On estimation of causal effects under the K-nearest-neighbor interference model with misspecified interference

by

Ruhania Tarannum

M.S., Kansas State University, 2019

A REPORT

submitted in partial fulfillment of the
requirements for the degree

Master of Science

Department of Statistics
College of Arts & Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2024

Approved by:

Major Professor
Dr. Michael Higgins

Copyright

Abstract

Interference in an experiment occurs when the response of a unit is not only affected by the treatment status given to that unit, but is also impacted by the interventions of other units around it. Interference in experiments is becoming increasingly common, especially in experiments performed within social networks. However, traditional causal models often assume no interference across experimental units. Estimating causal effects without considering interference when it is present may lead to biased estimation of causal quantities and may lead to incorrect conclusions. However, often it is impossible to know the interference structure prior estimating causal effects, and considerable work has been performed to improve estimation under unknown or misspecified interference structure.

In this report, we investigate misspecification of treatment interference under the K-Nearest Neighbor Interference Model (KNNIM). The KNNIM model assumes that the response of an unit is affected by its own treatment status and treatments applied to the unit's K "closest" units, where closeness may be a measure of the interaction between units. Of note, KNNIM allows for larger effects of treatment interference between units with stronger interactions. We perform a simulation study under KNNIM in which we inject uncertainty in the interaction measure to create misspecification in the interference structure. In our simulations, we assume that the magnitudes of interference effects are larger for closer units. We then perform Horvitz-Thompson estimation of direct, indirect, and nearest-neighbor causal effects, and we investigate how misspecification can impact estimates. We find, for our considered models, that the misspecification injects bias in estimates of the 1st nearest-neighbor effects, but have little impact on other causal effects. The bias of the 1st-nearest-neighbor effect grows with increasing sample sizes.

Table of Contents

List of Figures

List of Tables

Acknowledgments

First, I would like to acknowledge my indebtedness to my major professor Dr. Michael Higgins. I truly consider myself very fortunate to get the chance to do my master's research work under his supervision. I extend my profound sense of gratitude to Dr. Higgins for his immense support and guidance throughout my journey as a graduate student at K-state. This research work would not be possible without his expert advice and instructive comments. It has been a great experience working with him. I would like to express my sincere gratitude to Dr. Chris Vahl for advising me and guiding me always with his thoughtful suggestions during my tenure as a graduate student in the Department of Statistics. I acknowledge Dr.Shengwu Shang and Dr. Jieun Lee for their support and inspiration. I found both very compassionate while working as a GTA under their supervision. As a single parent of a toddler girl in a foreign country, it was not an easy task for me to pursue a Master's in statistics. I am grateful to all my faculties at the Department of Statistics who were considerate and always extended their helping hands to ease my journey at graduate school.

I am grateful to my family members and friends who supported me in every possible way during my journey at K-state in the last three years. In fall 2021, I came to USA with my nine-month-old daughter and started my graduate program at K-state. Raising a baby while studying at graduate school was a strenuous task. It was my mother, Hafiz Jahan, who accompanied me during that challenging time of my life. I could never thank her enough for what she did for me even at the autumn of her life. She continued to inspire me and supported me immensely by looking after my little daughter as long as she could. Her loving care enriched my daughter's upbringing and relieved me in my hectic days at graduate school.

I should acknowledge my elder sisters, Fahmida Nusrat, and Jannatul Ferdous, for their motivational words and their own fighting spirits towards odds that encouraged me always

to keep going and work on my incomplete tasks. I would love to express my special thanks to my husband, B M Samin Monowar Tonmoy, for his endless support and inspiration. Pursuing a Master's in statistics after becoming a mother would not be possible for me without his inspiration and continuous support. It was never easy for him and me as well, staying thousands of miles apart just after becoming parents of a beautiful baby, for my higher study abroad. I am truly grateful to my husband for understanding me and accepting the emotional struggle and sacrifice with a loving heart. My heartfelt thanks to my in-laws who have always been my well-wishers and inspired me a lot to pursue higher education abroad.

Doing Master's research as a graduate student in the Department of Statistics at Kansas State University has been a unique experience of my academic life. I learned a lot in every step I took in my graduate research work. That's why, above all, I am grateful to the almighty for giving me this opportunity to pursue my higher education in a reputed university of USA and making me capable of completing my master's successfully.

Chapter 1

Introduction

Causal inference is a growing branch of statistics aimed at investigating the effect of "treatments" on responses—for example, the effect of receiving a mailer on voting in the upcoming election or, in case of social networks, the effect of content placement on website engagement. Randomized experiments have been considered as the "gold standard" for causal inference (Imbens and Rubin, 2015). For example, online randomized controlled experiments are performed by big tech companies to study the effect of new feature or a new launched product on user engagement (Gui et al., 2015; Pouget-Abadie et al., 2019; Saveski et al., 2017). Additionally, researchers in epidemiology may study the effect of vaccination on a target population who are at risk of an infectious disease (Hudgens and Halloran, 2008).

Interference persists within an experiment when the treatment assigned to one unit affects the response of another unit. In traditional causal models, it is often assumed that there is no across experimental units. However, interference is very common in practice, especially in studies related to social networks. For example, on a social media platform, a user assigned to a new feature may interact to another user who is not assigned to it but their interactions may impact the latter user's engagement with the platform. Failure to account for interference in causal inference may lead to biased estimates of treatment effects and inaccurate conclusions (Sobel, 2006). With the increased popularity of these types of experiments, there has been considerable effort devoted to develop methodology for estimat-

ing causal effects under interference. For example, Hudgens and Halloran (2008) proposed estimands for direct, indirect, total and overall causal effects in the setting of Interference using potential outcomes framework. Tchetgen and VanderWeele (2012) extended the work of Hudgens and Halloran (2008) and developed a finite sample framework for causal inference under interference. Toulis and Kao (2013) also expanded potential outcomes framework to allow for interference in social networks by defining k-level estimand of interference effect. Under arbitrary form of interference Aronow and Samii (2017) estimated treatment effects using Horvitz-Thompson estimators. Aronow et al. (2021) reviewed methods developed for both interference in general network settings and partial interference in randomized experiments context. Karrer et al. (2021) introduced a framework accounting for interference by setting up cluster-randomized experiments. Sävje et al. (2021) investigated methods for consistent estimation of treatment effects under unknown interference structure.

However, causal estimands become complex to define and formulate in causal analysis under interference. Additionally, methods for causal inference under interference often require correctly specifying which units can interfere with each other. Without any assumptions on interference structure, estimation of causal quantities of interest could be impossible.

Controlled interference structure can be added into experimental design by defining an exposure mapping on the experimental units under study. An exposure mapping can be viewed a network where nodes of the network represent units and edges between vertices indicate the interactions between corresponding units. There may additionally be an interaction measure that weighs how strong or weak the interaction between two units.

A recently-developed causal model for specifying interference effects is the K-nearest neighbors interference model (KNNIM) (Alzubaidi and Higgins, 2024). This model assumes that the response of any experimental unit depends only on its treatment status and the treatment statuses of its K "closest" neighbors, where closeness is measured through some kind of interaction measure. Thus, the exposure mapping under KNNIM draws directed edges to a unit from the unit's K closest neighbors. In particular, KNNIM allows larger effects of treatment interference between units with stronger connections.

In many settings (including KNNIM), however, it is impossible to obtain a correct ex-

posure mapping. As a result, considerable work has been performed on the estimation of causal effects under misspecified or unknown network structure (Leung, 2022; Sävje, 2024). However, no work has been explicitly performed on the performance of KNNIM under misspecification.

In this report, we investigate the performance of causal estimators under KNNIM with a misspeficied exposure mapping. We perform a simulation study where random error is injected into the "closeness measure" used to define nearest neighbors. We then generate data satisfying KNNIM under the correct exposure mapping, and estimate causal effects assuming the erroneous exposure mapping. In particular, we assess both how much this type of error can impact the misspecification the exposure mapping and the consistency of estimates of treatment effects due to interference.

1.1 An overview of Causal Inference

Causal Inference refers to the process of inferring what would happen in future if the present actions are changed or inferring what would have happened if things had been done differently in the distant past (Rubin, 2019). A common aim of causal inference is to infer the causal effect of an intervention on units and how the intervention affects response variables.

1.1.1 Neyman-Rubin Causal Model (NRCM)

One of the most popular models of response in causal inference is Neyman-Rubin causal model (NRCM) (Holland, 1986; Imbens and Rubin, 2015; Neyman, 1923; Rubin, 1980). NRCM is developed based on the concept of *potential outcomes* which are hypothetical responses of units under each treatment condition. A *unit* can be a single physical object or person (i.e. a farm, a student etc.) or a collection of objects or individuals (i.e. classroom or market). The outcomes are referred as *potential* as only one outcome is ultimately realized—the one corresponding to the given treatment condition—and other potential outcomes cannot be not observed.

Suppose we have an experiment on N units, numbered $i = 1, 2, 3, \ldots, N$, where all units are given either a treatment or a control condition. Let W_i denote a treatment indicator for unit i, where $W_i = 1$ if unit i is given the treatment condition and $W_i = 0$ if it is given the control condition. The vector $\mathbf{W} = (W_1, W_2, W_3, ..., W_N) \in \{0, 1\}^N$ denotes the treatment assignment vector of all N units. The NRCM assumes that the response, Y_i of a unit i given either a treatment or a control follows the model

$$Y_i = y_i(1)W_i + y_i(0)(1 - W_i) \tag{1.1}$$

where $y_i(1)$ and $y_i(0)$ denote the potential outcomes of unit i under the treatment and control condition respectively.

Inherent in NRCM is the *stable unit treatment value assumption* (SUTVA) (Rubin, 1980). This assumption states that a unit's response is only affected by a single version of each treatment status and it is not affected by the treatment status of any other unit. Note that SUTVA is violated when treatment interference is present.

Under NRCM, the unit-level causal effect of treatment for a unit i is defined as the difference between the two potential outcomes of that unit:

$$\delta_i = y_i(1) - y_i(0). \tag{1.2}$$

Note that δ_i is not observable for any unit i as that would require knowing both potential outcomes for that unit. Thus, instead of estimating causal effect for one unit, the average treatment effect (ATE) is often considered instead (Imbens and Rubin, 2015): In a setting of independent treatment assignment $\boldsymbol{W}$ and finite population N where potential outcomes $y_i(1)$ and $y_i(0)$ are fixed quantities, nonrandom, observed outcomes depends on treatment assignment $\boldsymbol{W}$ and average treatment effect becomes:

$$\delta = \frac{1}{N}\sum_{i=1}^{N}\delta_i = \frac{1}{N}\sum_{i=1}^{N}\{y_i(1) - y_i(0)\} = \sum_{i=1}^{N} y_i(1) - \sum_{i=1}^{N} y_i(0). \tag{1.3}$$

While unit-level treatment effects are inestimable, the ATE can be decomposed into a difference of two population means, and hence, can be estimated (for example, by using the sample mean in a completely randomized experiment).

1.2 Causal Inference Under Interference

Treatment interference occurs when treatment status of one unit affects the response of another unit. Though interference is very common in studies on social networking, introduced in experiments as part of social interaction and influence, many researchers have been considering interference as a nuisance for accurate estimation of average treatment effects. A considerable amount of work has been performed on designing experiments dedicated to reduce the effect of treatment interference. For example, when the population of units can be partitioned into clusters and interference is assumed only to persist within clusters of units (e.g. neighborhoods, communities, etc.), experiments may assign treatment to these clusters, rather than individual units, to mitigate the effect of interference on the estimation of treatment effects (Eckles et al., 2017; Gui et al., 2015; Ugander et al., 2013). This interference setting is referred to as *partial interference* (Hudgens and Halloran, 2008; Rosenbaum, 2007; Sobel, 2006). Karrer et al. (2021) presented a framework accounting for interference through cluster-randomization experiments. Behavior of standard estimators for average treatment effects under weak interference was investigated by Sävje et al. (2021).

More recently, estimands of causal effects under interference have become of primary interest. For example, Sobel (2006) and Hudgens and Halloran (2008) proposed causal estimands under a partial interference assumption. Some work also has considered neighborhood interference where interference is not allowed to persist outside of a small neighborhood of a unit (Leung, 2022; Sussman and Airoldi, 2017). After specifying the nature of treatment interference, estimators can then be defined to estimate both *direct effects*—differences in a unit's response due to the treatment condition assigned to that unit—and *indirect effects*—differences in a unit's response due to the treatment conditions assigned to interfering units (Aronow and Samii, 2017).

1.3 Exposure mapping

It is impossible to estimate causal estimands of interest without making some assumption on interference structure. For example, under the unconstrained interference setting—where any unit can interfere with any other unit, each unit will have 2^N potential outcomes. That is, there will be too many unknown parameters in such a model to reliably estimate causal quantities, if such estimation is even possible (Alzubaidi, 2022; Aronow and Samii, 2017; Sussman and Airoldi, 2017; Toulis and Kao, 2013; Ugander et al., 2013). This is why researchers often make assumptions on how interference propagates across units.

The NRCM model of response given in (1.1) can be naturally extended to settings exhibiting interference. Recall that $\boldsymbol{W} = \{W_1, W_2, W_3, ..., W_N\}$ denotes a treatment allocation to all N units, and let Ω denote the set of all possible treatment allocations. Additionally, let $y_i(\boldsymbol{W})$ denote the potential outcome of unit i under the treatment allocation $\boldsymbol{W}$. The response Y_i of unit i can then be modeled as

$$Y_i = \sum_{\boldsymbol{W} \in \Omega} y_i(\omega) \mathbf{1}(\boldsymbol{W} = \omega), \tag{1.4}$$

where $\mathbf{1}(\boldsymbol{W} = \omega)$ is indicator variable which is equal to 1 if and only if when the observed treatment allocation is ω. This means that the response of unit i is only depending on the potential outcome of unit i and observed treatment allocation across the remaining $N - 1$ units.

Restricted interference can be imposed in an experimental design by constructing an exposure mapping. Aronow and Samii (2017) defines exposure mapping as a function that maps an assignment vector and unit i specific traits to an exposure value where there is a finite set of exposure values. Intuitively, an exposure mapping denotes which treatment allocations will yield the same potential outcome for a unit i (Sävje, 2024). Causal effects can then be defined as the difference between potential outcomes produced by exposures in a properly specified exposure mapping. For an example of an exposure mapping, see Section 1.4.

1.3.1 Misspecification in Exposure Mapping

Often, it is impossible to know precisely which units interfere with each other, and misspecification of the exposure mapping is common (Weinstein and Nevo, 2023). As mentioned previously, there has been substantial work to effectively estimate causal estimands under misspecification. For example, Aronow and Samii (2017), Sävje et al. (2021),Sävje (2024), and Leung (2022) have all considered causal inference under unknown or misspecified exposure mappings.

1.4 K-Nearest Neighbors Interference Model

The K-Nearest Neighbors Interference Model (KNNIM) is a recently developed model of treatment interference that proposes the response of a unit is affected only by the treatment given to the unit and treatment statuses of its K Nearest Neighbors (Alzubaidi, 2022; Alzubaidi and Higgins, 2023, 2024). Or, in other words, the treatment status of a unit j can affect the response of a unit i only if j is one of i's K-nearest neighboring units.

In KNNIM, units under study may be viewed as a mathematical directed graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ where each vertex $i \epsilon V$ represents a unit under study and an edge $\vec{ij} \in \boldsymbol{E}$ denotes the interaction or interference from unit i to unit j—that is, unit i's treatment status may affect the response of unit j. These interactions between units can be any type of relationships, for example, friendship in social media, membership to the same group, geographic proximity. etc. (Forastiere et al., 2021). These interactions may not be communicative which means if treatment status of unit i affects on unit j's response, it is not necessary that unit j's treatment status may also have an effect on the response of unit i. For the ease of exposition, it is assumed in this model that the degree of each vertex i is at least K.

The $N \times N$ adjacency matrix of G is denoted by $\boldsymbol{A}$. That is $A_{ij} = 1$ if $\vec{ij} \in \boldsymbol{E}$ and $A_{ij} = 0$ otherwise. The diagonal elements of adjacency matrix $A_{ii} = 0$ for all i as there is no self-loops in G.

The interaction measure $d(i, j)$ denotes the strength of the interaction between units i

and j. It is assumed that $d(i,j)$ is only computed for pairs of units with $A_{ij} = 1$. In this report, we make the decision to denote stronger interactions between units with smaller values of $d(i,j)$. The j^{th} smallest value of $\{d(i,j), j \neq i\}$ is denoted by $d(i,(j))$ so that $d(i,(1)) \leq d(i,(2)) \leq \ldots$. For ease of exposition, it is assumed that all values $d(i,j)$ are unique (in practice, ties may be broken at random).

The K- *neighborhood* of unit i denoted by N_{ik} is the set of the K "closest" units around unit i:

$$N_{ik} = \{j : d(i,j) \leq d(i,(K)), j = 1, 2, ..., K, j \neq i\}. \tag{1.5}$$

The quantity $N_{-ik} = \boldsymbol{V} \backslash (i \cup N_{ik})$ is defined as all the units of $\boldsymbol{V}$ that do not belong to the closed K-neighborhood of unit i. Note that the sets $\{i, N_{ik}, N_{-ik}\}$ form a partition of $\boldsymbol{V}$. For a given treatment allocation $\boldsymbol{W} = \{W_1, W_2, W_3, ..., W_N\}$, let $\boldsymbol{W_{N_{ik}}}$ denote the treatment statuses within the K-neighborhood of i and let $\boldsymbol{W_{N_{-ik}}}$ denote the treatment statuses of all units outside of i's closed K-neighborhood. Hence, any treatment allocation $\boldsymbol{W}$ can be decomposed into $\mathbf{W} = (W_i, \boldsymbol{W_{N_{ik}}}, \boldsymbol{W_{N_{-ik}}})$.

We now state the defining property of KNNIM:

K-Neighborhood Interference assumption (K-NIA): For all units i, the potential outcomes $y_i(\boldsymbol{W})$ satisfy

$$y_i(W_i, \boldsymbol{W_{N_{ik}}}, \boldsymbol{W_{N_{-ik}}}) = y_i(W_i, \boldsymbol{W_{N_{ik}}}, \boldsymbol{W'_{N_{-ik}}}). \tag{1.6}$$

K-NIA states that the potential outcome of unit i is only affected by the treatment status of the unit and the treatments applied to its K-nearest neighbors. The treatment statuses of the units outside of K-neighborhood have no impact on the potential outcome of the unit i. This assumption limits the number of potential outcomes to be 2^{K+1} for each unit. For ease of exposition, for treatment allocations under KNNIM, we may refer to particular potential outcomes of unit i by the treatment statuses given to i and its K-neighborhood, that is, we refer to the potential outcome $y_i(W_i, \boldsymbol{W_{N_{ik}}}, \boldsymbol{W_{N_{-ik}}})$ as $y_i(W_i, \boldsymbol{W_{N_{ik}}})$.

In KNNIM, the choice of K is ultimately left to the researchers. But it is also noted that larger values of K might not be sufficiently restrictive to allow for reliable causal estimates and inferences. Alzubaidi and Higgins (2024) described in detail how to choose K. It is found, as a useful rule of thumb, that the number of K should be chosen such that each exposure used in the estimation of treatment effects is observed at least 30 times.

In this research work, misspecification of KNNIM under interference is investigated through a simulation study. Horvitz-Thompson (HT) estimates are defined estimators for different causal quantities . HT estimates of direct, indirect and K-nearest neighbor indirect effect are also investigated in this research for varying sample sizes and different variances to obtain insights of causal estimates under interference model.

Chapter 2

Study on Misspecification & Causal Estimates Under KNNIM Model

2.1 Causal estimands Under KNNIM

Causal estimands are defined under KNNIM based on the potential outcomes framework and following Hudgens and Halloran (2008) and Alzubaidi and Higgins (2023). In the following subsection, different causal estimands under KNNIM are described briefly.

2.1.1 Direct, Indirect & Total Effects

The average direct effect (ADE), δ_{dtr} is defined as the average difference in a unit's potential outcomes when changing the unit's treatment status from control to treatment and holding all other units treatment statuses fixed. For this report, it is defined as

$$\delta_{dtr} = \frac{1}{N}\sum_{i=1}^{N}(y_i(1,\mathbf{1}) - y_i(0,\mathbf{1}), \tag{2.1}$$

Here, $\mathbf{1}$ denotes a vector of 1's of length K.

The average indirect effect (AIE), δ_{ind} is measured as the average difference in a unit's potential outcomes when holding the unit's treatment status fixed and altering the treatment

statuses of rest of the units. For this report, it is defined as

$$\delta_{ind} = \frac{1}{N}\sum_{i=1}^{N}(y_i(0,\mathbf{1}) - y_i(0,\mathbf{0}), \tag{2.2}$$

Here, $\mathbf{0}$ denotes a vector of 0's of length K.

The average total effect (ATOT) δ_{tot} is defined as the average difference in potential outcomes when all units receiving treatment and all units receiving control.

$$\delta_{tot} = \frac{1}{N}\sum_{i=1}^{N}(y_i(1,\mathbf{1}) - y_i(0,\mathbf{0}), \tag{2.3}$$

Note that these quantities are defined to satisfy $\delta_{tot} = \delta_{dtr} + \delta_{ind}$. Additionally, when SUTVA holds, $\delta_{tot} = \delta_{dtr}$ and $\delta_{ind} = 0$. Additionally, these are not the only way that these quantities may be defined. For example, δ_{dir} may be defined as $\delta_{dir} = N^{-1}\sum_{i=1}^{N}(y_i(1,\mathbf{0}) - y_i(0,\mathbf{0}))$ and δ_{ind} as $\delta_{ind} = N^{-1}\sum_{i=1}^{N}(y_i(1,\mathbf{1}) - y_i(1,\mathbf{0}))$ while still ensuring the total effect decomposition into direct and indirect effects. Moreover, direct effects may be defined for each possible K neighborhood exposure $\mathbf{W}_{-i}$—for example $\tau_{dir}(\mathbf{W}_{-i}) = N^{-1}\sum_{i=1}^{N}(y_i(1,\mathbf{W}_{-i}) - y_i(0,\mathbf{W}_{-i}))$. However, such definitions may prevent a decomposition of the total effect into direct and indirect effects (Hudgens and Halloran, 2008).

2.1.2 The ℓ^{th} Nearest Neighbor Indirect Effect

Under KNNIM, nearest neighbor indirect effects are of primary interest. Let $\boldsymbol{W}^*_\ell = (W^*_{\ell,1}, W^*_{\ell,2}, ...W^*_{\ell,K}) \in \{0,1\}^K$ denote the treatment assingment where the first ℓ nearest neighbors are given treatment and rest are given control:

$$W^*_{\ell,j} = \begin{cases} 1, & j \leq \ell \\ 0, & \text{Otherwise} \end{cases} \tag{2.4}$$

Note that $\boldsymbol{W}^*_K = \mathbf{1}$, and define $\boldsymbol{W}^*_0 = \mathbf{0}$. The *average ℓ^{th} nearest neighbor indirect effect*

(AℓNNE) is then defined as follows:

$$\delta_\ell = \frac{1}{N}\sum_{i=1}^{N}(y_i(0, \boldsymbol{W}^*_\ell)) - y_i(0, \boldsymbol{W}^*_{\ell-1})) \tag{2.5}$$

Note that $\boldsymbol{W}^*_\ell$ and $\boldsymbol{W}^*_{\ell-1}$ are identical except $W^*_{\ell,\ell} = 1$ and $W^*_{\ell-1,\ell} = 0$. Thus δ_ℓ can be interpreted as the average difference in response to the treatment status of ℓ^{th}- nearest neighbor. Also, under KNNIM, the AIE is the sum of all AℓNNE.

2.2 Horvitz-Thompson Estimators

Alzubaidi and Higgins (2023) defined Horvitz-Thompson estimator for the estimands described in previous section by closely following the approach in (Aronow and Samii, 2017).

Let $\bar{y}(W, W_{N_K})$ denote the average potential outcome across all N units for treatment allocation $(W, W_{N_K}$:

$$\bar{y}(W, W_{N_K}) = \frac{1}{N}\sum_{i=1}^{N} y_i(W, W_{N_K}). \tag{2.6}$$

Let $\pi_i(W, W_{N_K})$ denote the marginal probability that unit i is given exposure (W, W_{N_K}). Define indicator variables $I_i(W, W_{N_K})$ that are equal to 1 if unit i is given exposure $(W, W_{N_K}$ and are 0 otherwise.

The Horvitz-Thompson (HT) estimator (Horvitz and Thompson, 1952) for $\bar{y}(W, W_{N_K})$ is

$$\bar{Y}^{obs}_{HT}(W, W_{N_K}) = \frac{1}{N}\sum_{i=1}^{N} I_i(W, W_{N_K})\frac{Y_i}{\pi_i(W, W_{N_K})}. \tag{2.7}$$

This estimator is unbiased for $\bar{y}(W, W_{N_K})$,

$$E(\bar{Y}^{obs}_{HT}(W, W_{N_K})) = \bar{y}(W, W_{N_K}) \tag{2.8}$$

Using the HT estimator the unbiased estimators for the ADE, AIE, ATOT and the AℓNNIE are obtained as follows:

$$\hat{\delta}_{HT,dir} = \bar{Y}^{obs}_{HT}(1, \mathbf{1}) - \bar{Y}^{obs}_{HT}(1, \mathbf{1}), \tag{2.9}$$

$$\hat{\delta}_{HT,ind} = \bar{Y}^{obs}_{HT}(0, \mathbf{1}) - \bar{Y}^{obs}_{HT}(0, \mathbf{0}), \tag{2.10}$$

$$\hat{\delta}_{HT,tot} = \bar{Y}^{obs}_{HT}(1, \mathbf{1}) - \bar{Y}^{obs}_{HT}(0, \mathbf{0}), \tag{2.11}$$

$$\hat{\delta}_{HT,\ell} = \bar{Y}^{obs}_{HT}(0, \boldsymbol{W}^*_{\ell}) - \bar{Y}^{obs}_{HT}(1, \boldsymbol{W}^*_{\ell-1}). \tag{2.12}$$

2.3 Misspecification of KNNIM

Misspecification of KNNIM occurs when nearest neighbors of a unit i are not identified correctly. Specifically, consider the true dissimilarity measure $d(i, j)$ and an estimated dissimilarity measure $d^*(i, j)$. For a unit i, suppose that $d(i, j_{i1}) \leq d(i, j_{i2}) \leq \ldots \leq d(i, j_{iK})$ and $d^*(i, j^*_{i1}) \leq d^*(i, j^*_{i2}) \leq \ldots \leq d^*(i, j^*_{iK})$. Misspecification occurs if, for some unit i and for some $\ell \in \{1, \ldots, K\}$, we have $j_{i\ell} \neq j^*_{i\ell}$. This misspecification could take several forms. It could be the case that the misspecification assumes that i is one of j's K-nearest neighbors, just a closer or further away neighbor. It could also be the case that the misspecification incorrectly identifies j as one of i's K-nearest-neighbors, and thereby, fails to correctly identify one of i's nearest neighbors.

2.4 Simulation

2.4.1 Data Generation

We now perform a simulation study to generate data under a misspecified KNNIM model At the beginning of the simulation study, responses were generated under the following model which satisfies KNNIM:

$$Y_i = X_{i1} + X_{i2} + X_{i3} + \beta_1 W_{i1} + \beta_2 W_{i2} + \beta_3 W_{i3} + \beta_d W_i \tag{2.13}$$

In this model, $X_j, j = 1, 2, 3$, are the covariates which are independent and identically distributed $N(0, 1)$ random variables. The $W_{i\ell}$ terms are treatment indicators for the ℓ^{th} nearest neighbor of unit i. The coefficients $\beta_1, \beta_2, \beta_3$ denote the 1st, 2nd, and 3rd AℓNNIE. The coefficient β_d denotes the ADE. Thus, this model assumes that the response of unit i, Y_i, is affected by the treatment given to unit i and the treatment statuses of the three closest neighbors ($i.e. K = 3$).

We assume that covariates of similar values are more likely to interact with each other. In this simulation design, we assume the dissimilarity measure $d(i, j)$ to be the Euclidean distance between the vectors of covariates $\mathbf{X}_i$ and $\mathbf{X}_j$. Treatments were considered completely randomized across all N units during the data generation process with half of the units receiving treatments and the other half receiving controls.

We ensure misspecification of the dissimilarity measure by injecting random error into the dissimilarity measure that identifies the nearest neighboring units. That is, we assume the observed dissimilarity measure $d^*(i, j) = d(i, j) + \epsilon_{ij}$ where ϵ_{ij} are independent and identically distributed $N(0, \sigma)$ random variables. In this simulation study, response data were generated under models with varying values of coefficients, $\beta = (\beta_1, \beta_2, \beta_3, \beta_d)$, N and σ. HT estimates of these causal effects are computed assuming the misspecified dissimilarity measure $d^*(i, j)$.

Sample sizes $N = 100, 200, 300, 400, 500, 1000, 2000$ and standard deviations $\sigma = (0.02, 0.05, 0.1)$ were considered for data generation. For each choice of sample size and standard deviation, four different interference models are considered. These models are described in terms of $\boldsymbol{\beta}$ coefficients in Table 2.1. In all models considered, the closer the relationship or the nearer to unit i, the greater the indirect effect: $|\beta_1| \geq |\beta_2| \geq |\beta_3|$. We generate 100 realizations of potential for each model for all considered cases.

2.5 Study on misspecifications in KNNIM

First, we investigate the misspecification rate as the standard deviation σ varies across models. We consider $\sigma \in (0.02, 0.05, 0.1)$. Referring to our notation in Section 2.3, an

instance of misspecification occurs when $j_{i\ell} \neq j^*_{i\ell}$ for some i and for some $\ell \in \{1, \ldots, K\}$. The *misspecification rate* is the fraction of occurrences for which $j^*_{i\ell} \neq j_{i\ell}$.

Since the misspecification rate is not impacted by the responses Y_i, we use Model 1 ($w = 1, \boldsymbol{\beta} = (0, 0, 0, 0)$ to generate responses with a sample size of $N = 500$.

Figure 2.1 shows that the misspecification rate increases significantly with increasing standard deviation. Higher misspecification rates with increased standard deviation of injected error are anticipated as the increased spread of random error results in more randomness in nearest neighbor measure causing incorrect specification of neighboring units. Misspecification rates for KNNIM model over different sample sizes were also investigated. In figure 2.2, the distribution of misspecification rates is presented over varying sample sizes in two different standard deviation conditions ($\sigma = 0.02$ and $\sigma = 0.05$). The boxplots of misspecification rates in Figures 2.2a and 2.2b also suggest that larger sample size might result greater misspecification of nearest neighbors. In both cases with $\sigma = 0.02$ and $\sigma = 0.05$, boxplots presenting the misspecification rates distribution for sample sizes 400 and 500 show higher median values compared to the boxplot medians for sample sizes 100 and 200.

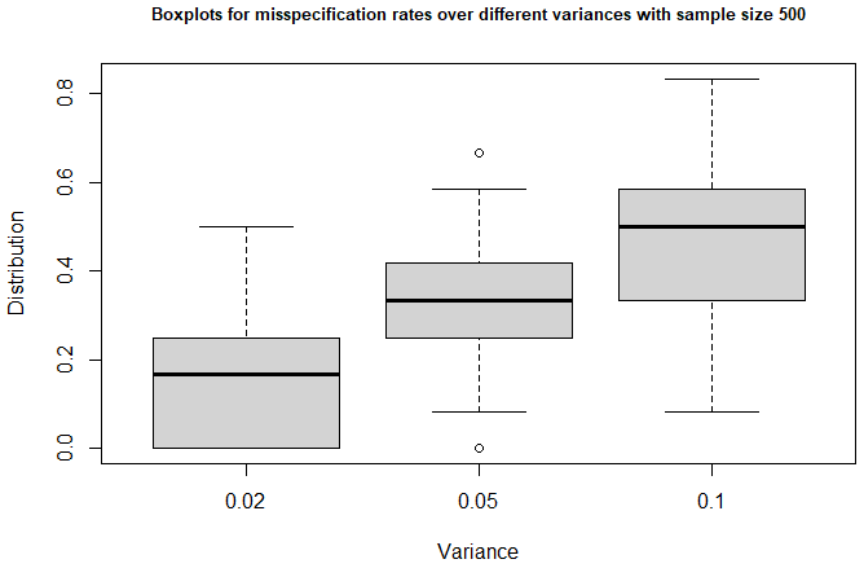
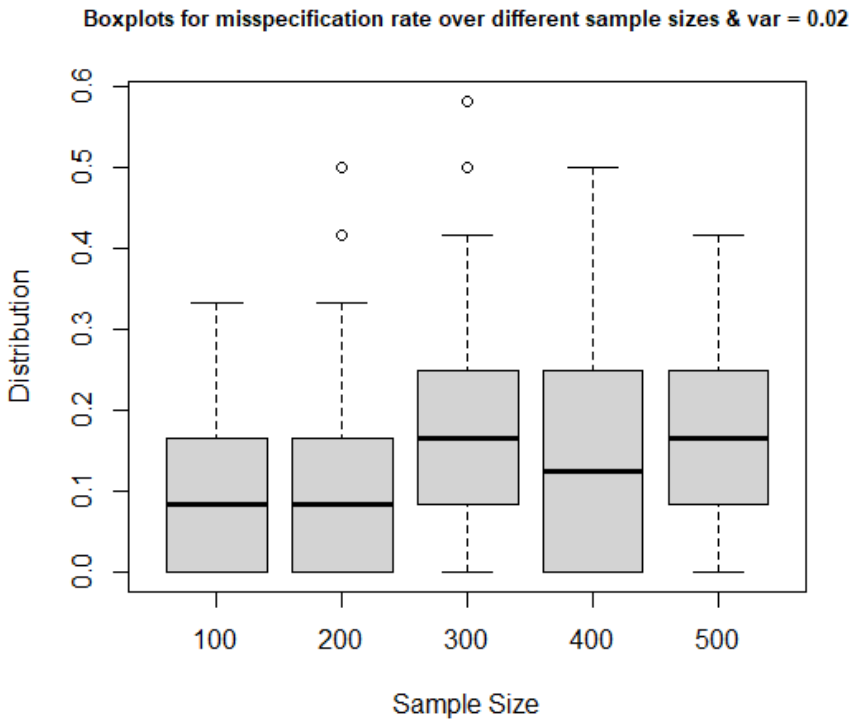
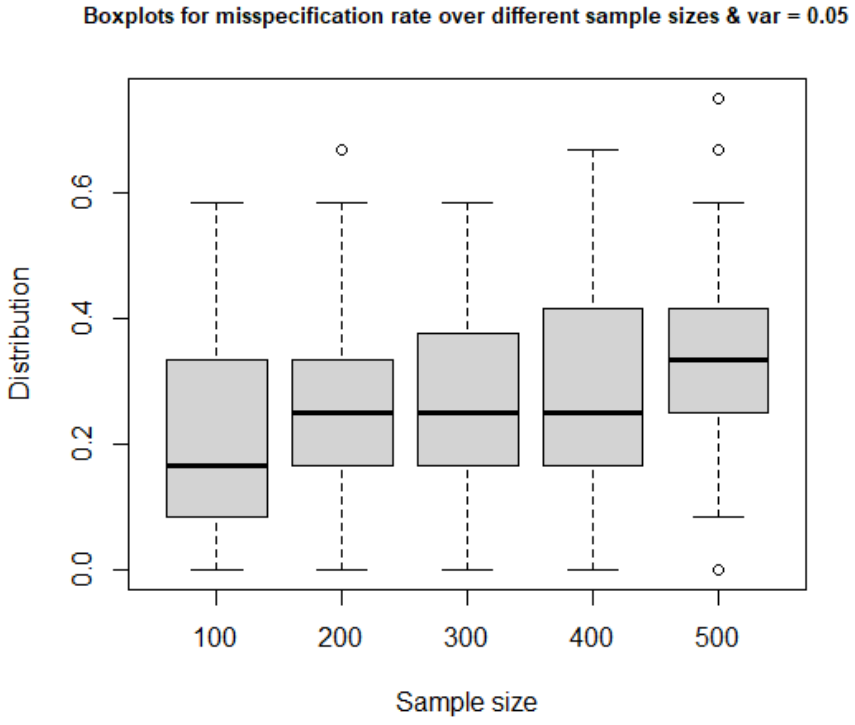


Figure 2.1: Distribution of misspecification rates of KNNIM over standard deviations



(a)



(b)

Figure 2.2: Boxplots for misspecification rates over different sample size, w = 1 with standard deviation (a) 0.02 & (b) 0.05

Table 2.1: Interference models

Models	$(\beta_1, \beta_2, \beta_3, \beta_d)$
Model 1	$(0, 0, 0, 0)$
Model 2	$(0, 0, 0, 4)$
Model 3	$(2, 1, 0.5, 1)$
Model 4	$(3, 2, 1, 4)$

2.6 Study on causal estimates

We now investigate the impact that misspecification has on causal estimates under KNNIM. For each simulation, we compute Horvitz-Thompson (HT) estimates of the average direct effect (ADE), the average indirect effect (AIE), and the average ℓ^{th} nearest neighbor indirect effects, $\ell = 1, 2, 3$.

The first KNNIM Model ($w = 1, \boldsymbol{\beta} = (0, 0, 0, 0)$) provides a control setting as this model assumes no treatment effect whatsoever on experimental units. Simulations were performed to study HT estimates for this model over different sample sizes and two different injected misspecifications with error standard deviations of 0.05 and 0.1 respectively. Figures 2.3 and 2.4 present the boxplots for distributions of different HT estimates under Model 1. Boxplots in these figures provide the expected result of no treatment effects under different misspecifications, with the median estimates of ADE, AIE, 1st AℓNNE, 2nd AℓNNE and 3rd AℓNNE being aligned with true values.

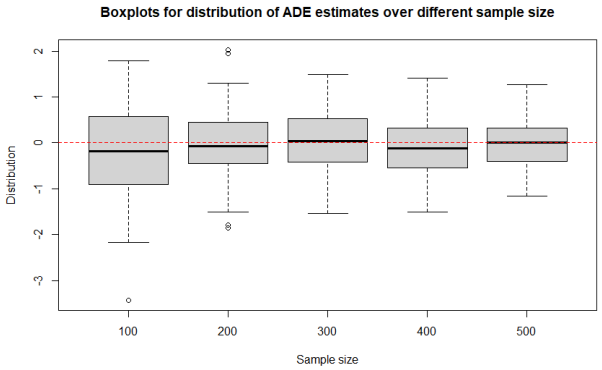
In the second KNNIM Model ($w = 2, \boldsymbol{\beta} = (0, 0, 0, 4)$) under study ,it was considered that the only treatment effect available was due to the treatment status of the unit of interest. In other words, only the direct treatment effect was present and there were no indirect causal effects from the nearest neighboring units. Thus, Model 2 should also provide a control setting, as misspecification of neighbors should have no effect due to no neighbors having any impact on response. Figures 2.5 and 2.6 provide visual presentations of different HT estimates distribution for Model 2 over changing sample size and injected misspecification. From the boxplots of HT causal estimates, it is noticeable that the ADE estimate aligns with the true value while the medians of other causal effects lie near the true value zero over different sample sizes and error standard deviation of misspecifications.

Figures 2.7 and 2.8 present H-T causal estimates distribution for KNNIM Model 3 ($w = 3, \boldsymbol{\beta} = (2, 1, 0.5, 1)$) over varying sample size and different standard deviations of random errors injected in units closeness measure. In this model a weak interaction was considered between experimental units and all direct and indirect treatment effects were expected to appear in this interference settings. In figure 2.7, HT estimates boxplots show that misspecification inserts significant bias in estimation of the indirect effect of 1st nearest neighbor (1st AℓNN) and eventually the average indirect effect (AIE) compared to other causal estimates. This bias seems to increase with increasing sample size and increase in standard deviation of applied misspecification.

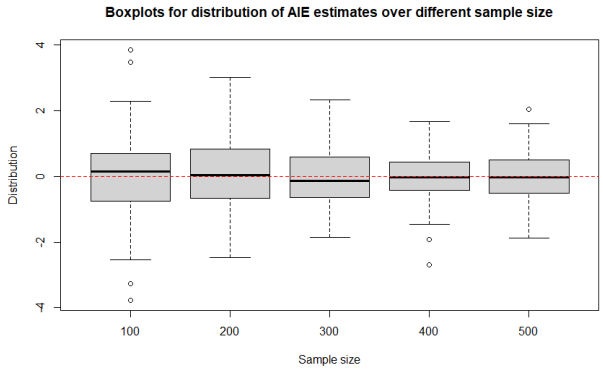
To study the performance of HT causal estimates, strong interactions between experimental units were considered under the fourth KNNIM model ($w = 4, \boldsymbol{\beta} = (3, 2, 1, 4)$). The model was investigated for different sample sizes $N = 100, 200, 300, 400, 500, 1000, 2000$ with three different injected error standard deviations ($\sigma = 0.02, 0.05, 0.1$). Figures 2.9, 2.10, and 2.11 include boxplots of different HT estimates for Model 4. It is evident from the plots that similar to the performances of HT estimates under weak interaction, more uncertainty is present in estimating indirect effect of the nearest unit while other causal effect estimates are not significantly biased by the misspecification inserted in KNNIM model. Boxplots in these figures also suggest that larger samples and greater standard deviation in injected error result more bias in estimation of 1st AℓNN as well as AIE.

It is noticeable in both weak and strong interference (Model 4 and Model 5), average indirect effect (1st AℓNN) of 1st nearest neighbor showed more bias compared to the average indirect effects of second and third nearest neighbor (2nd AℓNN, 3rd AℓNN). A natural extension K-nearest neighbor interference model proposes that the nearest neighboring unit or 1st the closest unit has the maximum indirect effect on the unit of interest. Therefore the magnitudes of indirect effect decrease gradually for the second and third closest units. When misspecification is injected in any interference structure, it results incorrect identification of neighboring units which impacts the indirect causal effects. As the closest most unit has the maximum indirect effect, misspecification or wrong identification of first nearest neighbor causes more bias in the estimation of first nearest neighbor indirect effect ((1st AℓNNIE).

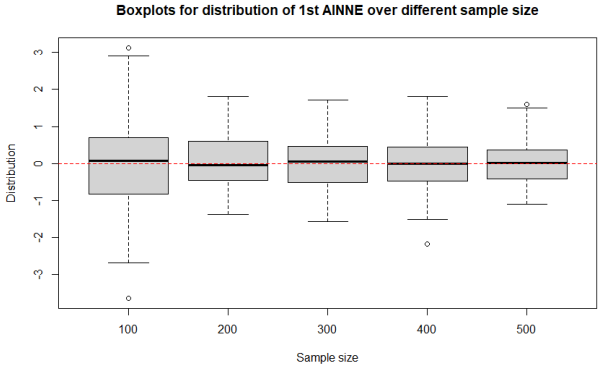
Magnitudes of respective indirect effects due to the second closest and third closest units in neighborhood drops with decreasing closeness to the unit of interest. Moreover, incorrect specification of, say, the 2nd nearest neighbor could either cause an increase (if the 1st nearest neighbor was incorrectly specified as the 2nd nearest neighbor) or decrease (if the 3rd nearest neighbor was incorrectly specified as the 2nd nearest neighbor) the estimate of the AℓNNIE, whereas incorrect specification of the 1st nearest neighbor will only lead to underestimates of the parameter. Therefore under weaK and strong interference, estimation of second or third nearest neighbor indirect effect ((2nd and 3rd AℓNN) were not significantly biased as the first nearest neighbor indirect effect estimation (1st AℓNN).



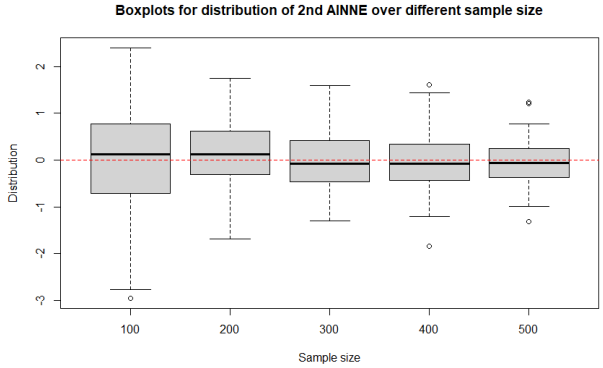
(a) ADE estimate



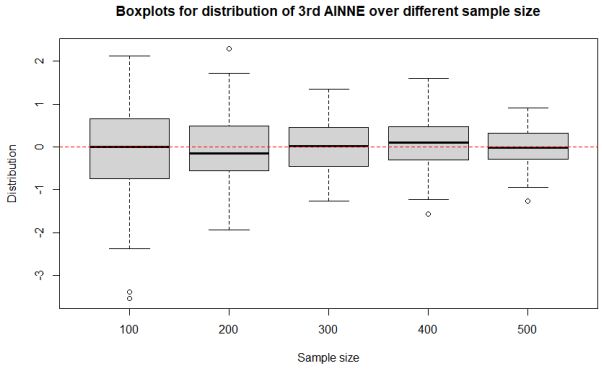
(b) AIE estimate



(c) 1st AlNNE estimate

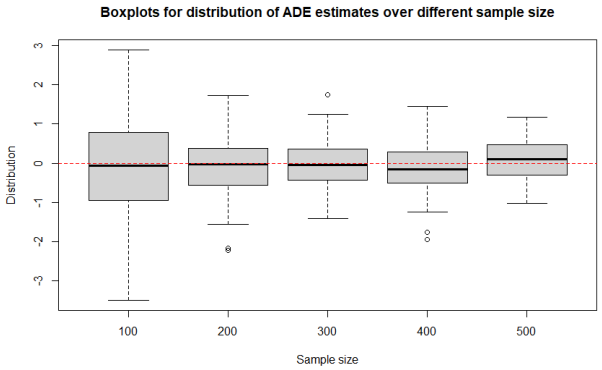


(d) 2nd AlNNE estimates

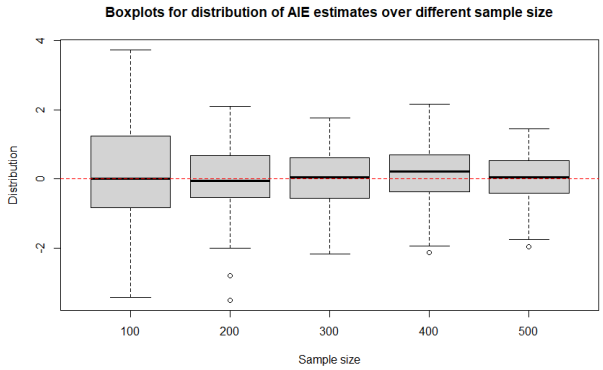


(e) 3rd AlNNE estimate

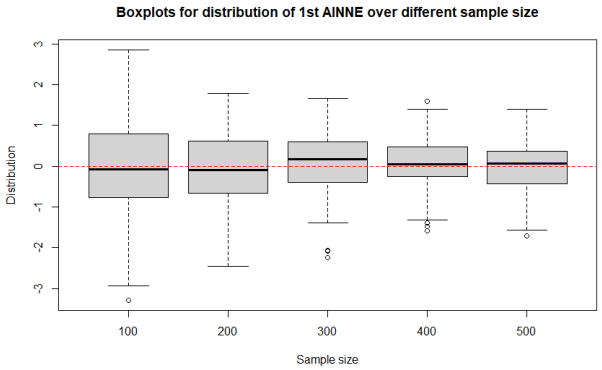
Figure 2.3: Boxplots for HT estimates over different sample size, w = 1, standard deviation = 0.05



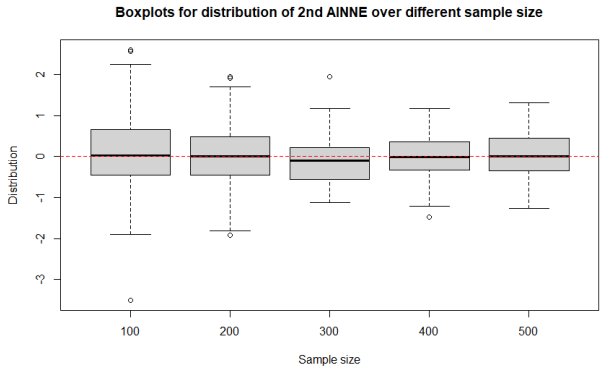
(a) ADE estimate



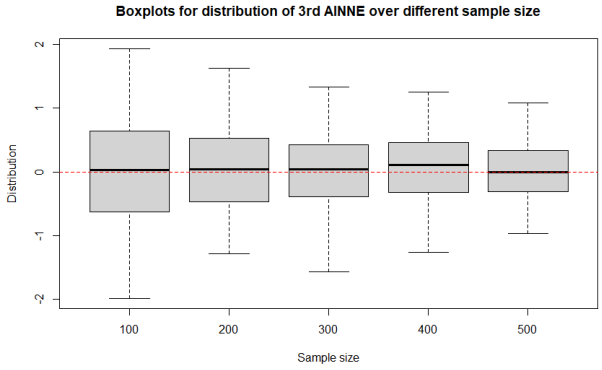
(b) AIE estimate



(c) 1st AlNNE estimate

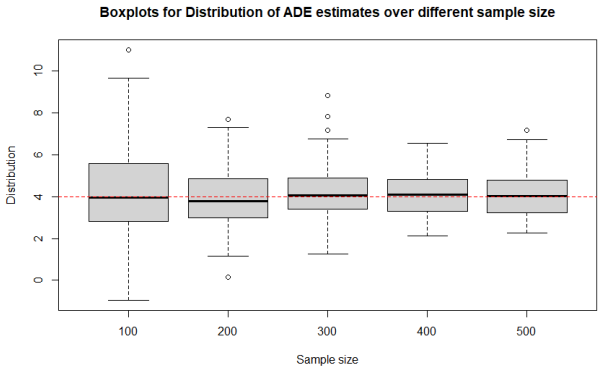


(d) 2nd AlNNE estimates

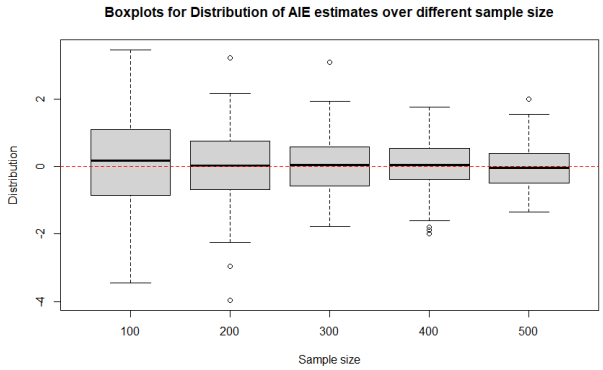


(e) 3rd AlNNE estimate

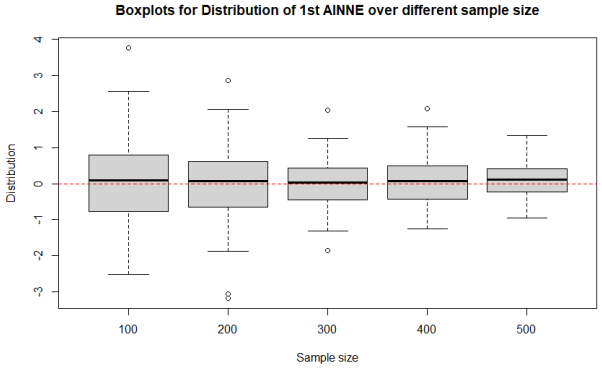
Figure 2.4: Boxplots for HT estimates over different sample size, w = 1, standard deviation = 0.1



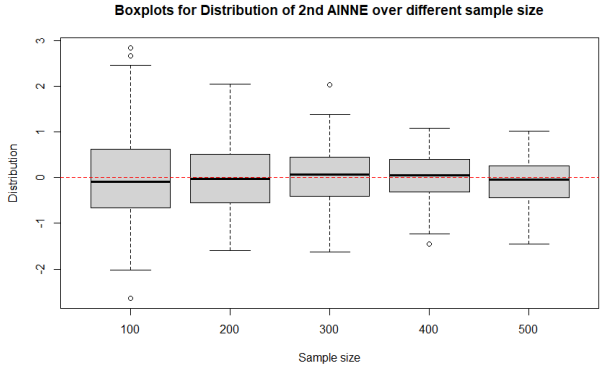
(a) ADE estimate



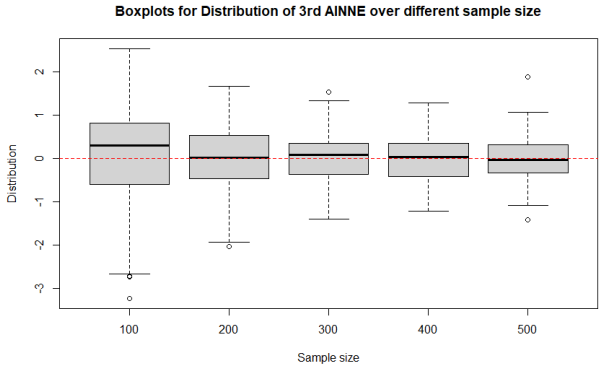
(b) AIE estimate



(c) 1st AlNNE estimate

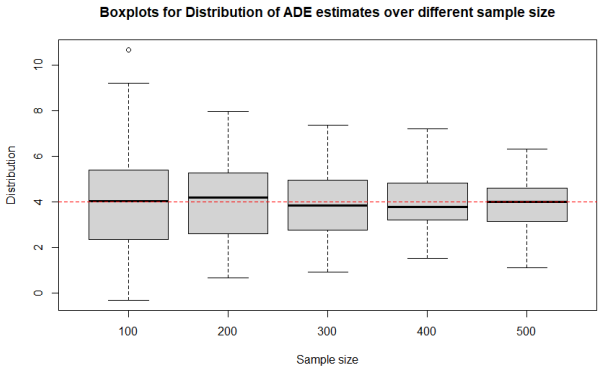


(d) 2nd AlNNE estimates

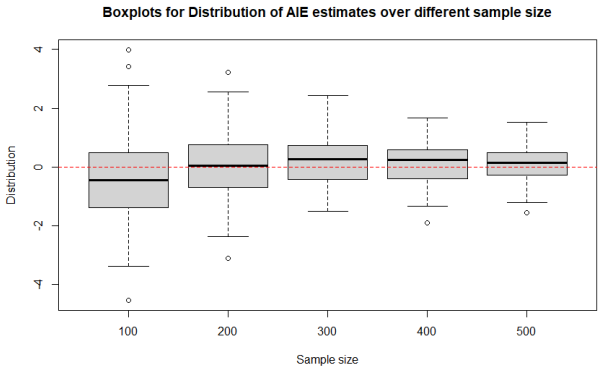


(e) 3rd AlNNE estimate

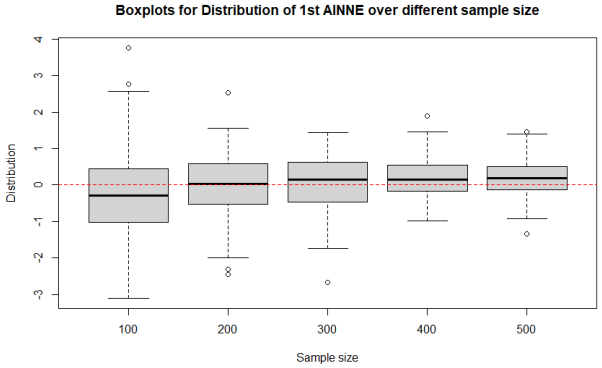
Figure 2.5: Boxplots for HT estimates over different sample size, w = 2, standard deviation = 0.05



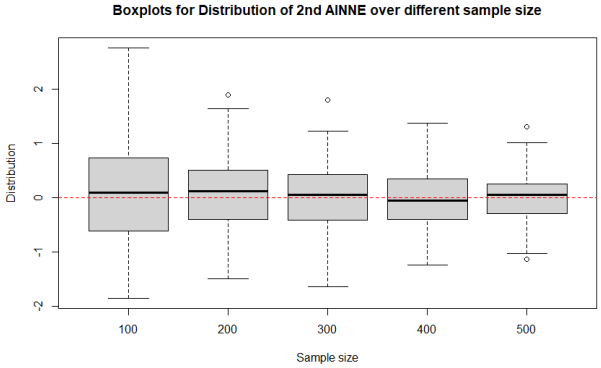
(a) ADE estimate



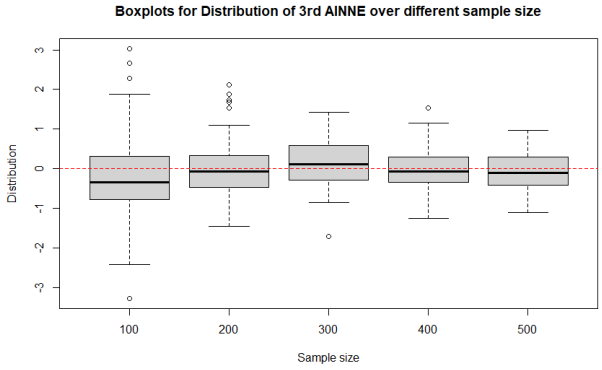
(b) AIE estimate



(c) 1st AlNNE estimate

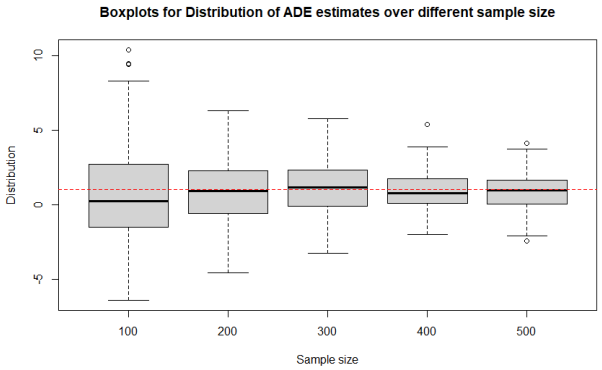


(d) 2nd AlNNE estimates

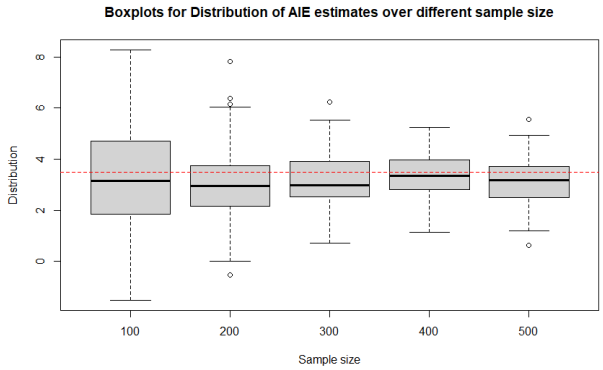


(e) 3rd AlNNE estimate

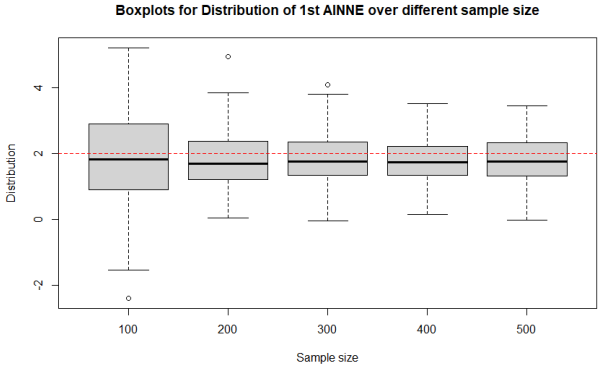
Figure 2.6: Boxplots for HT estimates over different sample size, w = 2, standard deviation = 0.1



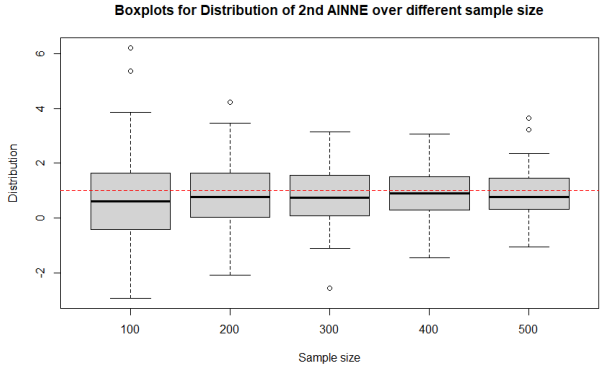
(a) ADE estimate



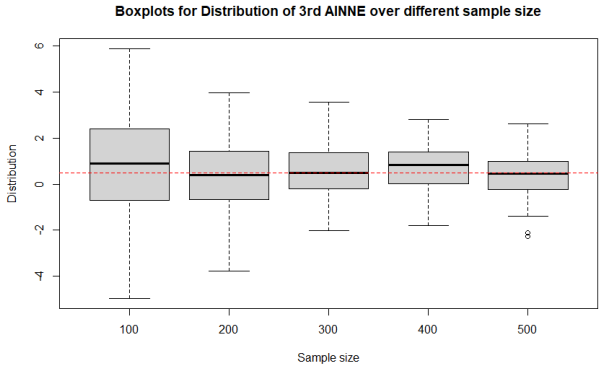
(b) AIE estimate



(c) 1st AlNNE estimate

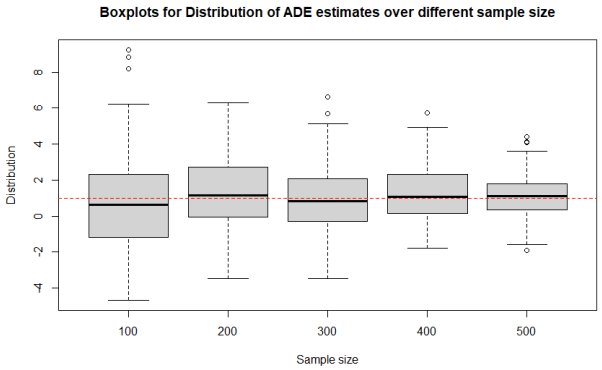


(d) 2nd AlNNE estimates

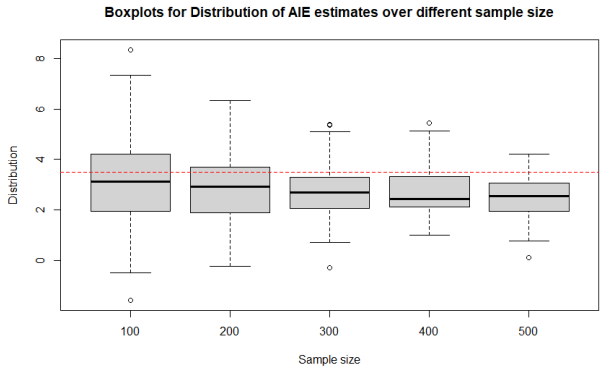


(e) 3rd AlNNE estimate

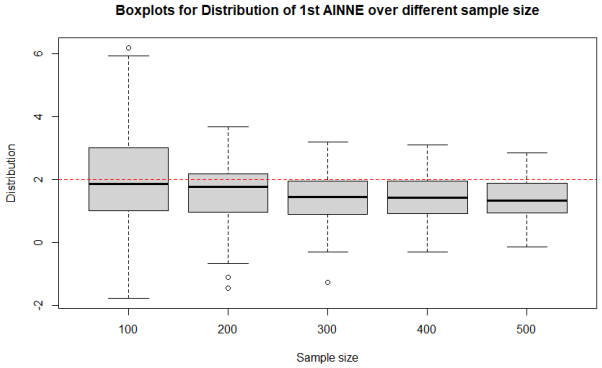
Figure 2.7: Boxplots for HT estimates over different sample size, w = 3, standard deviation = 0.05



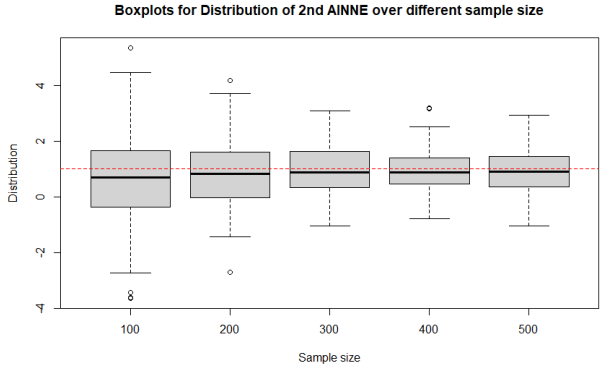
(a) ADE estimate



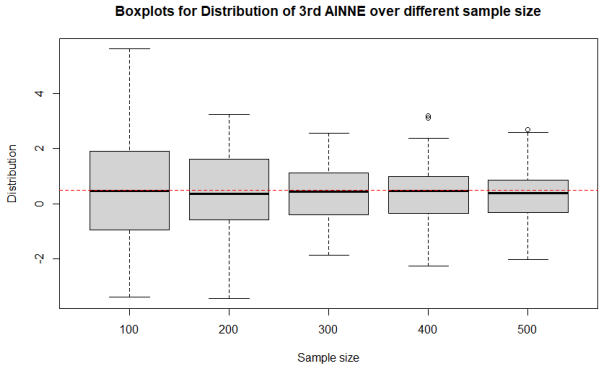
(b) AIE estimate



(c) 1st AlNNE estimate

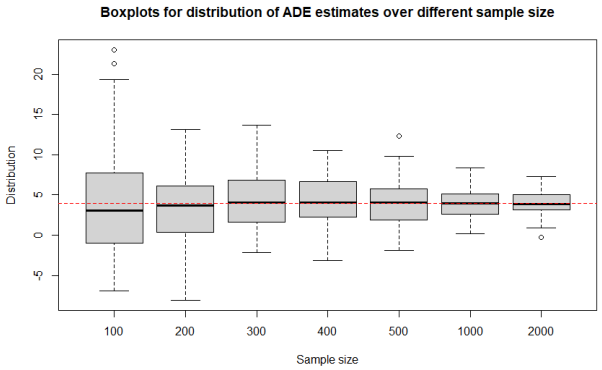


(d) 2nd AlNNE estimates

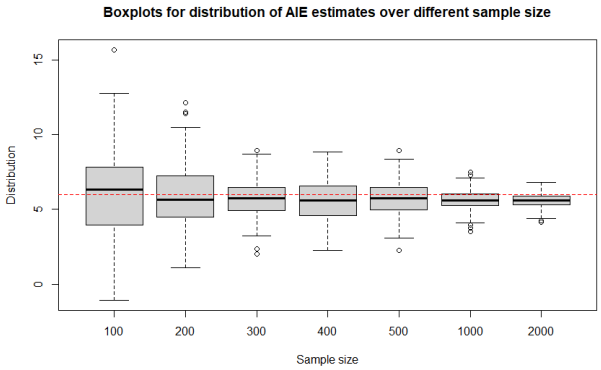


(e) 3rd AlNNE estimate

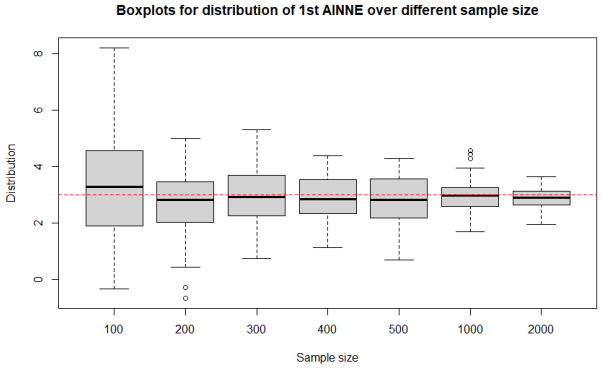
Figure 2.8: Boxplots for HT estimates over different sample size, w = 3, standard deviation = 0.1



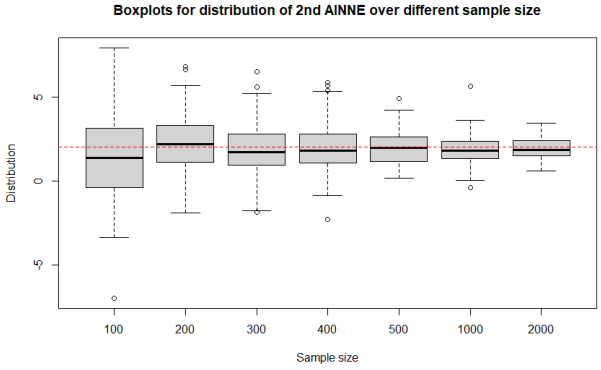
(a) ADE estimate



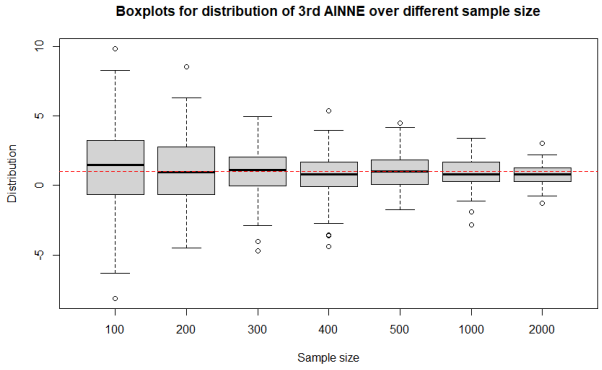
(b) AIE estimate



(c) 1st AlNNE estimate

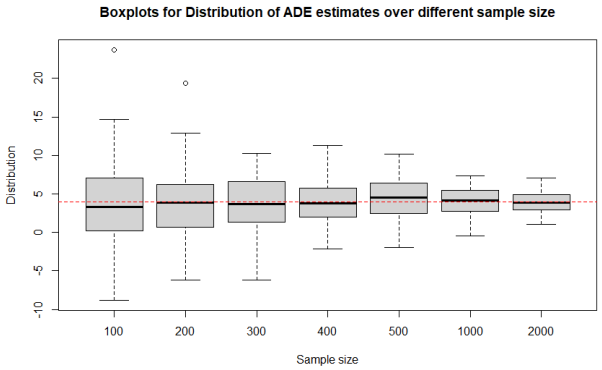


(d) 2nd AlNNE estimates

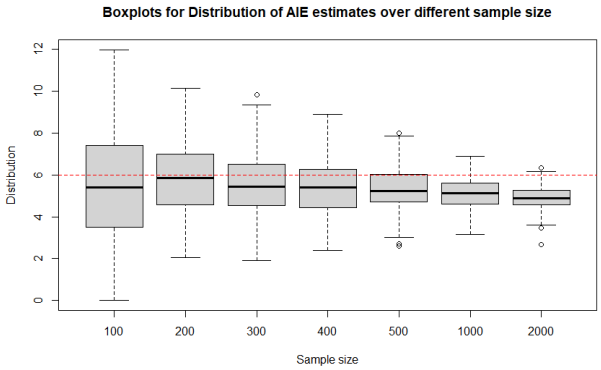


(e) 3rd AlNNE estimate

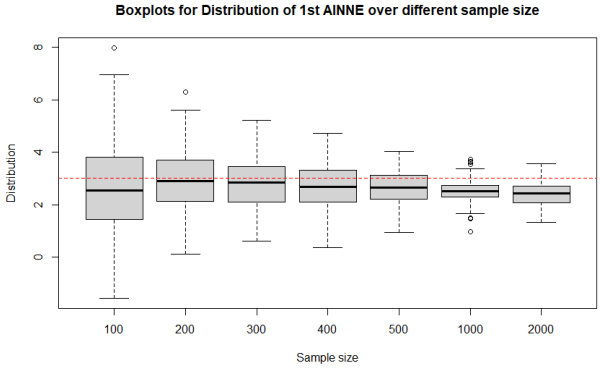
Figure 2.9: Boxplots for HT estimates over different sample size, w = 4, standard deviation = 0.02



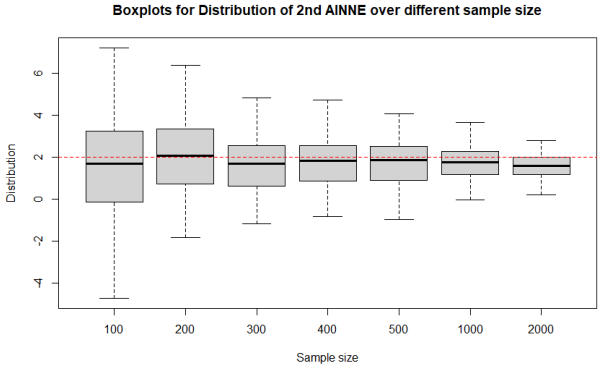
(a) ADE estimate



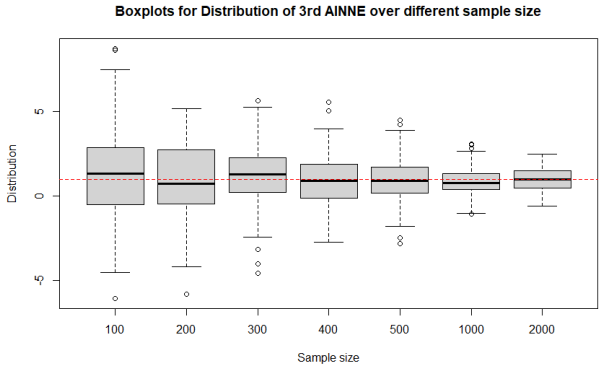
(b) AIE estimate



(c) 1st AlNNE estimate

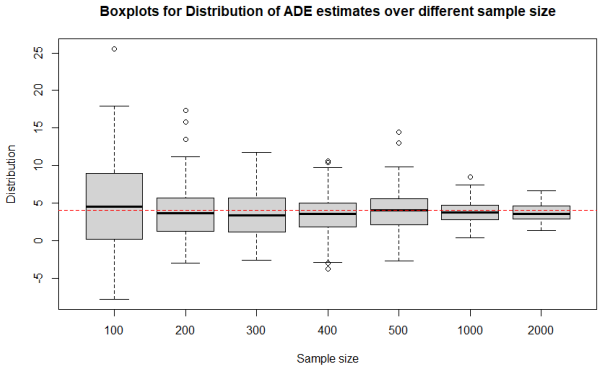


(d) 2nd AlNNE estimates

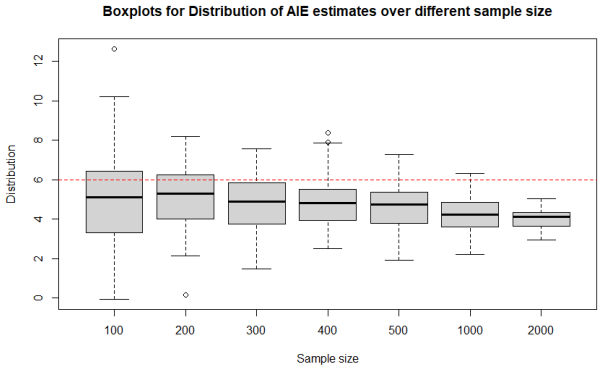


(e) 3rd AlNNE estimate

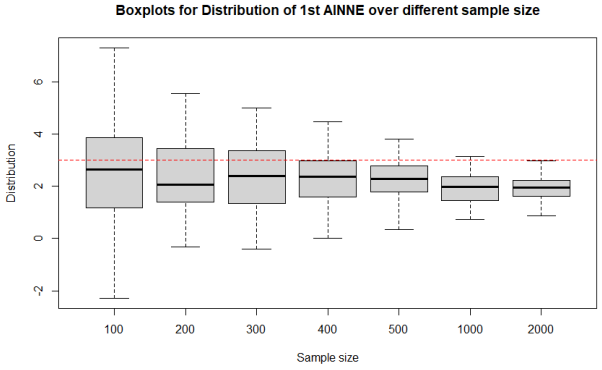
Figure 2.10: Boxplots for H-T estimates over different sample size, w = 4, standard deviation = 0.05



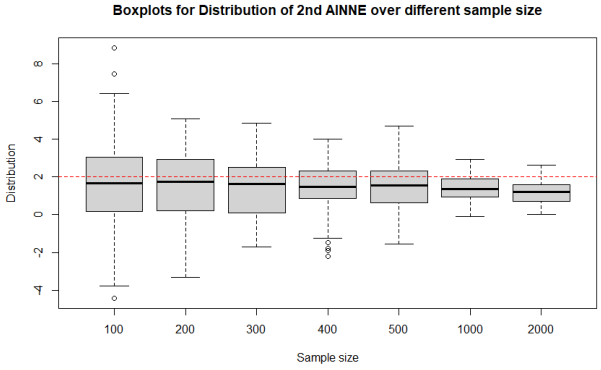
(a) ADE estimate



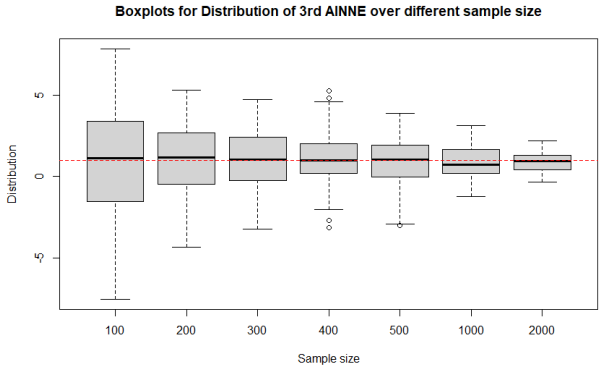
(b) AIE estimate



(c) 1st AlNNE estimate



(d) 2nd AlNNE estimates



(e) 3rd AlNNE estimate

Figure 2.11: Boxplots for HT estimates over different sample size, w = 4, standard deviation = 0.1

Chapter 3

Conclusion

3.1 Conclusion

In causal inference, interference comes into play when the effect of treatment is not only limited to the experimental unit that receives the treatment, rather it can also affect other units responses that have interactions with the original units. Over the last two decades, a substantial amount of research work have been devoted in developing models and techniques for studying causal inference under interference. The K-Nearest Neighbor Model (KNNIM) is one of these causal inference models that incorporate interference in causal analysis.

The KNNIM model proposed by Alzubaidi and Higgins (2024) supposes that the response of one unit can be affected the treatment statuses of the K "closest" units to the original unit. Therefore, interference is considered to be restricted to nearest neighboring units and a controlled interference structure is integrated in KNNIM model by defining exposure mapping. Exposure mapping can be considered as a network where nodes of the network represent experimental units and edges indicate units' possible interactions with each other. A "closeness measure" may quantify the strength of interaction between two units. The closer the units, the stronger the interactions. But incorrect specification of exposure mapping—or in other words, wrong identification of nearest neighboring units—result misspecification in interference structure.

In this research work, estimation of causal effects under misspecification of KNNIM was examined through a simulation study. Random error was injected in closeness measure of the K nearest neighbors and response data was generated for various sample sizes ($N = 100, 200, 300, 400, 500$) and changing variances (0.02, 0.05 and 0.1) of random error. From the boxplots of misspecification rates over different variances for injected error and sample size of 500, it was visualized that misspecification rates significantly increased with increasing standard deviation while misspecification rates boxplots over different sample sizes with constant standard deviation of random error showed larger samples might result greater misspecification in interference structure of neighboring units.

Horvitz-Thompson (HT) estimates average direct treatment effect (ADE), average indirect treatment effect (AIE), and three individual nearest neighbors' indirect effects (1st, 2nd & 3rd AℓNNE) were provided. To study the performances of these HT estimates, four different KNNIM models were investigated considering four different conditions of "no treatment effect", "only direct treatment effect", "weak interference" and "strong interference" respectively. KNNIM models were simulated over varying sample sizes and changing variances of injected misspecification. From the simulation results it was found that "under no treatment effect" and "only direct treatment effect" conditions all HT estimates closely align with true values of causal effects. In cases of "weak interference" and "strong interference" conditions, misspecifications seemed to impact in biased estimation of the first nearest neighbor's indirect effect and the total average indirect effect leaving other causal estimates unaffected. For both conditions of the KNNIM models, these two HT estimates showed more uncertainty with increasing sample size and variance of injected misspecification.

3.2 Future Work

In this research work, misspecification of KNNIM and performances of HT estimates for direct, indirect and nearest neighbor indirect causal effect were investigated in a completely randomized design. In future, a research extension is to consider Bernoulli randomization for further study of misspecified interference and HT causal estimates under KNNIM and

to compare the results with the findings of complete a randomized design. Additionally, this study detected substantial underestimation of the 1st-nearest-neighbor treatment effect. In future work, we may consider adjustments to estimates of this causal estimand under misspecification —for example, through inflating this estimate by some factor—to reduce bias.

Bibliography

Samirah Alzubaidi. *Causal inference under the K-nearest neighbors interference model*. Kansas State University, 2022.

Samirah Alzubaidi and Michael J Higgins. Estimation of causal effects under k-nearest neighbors interference. *arXiv preprint arXiv:2307.15204*, 2023.

Samirah H Alzubaidi and Michael J Higgins. Detecting treatment interference under k-nearest-neighbors interference. *Journal of Causal Inference*, 12(1), 2024.

Peter M Aronow and Cyrus Samii. Estimating average causal effects under general interference, with application to a social network experiment. 2017.

Peter M Aronow, Dean Eckles, Cyrus Samii, and Stephanie Zonszein. Spillover effects in experimental data. *Advances in experimental political science*, 289:319, 2021.

Dean Eckles, Brian Karrer, and Johan Ugander. Design and analysis of experiments in networks: Reducing bias from interference. *Journal of Causal Inference*, 5(1):20150021, 2017.

Laura Forastiere, Edoardo M Airoldi, and Fabrizia Mealli. Identification and estimation of treatment and interference effects in observational studies on networks. *Journal of the American Statistical Association*, 116(534):901–918, 2021.

Huan Gui, Ya Xu, Anmol Bhasin, and Jiawei Han. Network a/b testing: From sampling to estimation. In *Proceedings of the 24th International Conference on World Wide Web*, pages 399–409, 2015.

Paul W Holland. Statistics and causal inference. *Journal of the American statistical Association*, 81(396):945–960, 1986.

Daniel G Horvitz and Donovan J Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, 47(260): 663–685, 1952.

Michael G Hudgens and M Elizabeth Halloran. Toward causal inference with interference. *Journal of the American Statistical Association*, 103(482):832–842, 2008.

Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge university press, 2015.

Brian Karrer, Liang Shi, Monica Bhole, Matt Goldman, Tyrone Palmer, Charlie Gelman, Mikael Konutgan, and Feng Sun. Network experimentation at scale. In *Proceedings of the 27th acm sigkdd conference on knowledge discovery & data mining*, pages 3106–3116, 2021.

Michael P Leung. Causal inference under approximate neighborhood interference. *Econometrica*, 90(1):267–293, 2022.

J Neyman. On the application of probability theory to agricultural experiments: Principles (translated from polish original). *Roczniki Nauk Rolniczch*, 10(1):21–51, 1923.

Jean Pouget-Abadie, Guillaume Saint-Jacques, Martin Saveski, Weitao Duan, Souvik Ghosh, Ya Xu, and Edoardo M Airoldi. Testing for arbitrary interference on experimentation platforms. *Biometrika*, 106(4):929–940, 2019.

Paul R Rosenbaum. Interference between units in randomized experiments. *Journal of the american statistical association*, 102(477):191–200, 2007.

Donald B Rubin. Randomization analysis of experimental data: The fisher randomization test comment. *Journal of the American statistical association*, 75(371):591–593, 1980.

Donald B Rubin. Essential concepts of causal inference: a remarkable history and an intriguing future. *Biostatistics & Epidemiology*, 3(1):140–155, 2019.

Martin Saveski, Jean Pouget-Abadie, Guillaume Saint-Jacques, Weitao Duan, Souvik Ghosh, Ya Xu, and Edoardo M Airoldi. Detecting network effects: Randomizing over randomized experiments. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1027–1035, 2017.

Fredrik Sävje. Causal inference with misspecified exposure mappings: separating definitions and assumptions. *Biometrika*, 111(1):1–15, 2024.

Fredrik Sävje, Peter Aronow, and Michael Hudgens. Average treatment effects in the presence of unknown interference. *Annals of statistics*, 49(2):673, 2021.

Michael E Sobel. What do randomized studies of housing mobility demonstrate? causal inference in the face of interference. *Journal of the American Statistical Association*, 101 (476):1398–1407, 2006.

Daniel L Sussman and Edoardo M Airoldi. Elements of estimation theory for causal effects in the presence of network interference. *arXiv preprint arXiv:1702.03578*, 2017.

Eric J Tchetgen Tchetgen and Tyler J VanderWeele. On causal inference in the presence of interference. *Statistical methods in medical research*, 21(1):55–75, 2012.

Panos Toulis and Edward Kao. Estimation of causal peer influence effects. In *International conference on machine learning*, pages 1489–1497. PMLR, 2013.

Johan Ugander, Brian Karrer, Lars Backstrom, and Jon Kleinberg. Graph cluster randomization: Network exposure to multiple universes. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 329–337, 2013.

Bar Weinstein and Daniel Nevo. Causal inference with misspecified network interference structure. *arXiv preprint arXiv:2302.11322*, 2023.

Appendix A

Simulation code to generate response data and study misspecification rate of KNNIM

```
# R Code for generating response data with injected error and without injected error
#true_response is response without injecting error
#response_with_error is response under error
#N=sample size
#K=no of nearest neighbor
#U=mean of random injected error
#V= variance of random injected error
#NN.coeff=matrix of coefficient vectors for near neighbors

#function to generate response data with injected error and without injected error
generate_data=function(N, K, U, V, w, NN.Coeff){
  #number of treated units
  N.t=N/2
  #Number of neighbors including treated unit
```

```
S=K+1

# generation of covariates
cov1<-rnorm(N)
cov2<-rnorm(N)
cov3<-rnorm(N)
#covariates matrix
mycov<-cbind(cov1, cov2, cov3)

# distance in covariates without injecting error
trudist = dist(mycov)
dismat <-as.matrix(trudist)

#injected error
errmat = matrix(0,nrow = N, ncol = N)
E= rnorm(
  N*(N-1)/2, U, V)
errmat[upper.tri(errmat)] = E
errmat = errmat + t(errmat)

# distance in covariates without injecting error
randdismat = dismat + errmat
#randdismat <-as.matrix(randdist)

#nearest neighbours
nearneigh <- apply(dismat,2,function(x){order(x)[c(2:S,1)]})


#nearest neighbours with injected error
```

```
nearneighmis <- apply(randdismat,2,function(x){order(x)[c(2:S,1)]})

#treatment given to units
hypcomprand = sample(c(rep(0,N/2),rep(1,N/2)))

indirect.effect= NN.Coeff[w,1:3]
direct.effect = NN.Coeff[w,4]
coeffvec =c(indirect.effect,direct.effect)

#treatment status of unit and nearest neighbors without injected error
trtstat = apply(nearneigh, 2, function(x){hypcomprand[x]})

#treatment status of unit and nearest neighbors with injected error
trtstat_mis = apply(nearneighmis, 2, function(x){hypcomprand[x]})

#response for any treatment assignment
getResponse = function(covs,trt, coeffvec){
  1*covs[1]+1*covs[2] + 1*covs[3] + sum(coeffvec*trt)
}

#function to get one potential outcome
getWhichPotOut = function(mytrt,possassign){
  which(apply(possassign,1,function(x){
    isTRUE(all.equal(mytrt,x,check.attributes = FALSE))}))
}

#generate potential outcome
getPotout = function(covs,possassign, coeffvec){
  apply(possassign, 1, function(x){apply(covs,1,function(y){
```

```
    getResponse(y,x,coeffvec)})})
}

#possible treatment assignment
expandlist = NULL
for(i in 1:S){
  expandlist[[i]] = c(0,1)
}
mypossassign = expand.grid(expandlist)

mypotout=getPotout(mycov,mypossassign,coeffvec)

whichPotOut = apply(trtstat, 2, getWhichPotOut, possassign = mypossassign)
whichPotOutmat = matrix(whichPotOut,nrow = N, ncol = 1)

whichPotOut_with_error = apply(trtstat_mis, 2, getWhichPotOut, possassign = mypossassi
whichPotOutmat_with_error = matrix(whichPotOut_with_error,nrow = N, ncol = 1)

#true response
true_response = rep(0,N)
for(i in 1:N){
  true_response[i] = mypotout[i,whichPotOut[i]]
}

#response with error
response_with_error = rep(0,N)
for(i in 1:N){
  response_with_error[i] = mypotout[i,whichPotOut_with_error[i]]
}
```

```
  # #misspecification rate
  misspecification_rate=1-mean((nearneigh == nearneighmis)[,1:K])
  misspecification_rate_df =as.data.frame(misspecification_rate)
  names(misspecification_rate_df)=c('misspecification rate')

  nearneighdf = as.data.frame(t(nearneigh))
  names(nearneighdf) = c("NN1", "NN2", "NN3", "UnitID")

  nearneighmisdf = as.data.frame(t(nearneighmis))
  names(nearneighmisdf) = c("NN1Err", "NN2Err", "NN3Err", "UnitIDErr")

  trtstatdf = as.data.frame(t(trtstat))
  names(trtstatdf) = c("NN1Trt", "NN2Trt","NN3Trt", "UnitTrt" )

  trtstat_misdf = as.data.frame(t(trtstat_mis))
  names(trtstat_misdf) = c("NN1ErrTrt", "NN2ErrTrt","NN3ErrTrt", "UnitErrTrt" )


  #generated data
  mydata=data.frame(mycov,nearneighdf,response_with_error, nearneighmisdf, hypcomprand,

  mydata
}



#Create a list of all possible exposures:
#Exposures
```

```
Exposures = list(c(0,0,0,0), c(1,0,0,0), c(0,1,0,0), c(1,1,0,0), c(0,0,1,0),c(1,0,1,0),c

#function for calculating Exact Marginal Probabilities
generate_pi=function(N,K, Exposures){pi = sapply(Exposures,function(x){N.kt = sum(x)
N.t=N/2
pi = choose(N-K-1,N.t-N.kt) / choose(N,N.t)})
pi}



#response
#TRT STATUS OF THE NEAREST NEIGHBORS
# neartrt = matrix(trt[nearneigh],nrow = nrow(nearneigh), ncol = ncol(nearneigh))
# mydata$N1 = neartrt[1,]
# mydata$N2 = neartrt[2,]
# mydata$N3 = neartrt[3,]
#
#
# numexp = rep(0,N)
# for(i in 1:N){
#   alleq = lapply(Exposures, function(x){all(x == neartrt[,i])})
#   numexp[i] = which(unlist(alleq))
# }
# mydata$exposure = numexp


#function to compute Horvitz-Thompson Estimator
#Use number 1-16 for exposure, using the exposure list.
HT_estimator<- function(exposure, pi, thedata){
```

```
  whichexp = which(thedata$whichPotOutmat == exposure)
  (1/nrow(thedata))*sum(thedata[whichexp,8]/pi[exposure])
}

#Horvitz-Thompson estimates distribution for different sample sizes
#initialize variables

A = rep(0, 100)
B = rep(0, 100)
C = rep(0, 100)
D = rep(0, 100)
E = rep(0, 100)
G = rep(0, 100)
H = rep(0, 100)
I = rep(0, 100)
J = rep(0, 100)
K = rep(0, 100)
L = rep(0, 100)
M = rep(0, 100)
N_0 = rep(0, 100)
O = rep(0, 100)
P = rep(0, 100)
Q_1 = rep(0, 100)
R_1 = rep(0, 100)
S_1 = rep(0, 100)
U_1 = rep(0, 100)
V_1 = rep(0, 100)
Q_2 = rep(0, 100)
R_2 = rep(0, 100)
```

```
S_2 = rep(0, 100)
U_2 = rep(0, 100)
V_2 = rep(0, 100)
Q_3 = rep(0, 100)
R_3 = rep(0, 100)
S_3 = rep(0, 100)
U_3 = rep(0, 100)
V_3 = rep(0, 100)
mis_sp_N_500_V_0.02 =rep(0, 100)
mis_sp_N_500_V_0.05 =rep(0, 100)
mis_sp_N_500_V_0.1 =rep(0, 100)


coeff=matrix(c(0,0,0,0,  0,0,0,1, 0,0,0,4, 2,1,0.5,0, 2,1,0.5,1,
               2,1,0.5,4, 3,2,1,0, 3,2,1,1, 3,2,1,4), nrow=9,ncol=4, byrow  = TRUE)
for(i in 1:100){
  #generate data for different sample sizes
  data_N_500_V_0.02=generate_data(N=500, K=3, U=0, V= 0.02, w=1, NN.Coeff=coeff)
  data_N_500_V_0.05=generate_data(N=500, K=3, U=0, V= 0.05, w=1, NN.Coeff=coeff)
  data_N_500_V_0.1=generate_data(N=500, K=3, U=0, V= 0.1, w=1, NN.Coeff=coeff)
  #
  #misspecification rate
  mis_sp_N_500_V_0.02[i]=data_N_500_V_0.02[1,24]
  mis_sp_N_500_V_0.05[i]=data_N_500_V_0.05[1,24]
  mis_sp_N_500_V_0.1[i]=data_N_500_V_0.1[1,24]

  # pi_N_100=generate_pi(N=100, K=3, Exposures)
  # pi_N_200=generate_pi(N=200, K=3, Exposures)
  # pi_N_300=generate_pi(N=300, K=3, Exposures)
```

```
# pi_N_400=generate_pi(N=400, K=3, Exposures)
# pi_N_500=generate_pi(N=500, K=3, Exposures)
#
# #HT
# A[i] = HT_estimator(exposure = 12, pi_N_100,data_N_100)
# B[i] = HT_estimator(exposure = 12, pi_N_200, data_N_200)
# C[i] = HT_estimator(exposure = 12, pi_N_300, data_N_300)
# D[i] = HT_estimator(exposure = 12, pi_N_400, data_N_400)
# E[i] = HT_estimator(exposure = 12, pi_N_500, data_N_500)
# #ADE
# G[i] = HT_estimator(exposure = 16, pi_N_100,data_N_100)-HT_estimator(exposure = 8, p
# H[i] = HT_estimator(exposure = 16, pi_N_200, data_N_200)-HT_estimator(exposure = 8,
# I[i] = HT_estimator(exposure = 16, pi_N_300, data_N_300)-HT_estimator(exposure = 8,
# J[i] = HT_estimator(exposure = 16, pi_N_400, data_N_400)-HT_estimator(exposure = 8,
# K[i] = HT_estimator(exposure = 16, pi_N_500, data_N_500)-HT_estimator(exposure = 8,
# #AIE
# L[i] = HT_estimator(exposure = 8, pi_N_100,data_N_100)-HT_estimator(exposure = 1, pi
# M[i] = HT_estimator(exposure = 8, pi_N_200, data_N_200)-HT_estimator(exposure = 1, p
# N_0[i] = HT_estimator(exposure = 8, pi_N_300, data_N_300)-HT_estimator(exposure = 1,
# O[i] = HT_estimator(exposure = 8, pi_N_400, data_N_400)-HT_estimator(exposure = 1, p
# P[i] = HT_estimator(exposure = 8, pi_N_500, data_N_500)-HT_estimator(exposure = 1, p
# #1st AlNNE
# Q_1[i] = HT_estimator(exposure = 2, pi_N_100,data_N_100)-HT_estimator(exposure = 1,
# R_1[i] = HT_estimator(exposure = 2, pi_N_200, data_N_200)-HT_estimator(exposure = 1,
# S_1[i] = HT_estimator(exposure = 2, pi_N_300, data_N_300)-HT_estimator(exposure = 1,
# U_1[i] = HT_estimator(exposure = 2, pi_N_400, data_N_400)-HT_estimator(exposure = 1,
# V_1[i] = HT_estimator(exposure = 2, pi_N_500, data_N_500)-HT_estimator(exposure = 1,
# #2nd AlNNE
# Q_2[i] = HT_estimator(exposure = 4, pi_N_100,data_N_100)-HT_estimator(exposure = 2,
```

```
  # R_2[i] = HT_estimator(exposure = 4, pi_N_200, data_N_200)-HT_estimator(exposure = 2,
  # S_2[i] = HT_estimator(exposure = 4, pi_N_300, data_N_300)-HT_estimator(exposure = 2,
  # U_2[i] = HT_estimator(exposure = 4, pi_N_400, data_N_400)-HT_estimator(exposure = 2,
  # V_2[i] = HT_estimator(exposure = 4, pi_N_500, data_N_500)-HT_estimator(exposure = 2,
  # #3rd AlNNE
  # Q_3[i] = HT_estimator(exposure = 8, pi_N_100,data_N_100)-HT_estimator(exposure = 4,
  # R_3[i] = HT_estimator(exposure = 8, pi_N_200, data_N_200)-HT_estimator(exposure = 4,
  # S_3[i] = HT_estimator(exposure = 8, pi_N_300, data_N_300)-HT_estimator(exposure = 4,
  # U_3[i] = HT_estimator(exposure = 8, pi_N_400, data_N_400)-HT_estimator(exposure = 4,
  # V_3[i] = HT_estimator(exposure = 8, pi_N_500, data_N_500)-HT_estimator(exposure = 4,

}


#boxplotes of misspecification rates of KNNIM over variances

plot_data_misspecification_rate_diff_Variance<- cbind('0.02'=mis_sp_N_500_V_0.02,'0.05'=
par(cex.main = 0.8)
boxplot(as.data.frame(plot_data_misspecification_rate_diff_Variance), xlab='Variance', m
```

Appendix B

Simulation code to study Horvitz-Thompson causal estimates under KNNIM

```
#Sample simulation code to study Horvitz_Thompson causal estimates under KNNIM
# Sample simulation code for KNNIM: Model 3 ("weak interference" condition)
# R Code for generating response data with injected error and without injected error
#true_response is response without injecting error
#response_with_error is response under error
#N=sample size
#K=no of nearest neighbor
#U=mean of random injected error
#V= variance of random injected error
#NN.coeff=matrix of coefficient vectors for near neighbors

#function to generate response data with injected error and without injected error
generate_data=function(N, K, U, V, w, NN.Coeff){
  #number of treated units
```

```
N.t=N/2
#Number of neighbors including treated unit
S=K+1

# generation of covariates
cov1<-rnorm(N)
cov2<-rnorm(N)
cov3<-rnorm(N)
#covariates matrix
mycov<-cbind(cov1, cov2, cov3)

# distance in covariates without injecting error
trudist = dist(mycov)
dismat <-as.matrix(trudist)

#injected error
errmat = matrix(0,nrow = N, ncol = N)
E= rnorm(
  N*(N-1)/2, U, V)
errmat[upper.tri(errmat)] = E
errmat = errmat + t(errmat)

# distance in covariates without injecting error
randdismat = dismat + errmat
#randdismat <-as.matrix(randdist)

#nearest neighbours
nearneigh <- apply(dismat,2,function(x){order(x)[c(2:S,1)]})
```

```
#nearest neighbours with injected error
nearneighmis <- apply(randdismat,2,function(x){order(x)[c(2:S,1)]})

#treatment given to units
hypcomprand = sample(c(rep(0,N/2),rep(1,N/2)))

indirect.effect= NN.Coeff[w,1:3]
direct.effect = NN.Coeff[w,4]
coeffvec =c(indirect.effect,direct.effect)

#treatment status of unit and nearest neighbors without injected error
trtstat = apply(nearneigh, 2, function(x){hypcomprand[x]})

#treatment status of unit and nearest neighbors with injected error
trtstat_mis = apply(nearneighmis, 2, function(x){hypcomprand[x]})

#response for any treatment assignment
getResponse = function(covs,trt, coeffvec){
  1*covs[1]+1*covs[2] + 1*covs[3] + sum(coeffvec*trt)
}

#function to get one potential outcome
getWhichPotOut = function(mytrt,possassign){
  which(apply(possassign,1,function(x){
    isTRUE(all.equal(mytrt,x,check.attributes = FALSE))}))
}

#generate potential outcome
```

```
getPotout = function(covs,possassign, coeffvec){
  apply(possassign, 1, function(x){apply(covs,1,function(y){
    getResponse(y,x,coeffvec)})})
}

#possible treatment assignment
expandlist = NULL
for(i in 1:S){
  expandlist[[i]] = c(0,1)
}
mypossassign = expand.grid(expandlist)

mypotout=getPotout(mycov,mypossassign,coeffvec)

whichPotOut = apply(trtstat, 2, getWhichPotOut, possassign = mypossassign)
whichPotOutmat = matrix(whichPotOut,nrow = N, ncol = 1)

whichPotOut_with_error = apply(trtstat_mis, 2, getWhichPotOut, possassign = mypossassi
whichPotOutmat_with_error = matrix(whichPotOut_with_error,nrow = N, ncol = 1)

#true response
true_response = rep(0,N)
for(i in 1:N){
  true_response[i] = mypotout[i,whichPotOut[i]]
}

#response with error
response_with_error = rep(0,N)
for(i in 1:N){
```

```
    response_with_error[i] = mypotout[i,whichPotOut_with_error[i]]
  }

  nearneighdf = as.data.frame(t(nearneigh))
  names(nearneighdf) = c("NN1", "NN2", "NN3", "UnitID")

  nearneighmisdf = as.data.frame(t(nearneighmis))
  names(nearneighmisdf) = c("NN1Err", "NN2Err", "NN3Err", "UnitIDErr")

  trtstatdf = as.data.frame(t(trtstat))
  names(trtstatdf) = c("NN1Trt", "NN2Trt","NN3Trt", "UnitTrt" )

  trtstat_misdf = as.data.frame(t(trtstat_mis))
  names(trtstat_misdf) = c("NN1ErrTrt", "NN2ErrTrt","NN3ErrTrt", "UnitErrTrt" )


  #generated data
  mydata=data.frame(mycov,nearneighdf,response_with_error, nearneighmisdf, hypcomprand,

  mydata
}


#Create a list of all possible exposures:
#Exposures
Exposures = list(c(0,0,0,0), c(1,0,0,0), c(0,1,0,0), c(1,1,0,0), c(0,0,1,0),c(1,0,1,0),c

#function for calculating Exact Marginal Probabilities
generate_pi=function(N,K, Exposures){pi = sapply(Exposures,function(x){N.kt = sum(x)
```

```
N.t=N/2
pi = choose(N-K-1,N.t-N.kt) / choose(N,N.t)})
pi}


#function to compute Horvitz-Thompson Estimator
#Use number 1-16 for exposure, using the exposure list.
HT_estimator<- function(exposure, pi, thedata){
  whichexp = which(thedata$whichPotOutmat == exposure)
  (1/nrow(thedata))*sum(thedata[whichexp,8]/pi[exposure])
}

#Horvitz-Thompson estimates distribution for different sample sizes
#Initialize variable

A = rep(0, 100)
B = rep(0, 100)
C = rep(0, 100)
D = rep(0, 100)
E = rep(0, 100)
G = rep(0, 100)
H = rep(0, 100)
I = rep(0, 100)
J = rep(0, 100)
K = rep(0, 100)
L = rep(0, 100)
M = rep(0, 100)
N_0 = rep(0, 100)
O = rep(0, 100)
```

```
P = rep(0, 100)
Q_1 = rep(0, 100)
R_1 = rep(0, 100)
S_1 = rep(0, 100)
U_1 = rep(0, 100)
V_1 = rep(0, 100)
Q_2 = rep(0, 100)
R_2 = rep(0, 100)
S_2 = rep(0, 100)
U_2 = rep(0, 100)
V_2 = rep(0, 100)
Q_3 = rep(0, 100)
R_3 = rep(0, 100)
S_3 = rep(0, 100)
U_3 = rep(0, 100)
V_3 = rep(0, 100)

coeff=matrix(c(0,0,0,0,  0,0,0,1, 0,0,0,4, 2,1,0.5,0, 2,1,0.5,1,
               2,1,0.5,4, 3,2,1,0, 3,2,1,1, 3,2,1,4), nrow=9,ncol=4, byrow  = TRUE)
for(i in 1:100){
  #generate data for different sample sizes
  data_N_100=generate_data(N=100, K=3, U=0, V= 0.05, w=5, NN.Coeff=coeff)
  data_N_200=generate_data(N=200, K=3, U=0, V= 0.05, w=5, NN.Coeff=coeff)
  data_N_300=generate_data(N=300, K=3, U=0, V= 0.05, w=5, NN.Coeff=coeff)
  data_N_400=generate_data(N=400, K=3, U=0, V= 0.05, w=5, NN.Coeff=coeff)
  data_N_500=generate_data(N=500, K=3, U=0, V= 0.05, w=5, NN.Coeff=coeff)

  #generate exact marginal probabilities for different sample sizes
  pi_N_100=generate_pi(N=100, K=3, Exposures)
```

```
pi_N_200=generate_pi(N=200, K=3, Exposures)
pi_N_300=generate_pi(N=300, K=3, Exposures)
pi_N_400=generate_pi(N=400, K=3, Exposures)
pi_N_500=generate_pi(N=500, K=3, Exposures)

#HT
A[i] = HT_estimator(exposure = 12, pi_N_100,data_N_100)
B[i] = HT_estimator(exposure = 12, pi_N_200, data_N_200)
C[i] = HT_estimator(exposure = 12, pi_N_300, data_N_300)
D[i] = HT_estimator(exposure = 12, pi_N_400, data_N_400)
E[i] = HT_estimator(exposure = 12, pi_N_500, data_N_500)
#ADE
G[i] = HT_estimator(exposure = 16, pi_N_100,data_N_100)-HT_estimator(exposure = 8, pi_
H[i] = HT_estimator(exposure = 16, pi_N_200, data_N_200)-HT_estimator(exposure = 8, pi
I[i] = HT_estimator(exposure = 16, pi_N_300, data_N_300)-HT_estimator(exposure = 8, pi
J[i] = HT_estimator(exposure = 16, pi_N_400, data_N_400)-HT_estimator(exposure = 8, pi
K[i] = HT_estimator(exposure = 16, pi_N_500, data_N_500)-HT_estimator(exposure = 8, pi
#AIE
L[i] = HT_estimator(exposure = 8, pi_N_100,data_N_100)-HT_estimator(exposure = 1, pi_N
M[i] = HT_estimator(exposure = 8, pi_N_200, data_N_200)-HT_estimator(exposure = 1, pi_
N_0[i] = HT_estimator(exposure = 8, pi_N_300, data_N_300)-HT_estimator(exposure = 1, p
O[i] = HT_estimator(exposure = 8, pi_N_400, data_N_400)-HT_estimator(exposure = 1, pi_
P[i] = HT_estimator(exposure = 8, pi_N_500, data_N_500)-HT_estimator(exposure = 1, pi_
#1st AlNNE
Q_1[i] = HT_estimator(exposure = 2, pi_N_100,data_N_100)-HT_estimator(exposure = 1, pi
R_1[i] = HT_estimator(exposure = 2, pi_N_200, data_N_200)-HT_estimator(exposure = 1, p
S_1[i] = HT_estimator(exposure = 2, pi_N_300, data_N_300)-HT_estimator(exposure = 1, p
U_1[i] = HT_estimator(exposure = 2, pi_N_400, data_N_400)-HT_estimator(exposure = 1, p
V_1[i] = HT_estimator(exposure = 2, pi_N_500, data_N_500)-HT_estimator(exposure = 1, p
```

```
  #2nd AlNNE
  Q_2[i] = HT_estimator(exposure = 4, pi_N_100,data_N_100)-HT_estimator(exposure = 2, pi
  R_2[i] = HT_estimator(exposure = 4, pi_N_200, data_N_200)-HT_estimator(exposure = 2, p
  S_2[i] = HT_estimator(exposure = 4, pi_N_300, data_N_300)-HT_estimator(exposure = 2, p
  U_2[i] = HT_estimator(exposure = 4, pi_N_400, data_N_400)-HT_estimator(exposure = 2, p
  V_2[i] = HT_estimator(exposure = 4, pi_N_500, data_N_500)-HT_estimator(exposure = 2, p
  #3rd AlNNE
  Q_3[i] = HT_estimator(exposure = 8, pi_N_100,data_N_100)-HT_estimator(exposure = 4, pi
  R_3[i] = HT_estimator(exposure = 8, pi_N_200, data_N_200)-HT_estimator(exposure = 4, p
  S_3[i] = HT_estimator(exposure = 8, pi_N_300, data_N_300)-HT_estimator(exposure = 4, p
  U_3[i] = HT_estimator(exposure = 8, pi_N_400, data_N_400)-HT_estimator(exposure = 4, p
  V_3[i] = HT_estimator(exposure = 8, pi_N_500, data_N_500)-HT_estimator(exposure = 4, p

}

#boxplotes of hoevitz-thompson estimates
#plot_data_HT<-cbind('100'=A,'200'=B, '300'=C, '400'=D, '500'=E)
#boxplot(as.data.frame(plot_data_HT), xlab='sample size', main='Boxplots for Distributio
#abline(h=0, col='red', lwd=1, lty=2)

plot_data_ADE<-cbind('100'=G,'200'=H, '300'=I, '400'=J, '500'=K)
boxplot(as.data.frame(plot_data_ADE), xlab='Sample size', main='Boxplots for Distributio
abline(h=1, col='red', lwd=1, lty=2)

plot_data_AIE<-cbind('100'=L,'200'=M, '300'=N_0, '400'=O, '500'=P)
boxplot(as.data.frame(plot_data_AIE), xlab='Sample size', main='Boxplots for Distributio
abline(h=3.5, col='red', lwd=1, lty=2)

plot_data_1st_AlNNE<-cbind('100'=Q_1,'200'=R_1, '300'=S_1, '400'=U_1, '500'=V_1)
```

```
boxplot(as.data.frame(plot_data_1st_AlNNE), xlab='Sample size', main='Boxplots for Distr
abline(h=2, col='red', lwd=1, lty=2)

plot_data_2nd_AlNNE<-cbind('100'=Q_2,'200'=R_2, '300'=S_2, '400'=U_2, '500'=V_2)
boxplot(as.data.frame(plot_data_2nd_AlNNE), xlab='Sample size', main='Boxplots for Distr
abline(h=1, col='red', lwd=1, lty=2)

plot_data_3rd_AlNNE<-cbind('100'=Q_3,'200'=R_3, '300'=S_3, '400'=U_3, '500'=V_3)
boxplot(as.data.frame(plot_data_3rd_AlNNE), xlab='Sample size', main='Boxplots for Distr
abline(h=0.5, col='red', lwd=1, lty=2)
```