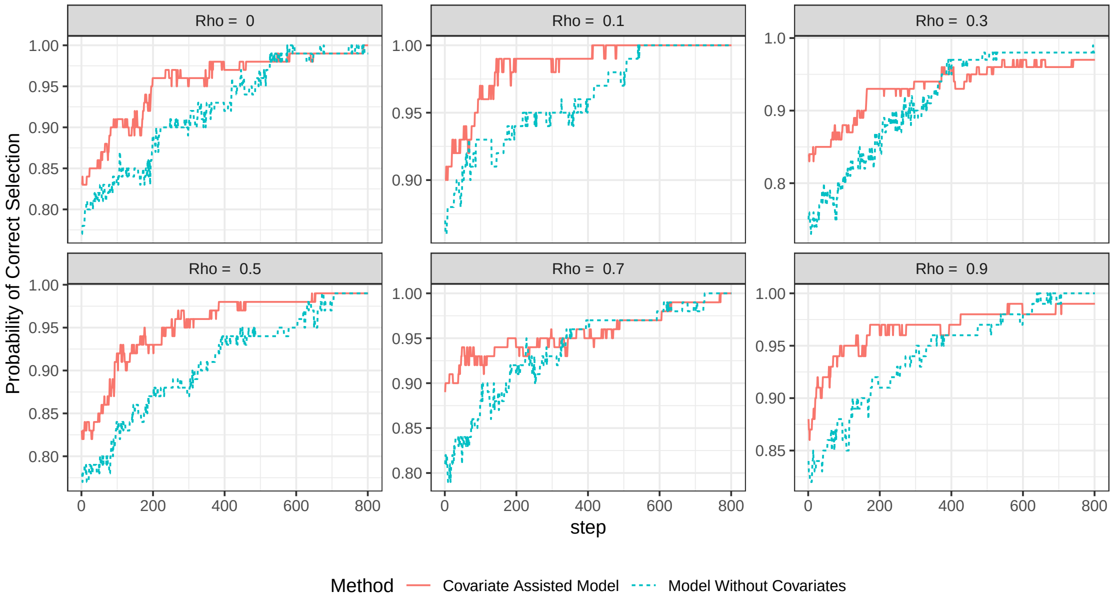

Figure 4.24: Probability of correct selection calculated at each step of the procedure, using noise-to-signal ratio of 100 and a subset of covariates over a Facebook ego network of size 1034 and density 0.05

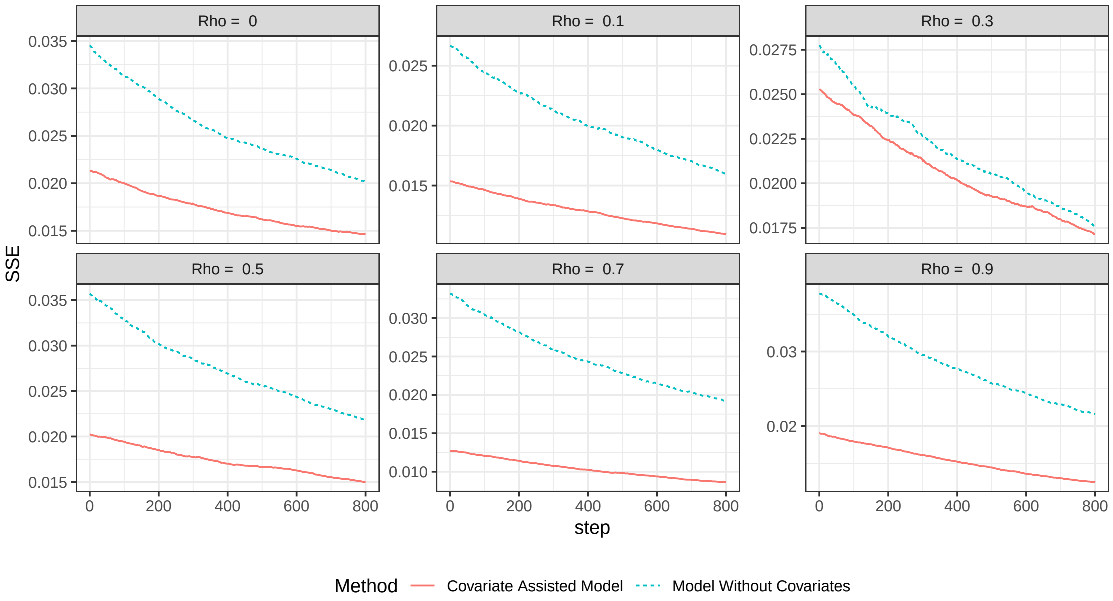

Figure 4.25: SSE calculated at each step of the procedure, using noise-to-signal ratio of 10 and a subset of covariates over a Facebook ego network of size 1034 and density 0.05

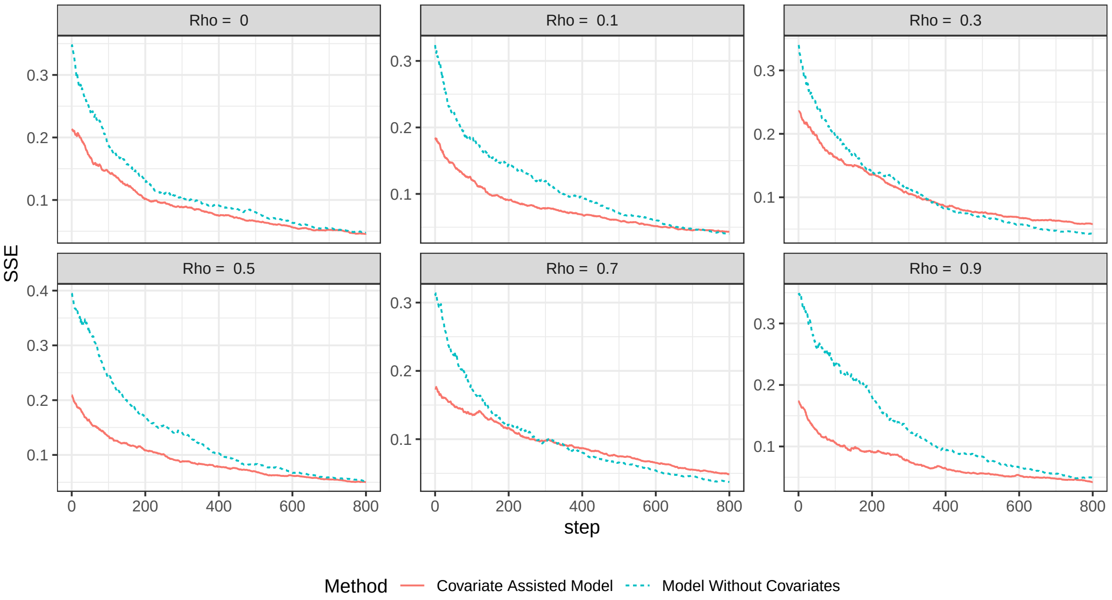

Figure 4.26: SSE calculated at each step of the procedure, using noise-to-signal ratio of 100 and a subset of covariates over a Facebook ego network of size 1034 and density 0.05

to-signal ratio. We observe that SSE is the smallest for a covariate assisted model regardless the choice of ρ and noise-to-signal ratio. We can conclude that a covariate assisted model outperforms a model without covariates in terms of P_{cs} and SSE, as tested on this real-world network, which conforms with the results in Section 4.5.

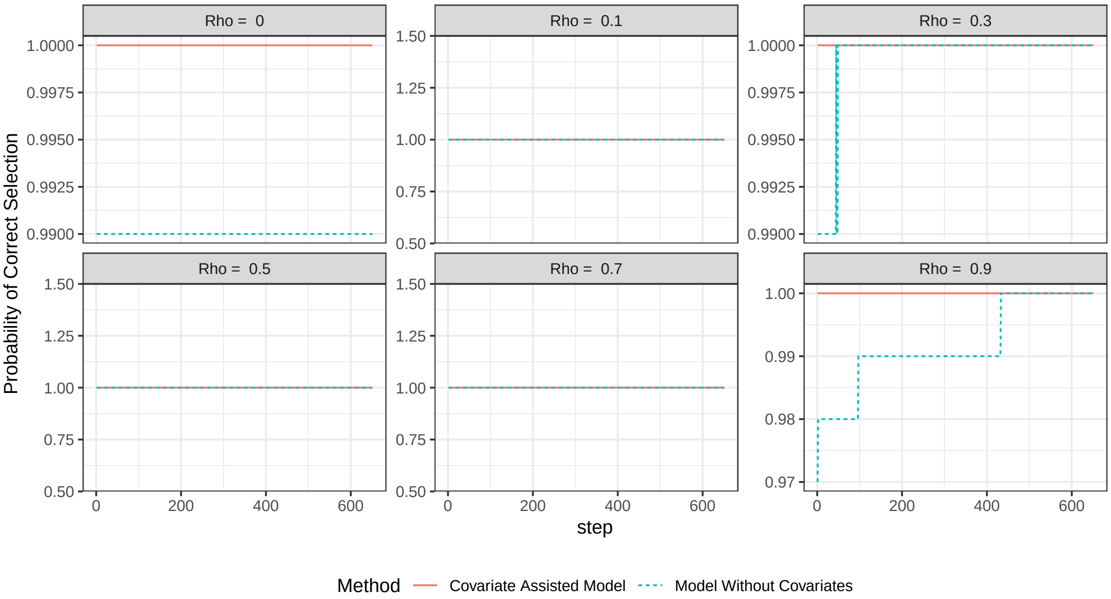

Figure 4.27: Probability of correct selection calculated at each step of the procedure, using noise-to-signal ratio of 10 over one Facebook ego network of size 747 and density 0.05

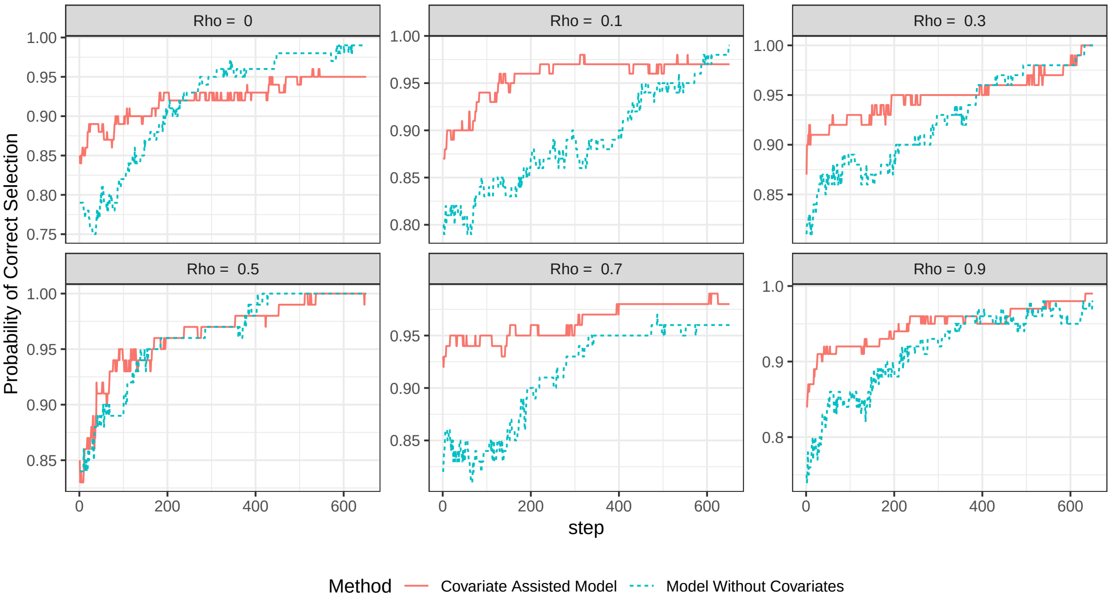

Figure 4.28: Probability of correct selection calculated at each step of the procedure, using noise-to-signal ratio of 100 over one Facebook ego network of size 747 and density 0.05

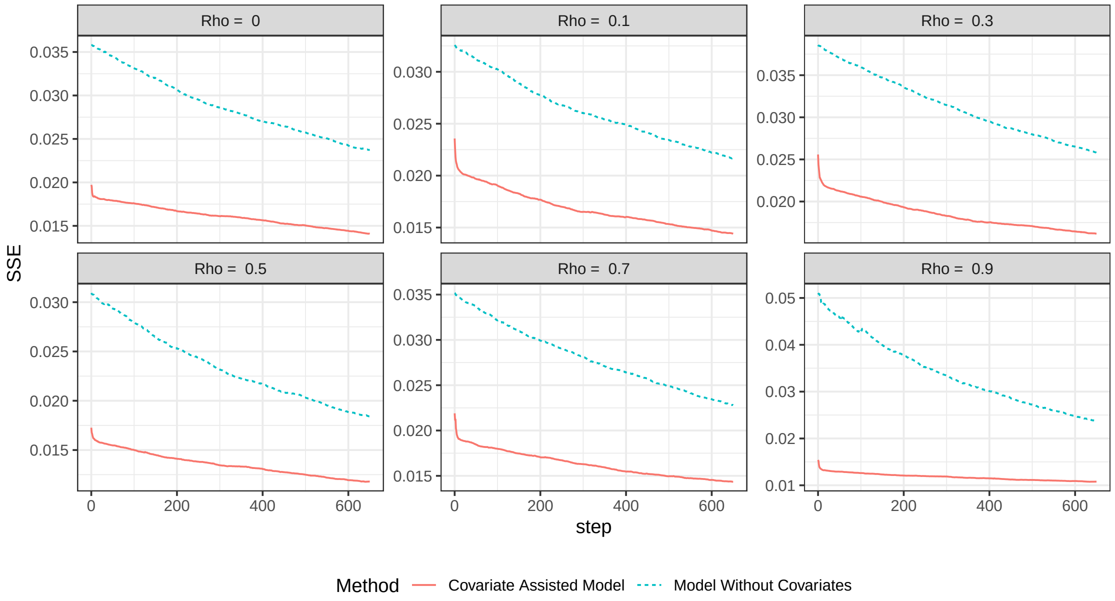

Figure 4.29: SSE calculated at each step of the procedure, using noise-to-signal ratio of 10 over one Facebook ego network of size 747 and density 0.05

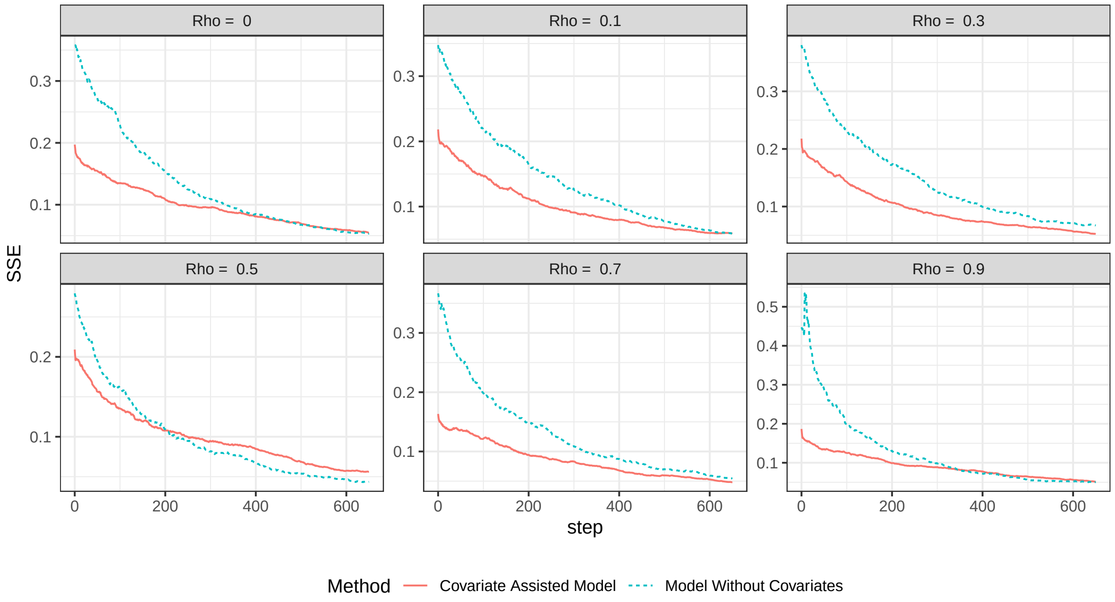

Figure 4.30: SSE calculated at each step of the procedure, using noise-to-signal ratio of 100 over one Facebook ego network of size 747 and density 0.05

Chapter 5

Conclusion

In this research we worked on creating three experimental statistical designs for A/B testing on social networks. For all three projects we assume that we can describe the relationship of network members with each other using CAR model, and that responses y to treatments, A or B, are continuous. In Chapter 2 we formulated a mixed integer program or MIP to find an exact D-optimal design for network A/B testing, that we call an Original-MIP, and used GUROBI commercial solver to compute the design. We also proposed an approximation to the design or Modified-MIP, that not only has significant savings in computational time, but also has a very comparable performance to Original-MIP. Next we tested our proposed methods on synthetic and real networks and compared their performance to random designs. What we observed is that both methods have smaller variance of the treatment effect than random when network correlation parameter ρ grows large. When ρ is small, the performance of our methods was comparable to random designs.

As network size grows large, it becomes very difficult if not impossible for the solver to bring the optimality gap to zero in reasonable time. However we observed that even the solution for the designs with high optimality gaps still resulted in smaller variances for the treatment effect during our testing. Finally, for cases of very large networks we proposed to separate the network into relatively small networks, compute the optimal design for each small network, and then combine the allocation results together. We tested this algorithm on a sample from Facebook network and observed very similar result, that as ρ grows large, the variance of the treatment effect calculated using Modified-MIP was significantly smaller than variance calculated from purely random designs. Overall the results proved to be very favorable to the proposed methods as long as the model assumptions hold. When model assumptions are violated, the use of random designs are preferred.

In Chapter 3 we continue to work with D-optimal designs. First we showed that the terms of the MIP formulation for D-optimal design follow well known statistical distributions. Based on that information we proposed an algorithm to create re-randomization design where the rejection criteria is set up based on the distribution of the terms in original MIP formulation for D-optimal designs (3.3). Then we tested the proposed method on synthetic networks and compared its performance to Original-MIP, Modified-MIP and purely random designs. We observed that on small networks re-randomization designs have the best performance in terms of the variance of treatment effect when the correlation parameter ρ is small, but as ρ increases, Original and Modified-MIP designs outperform re-randomization design. As network size increases, however, the performance of re-randomization designs become more comparable and show marginal improvement over random designs. Finally, in terms of computational time and model misspecification, the proposed re-randomization design outperforms Original and Modified-MIP designs in all considered scenarios.

In Chapter 4 we took a Bayesian optimization approach to creating sequential designs for network A/B testing. Here we were using normal conjugate priors to update our beliefs about the distribution of the treatment effect, and we included network information in the model by means of network covariates. Then we compared the Bayesian model without network covariates to the proposed model that uses network covariates on synthetic and real networks. We have done numerous testing on multiple and single networks, varying the network correlation parameter, ρ, and noise-to-signal ratio, τ/β, that we used in response generation. Our main observation was that the structure of the network was of the most importance, and regardless of the choice of ρ and τ/β, our proposed method performed better than a model without covariates only on real or clustered networks. The testing done on synthetic random networks showed that a model without covariates had smaller SSE and higher probability of correct selection, P_{cs}, for any combinations of ρ and τ/β.

Bibliography

Agrawal, S. and Goyal, N. (2012), "Analysis of thompson sampling for the multi-armed bandit problem," in *Conference on Learning Theory*, pp. 39–1.

Anscombe, F. J. (1948), "The validity of comparative experiments," *Journal of the royal statistical society. series A (General)*, 111, 181–211.

Arnold, G. (1986), "Randomization: A historic controversy," in *The fascination of Statistics*, eds. Brook, R. J., Arnold, G. C., Hassard, T. H., and Pringle, R. M., New York, NY: Marcel Dekker, chap. 18, pp. 231–244.

Aronow, P. M., Samii, C., et al. (2017), "Estimating average causal effects under general interference, with application to a social network experiment," *The Annals of Applied Statistics*, 11, 1912–1947.

Atkinson, A., Donev, A., and Tobias, R. (2007), *Optimum experimental designs, with SAS*, vol. 34, Oxford University Press.

Atkinson, A. C. and Woods, D. C. (2015), *Handbook of Design and Analysis of Experiments*, Boca Raton, FL: Chapman & Hall/CRC.

Atwood, C. L. (1969), "Optimal and Efficient Designs of Experiments," *Ann. Math. Statist.*, 40, 1570–1602.

Azimi, J., Jalali, A., and Fern, X. (2012), "Hybrid batch Bayesian optimization," *arXiv preprint arXiv:1202.5597*.

Bailey, R. (1983), "Restricted randomization," *Biometrika*, 70, 183–198.

Bailey, R. and Rowley, C. (1987), "Valid randomization," *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 410, 105–124.

Bartlett, M. S. (1951), "An inverse matrix adjustment arising in discriminant analysis," *The Annals of Mathematical Statistics*, 22, 107–111.

Basse, G. W. and Airoldi, E. M. (2015), "Optimal model-assisted design of experiments for network correlated outcomes suggests new notions of network balance," *arXiv preprint arXiv:1507.00803*.

— (2018a), "Limitations of design-based causal inference and A/B testing under arbitrary and network interference," *Sociological Methodology*, 48, 136–151.

— (2018b), "Model-assisted design of experiments in the presence of network-correlated outcomes," *Biometrika*, 105, 849–858.

Bergstra, J. S., Bardenet, R., Bengio, Y., and Kégl, B. (2011), "Algorithms for hyper-parameter optimization," in *Advances in neural information processing systems*, pp. 2546–2554.

Bertsimas, D., Johnson, M., and Kallus, N. (2015), "The power of optimization over randomization in designing experiments involving small samples," *Operations Research*, 63, 868–876.

Besag, J. (1974), "Spatial interaction and the statistical analysis of lattice systems," *Journal of the Royal Statistical Society: Series B (Methodological)*, 36, 192–225.

Bhat, N., Farias, V. F., Moallemi, C. C., and Sinha, D. (2017), "Near Optimal AB Testing," Columbia Business School.

Bivand, R., Bernat, A., Carvalho, M., Chun, Y., Dormann, C., Dray, S., Halbersma, R., Lewin-Koh, N., Ma, J., Millo, G., et al. (2005), "The spdep package," *Comprehensive R Archive Network, Version*, 05–83.

Brook, D. (1964), "On the distinction between the conditional probability and the joint probability approaches in the specification of nearest-neighbour systems," *Biometrika*, 51, 481–483.

Bruhn, M. and McKenzie, D. (2009), "In pursuit of balance: Randomization in practice in development field experiments," *American economic journal: applied economics*, 1, 200–232.

Chapelle, O. and Li, L. (2011), "An empirical evaluation of thompson sampling," in *Advances in neural information processing systems*, pp. 2249–2257.

Chen, Y., Qi, Y., Liu, Q., and Chien, P. (2018), "Sequential sampling enhanced composite likelihood approach to estimation of social intercorrelations in large-scale networks," *Quantitative Marketing and Economics*, 16, 409–440.

Cox, D. (2009), "Randomization in the design of experiments," *International Statistical Review*, 77, 415–429.

Draper, N. and Smith, H. (1966), *Applied regression analysis*, New York: Wiley.

Eckles, D., Karrer, B., and Ugander, J. (2017), "Design and analysis of experiments in networks: Reducing bias from interference," *Journal of Causal Inference*, 5.

Fedorov, V. (2010), "Optimal experimental design," *Wiley Interdisciplinary Reviews: Computational Statistics*, 2, 581–589.

Fisher, R. A. (1992), "The arrangement of field experiments," in *Breakthroughs in statistics*, eds. Kotz, S. and Johnson, N., Springer, pp. 82–91.

Goldenberg, A., Zheng, A. X., Fienberg, S. E., Airoldi, E. M., et al. (2010), "A survey of statistical network models," *Foundations and Trends® in Machine Learning*, 2, 129–233.

Gosset (1938), "Comparison between balanced and random arrangements of field plots," *Biometrika*, 363–378.

Greenberg, B. (1951), "Why randomize?" *Biometrics*, 7, 309–322.

Grundy, P. and Healy, M. (1950), "Restricted randomization and quasi-Latin squares," *Journal of the Royal Statistical Society. Series B (Methodological)*, 12, 286–291.

Gui, H., Xu, Y., Bhasin, A., and Han, J. (2015), "Network a/b testing: From sampling to estimation," in *Proceedings of the 24th International Conference on World Wide Web*, International World Wide Web Conferences Steering Committee, pp. 399–409.

Harville, D. (1975), "Experimental Randomization: Who Needs It?" *The American Statistician*, 29, 27–31.

Hoffman, M., Shahriari, B., and Freitas, N. (2014), "On correlation and budget constraints in model-based bandit optimization with application to automatic machine learning," in *Artificial Intelligence and Statistics*, pp. 365–374.

Holschuh, N. (1980), "Randomization and design: I," in *RA Fisher: An Appreciation. Lecture Notes in Statistics*, eds. Fienberg, S. E. and Hinkley, D. V., Springer, vol. 1, pp. 35–45.

Hore, S., Dewanji, A., and Chatterjee, A. (2014), "Design issues related to allocation of experimental units with known covariates into two treatment groups," *Journal of Statistical Planning and Inference*, 155, 117–126.

Hudgens, M. G. and Halloran, M. E. (2008), "Toward causal inference with interference," *Journal of the American Statistical Association*, 103, 832–842.

Hutter, F., Hoos, H. H., and Leyton-Brown, K. (2011), "Sequential model-based optimization for general algorithm configuration," in *International conference on learning and intelligent optimization*, Springer, pp. 507–523.

Imai, K., King, G., and Stuart, E. A. (2008), "Misunderstandings between experimentalists and observationalists about causal inference," *Journal of the royal statistical society: series A (statistics in society)*, 171, 481–502.

Kempthorne, O. (1955), "The randomization theory of experimental inference," *Journal of the American Statistical Association*, 50, 946–967.

Kiefer, J., Wolfowitz, J., et al. (1959), "Optimum designs in regression problems," *The Annals of Mathematical Statistics*, 30, 271–294.

Kohavi, R., Longbotham, R., Sommerfield, D., and Henne, R. M. (2009), "Controlled experiments on the web: survey and practical guide," *Data mining and knowledge discovery*, 18, 140–181.

Kolaczyk, E. D. and Csárdi, G. (2014), *Statistical analysis of network data with R*, vol. 65, Springer.

Krause, M. S. and Howard, K. I. (2003), "What random assignment does and does not do," *Journal of Clinical Psychology*, 59, 751–766.

Leskovec, J. and Mcauley, J. J. (2012), "Learning to discover social circles in ego networks," in *Advances in neural information processing systems*, pp. 539–547.

Lizotte, D. J., Wang, T., Bowling, M. H., and Schuurmans, D. (2007), "Automatic Gait Optimization with Gaussian Process Regression." in *IJCAI*, vol. 7, pp. 944–949.

Maclure, M., Nguyen, A., Carney, G., Dormuth, C., Roelants, H., Ho, K., and Schneeweiss, S. (2006), "Measuring prescribing improvements in pragmatic trials of educational tools for general practitioners," *Basic & clinical pharmacology & toxicology*, 98, 243–252.

Mahendran, N., Wang, Z., Hamze, F., and De Freitas, N. (2012), "Adaptive MCMC with Bayesian optimization," in *Artificial Intelligence and Statistics*, pp. 751–760.

Manski, C. F. (2013), "Identification of treatment response with social interactions," *The Econometrics Journal*, 16, S1–S23.

Marchant, R. and Ramos, F. (2012), "Bayesian optimisation for intelligent environmental monitoring," in *2012 IEEE/RSJ international conference on intelligent robots and systems*, IEEE, pp. 2242–2249.

Martinez-Cantin, R., de Freitas, N., Doucet, A., and Castellanos, J. A. (2007), "Active policy learning for robot planning and exploration under uncertainty." in *Robotics: Science and Systems*, vol. 3, pp. 321–328.

McPherson, M., Smith-Lovin, L., and Cook, J. M. (2001), "Birds of a feather: Homophily in social networks," *Annual review of sociology*, 27, 415–444.

Morgan, K. L. and Rubin, D. B. (2012), "Rerandomization to improve covariate balance in experiments," *The Annals of Statistics*, 40, 1263–1282.

Moulton, L. H. (2004), "Covariate-based constrained randomization of group-randomized trials," *Clinical Trials*, 1, 297–305.

Nandy, P., Basu, K., Chatterjee, S., and Tu, Y. (2019), "A/B Testing in Dense Large-Scale Networks: Design and Inference," *arXiv preprint arXiv:1901.10505*.

Ogburn, E. L., Sofrygin, O., Diaz, I., and van der Laan, M. J. (2017), "Causal inference for social network data," *arXiv preprint arXiv:1705.08527*.

Ogburn, E. L. and VanderWeele, T. J. (2017), "Vaccines, contagion, and social networks," *The Annals of Applied Statistics*, 11, 919–948.

Pokhilko, V., Zhang, Q., Kang, L., and Mays, D. (2019), "D-optimal Design for Network A/B Testing," *Journal of Statistical Theory and Practice*, 13.

Pouget-Abadie, J., Saveski, M., Saint-Jacques, G., Duan, W., Xu, Y., Ghosh, S., and Airoldi, E. M. (2017), "Testing for arbitrary interference on experimentation platforms," *arXiv preprint arXiv:1704.01190*.

Pukelsheim, F. (1993), *Optimal Design of Experiments*, vol. 50, SIAM.

Rosenbaum, P. R. (2007), “Interference between units in randomized experiments,” *Journal of the American Statistical Association*, 102, 191–200.

Rosenberger, W. F. and Lachin, J. M. (2015), *Randomization in clinical trials: theory and practice*, John Wiley & Sons.

Rosenberger, W. F., Sverdlov, O., et al. (2008), “Handling covariates in the design of clinical trials,” *Statistical Science*, 23, 404–419.

Rubin, D. B. (1974), “Estimating causal effects of treatments in randomized and nonrandomized studies,” *Journal of educational Psychology*, 66, 688.

— (2008), “Comment: The design and analysis of gold standard randomized experiments,” *Journal of the American Statistical Association*, 103, 1350–1353.

Saveski, M., Pouget-Abadie, J., Saint-Jacques, G., Duan, W., Ghosh, S., Xu, Y., and Airoldi, E. M. (2017), “Detecting network effects: Randomizing over randomized experiments,” in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, ACM, pp. 1027–1035.

Schmidt, A. M. and Nobre, W. S. (2014), “Conditional Autoregressive (CAR) Model,” *Wiley StatsRef: Statistics Reference Online*, 1–11.

Scott, S. L. (2010), “A modern Bayesian look at the multi-armed bandit,” *Applied Stochastic Models in Business and Industry*, 26, 639–658.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. (2016), “Taking the human out of the loop: A review of bayesian optimization,” *Proceedings of the IEEE*, 104, 148–175.

Shalizi, C. R. and Thomas, A. C. (2011), “Homophily and contagion are generically confounded in observational social network studies,” *Sociological methods & research*, 40, 211–239.

Sherman, J. and Morrison, W. J. (1950), "Adjustment of an inverse matrix corresponding to a change in one element of a given matrix," *The Annals of Mathematical Statistics*, 21, 124–127.

Snoek, J., Larochelle, H., and Adams, R. P. (2012), "Practical bayesian optimization of machine learning algorithms," in *Advances in neural information processing systems*, pp. 2951–2959.

Soares, J. and Wu, C. (1985), "Optimality of random allocation design for the control of accidental bias in sequential experiments," *Journal of Statistical Planning and Inference*, 11, 81–87.

Sprott, D. and Farewell, V. (1993), "Randomization in experimental science," *Statistical Papers*, 34, 89–94.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. (2009), "Gaussian process optimization in the bandit setting: No regret and experimental design," *arXiv preprint arXiv:0912.3995*.

Swersky, K., Snoek, J., and Adams, R. P. (2013), "Multi-task bayesian optimization," in *Advances in neural information processing systems*, pp. 2004–2012.

Thompson, W. R. (1933), "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, 25, 285–294.

Thornton, C., Hutter, F., Hoos, H. H., and Leyton-Brown, K. (2013), "Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 847–855.

Toulis, P. and Kao, E. (2013), "Estimation of causal peer influence effects," in *International conference on machine learning*, pp. 1489–1497.

Tukey, J. W. (1993), "Tightening the clinical trial," *Controlled clinical trials*, 14, 266–285.

Ugander, J., Karrer, B., Backstrom, L., and Kleinberg, J. (2013), "Graph cluster randomization: Network exposure to multiple universes," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 329–337.

Urbach, P. (1985), "Randomization and the design of experiments," *Philosophy of Science*, 52, 256–273.

Wall, M. M. (2004), "A close look at the spatial structure implied by the CAR and SAR models," *Journal of statistical planning and inference*, 121, 311–324.

Wang, Z., Hutter, F., Zoghi, M., Matheson, D., and de Feitas, N. (2016), "Bayesian optimization in a billion dimensions via random embeddings," *Journal of Artificial Intelligence Research*, 55, 361–387.

Wang, Z., Shakibi, B., Jin, L., and de Freitas, N. (2014), "Bayesian Multi- Scale Optimistic Optimization," in *International Conference on Artificial Intelligence and Statistics*, pp. 1005–1014.

Wolsey, L. A. and Nemhauser, G. L. (2014), *Integer and combinatorial optimization*, John Wiley & Sons.

Woods, D. (2005), "Designing experiments under random contamination with application to polynomial spline regression," *Statistica Sinica*, 15, 619.

Worrall, J. (2010), "Evidence: philosophy of science meets medicine," *Journal of evaluation in clinical practice*, 16, 356–362.

Wu, C. J. and Hamada, M. S. (2011), *Experiments: planning, analysis, and optimization*, vol. 552, Hoboken, New Jersey: John Wiley & Sons.

Wu, J. (2017), "Knowledge Gradient Methods for Bayesian Optimization," Ph.D. thesis, copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2019-10-18.

Xia, Y., Liu, C., Li, Y., and Liu, N. (2017), "A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring," *Expert Systems with Applications*, 78, 225–241.

Xu, Y., Chen, N., Fernandez, A., Sinno, O., and Bhasin, A. (2015), "From infrastructure to culture: A/b testing challenges in large scale social networks," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp. 2227–2236.

Yang, M., Biedermann, S., and Tang, E. (2013), "On Optimal Designs for Nonlinear Models: A General and Efficient Algorithm," *Journal of the American Statistical Association*, 108, 1411–1420.

Yates, F. (1939), "The comparative advantages of systematic and randomized arrangements in the design of agri-cultural and biological experiments," *Biometrika*, 30, 440–466.

— (1964), "Sir Ronald Fisher and the design of experiments," *Biometrics*, 20, 307–321.

Vita

Victoria V. Pokhilko was born on July 16, 1980, in Stavropol, Russia. She graduated from Stavropol State University, in 2002, and immigrated to the United States of America shortly after her graduation. She received her Master of Business Administration from Virginia Commonwealth University in Richmond, Virginia, in 2010 while working at Hamilton Beach Brands. In 2015 she graduated with M.S. in Mathematical Sciences / Statistics from Virginia Commonwealth University. She was a Graduate Teaching Assistant, taught classes at VCU and American University and had internships at Travelers and Averhealth while working on her Ph.D. at Virginia Commonwealth University.