Master’s thesis

Reducing Bias in A/B Testing on Social Network Services

（ソーシャルネットワークサービス上のＡ／Ｂテストにおけるバイアス削減手法）

February 6, 2018

Supervisor
Associate Professor Masashi Toyoda

Department of Information and Communication Engineering
Graduate School of Information Science and Technology
The University of Tokyo

48-166462 Jian CHEN

Abstract

A/B testing is the most widely used method to estimate the effect of an intervention. For example, it is often used to estimate the effect of a new drug on a certain disease. It achieves high estimation accuracy when SUTVA (Stable Unit Treatment Value Assumption) holds. SUTVA states that the outcome of a unit depends only on its own, and is not affected by other units. However, in social network, users often interact with each other and the outcome of one user may be interfered by other users, resulting in the decrease of estimation accuracy. In our research, we propose various methods to reduce the estimation bias. To this end, we first propose a new graph partitioning method, which is of great importance for reducing the interference between the treatment group and control group. Since existing methods tend to underestimate the effect, we also propose methods that try to correct the bias. We do the bias correction in two ways, one of which is to make the most of the structure of the network, and the other is to assume the outcome function and then estimate its parameters, by which the effect can be further estimated.

Keyword: A/B testing, social network, bias correction, network effects

Acknowledgements

I would like to thank all the members in Kitsuregawa,Toyoda,Nemoto,Yoshinaga Lab. I am especially grateful to my advisor Associate Professor Masashi Toyoda, who accepted me as a research student, making it possible for me to enter the graduate school, and always give me a lot of valuable advice on my research. I am also grateful to Professor Masaru Kitsuregawa, who provides the top level research environment that benefits me a lot. I would like to thank Associate Professor Naoki Yoshinaga. In addition to the kind help on my research, he also traveled with me to many beautiful places, and introduced me Japanese culture. I would also like to thank Research Associate Junpei Komiyama and Specially Appointed Research Associate Daisuke Yokoyama for their precious advice. My sincere thanks also goes to my fellow students. The doctoral students, Shonosuke Ishiwatari, Hongshan Jin, Shoetsu Sato, helped me a lot since I joined the Lab. Kohei Ohara and Satoshi Akasaki, who are also master's students as me, always kindly remind me of important information about the courses and the Rinko.

I would also like to thank the foreign students who entered the graduate school at the same time with me and also my good friends. They helped me a lot even outside the school. Special thanks to Hasegawa International Scholarship Foundation for providing me the scholarship, which enabled me to be more concentrated on my study.

Finally, I must express my very profound gratitude to my parents and to my brother, without whom the accomplishment would not have been possible.

Thanks for all your encouragement! February 6, 2018

Contents

List of Figures

List of Tables

Chapter 1

Introduction to A/B Testing

A/B testing, sometimes called randomized experimentation, lies at the core of causal inference. Causal inference aims to estimate the effect of a certain intervention (treatment). The standard statistical analysis, such as regression, hypothesis testing, and interval estimation, essentially aims to estimate the parameters of a distribution from sample data. Then predictions can be made using these estimated parameters. However, these standard statistical methods can only work on static data, which means they may not be able to predict the outcome when there exist some interventions [1]. For example, since the weight and height of a person are associated and a regression model can be trained to predict one's height given the weight. However, if we want to know the answer that will one's height increase if his/her weight increases, the regression model cannot be applied, since weight is not the cause of height despite that they are associated. The concept of causality is closely related to that of association, but they are essentially different concepts. Causal inference is the research area dealing with the causality.

1.1 A/B Testing without Interference

Traditional A/B testing usually assumes that there are no interference among experiment units. This assumption is plausible in many cases and makes the estimation be simple. In this section, we first introduce the methods for A/B testing when the interference among units does not present, and in the next section we will consider the case that the interference among units presents.

1.1.1 Neyman–Rubin Causal Model and SUTVA

Suppose that we want to estimate the effect of a new medicine for a certain disease. The ideal way is to recruit some patients as volunteers. For each of them if we can obtain both the outcome under treatment Y^1 and outcome under control Y^0, then the *individual treatment effect* (ITE) for unit i is $Y^1(i) - Y^0(i)$. The average causal effect (ATE) is the average of the individual treatment effect over the experiment units, and is written as:

$$\delta = \frac{1}{N} \sum_{i=1}^{N} [Y^1(i) - Y^0(i)] \tag{1.1}$$

where δ denotes the ATE. Since the difference of the outcome is caused only by the new medicine, this result is the causal effect for this new medicine. TABLE 1.1 shows an example of the outcomes for this experiment. The ATE in this case is 1/4, which indicates the new medicine actually has effect on the disease[1].

However, in reality, it is impossible to know both the outcome under treatment and the outcome under control. Obtaining both of the outcomes is just like we are conducting the experiment in two "parallel universes". The outcomes of real experiments are like what is shown in TABLE 1.2. We can only obtain one kind of the outcome, either the outcome under treatment or the outcome under control. The other is unobservable and is called *potential outcome*. The Neyman–Rubin

[1] In fact, in order to draw this conclusion, we should have enough experiment units. But in this case, for illustration purpose, we only use 8 experiment units.

patient	Y^0	Y^1
A	0	1
B	0	1
C	0	0
D	0	1
E	1	0
F	0	1
G	1	0
H	1	1

Table 1.1: "Ideal" Randomized Experiment

patient	Y^0	Y^1
A	?	1
B	0	?
C	?	0
D	0	?
E	1	?
F	0	?
G	?	0
H	?	1

Table 1.2: Randomized Experiment in Reality

causal model is based on the framework of potential outcomes, and we will explain it in more detail later.

Given the reason we mentioned above, the ITE is impossible to obtain because it requires both the outcome under treatment and the outcome under control. Since the calculation of the ATE also depends on the ITE as shown in EQUATION 1.1, the ATE is also impossible to obtain. Instead, we need a method to estimate the ATE.

To make the estimation simple, there is an extremely important assumption for A/B testing. It is called *Stable Unit Treatment Value Assumption* (SUTVA), which states that one unit in the experiment cannot interfere with another unit with regard to the outcome, and the outcome of each unit only depends on its own assignment and has nothing to do with other units' assignments. This assumption is quite reasonable

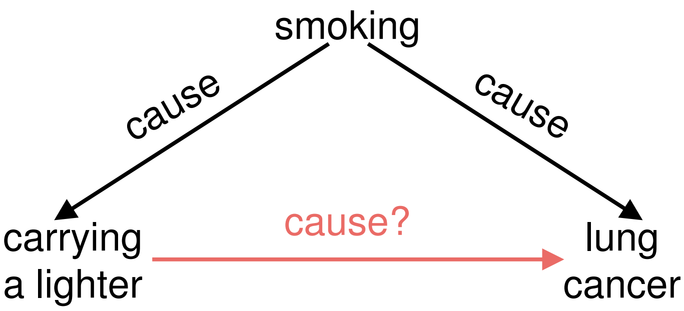


Figure 1.1: Causal graph

in many occasions. For example, for many diseases that are non contagious, the condition of a patient depends on his/her own treatments and won't be interfered by other patients. In this case, the SUTVA holds. In this section, we only discuss the estimation methods when SUTVA holds.

Neyman-Rubin causal model uses randomization to alleviate the problem caused by the absence of the potential outcomes. To explain the reason why randomization is important, we take another example. Suppose we need to investigate that if carrying a lighter in the pocket often can cause lung caner. We can conduct an experiment in the following way.

1. Recruiting some volunteers who often carry a lighter (termed group A) and some volunteers who do not often carry a lighter (termed group B).
2. Taking the ratio of volunteer who have lung cancer in group A as the probability of getting lung cancer if often carrying a lighter (denoting as P_1), and taking the ratio of volunteer who have lung cancer in group B as the probability of getting lung cancer if not often carrying a lighter (denoting as P_2).
3. Comparing P_1 with P_2.

The medical knowledge tells us that carrying a lighter cannot cause lung cancer[2]. However, the experiment designed above will show us that carrying a lighter has

[2] We did not try to find the evidence to support this statement, but it is enough to suppose it is true for our following analysis.

a higher probability to get lung cancer than not doing so. The reason is as the following. People who smoke often carrying a lighter, and they are also more likely to get lung cancer than people who do not smoke. So people carrying a lighter have a higher probability to be a smoker than nonsmoker, and thus have a higher probability to get lung cancer. But as we have mentioned above, carrying a lighter is not the cause of lung cancer. The reason for this counterintuitive example is the common cause. As shown in FIGURE 1.1, smoking is both the cause of carrying a lighter and the cause of lung cancer, and it makes people carrying a lighter have a higher probability to get lung cancer, while in fact they do not have a causal relationship.

Randomization can solve the problem cased by the common cause. We design a new experiment, making use of randomization.

1. Recruiting some volunteers who do not have lung cancer, and randomly assigning them to group A or group B.
2. Volunteers in group A are told to carry a lighter every day, while volunteers in group B are kept from carrying a lighter.
3. In the end of the experiment, taking the ratio of volunteer who get lung cancer in group A as the probability of getting lung cancer if carrying a lighter (denoting as P_1), and taking the ratio of volunteer who get lung cancer in group B as the probability of getting lung cancer if not carrying a lighter (denoting as P_2).
4. Comparing P_1 with P_2.

With randomization, volunteers in group A and group B have the same probability to be either a smoker or a nonsmoker. So the probability of getting lung cancer will also be the same, which indicates carrying a lighter cannot cause lung cancer and this result is what we expected. This is achieved by randomization, which eliminates the difference of common cause (smoking) in the two groups.

1.1.2 Estimation with Uniform Sampling

Using randomization, the assignment of an experiment unit is randomized decided. The most simple way to do this is to let the assignment $Z \sim \text{Bernoulli}(0.5)$, that is, to treat or to control a unit with the same probability. Then the *treatment group* is the group of users who are treated, denoting as $T = \{i \mid Z(i) = 1\}$, and the *control group* is the group of users who are controlled, denoting as $C = \{i \mid Z(i) = 0\}$. The ATE can thus be estimated as

$$\hat{\delta} = \frac{1}{n_t} \sum_{i \in T} Y^1(i) - \frac{1}{n_c} \sum_{i \in C} Y^0(i) \tag{1.2}$$

where n_t and n_c is the number of units in treatment group and control group respectively. This estimator is often called *difference-in-means estimator*.

The difference-in-means estimator is an unbiased estimator for the ATE. To prove the this, we first calculate the expected value of the left term in EQUATION 1.2.

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{n_t} \sum_{i \in T} Y^1(i)\right] &= \frac{1}{n_t} \mathbb{E}\left[\sum_{i \in T} Y^1(i)\right] \\
&= \frac{1}{n_t} n_t \bar{Y^1} \\
&= \bar{Y^1} \\
&= \frac{1}{N} \sum_{i=1}^{N} Y^1(i)
\end{aligned} \tag{1.3}$$

where $\bar{Y^1}$ is the mean value of Y^1 for all units and is unobservable.

we also calculate the expected value of the right term in EQUATION 1.2.

$$\begin{aligned}\mathbb{E}\left[\frac{1}{n_c}\sum_{i\in T}Y^0(i)\right] &= \frac{1}{n_c}\mathbb{E}\left[\sum_{i\in T}Y^0(i)\right] \\ &= \frac{1}{n_c}n_c\bar{Y^0} \\ &= \bar{Y^0} \\ &= \frac{1}{N}\sum_{i=1}^{N}Y^0(i)\end{aligned} \tag{1.4}$$

where $\bar{Y^0}$ is the mean value of Y^0 for all units and is unobservable.

Combining EQUATION 1.2 $\sim$ 1.4, we have

$$\begin{aligned}\mathbb{E}[\hat{\delta}] &= \mathbb{E}\left[\frac{1}{n_t}\sum_{i\in T}Y^1(i) - \frac{1}{n_c}\sum_{i\in C}Y^0(i)\right] \\ &= \mathbb{E}\left[\frac{1}{n_t}\sum_{i\in T}Y^1(i)\right] - \left[\frac{1}{n_c}\sum_{i\in C}Y^0(i)\right] \\ &= \frac{1}{N}\sum_{i=1}^{N}Y^1(i) - \frac{1}{N}\sum_{i=1}^{N}Y^0(i) \\ &= \delta\end{aligned} \tag{1.5}$$

Therefore, difference-in-means estimator with uniform sampling is a unbiased estimator for the ATE.

1.1.3 Estimation with Cluster Randomized Sampling

Using cluster randomized sampling, we first divide all the units into M clusters, $C_1, C_2, \ldots, C_M$, and then randomly assign treatment or control on cluster level. We denote the assignment of a cluster as W, $W \sim$ Bernoulli(0.5) and $\forall i \in C_j$, $Z(i) = W(j)$.

Although all units have the same probability of being treated or controlled, the difference-in-means estimator using cluster randomized sampling is no longer unbiased. We derive the bias in the remaining of this subsection following [2]. The difference-in-means estimator with cluster randomized sampling is written as

$$\hat{\delta} = \frac{\displaystyle\sum_{j\in C^1}\sum_{i=1}^{n_j} Y^1(ij)}{\displaystyle\sum_{j\in C^1} n_j} - \frac{\displaystyle\sum_{j\in C^0}\sum_{i=1}^{n_j} Y^0(ij)}{\displaystyle\sum_{j\in C^0} n_j} \tag{1.6}$$

where $Y(ij)$ is the outcome of the unit j in cluster i, n_j is the number of units in cluster j, C^0 and C^1 are the set of treated clusters and the set of controlled clusters respectively, and $|C^0| = m_c$, $|C^1| = m_t$. In this equation, $n_t = \sum_{j\in C^1} n_j$ and $n_c = \sum_{j\in C^0} n_j$ depend on the size of the clusters, and thus are random variables, while in EQUATION 1.2, n_t and n_c are fixed numbers. In general, for two random variables U and V $(V > 0)$, we have

$$\mathbb{E}[\frac{U}{V}] = \frac{1}{\mathbb{E}[V]}\Big[\mathbb{E}[U] - \mathrm{Cov}(\frac{U}{V}, V)\Big] \tag{1.7}$$

So the expected value of the estimator in EQUATION 1.6 is

$$
\begin{aligned}
\mathbb{E}[\hat{\delta}] &= \mathbb{E}\left[\frac{\sum\limits_{j\in C^1}\sum\limits_{i=1}^{n_j} Y^1(ij)}{\sum\limits_{j\in C^1} n_j}\right] - \left[\frac{\sum\limits_{j\in C^0}\sum\limits_{i=1}^{n_j} Y^0(ij)}{\sum\limits_{j\in C^0} n_j}\right] \\
&= \frac{1}{Nm_t/M}\left[\frac{\bar{Y^1} N m_t}{M} - \mathrm{Cov}\left(\frac{\sum\limits_{j\in C^1}\sum\limits_{i=1}^{n_j} Y^1(ij)}{\sum\limits_{j\in C^1} n_j}, \sum_{j\in C^1} n_j\right)\right] \\
&\quad - \frac{1}{Nm_c/M}\left[\frac{\bar{Y^0} N m_c}{M} - \mathrm{Cov}\left(\frac{\sum\limits_{j\in C^0}\sum\limits_{i=1}^{n_j} Y^0(ij)}{\sum\limits_{j\in C^0} n_j}, \sum_{j\in C^0} n_j\right)\right] \qquad (1.8) \\
&= (\bar{Y^1} - \bar{Y^0}) - \frac{M}{N}\left[\frac{1}{m_t}\mathrm{Cov}\left(\frac{\sum\limits_{j\in C^1}\sum\limits_{i=1}^{n_j} Y^1(ij)}{\sum\limits_{j\in C^1} n_j}, \sum_{j\in C^1} n_j\right)\right. \\
&\quad \left. - \frac{1}{m_c}\mathrm{Cov}\left(\frac{\sum\limits_{j\in C^0}\sum\limits_{i=1}^{n_j} Y^0(ij)}{\sum\limits_{j\in C^0} n_j}, \sum_{j\in C^0} n_j\right)\right]
\end{aligned}
$$

Since $\bar{Y^1} - \bar{Y^0} = \frac{1}{N}\sum_{i=1}^{N} Y^1(i) - \frac{1}{N}\sum_{i=1}^{N} Y^0(i) = \delta$, the bias of $\hat{\delta}$ is

$$
\hat{\delta} - \delta = \frac{M}{N}\left[\frac{1}{m_t}\mathrm{Cov}\left(\frac{\sum\limits_{j\in C^1}\sum\limits_{i=1}^{n_j} Y^1(ij)}{\sum\limits_{j\in C^1} n_j}, \sum_{j\in C^1} n_j\right) - \frac{1}{m_c}\mathrm{Cov}\left(\frac{\sum\limits_{j\in C^0}\sum\limits_{i=1}^{n_j} Y^0(ij)}{\sum\limits_{j\in C^0} n_j}, \sum_{j\in C^0} n_j\right)\right] \qquad (1.9)
$$

According to our analysis above, when cluster randomized sampling is used, the

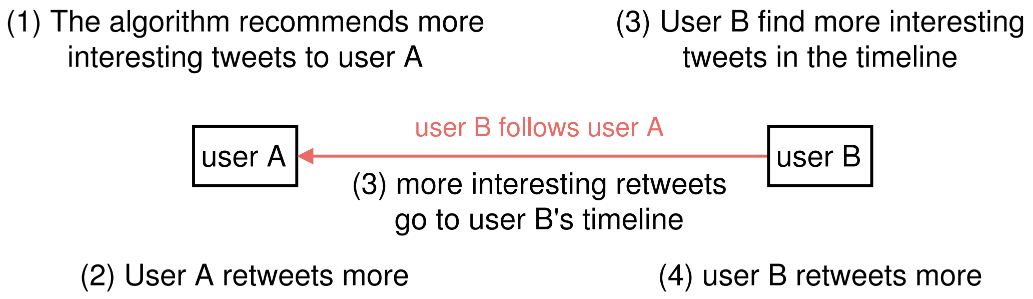

(1) The algorithm recommends more interesting tweets to user A
(3) User B find more interesting tweets in the timeline
user A
user B follows user A
user B
(3) more interesting retweets go to user B's timeline
(2) User A retweets more
(4) user B retweets more

FIGURE 1.2: Illustration for the example of recommend algorithm.

difference-in-means estimator is not unbiased and the bias is expressed in EQUATION 1.9. In section 2.3.2, we will discuss the Horvitz-Thompson estimator, which is an unbiased estimator for cluster randomized sampling when SUTVA holds.

1.2 A/B Testing with Interference

In many cases, SUTVA can hold as we discussed above. But there are also many cases that it is unreasonable to assume SUTVA, especially when we conduct A/B testing experiment in social network. For example, if we developed a recommendation algorithm that recommends interesting tweets to each user, and the outcome we are interested is the number of retweets of a user, then users are very likely to interfere with each other. To gain a better insight into this example, we assume the recommendation algorithm is indeed effective and users who are treated will retweet more. Then the users who follow the treated users can also find more interesting tweets in their timelines. As a result, their number of retweets will also increase. This example is illustrated in FIGURE 1.2. As ATE is the difference of the outcomes between the treatment group and control group, when the outcome of controlled units increase due to the interference, the ATE will be underestimated.

In this thesis, we mainly deal with the estimation for the ATE in A/B testing when interference is presented, and in this section, we first introduce some useful concepts.

1.2.1 Network Effects

Network effects are the effects on a unit that received from other units in the network. Depending on the context, network effects are also called “peer effects”, “spillover effects”, “social effects” and etc. This kind of effects is commonly observed in many the social and economical phenomena. For example, an individual’s demand for a product is influenced by other individual’s demand in the market [3].

The network effects are even more common in social network services (SNSs), such as Twitter, Facebook, and Instgram. In these SNSs, users can share contents freely, and those contents will then appear on the timeline of their friends or followers. Like the example of recommend algorithm we mentioned previously, one user’s behavior may be influenced by other users. This kind of effects can also propagate through the social network. That is, if user A is influenced by user B, and user B is influenced by user C, then user A is indirectly influenced by user C, which indicates that the influence propagates from user C to user A through user B. When the social network is large, the network effects are significant.

1.2.2 Outcome Function

In an A/B testing experiment, we need outcomes to estimate the ATE using estimator such as difference-in-means estimator. The whole process is like the following:

(1) sampling; (2) carrying out the experiment (treating and control corresponding units) and collecting the outcomes in the end of the experiment; (3) estimating the

1. Sampling: randomly assigning each experiment unit to either treatment group or control group.
2. Collecting outcomes: carrying out the experiment (applying treatment to the treatment group and controlling the units in the control group), and collecting the outcomes in the end of the experiment.

Representation	Example	Meaning
uppercase normal letter	X	random variable
uppercase bold letter	$\mathbf{X}$	random vector or matrix[3]
lowercase bold letter	$\mathbf{x}$	vector
lowercase normal letter	x	scalar

TABLE 1.3: Notations of Different Types of Symbols

3. Estimating: estimating the ATE.

The outcomes can always be observed in this case, and once the outcomes are available, the estimation can be made.

However, even through the outcomes are observable, we can only obtain the estimated ATE like what expressed in EQUATION 1.2, while the true ATE which expressed in EQUATION 1.1 is impossible to be obtained. In consequence, we do not have a ground truth to compare with.

This problem can be alleviated by using a synthetic outcome function. The outcomes can be expressed $\mathbf{Y}(\mathbf{Z}) = f(\mathbf{Z})$, where $\mathbf{Y}$ is a random vector and $\mathbf{Y}_i$ is the outcome of unit i, $\mathbf{Z}$ is also a random vector and $\mathbf{Z}_i$ is the assignment of unit i. For clarification purpose, we list the notations frequently used in this thesis in TABLE 1.3. Using a synthetic outcome function, the true ATE can be obtained as $\mathbf{Y}(\mathbf{Z} = \mathbf{1}) - \mathbf{Y}(\mathbf{Z} = \mathbf{0})$, and the outcomes under the experiment assignments $\mathbf{z}$ are $\mathbf{Y}(\mathbf{Z} = \mathbf{z})$.

Some design principles for outcome functions are discussed in [4]. In this thesis, we use the *linear-in-means model* [3][5], which is a model usually used to capture the interaction of social and economic phenomenon. The linear-in-means model can be written as the following Equation in matrix form.

$$\begin{aligned} \mathbf{Y}^{t*} &= \alpha + \lambda_1 \mathbf{Z} + \lambda_2 \frac{\mathbf{A}\mathbf{Y}^{t-1}}{\mathbf{D}} + \mathbf{U}^t \\ \mathbf{Y}^t &= g\big(\mathbf{Y}^{t*}\big) \end{aligned} \tag{1.10}$$

[3] This can be differentiated based on the context.

where $\mathbf{Y}$ is the outcome vector[4], α is a baseline value, λ_1 is the direct treatment effect, λ_2 is the network effect, $\mathbf{Z}$ is the assignment vector, $\mathbf{A}$ is the adjacency matrix (a binary matrix), $\mathbf{D}$ is diagonal matrix and $\mathbf{D}_{ii}$ is the degree (out degree for directed network) of unit i, $\mathbf{U}$ is a vector representing user specific traits and $\forall i \in [1, N]$, $\mathbf{U}_i \sim \mathcal{N}(0, 1)$.

The outcome is generated by running this model iteratively until the mean of $\mathbf{Y}$ converges, and $\mathbf{Y}$ is initialized as zero vector. The superscript 't' in Equation 1.10 is the iteration step. Hence, $\mathbf{Y}^{t=0} = \mathbf{0}$ according to our initialization.

To make Equation 1.10 more clear, we can rewrite it as the following Equation.

$$\begin{aligned} \mathbf{Y}_i^{t*} &= \alpha + \lambda_1 \mathbf{Z}_i + \lambda_2 \frac{1}{\mathbf{D}_{ii}} \sum_{\{j;\mathbf{A}_{ij}=1\}} \mathbf{Y}_j^{t-1} + \mathbf{U}_i^t \\ \mathbf{Y}^t &= g\big(\mathbf{Y}^{t*}\big) \end{aligned} \tag{1.11}$$

Here we can see the outcome $\mathbf{Y}$ is summed up by 4 components.

- α: α is the baseline value which is a constant here. It simply means that even if there is no treatment, the outcome may still be non-zero. For example, if the outcome is the number of retweets, it is non-zero even if a new feature is not added.

- $\lambda_1 \mathbf{Z}_i$: $\mathbf{Z}_i = 1$ if user i is treated, and $\mathbf{Z}_i = 0$ if controlled. Therefore, the outcome of a user will increase by λ_1 if it's treated compared with the case that it's controlled. So we call λ_1 *direct treatment effect.*

- $\lambda_2 \frac{1}{\mathbf{D}_{ii}} \sum_{\{j;\mathbf{A}_{ij}=1\}} \mathbf{Y}_j^{t-1}$: this component is the average outcome of user i's neighbors [5] at the previous iteration step multiplied by a coefficient λ_2. λ_2 is the *network effect.* A large λ_2 indicates the outcome of a user is interfered more by the neighbors, a small λ_2 indicates the outcome of a user is interfered less

[4] Vectors are represented as *column vectors*, unless otherwise stated.

[5] In directed graph, we use neighbors to mean the successors (nodes pointed to by directed edges from a starting node)

by the neighbors. In particular, when $\lambda_2 = 0$ the outcome of a user does not depend on other users, which is equal to SUTVA.

- $\mathbf{U}_i$: Since every user is reasonable to respond differently to the treatment due to some user specific traits, like the age, personality, occupation, etc., we use a Gaussian random variable to capture these traits.

1.2.3 Problems Caused by Network Effects

When the network effects are presented, the notation of ATE in EQUATION 1.1 also need to be changed. First the ITE (individual treatment effect) of unit i is $\mathbf{Y}_i(\mathbf{Z} = \mathbf{1}) - \mathbf{Y}_i(\mathbf{Z} = \mathbf{0})$. Note that $\mathbf{Z} = \mathbf{1}$ indicates all units are treated and $\mathbf{Z} = \mathbf{0}$ indicates all units are controlled. Since units can interfere with each other, the ITE is the difference of user i's outcomes between the case that all users are treated and the case that all users are controlled. Then the ATE is expressed as

$$\delta = \frac{1}{N}\mathbf{1}^{\intercal}[\mathbf{Y}(\mathbf{Z} = \mathbf{1}) - \mathbf{Y}(\mathbf{Z} = \mathbf{0})] \tag{1.12}$$

which is the average ITE over all units.

We discussed the estimation of ATE when SUTVA holds in the previous section. In that case, the difference-in-means estimator with uniform sampling is an unbiased estimator. But when there exist network effects, the estimator is biased. We give an example to explain the reason.

As shown in FIGURE 1.3(b), unit 2 follows unit 1 and unit 3, unit 3 follows unit 4, and unit 4 follows unit 2. Unit 1 and unit 2 are treated (marked as red), while unit 3 and unit 4 are controlled (marked as blue). So the assignment vector $\mathbf{z} = (1, 1, 0, 0)$. In FIGURE 1.3(a) all users are controlled and in FIGURE 1.3(c) all users are treated. From FIGURE 1.3(a) $\sim$ FIGURE 1.3(c), according to the outcome function in EQUATION 1.10 we have the following observations.

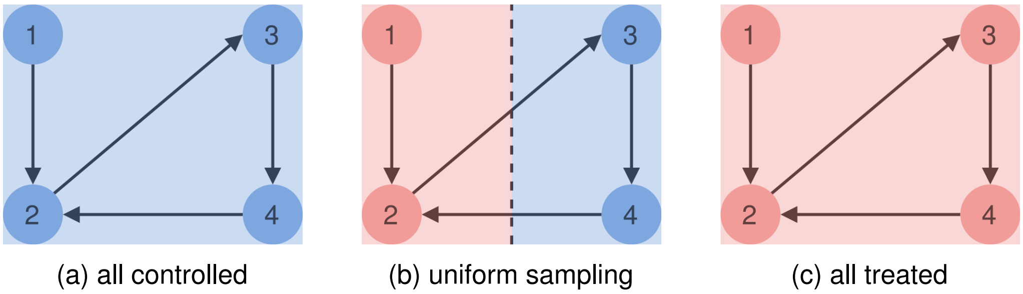


FIGURE 1.3: Illustration for The Problem with Network Effect

- In FIGURE 1.3(b), unit 4 follows unit 2 who is treated, so $\mathbf{Y}_{4,b} > \mathbf{Y}_{4,a}$, and thus $\mathbf{Y}_{3,b} > \mathbf{Y}_{3,a}$ because unit 3 follows unit 4. Therefore, the outcomes of units in the control group are larger than the outcomes of them when all users are controlled.

- In FIGURE 1.3(b), unit 2 follows unit 3 who is treated, so $\mathbf{Y}_{2,b} < \mathbf{Y}_{2,c}$, and thus $\mathbf{Y}_{1,b} < \mathbf{Y}_{1,c}$ because unit 1 follows unit 2. Therefore, the outcomes of units in the treatment group are smaller than the outcomes of them when all users are treated.

From the observations above, we have

$$\frac{1}{n_t} \sum_{\{i;\mathbf{Z}_i=1\}} \mathbf{Y}_i(\mathbf{Z}=\mathbf{z}) < \frac{1}{N} \sum_{i=1}^{N} \mathbf{Y}_i(\mathbf{Z}=\mathbf{1}) \tag{1.13}$$

$$\frac{1}{n_c} \sum_{\{i;\mathbf{Z}_i=0\}} \mathbf{Y}_i(\mathbf{Z}=\mathbf{z}) > \frac{1}{N} \sum_{i=1}^{N} \mathbf{Y}_i(\mathbf{Z}=\mathbf{0}) \tag{1.14}$$

Furthermore, we have the following result

$$\begin{aligned}\hat{\delta} &= \frac{1}{n_t}\sum_{\{i;\mathbf{Z}_i=1\}}\mathbf{Y}_i(\mathbf{Z}=\mathbf{z}) - \frac{1}{n_c}\sum_{\{i;\mathbf{Z}_i=0\}}\mathbf{Y}_i(\mathbf{Z}=\mathbf{z}) \\ &< N\sum_{i=1}^{N}\mathbf{Y}_i(\mathbf{Z}=\mathbf{1}) - N\sum_{i=1}^{N}\mathbf{Y}_i(\mathbf{Z}=\mathbf{0}) \\ &= \delta\end{aligned} \tag{1.15}$$

This result indicates the difference-in-means estimator always underestimates the ATE when linear-in-means outcome function is assumed.

1.3 Goal of Our Research

In the previous sections, we introduced A/B testing with and without interference, and discussed the problem that the difference-in-means estimator, which is an unbiased estimator combining with uniform sampling when no interference among units exists, tends to underestimate the ATE when the outcome function is linear-in-means model. So the goal of our research is to propose new estimation method to increase the estimation accuracy.

In Chapter 2 we introduce the related work about A/B testing in social network, and in Chapter 3 we introduce our proposed methods.

Chapter 2

Related Work

With the rise of social network services (SNSs), such as Twitter, Facebook and LinkedIn, more and more people are connected online. To improve the usability, many new features are continuously added to those SNSs. When adding a new feature, estimating the effect of the new feature is often necessary. If it has much positive effect, more features like this should be developed. If it has nearly no effect, some modifications are needed. And if it has noticeable negative effect, this kind of feature should be avoided. For some features, like adding a new button to the user interface, users usually do not affect each other, and the SUTVA holds. But for some other features, like adding an recommendation algorithm, which we mentioned in the previous chapter, make SUTVA not hold. Therefore, estimation methods taking the interference among units into consideration are of great importance to A/B testing in social network.

To estimate the average outcome when adding a new feature, which only takes effect when a user and at least d neighbors are treated, the problem called *Network Bucket Testing* is formulated and discussed in [6][7]. It differs from A/B testing in that its goal is to estimate the average outcome on a small portion of users before releasing the new feature, rather than to estimate the effect.

To reduce the estimation bias of ATE, the use of cluster randomized sampling was introduced [8][9][4], and some unbiased estimators are also proposed [2]. Although those unbiased estimators are based on cluster randomized sampling, they assume SUTVA. Therefore, they are not truly unbiased when there exist interferences among clusters. [9] also used bias correction to further reduce the estimation bias by assuming the outcome is a linear function of the assignment and the treated ratio of neighbors.

Other than the estimation of ATE, there are also some other work trying to estimate or test the existence of the network effect [10][11][12].

2.1 A/B Testing Process

The whole A/B testing process contains the following steps:

1. Sampling: randomly assigning each experiment unit to either treatment group or control group.
2. Collecting outcomes: carrying out the experiment (applying treatment to the treatment group and controlling the units in the control group), and collecting the outcomes in the end of the experiment.
3. Estimating: estimating the ATE.

In this section, we explain these steps in more detail.

2.1.1 Step 1: Sampling

Sampling is the process that deciding which unit to be treated and which unit to be controlled. In SECTION 1.1, we have introduced uniform sampling, which gives each unit the same probability of being treated and being controlled, and cluster randomized sampling, which partitions units into clusters and then uniformly samples on cluster level. We will talk more about the sampling methods in the next section.

Sampling methods also have close relationship with estimators. For different sampling methods, the probability of being treated and being controlled may be different, so to keep the estimation unbiased or to reduce the estimation bias, different estimator may be needed.

2.1.2 Step 2: Collecting Outcomes

In an A/B testing, the experiment is carried out by treating and controlling the corresponding units for a reasonable long period. For example, to estimate the effect of a new drug, when sampling is finished, units are assigned to treatment group and control group, and then units in the treatment group are treated using the new drug, while units in the control group are controlled. Depending on the actual case, this may be lasting for several days or even several years. In the end of the experiment, the outcomes, such as blood pressures and weights, can be collected. The outcomes are vital to the estimation of the ATE.

In SECTION 1.2 we also mentioned that although outcomes are observable, the ATE is unobservable, and thus a synthetic outcome function is needed. With a synthetic outcome function, not only the ATE can be obtained, but also the experimentation becomes simpler because in this way we do not need to carry out a real experiment and wait for a long time to collect the outcome data. In this thesis, we mainly focus on the outcome function expressed in EQUATION 1.10. With the assignment $\mathbf{Z} = \mathbf{z}$, the observed outcomes are $\mathbf{Y}(\mathbf{Z} = \mathbf{z})$. And the ATE is written as

$$\delta = \mathbf{Y}(\mathbf{Z} = \mathbf{1}) - \mathbf{Y}(\mathbf{Z} = \mathbf{0}) \tag{2.1}$$

2.1.3 Step 3: Estimating

Once the outcomes $\mathbf{Y}$ are available, the estimation can be made. We denote the estimated ATE as $\hat{\delta}$, and the estimation bias is thus $\hat{\delta} - \delta$. When interferences among units present, the choosing of sampling method and estimator is vital to the estimation. In other words, to reduce the estimation bias, we can try to change the sampling method or the estimator. In the remaining of this chapter, we introduce the sampling methods and estimators that used in related work.

2.2 Sampling Methods

In this section, we introduce 3 kinds of sampling methods, and explain the advantages of each of them.

2.2.1 Uniform Sampling

As we discussed in SECTION 1.1, the randomization is essential to Neyman–Rubin causal model for A/B testing, this is still the case when interferences among units present. Uniform sampling decides the assignment of each unit totally randomly by setting $\mathbf{Z}_i \sim \text{Bernoulli}(0.5)$.

When interferences present among units, this kind of full randomization brings much bias. For a treated unit, half of its neighbors are controlled, according to the outcome function in EQUATION 1.10, the outcome is smaller compared with the case that all its neighbors are treated. And likewise, the outcome of a controlled user is larger compared with the case that all its neighbors are controlled. Hence, a treated unit needs more treated neighbors and a controlled unit needs more controlled neighbors, and this is done by using cluster randomized sampling.

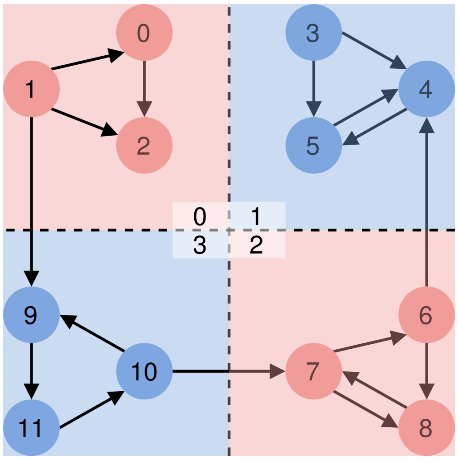


FIGURE 2.1: Illustration for Cluster Randomized Sampling

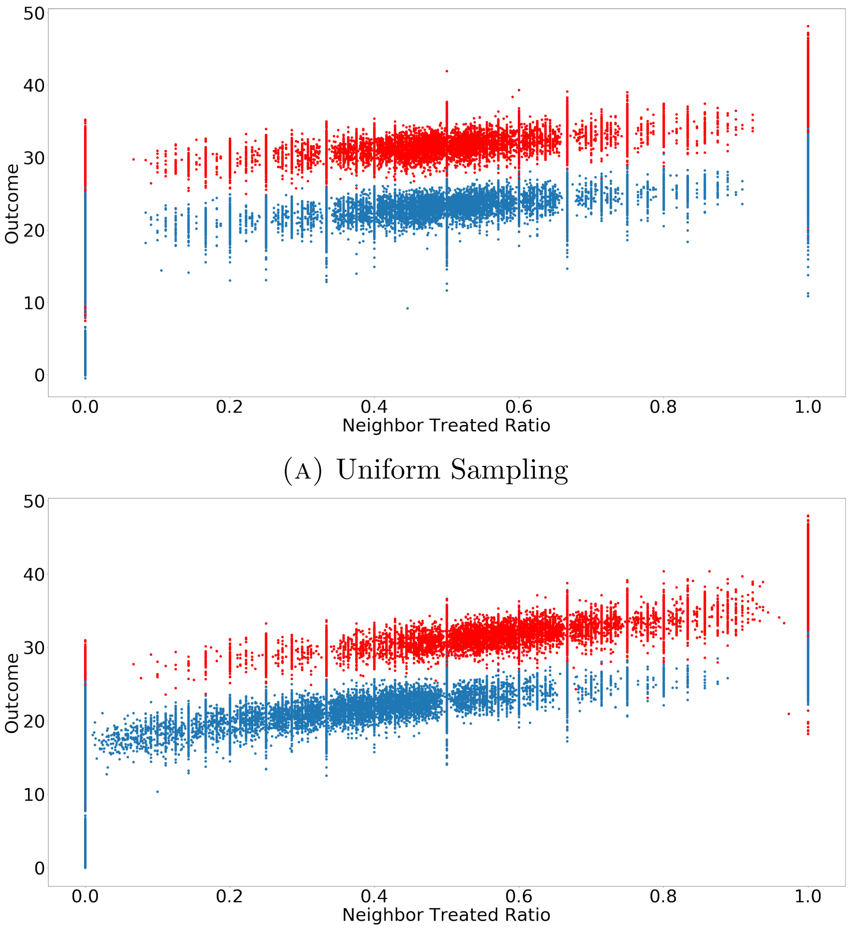
2.2.2 Cluster Randomized Sampling

As we mentioned above, although the sampling should be randomized, we need also let treated users, as well as controlled users, gathered as closely as possible. Cluster randomized sampling does this kind of trade-off.

We illustrate the cluster randomized sampling in FIGURE 2.1, in which there are 12 units and they are partitioned into 4 clusters. We use $\mathbf{W}$ to denote the assignment vector of the clusters, and since cluster 0 and cluster 2 (the top left and bottom right ones) are treated, and cluster 1 and cluster 3 (the top right and bottom left ones) are controlled, $\mathbf{W} = (1, 0, 1, 0)$ in this case. $\forall i \in C_j$, we let $\mathbf{Z}_i = \mathbf{W}_j$, where C_j is the jth cluster. So $\mathbf{Z} = (1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0)$ in this example.

Many ready-to-use graph partitioning or community detections algorithms [13][14] can be applied to partitioning the units for the cluster randomized sampling. In [8], the author also proposed a graph partitioning algorithm to reduce the estimation variance based on their analysis.

To show the difference between uniform sampling and cluster randomized sampling, we generated the sampling result using the linear-in-means outcome function and "soc-Slashdot0811" graph. The results are shown in FIGURE 2.2, where the



(A) Uniform Sampling

(B) Cluster Randomized Sampling

FIGURE 2.2: Outcome vs. neighbor treated portion plot using uniform sampling and cluster randomized sampling. Red points represent treated users and blue points represent controlled user. The data are generated using the linear-in-means outcome function in EQUATION 1.10 by setting $\lambda_0 = 3, \lambda_1 = 8, \lambda_2 = 0.6$. The graph data used is "soc-Slashdot0811" from [15].

outcome is plotted versus the *neighbor treated ratio*, which is the ratio of treated neighbors to all neighbors of a unit. We denote neighbor treated ratios as $\boldsymbol{\sigma}$, and define it as

$$\forall i \in [1, N], \quad \boldsymbol{\sigma}_i = \frac{\sum\limits_{\{j;\mathbf{A}_{ij}=1\}} \mathbf{Z}_j}{\mathbf{D}_{ii}} \tag{2.2}$$

We can observe that using uniform sampling, the points are centered at the position where $\sigma = 0.5$ and distribute similarly on the left and right side, and when using cluster randomized sampling, most red points are distributed on the right side which implies a higher neighbor treated ratio for treated units and most blue points are distributed on the left side which implies a lower neighbor treated ratio for controlled units. Therefore, cluster randomized sampling makes the treatment group more closer to the cases that all units are treated, and the control group more closer to the case that all units controlled.

2.2.3 Balanced Cluster Randomized Sampling

In SECTION 1.2, we analyzed the bias of difference-in-means estimator when cluster randomized sampling is used and SUTVA is assumed. The bias is expressed in EQUATION 1.9. We write it here again for clarification purpose.

$$\hat{\delta} - \delta = \frac{M}{N}\left[\frac{1}{m_t}\mathrm{Cov}\left(\frac{\sum_{j\in C^1}\sum_{i=1}^{n_j} Y^1(ij)}{\sum_{j\in C^1} n_j}, \sum_{j\in C^1} n_j\right) - \frac{1}{m_c}\mathrm{Cov}\left(\frac{\sum_{j\in C^0}\sum_{i=1}^{n_j} Y^0(ij)}{\sum_{j\in C^0} n_j}, \sum_{j\in C^0} n_j\right]\right. \tag{2.3}$$

The bias come from the correlation between the average outcome and the size of treatment group (control group). Therefore, if the cluster sizes are balanced, which means all clusters have almost the same size, then the size of treatment group (control group) is a constant, and hence the correlation is 0, making the difference-in-means estimator an unbiased estimator in this case[1].

To make the cluster sizes balanced, we need balanced graph partitioning algorithms. [9] used a label-swap based method. They first randomly partition the graph into equally sized clusters and then repeat the following two steps until convergence:

1. For each pair of units, if swapping the cluster labels of them can decrease the cross-cluster cuts, then swap them.

[1] It is unbiased when we assume SUTVA.

2. Randomly swap the labels of 5% pairs of units to break the local minimum.

There are also some streaming balanced graph partitioning algorithms that also aim to speed up the partition task when the target graph is very large [16][17][18].

2.3 Estimators

In this section, we introduce 3 kinds of estimators, and explain the advantages and disadvantages of them.

2.3.1 Difference-in-means Estimator

We have mentioned the difference-in-means estimators many times, which estimates the ATE using the difference of average outcomes between treated users and controlled users. This estimator is written as

$$\hat{\delta} = \frac{1}{n_t} \sum_{\{i;\mathbf{Z}_i=1\}} \mathbf{Y}_i(\mathbf{Z} = \mathbf{z}) - \frac{1}{n_c} \sum_{\{i;\mathbf{Z}_i=0\}} \mathbf{Y}_i(\mathbf{Z} = \mathbf{z}) \tag{2.4}$$

So taking FIGURE 2.2 as an example, the estimate ATE using difference-in-means estimator is the difference of average outcomes between the red points and the blue points.

The neighborhood exposure conditions can also be defined as an approach to reducing the estimation bias [8][9]. For example, the following three neighborhood exposure conditions can be used.

- Full neighborhood exposure: unit i and all its neighbors receive the same assignment.
- Absolute k-neighborhood exposure: unit i and at least k neighbors of i receive the same assignment.

- Fractional q-neighborhood exposure: unit i and at least $q * \mathbf{D}_{ii}$ neighbors of i receive the same assignment[2].

Based on the fractional q-neighborhood exposure condition, the estimated ATE is

$$\hat{\delta} = \frac{1}{n_t^*} \sum_{\{i;\boldsymbol{\sigma}_i \geq q\}} \mathbf{Y}_i(\mathbf{Z} = \mathbf{z}) - \frac{1}{n_c^*} \sum_{\{i;\boldsymbol{\sigma}_i \leq 1-q\}} \mathbf{Y}_i(\mathbf{Z} = \mathbf{z}) \tag{2.5}$$

where n_t^* is the number of treated units who satisfy the q-neighborhood exposure condition, n_c^* is the number of controlled units who satisfy the q-neighborhood exposure condition, $\boldsymbol{\sigma}$ is the neighbor treated ratios defined in EQUATION 2.2.

The reason that exposure conditions are used is that when units have more neighbors who receive the same assignment as them, it is more closer to the case that all users are treated or controlled. This is also the reason we make use of cluster randomized sampling. The exposure conditions also introduce some new bias because the data of units who do not satisfy the condition are disposed.

The advantage of the difference-in-means estimator is that it is applicable to almost any A/B testing task, while the disadvantage is that the bias of it may be large, and we will show this in the experiment part in CHAPTER 4.

2.3.2 Horvitz-Thompson Estimator

When the cluster sizes are not balanced, the difference-in-means estimator is biased. Again here we say an estimator is biased or unbiased based on SUTVA. In the next subsection we will introduce the linear model estimator which correct the bias without assuming SUTVA, and in CHAPTER 3 we will introduce our proposed bias correction methods which also do not assume SUTVA.

[2] Recall that $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}_{ii}$ being the number of neighbors of unit i.

Horvitz-Thompson estimator is an unbiased estimator when cluster randomized sampling is used and SUTVA is assumed [2]. It is written as

$$\hat{\delta} = \frac{M}{N}\left[\frac{1}{m_t}\sum_{\{i;\mathbf{Z}_i=1\}} \mathbf{Y}_i(\mathbf{Z}=\mathbf{z}) - \frac{1}{m_c}\sum_{\{i;\mathbf{Z}_i=0\}} \mathbf{Y}_i(\mathbf{Z}=\mathbf{z})\right] \tag{2.6}$$

where m_t is the number of treated clusters and m_c is the number of controlled clusters. And we prove its unbiasedness by

$$\begin{aligned}
\mathbb{E}[\hat{\delta}] &= \frac{M}{N}\mathbb{E}\left[\frac{1}{m_t}\sum_{\{i;\mathbf{Z}_i=1\}} \mathbf{Y}_i(\mathbf{Z}=\mathbf{z})\right] - \frac{M}{N}\mathbb{E}\left[\frac{1}{m_c}\sum_{\{i;\mathbf{Z}_i=0\}} \mathbf{Y}_i(\mathbf{Z}=\mathbf{z})\right] \\
&= \frac{M}{N}\overline{Y^{1,C}} - \frac{M}{N}\overline{Y^{0,C}} \\
&= \overline{Y^1} - \overline{Y^0} \\
&= \mathbb{E}[Y^1] - \mathbb{E}[Y^0] \\
&= \delta
\end{aligned} \tag{2.7}$$

where $\overline{Y^{1,C}}$ ($\overline{Y^{0,C}}$) is the average outcome of a cluster when all clusters are treated (controlled), and is thus unobservable.

Remember that Horvitz-Thompson estimator is unbiased based on SUTVA. Compared with the difference-in-means estimator, it does not require the cluster size be balanced. However, it usually produce larger estimation variance than the difference-in-means estimator does.

2.3.3 Linear Model Estimator

Both the difference-in-means estimator and Horvitz-Thompson estimator try to estimate the ATE by assuming SUTVA. Even though they can be unbiased based on SUTVA, when there are interferences among units, especially in social network, the estimation bias can still be large.

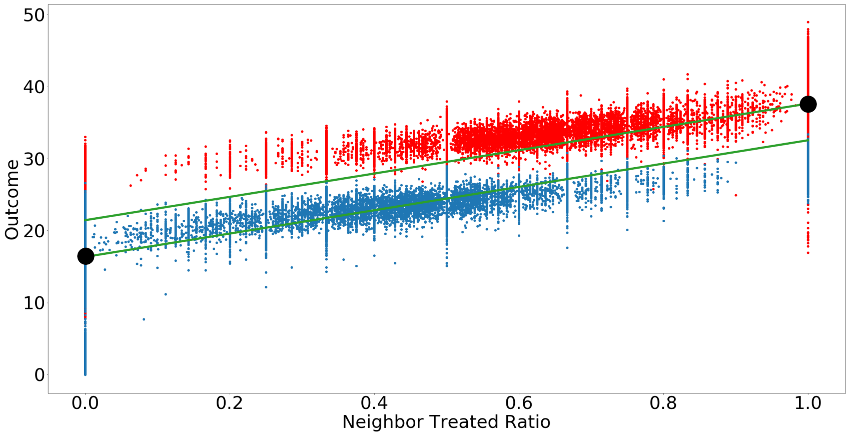


FIGURE 2.3: Illustration for the linear model estimator. The two green lines are the trained linear model when setting $\mathbf{Z}_i$ as 1 and 0 respectively. The right top black point is the predicted value when $\mathbf{Z}_i = 1, \boldsymbol{\sigma}_i = 1$, and the left bottom black point is the predicted value when $\mathbf{Z}_i = 0, \boldsymbol{\sigma}_i = 0$.

A linear model estimator is proposed in [9]. It differs with other estimators in that it is not the pure statistic obtained from the collected data, but the predicted value based on the assumption of outcome function. It assumes the outcome function is a linear function which depends on the assignment Z and the neighbor treated ratio σ. It is written as

$$\mathbf{Y}_i = \alpha + \beta \mathbf{Z}_i + \gamma \boldsymbol{\sigma}_i \tag{2.8}$$

Since $\mathbf{Y}$, $\mathbf{Z}$ and $\boldsymbol{\sigma}$ are all observable, the parameters α, β and γ can be estimated using linear regression as shown in Figure 2.3. Since when all user are treated, we have $\mathbf{Z} = \mathbf{1}, \boldsymbol{\sigma} = \mathbf{1}$, the average outcome when all users are treated is predicted by setting both $\mathbf{Z}$ and $\boldsymbol{\sigma}$ to $\mathbf{1}$, which is the right top black point in Figure 2.3. And likewise, the average outcome when all users are controlled is predicted by setting both $\mathbf{Z}$ and $\boldsymbol{\sigma}$ to $\mathbf{0}$, which is the left bottom black point in Figure 2.3. Finally, the ATE is estimated as the difference between those two values. Therefore, the estimated ATE is written as

$$
\begin{aligned}
\hat{\delta} &= \frac{1}{N}\sum_{i=1}^{N}\left[\hat{\mathbf{Y}}_i(\mathbf{Z}_i = 1, \boldsymbol{\sigma}_i = 1) - \hat{\mathbf{Y}}_i(\mathbf{Z}_i = 0, \boldsymbol{\sigma}_i = 0)\right] \\
&= \frac{1}{N}\sum_{i=1}^{N}\left[(\hat{\alpha} + \hat{\beta} + \hat{\gamma}) - \hat{\alpha}\right] \\
&= \hat{\beta} + \hat{\gamma}
\end{aligned}
\tag{2.9}
$$

As we discussed in SECTION 1.2.3, the difference-in-means estimator tends to underestimate the ATE, which is shown in EQUATION 1.15. Since the predicted average outcome when all users are treated is larger than the average observed outcome of treated users, and the predicted average outcome when all users are controlled is smaller than the average observed outcome of controlled users, the estimated ATE using linear model estimator is more close to the true ATE than that using the difference-in-means estimator, and thus the bias is smaller.

The linear model estimator further increase the estimation accuracy by making an assumption of the outcome function, which is known to the us. But the disadvantage of the linear model estimator is that when the real outcome function is quite different from the linear function in EQUATION 2.8, the bias may be even larger than the estimators which make no assumption of the outcome function. We can also see that making a correct assumption of the outcome function is a good way to improve the estimation accuracy.

Chapter 3

Proposed Methods

In the previous chapters, we explained the problem of estimating the ATE in A/B testing when interferences among units present. We also introduced two main ways to reduce the estimation bias: improving the sampling method and improving the estimator. In this chapter, we introduce our proposed bias reduction methods that are based on these two kinds of approaches.

3.1 Proposed Sampling Method

To reduce the estimation bias, our sampling methods should obey the following two guidelines.

1. The assignment of a unit should be independent of the assignments of other units as far as possible, this is for the purpose of randomization.
2. There should be as few interactions across the treatment group and the control group as possible, and equivalently as many interactions inside the treatment group and the control group. This is to make the treatment group more similar

to the case that all users are treated, and also make the control group more similar to the case that all users are controlled.

The first guideline is contradicting with the second guideline because more randomization brings more interactions across the treatment group and control group. Uniform sampling is an example of full randomization, and in this case every unit has about half neighbors who are treated and about half neighbors who are controlled and there are a lot of interactions across the two groups. On the other hand, if we partition the graph to two clusters by minimizing the cross-cluster cuts, the interactions are largely reduced, but the assignments of all units in the same cluster are correlated, resulting in the insufficiency of randomization[1].

The cluster randomized sampling does the trade-off between these two guidelines. Producing more clusters (smaller cluster size) ensures more randomization, and producing less clusters (larger cluster size) reduces the cross-group interactions. In this section, we propose weighted cluster randomized sampling to further reduce the cross-group interactions when producing the same number of clusters.

3.1.1 Weighted Cluster Randomized Sampling

FIGURE 3.1 shows an example of a directed network. Unit 2 has three out edges, which indicates that it is influenced by three other units, unit 0, 1, and 3. On the contrary, unit 4 only has one edge pointing to unit 3, which indicates unit 4 only receives influence from unit 3. When a unit has more out edges, the influence propagated through each of those edges is likely to be less. Therefore, when we partition the graph, each edge should not have equal importance.

[1] Imaging that the two clusters are men and women in the social network, since men and women may react very differently to the same feature, this sampling method brings a lot of bias.

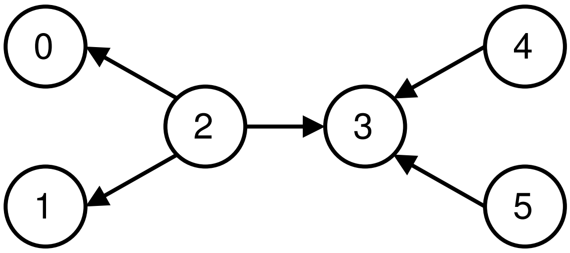


FIGURE 3.1: Example of a directed network

For many graph partitioning algorithms [9][16][17][18], the objective is to minimize the number of edges across clusters, and it can be expressed as

$$\text{minimize} \quad \left|\{e_{ij} \mid C(i) \neq C(j), \mathbf{A}_{ij} = 1, \forall i, j \in [1, N]\}\right| \tag{3.1}$$

where e_{ij} is the edge pointing from unit i to unit j, $C(i)$ is the cluster to which unit i belongs, $\mathbf{A}$ is the adjacency matrix.

We assign each edge a weight to represent the its importance. Then our objective is to minimize the total weight across clusters, and it is expressed as

$$\text{minimize} \quad \sum_{\{e_{ij};C(i)\neq C(j)\}} w(e_{ij}) \tag{3.2}$$

where $w(e_{ij})$ is the weight of e_{ij}. When $w(e) = 1, \forall e \in E$, where E is the set of all edges, EQUATION 3.2 degenerates to EQUATION 3.1. In the remaining of this section, we propose two weighted cluster randomized sampling methods based on different weight assignment strategies.

3.1.2 Degree Based Weighted Cluster Randomized Sampling

If we assume that for a unit, every out edge of it has the same importance, then the weight of edge e_{ij} is $w(e_{ij}) = \frac{1}{\mathbf{D}_{ii}}$, where $\mathbf{D}_{ii}$ is the number of out edges of unit

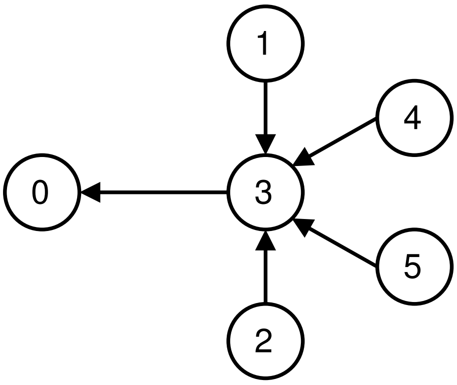


FIGURE 3.2: Illustration for the LinkRank based weighted cluster randomized sampling

i. Using this weight assignment strategy, the weight of each edge in FIGURE 3.1 are $e_{2,0} = e_{2,1} = e_{2,3} = 1/3$, $e_{4,3} = 1$, $e_{5,3} = 1$.

3.1.3 LinkRank Based Weighted Cluster Randomized Sampling

In FIGURE 3.2, if we use degree based weighted cluster randomized sampling, the weight of all edges is 1, but through $e_{1,3}$ only unit 1 is influenced, while through $e_{3,0}$ unit 3 is influenced and the influence is further propagated through unit 3. Therefore, more influence is propagated through $e_{3,0}$ than through $e_{1,3}$.

Borrowing the terminology in social network services, *Followers* of a unit are the units who have an edge pointing to that unit. Unit 3 has more followers than unit 1 does. So unit 3 can influence more units, and we can say unit 3 is more important than unit 1. Although the number of followers can be used to indicate the importance of an unit, PageRank [19] is more suitable, which is originally proposed to evaluate the importance of a web page. Based on PageRank, we can assign weight to each edge using LinkRank [20], which evaluates the importance of the

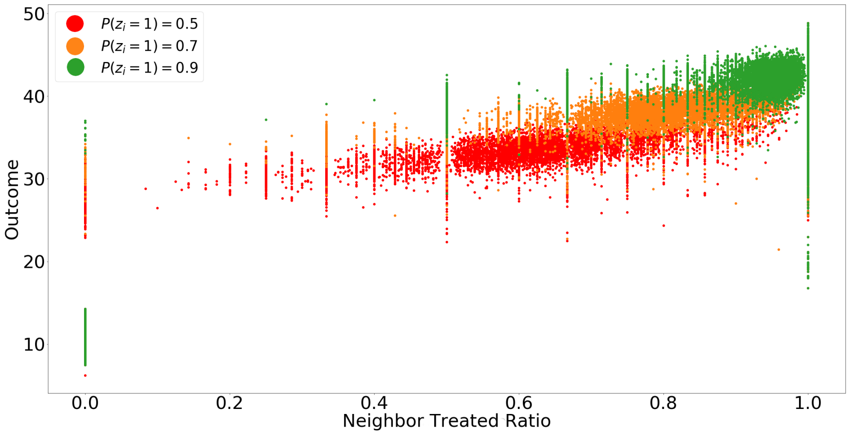


FIGURE 3.3: Different outcomes obtained by setting different treatment probability.

edges. LinkRank is defined as

$$w(e_{ij}) = \boldsymbol{\pi}_i \mathbf{G}_{ij} \tag{3.3}$$

where $\boldsymbol{\pi}$ is the PageRank vector, $\mathbf{G}$ is Google Matrix.

3.2 Proposed Estimators

We explained the problem of underestimation in SECTION 1.2.3. Even using cluster randomized sampling, the cross-group interactions cannot be fully eliminated, which means for a unit, there are still many neighbors who receive the opposite assignment. The example of neighbor treated ratio distribution can be found in Figure 2.2(b) for cluster randomized sampling. Most treated units have roughly 40% $\sim$ 80% treated neighbors, which indicates they have about 20% $\sim$ 60% controlled neighbors. To put it simply, the observed average outcome of the treatment group is smaller than the average outcome when all users are treated, and likewise, the observed average outcome of the control group is larger than the average outcome when all users are controlled, resulting the underestimation of ATE, as shown in EQUATION 1.15.

In Figure 3.3, we show different outcomes in treatment group obtained by setting different treatment probability (0.5, 0.7, 0.9). When we conduct an A/B testing experiment, usually the treatment probability is 0.5, and the outcomes are plotted in red points. When we increase the treatment probability, the average outcome also increases, as the yellow and green points show. When all users are treated, the treatment probability is 1, and therefore the average outcome will be even larger in the treatment group.

Therefore, we need estimators that can correct this kind of underestimation. In the section, we propose two estimators, one of correct the bias based on the network structure, and the other is based on the assumption of the outcome function.

3.2.1 Bias Correction Based on The Network Structure

In SECTION 2.3.3, we introduced the linear model estimator, which assumes the outcome function is a linear function depending on the assignment and neighbor treated ratio. However, since the influence can propagate through edges, a unit can be influenced by not only its neighbors, but also its neighbors of neighbors, its neighbors of neighbors of neighbors, and so on. To take the influence of other units into consideration, we define the *neighbor treated strength* as

$$\boldsymbol{\sigma}_i^* = \frac{1}{\mathbf{D}_{ii}} \sum_{\{j;\mathbf{A}_{ij}=1\}} \left(p_1 \mathbf{Z}_j + p_2 \frac{1}{\mathbf{D}_{jj}} \sum_{\{k;\mathbf{A}_{jk}=1\}} \boldsymbol{\sigma}_k^* \right) \tag{3.4}$$

where p_1 and p_2 are two parameters satisfying $p_1 > 0$, $p_2 > 0$, $p_1 + p_2 = 1$. p_1 controls the weight of the assignment of the neighbor, and p_2 controls the weight of the neighbor treated strength of the neighbor's neighbors. When $p_1 = 1, p_2 = 0$, this is exactly the same as the neighbor treated ratio defined in 2.2.

The linear model estimator in EQUATION 2.8 can then be rewritten as

$$\mathbf{Y}_i = \alpha + \beta \mathbf{Z}_i + \gamma \boldsymbol{\sigma}_i^* \tag{3.5}$$

EQUATION 2.9 which is used to estimate the ATE still holds.

The definition of neighbor treated strength requires us choose the parameters p_1 and p_2. Setting $p_1 = 1$ and $p_2 = 0$, it is equal to the linear model estimator with neighbor treated ratio. When the network effect is large, setting a larger p_2 can reduce the estimation bias. But since the network effect is unknown, some prior knowledge of it is needed.

3.2.2 Bias Correction Based on The Assumption of Outcome Function

I propose a new linear model estimator by assuming the outcome is a linear function depending on the assignment and the outcome of the neighbors.

$$\mathbf{Y}_i = \alpha + \beta \mathbf{Z}_i + \gamma \mathbf{Y}_i^{\mathrm{nbr}} \tag{3.6}$$

where $\mathbf{Y}_i^{\mathrm{nbr}} = (\frac{\mathbf{AY}}{\mathbf{D}})_i$ is the average outcome of unit i's neighbors. Since $\mathbf{Y}$, $\mathbf{Z}$ and $\mathbf{Y}^{\mathrm{nbr}}$ are all observable, the parameters α, β and γ can all be estimated using linear regression in the same way as the linear model estimator in EQUATION 2.8 does. For the sake of readability, it is written here again:

$$\mathbf{Y}_i = \alpha + \beta \mathbf{Z}_i + \gamma \boldsymbol{\sigma}_i \tag{3.7}$$

Since when all units are treated, $\mathbf{Z} = \mathbf{1}$, $\boldsymbol{\sigma} = \mathbf{1}$, and when all units are controlled, $\mathbf{Z} = \mathbf{0}$, $\boldsymbol{\sigma} = \mathbf{0}$, the estimator in EQUATION 3.7, when all the parameters are estimated, ATE is estimated as:

$$\begin{aligned}\hat{\delta} &= \frac{1}{N}\sum_{i=1}^{N}\left[\hat{\mathbf{Y}}_i(\mathbf{Z}_i = 1, \boldsymbol{\sigma}_i = 1) - \hat{\mathbf{Y}}_i(\mathbf{Z}_i = 0, \boldsymbol{\sigma}_i = 0)\right] \\ &= \frac{1}{N}\sum_{i=1}^{N}\left[(\hat{\alpha} + \hat{\beta} + \hat{\gamma}) - \hat{\alpha}\right] \\ &= \hat{\beta} + \hat{\gamma}\end{aligned} \tag{3.8}$$

However, if using the proposed linear model, although $\mathbf{Z}$ can be set to $\mathbf{1}$ when all units are treated and $\mathbf{0}$ when all units are controlled, $\mathbf{Y}^{\text{nbr}}$ is unknown. So the estimation cannot be made in the same way.

To estimate the ATE using the new proposed linear model estimator, first let the following equation be assumed:

$$\mathbb{E}[Y] \approx \mathbb{E}[Y^{\text{nbr}}] \tag{3.9}$$

Then the ATE can be estimated in the following way:

$$\begin{aligned}\mathbb{E}[Y^1] &= a + b + c\mathbb{E}[Y^{1,\text{nbr}}] \approx a + b + c\mathbb{E}[Y^1] \\ \Rightarrow \mathbb{E}[Y^1] &\approx \frac{a+b}{1-c}\end{aligned} \tag{3.10}$$

$$\begin{aligned}\mathbb{E}[Y^0] &= a + c\mathbb{E}[Y^{0,\text{nbr}}] \approx a + c\mathbb{E}[Y^0] \\ \Rightarrow \mathbb{E}[Y^0] &\approx \frac{a}{1-c}\end{aligned} \tag{3.11}$$

$$\delta = \mathbb{E}[Y^1] - \mathbb{E}[Y^0] \approx \frac{a+b}{1-c} - \frac{a}{1-c} = \frac{b}{1-c} \tag{3.12}$$

$$\hat{\delta} \approx \frac{\hat{b}}{1-\hat{c}} \tag{3.13}$$

Although $\mathbf{Y}$ is impossible to predict using the proposed linear model, the expected value $\mathbb{E}[Y^1]$ and $\mathbb{E}[Y^0]$ can be approximated. So the ATE can sill be estimated using EQUATION 3.13.

There are two problems with the proposed linear estimator. The first problem is

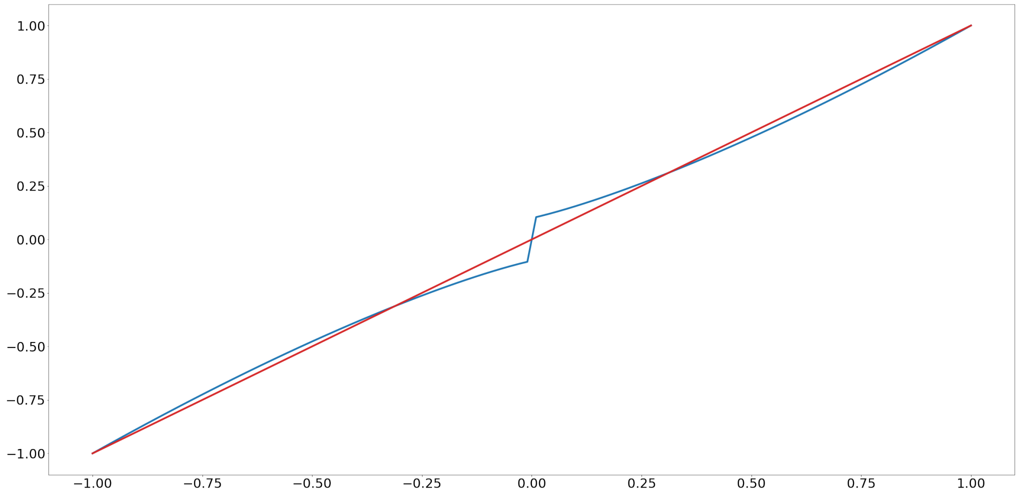


FIGURE 3.4: Plot for the smoothing function. The red line is the function $y = x$, and the curve line is the smoothing function. Although its domain is [0, 1], it is plotted in the domain of [−1, 1] to make it more clear.

that when c is close to 1, $1 - c$ will be close to 0, and the estimated ATE $\hat{\delta} = \frac{b}{1-c}$ will be extremely large. To prevent this unexpected result, we need a smoothing function $h(x)$ such that when x is too small, it can be scaled up, and when x is not too small, it is kept unchanged. The domain is [0, 1], and $h(x)$ should be a strictly increasing function.

We use the following function as the smoothing function:

$$h(x) = 0.1x^{0.0001} + 0.4x + 0.5x^{1.5} \tag{3.14}$$

This smoothing function is plotted in FIGURE 3.4 along with the linear function $y = x$.

The second problem is that when the estimated $\hat{c}$ is greater than or equal to 1, the proposed estimator does not work properly because $\hat{\delta} = \infty$ if $c = 1$ and $\hat{\delta} < 0$ if $c > 1$[2]. So if $c >= 1$, the linear model estimator based on neighbor treated ratio is used.

[2] The ATE can be negative when λ_1 is negative, but whether λ_1 is negative is related to the parameter b in the proposed linear model estimator, and has noting to do with c.

Chapter 4

Evaluation

In this chapter, the proposed methods are evaluated by comparing with baseline methods on various data sets, with respect to the estimation bias.

4.1 Experiment Settings

Before showing the experiment results, in this section, I first introduce the experiment settings.

4.1.1 Datasets

The graph data sets used in the experiment are from [15]. The graph dataset information is summarized in TABLE 4.1. These graph are originally directed graphs. For the evaluation of sampling method, When comparing the proposed estimators, they are converted to undirected graphs, and dangling nodes (nodes with degree being 0) are removed.

Graph Name	Nodes	Edges	Description
wiki-Vote	7,115	103,689	Wikipedia who-votes-on-whom network
soc-Epinions1	75,879	508,837	Who-trusts-whom network of Epinions.com
soc-Slashdot0811	77,360	905,468	Slashdot social network from November 2008

TABLE 4.1: Graph dataset information

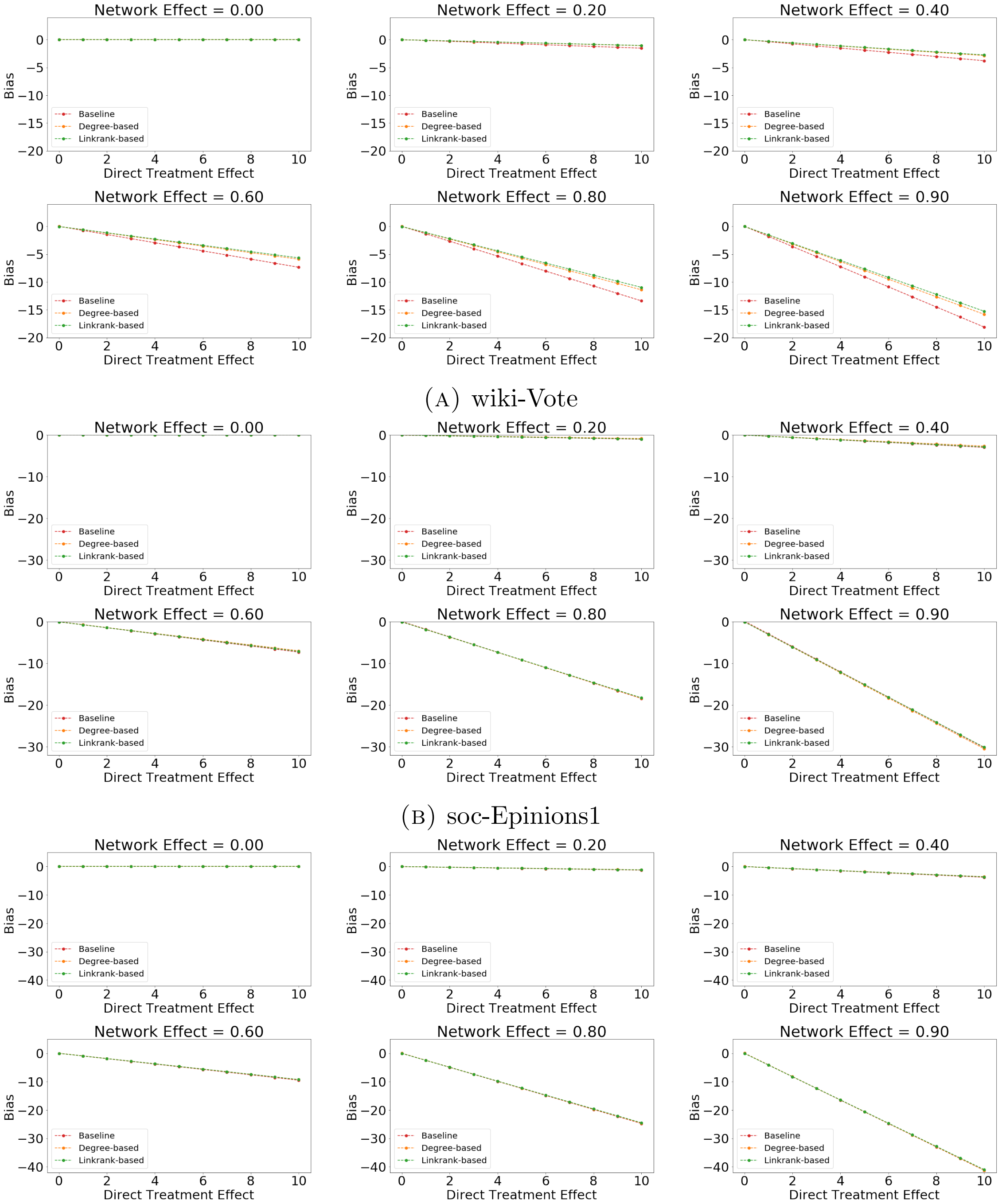
4.1.2 Outcome Generation

A synthetic outcome function is used to generate the outcomes in the experiment. The reason why a synthetic outcome function is used is discussed in SECTION 1.2.2. The synthetic outcome function in EQUATION 1.10 is used. And it is written here again.

$$\begin{aligned} \mathbf{Y}^{t*} &= \alpha + \lambda_1 \mathbf{Z} + \lambda_2 \frac{\mathbf{A}\mathbf{Y}^{t-1}}{\mathbf{D}} + \mathbf{U}_{(t)} \\ \mathbf{Y}^{t} &= g\big(\mathbf{Y}^{t*}\big) \end{aligned} \tag{4.1}$$

By setting different regularization function g and different parameters, the following 4 different outcome functions are used in the experiment.

1. Outcome function f_1: $g(x) = x$, $\alpha = 3$
2. Outcome function f_2: $g(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases}$, $\alpha = -1.5$
3. Outcome function f_3: $g(x) = 0.7x$, $\alpha = 3$
4. Outcome function f_4: $g(x) = x^{0.7}$, $\alpha = 3$

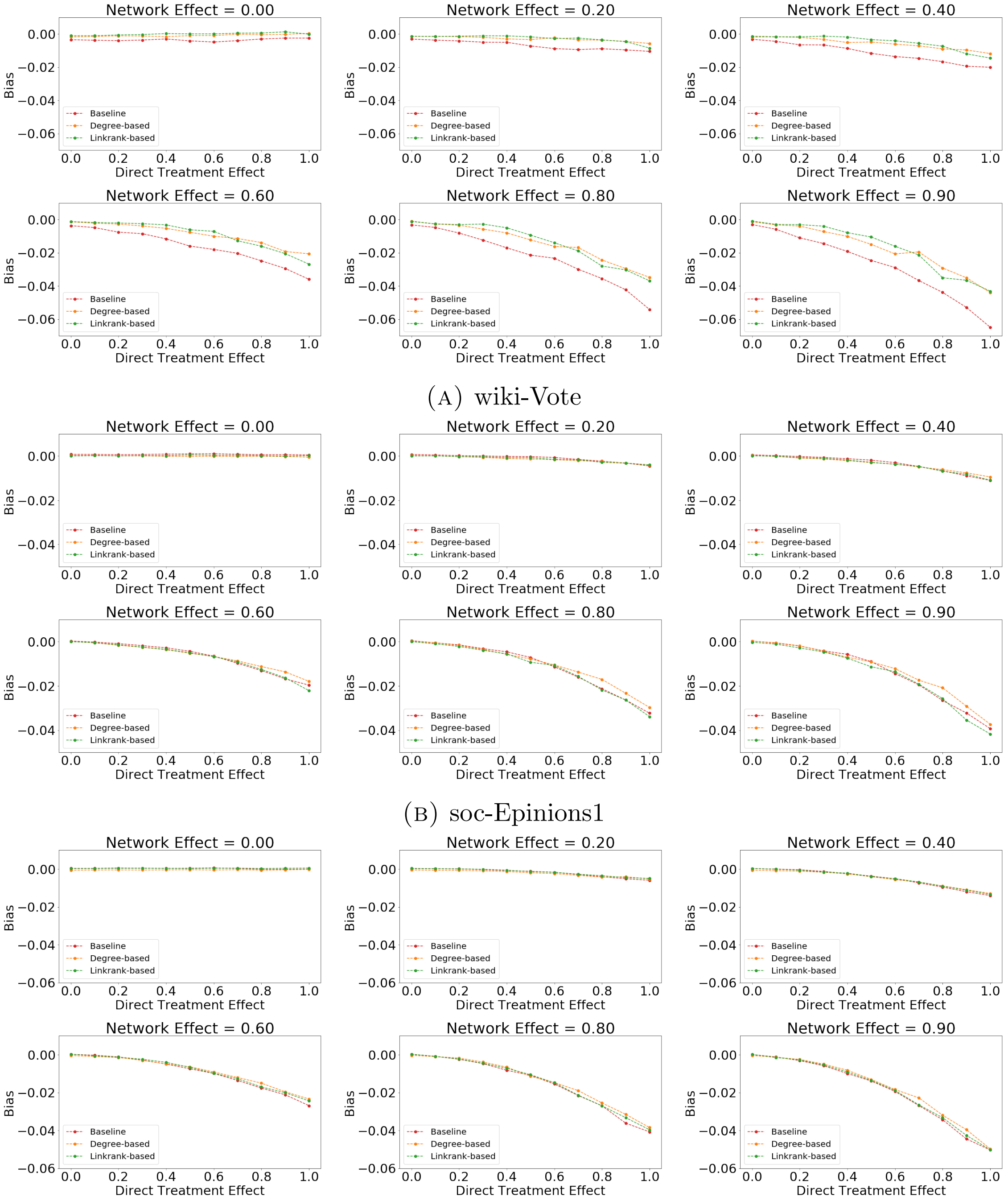


(A) wiki-Vote

(B) soc-Epinions1

(C) soc-Slashdot0811

Figure 4.1: Results of different sampling methods for outcome function f_1



(A) wiki-Vote

(B) soc-Epinions1

(C) soc-Slashdot0811

FIGURE 4.2: Results of different sampling methods for outcome function f_2.

Method	Weight	Estimator
Baseline method	1	difference-in-means estimator
Degree based method	$1/d$	difference-in-means estimator
Linkrank based method	Linkrank value	difference-in-means estimator

TABLE 4.2: The summarization of methods using different sampling method

4.2 Results of Proposed Sampling Methods

In this section, the proposed degree based randomized cluster sampling method and Linkrank based randomized cluster sampling method are evaluated. These methods along with the baseline method are compared on the directed graphs.

4.2.1 Baseline Method

The graph partitioning algorithm used in this experiment is the label swap based method proposed in [9]. The baseline method sets the weight of all edges to 1, the proposed degree based method sets the weight of an edge to $1/d$, where d is its out-degree, the proposed Linkrank based method sets the weight of an edge to its Linkrank value. These methods are summarized in TABLE 4.2.

Although the linear model estimator can usually achieve higher estimation accuracy than the difference-in-means estimator, the latter is used in this experiment. The reason is that when linear model estimator is used, the bias correction made the sampling method less important, and it is harder to see the difference among these sampling methods. Although the estimation accuracy is lower, the difference-in-means estimator is applicable without any assumption of the outcome function, while the linear model estimator performs well when the outcome can be indeed assumed to be a linear function which depends on the neighbor treated ratio.

Graph Name	Edges	Edges in largest SCC	Edges in largest WCC
wiki-Vote	103,689	39,456 (0.381)	103,663 (1.000)
soc-Epinions1	508,837	443,506 (0.872)	508,836 (1.000)
soc-Slashdot0811	9,054,680	888,662 (0.981)	905,468 (1.000)

Table 4.3: Information about the largest strongly connected component in the graph datasets.

4.2.2 Results

The results for outcome function f_1 is shown in FIGURE 4.1 and the results for outcome function f_2 is shown in FIGURE 4.2. As shown in the FIGURES, on the wiki-Vote dataset, the proposed degree based method and link based method outperforms the baseline methods for both f_1 and f_2. On the other two datasets, the three methods achieved almost the same estimation accuracy with respect to the bias. The two proposed methods produce similar estimation bias on all datasets. Since for f_3 and f_4, the experiment demonstrates that the results are similar to that of f_1 and f_2, the results are not shown here.

One possible explanation for the result that the proposed methods only outperform the baseline method on wiki-Vote is that wiki-Vote is less strongly connected than the other two graphs. As shown in TABLE 4.3, the largest SCC (strongly connected component) contains 38.1% of all the nodes in the graph, which is far less than that of the other two graphs. With respect to the largest WCC (weakly connected component), for all there datasets, almost all edges are contained in the largest WCC. So the number of edges in the largest SCC being small indicates there are some edges propagating the influence from one SCC to another but no edge propagating the influence in the reverse direction. If this kind of edges are assigned smaller weight, then they will be cut when partitioning the graph, and the SCCs is more likely to be partitioned to different clusters, result in the reduction of interference among clusters.

4.3 Results of Proposed Estimators

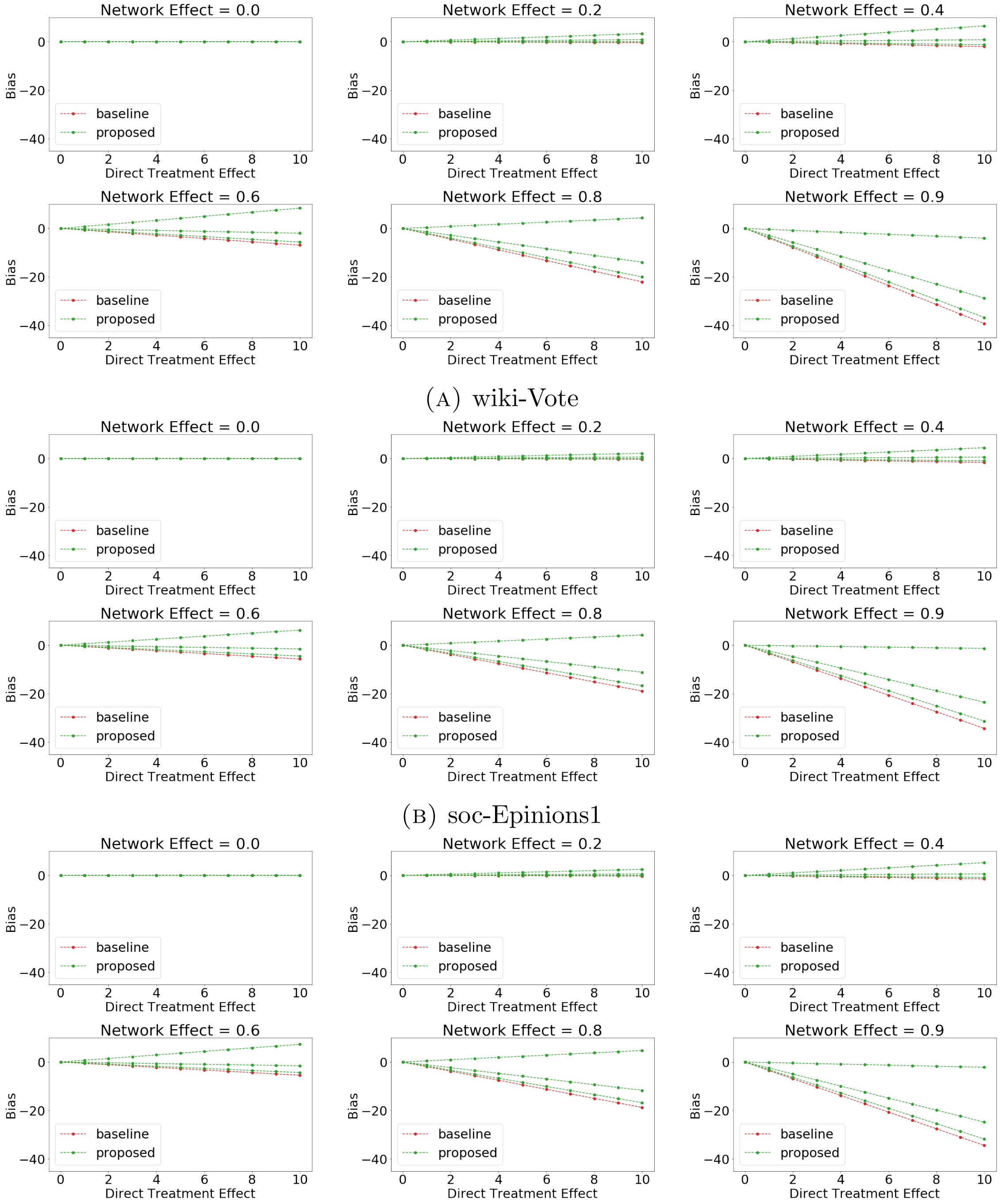
In this section, the two proposed methods are compared with the baseline method separately. In this experiment, the graphs are converted into undirected graph and dangling nodes are removed, because for the directed graph, if it is not strongly connected, the influence cannot fully propagated. All these estimator are based on the baseline cluster randomized sampling in the previous section.

4.3.1 Baseline Method

The baseline method is the linear model estimator in EQUATION 2.8. The two proposed methods: the bias corrected estimator based on network structure and the bias corrected estimator based on assumption, are compared with the baseline method separately in the remaining of this section.

4.3.2 Results of Bias Corrected Estimator Based on Network Structure

Experiments conducted on the four outcome functions are shown in FIGURE 4.3 $\sim$ 4.6. In each of the graphs, there are three green lines, which are the results of the proposed method plotted by setting different parameter (p_2 in EQUATION 3.4). As shown in the results, by choosing a good parameter p_2, the bias can be effectively corrected, but if p_2 is too large, the proposed estimator will overestimate the ATE. Therefore, choosing a small p_2 is less like to result in overestimation and in particular, when $p_2 = 0$, the proposed method is equivalent to the baseline method.

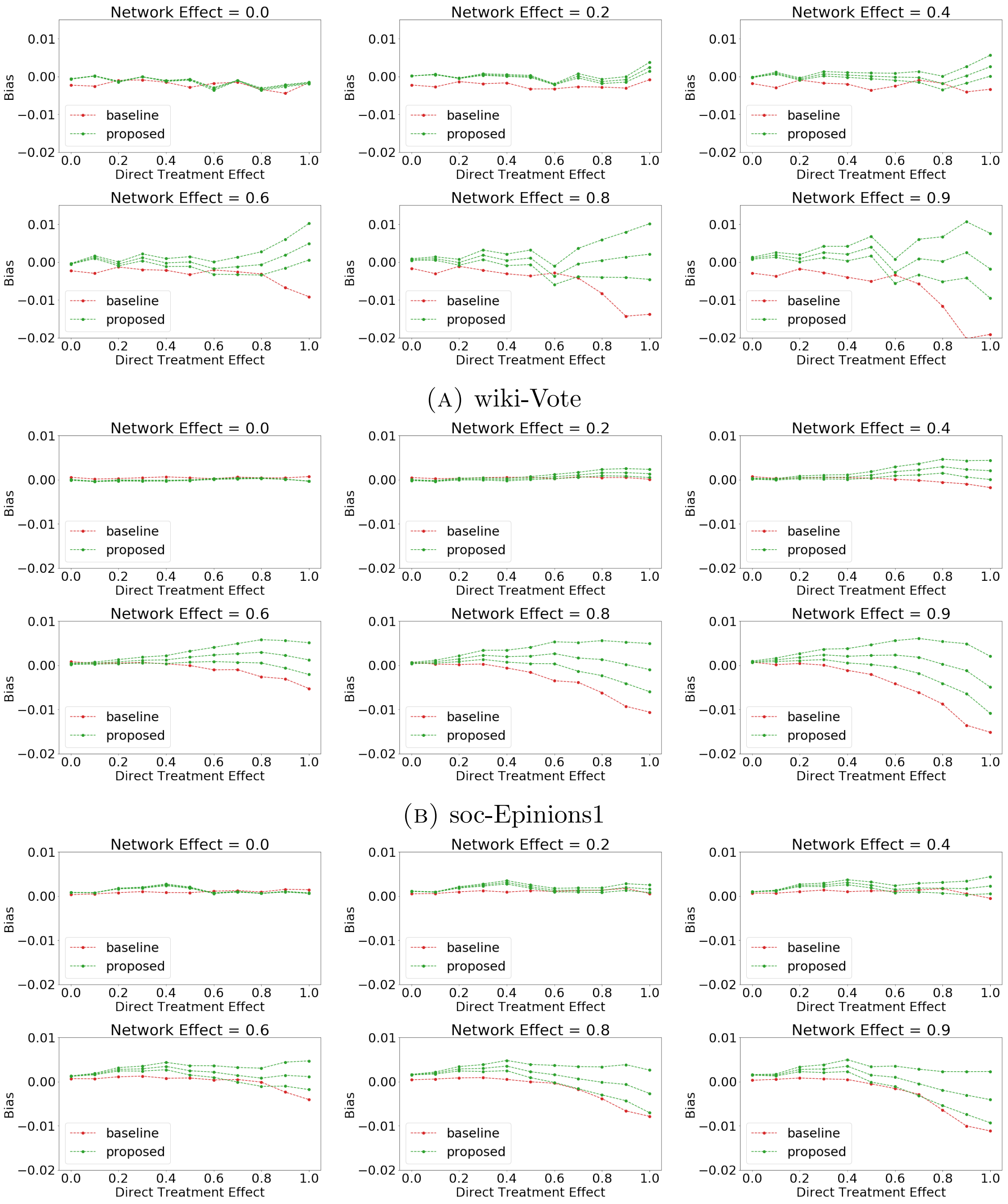


(A) wiki-Vote

(B) soc-Epinions1

(C) soc-Slashdot0811

FIGURE 4.3: Results of bias corrected estimator based on network structure for outcome function f_1. Proposed method are plotted by setting p_2 to 0.2, 0.5, 0.8. The larger the p_2, the higher of the position of the green line.

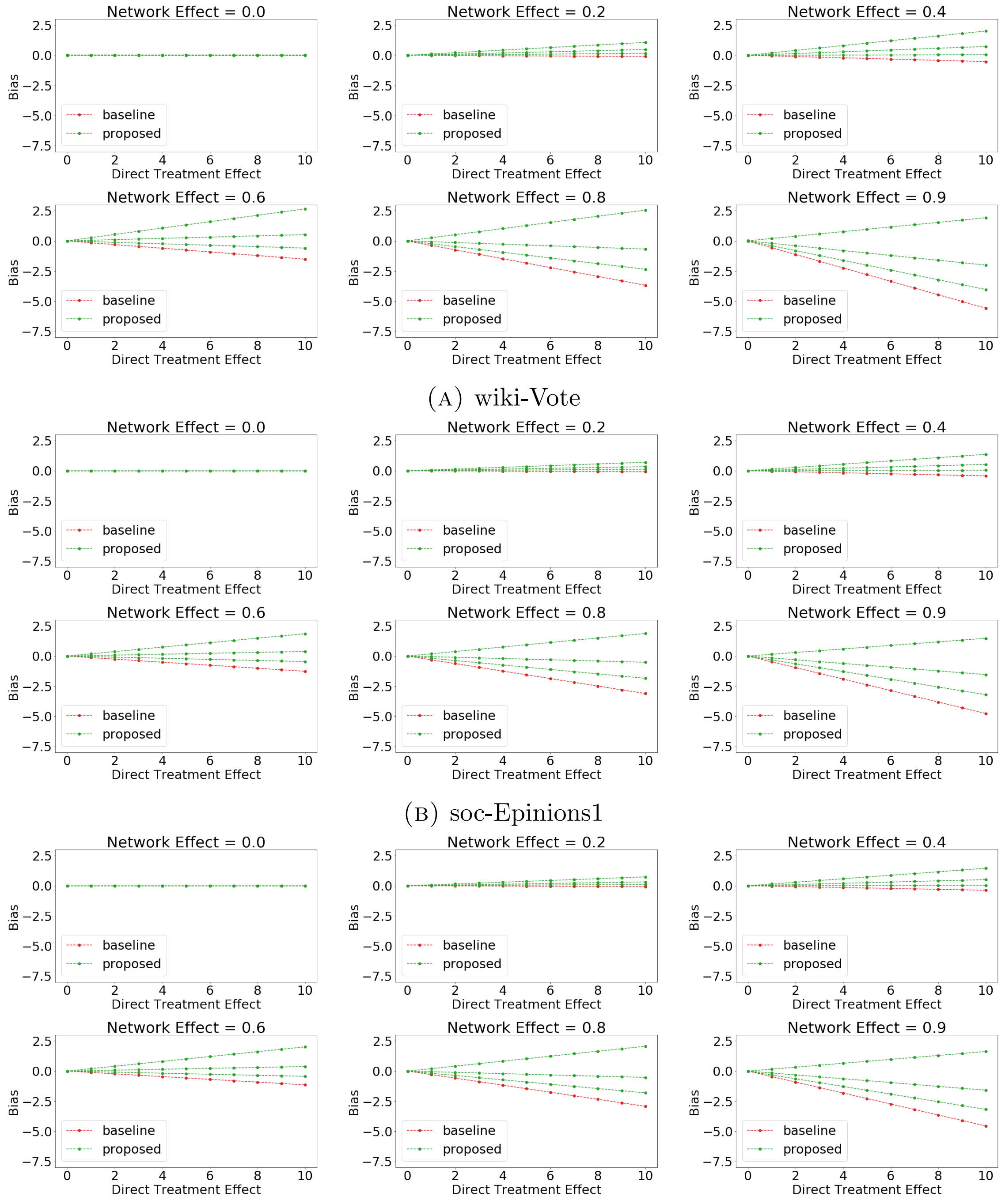


(A) wiki-Vote

(B) soc-Epinions1

(C) soc-Slashdot0811

Figure 4.4: Results of bias corrected estimator based on network structure for outcome function f_2. Proposed method are plotted by setting p_2 to 0.1, 0.2, 0.3. The larger the p_2, the higher of the position of the green line.

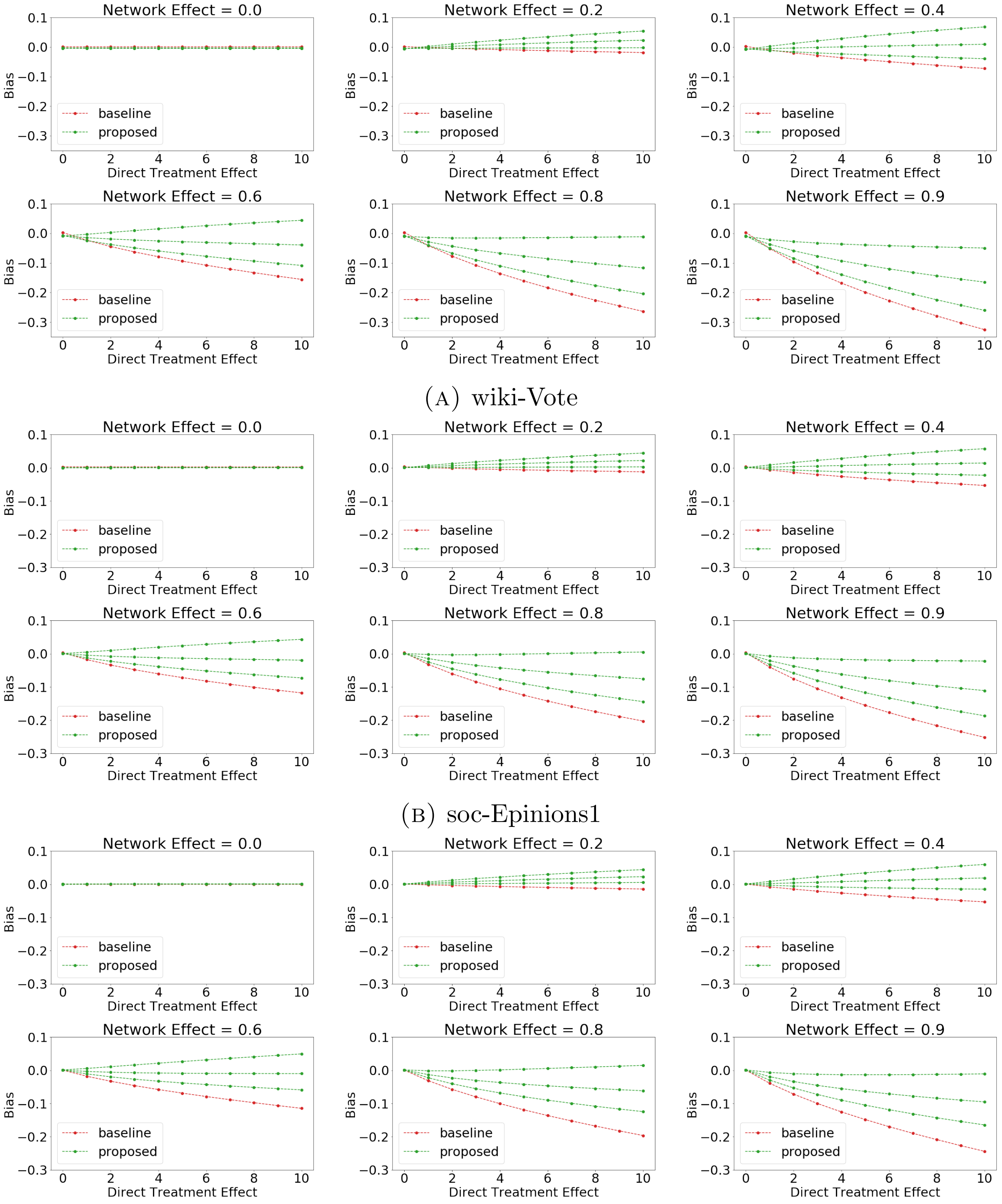


(A) wiki-Vote

(B) soc-Epinions1

(C) soc-Slashdot0811

Figure 4.5: Results of bias corrected estimator based on assumption for outcome function f_3. Proposed method are plotted by setting p_2 to 0.3, 0.5, 0.7. The larger the p_2, the higher of the position of the green line.



(A) wiki-Vote

(B) soc-Epinions1

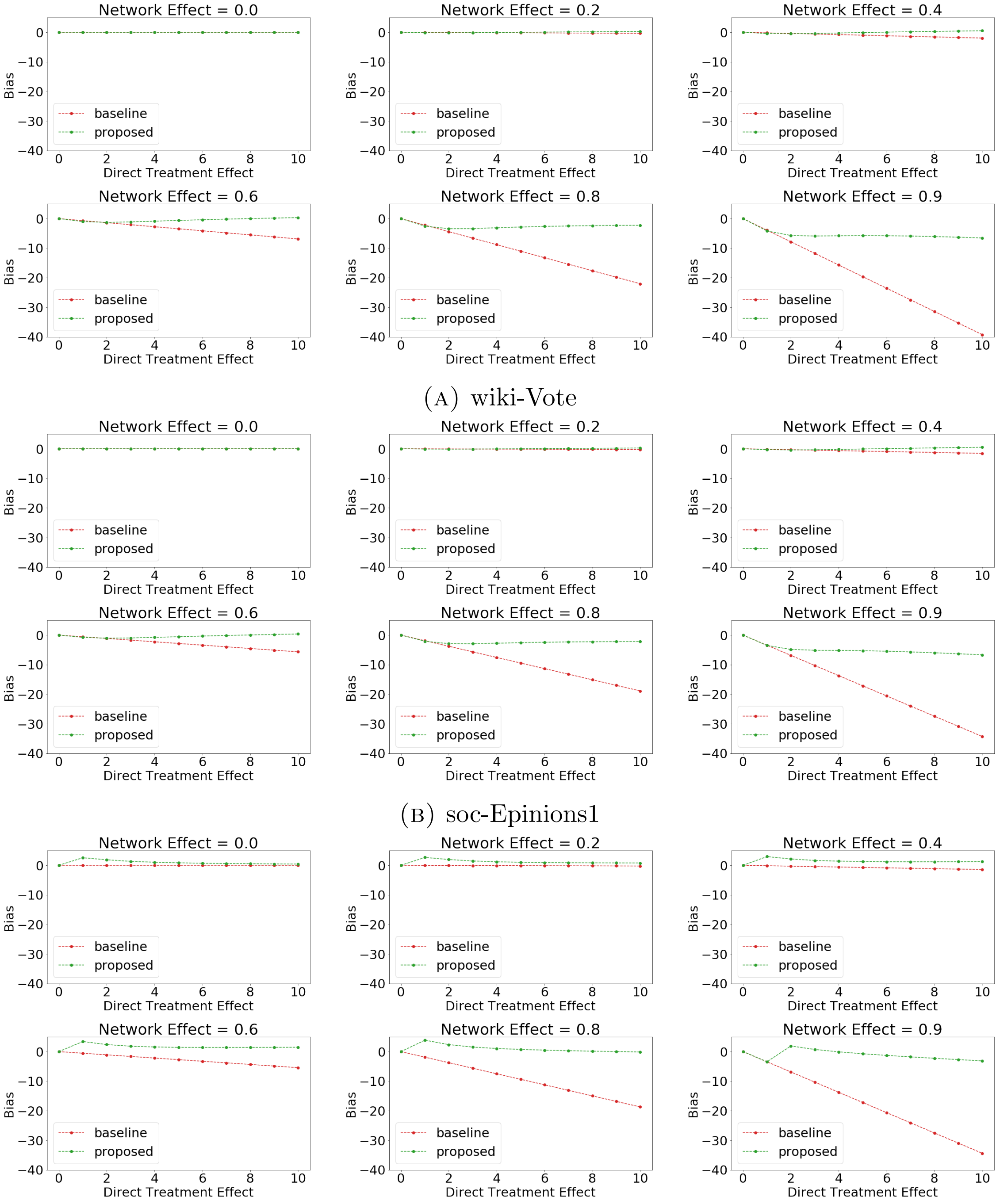
(C) soc-Slashdot0811

Figure 4.6: Results of bias corrected estimator based on assumption for outcome function f_4. Proposed method are plotted by setting p_2 to 0.1, 0.2, 0.3. The larger the p_2, the higher of the position of the green line.

4.3.3 Results of Bias Corrected Estimator Based on Assumption

Experiments conducted on the four outcome functions are shown in FIGURE 4.7 ∼ 4.10. As shown in the results, for f_1, the proposed method outperforms the baseline method on all three datasets, but for f_2, the baseline method is better. For f_3 and f_4, the proposed method wins on wiki-Vote and soc-Epinions, and is defeated on soc-Slashdot0811.

It can also be observed that when the direct treatment effect is small, the proposed method has relatively large bias. The possible reason is that when the direct treatment effect is small, the outcome and average neighbor outcome is dominated by the Gaussian noise, and in consequence, the proposed linear model assuming the linear relationship between the outcome and average neighbor outcome cannot work properly.

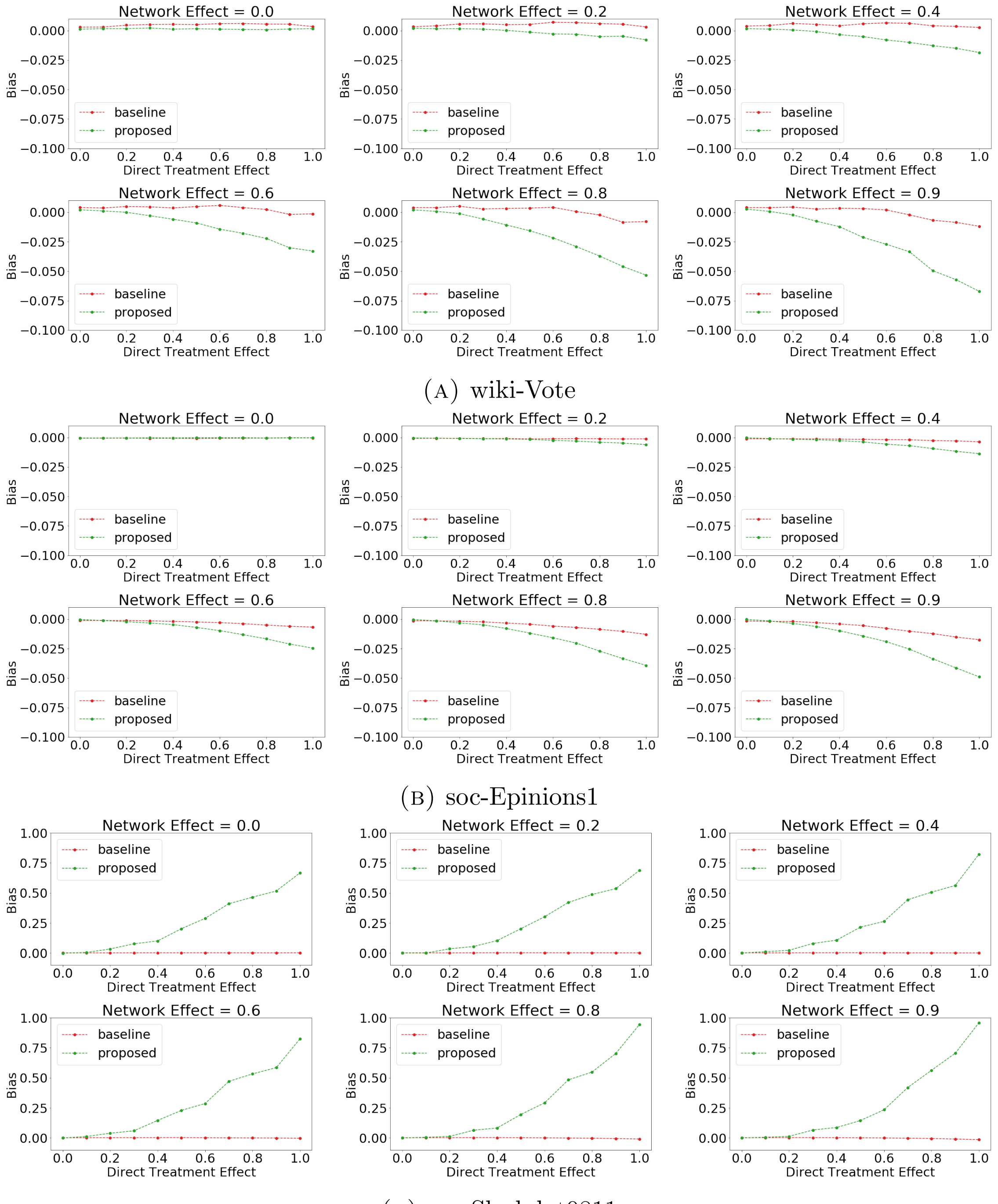


(A) wiki-Vote

(B) soc-Epinions1

(C) soc-Slashdot0811

FIGURE 4.7: Results of bias corrected estimator based on assumption for outcome function f_1.

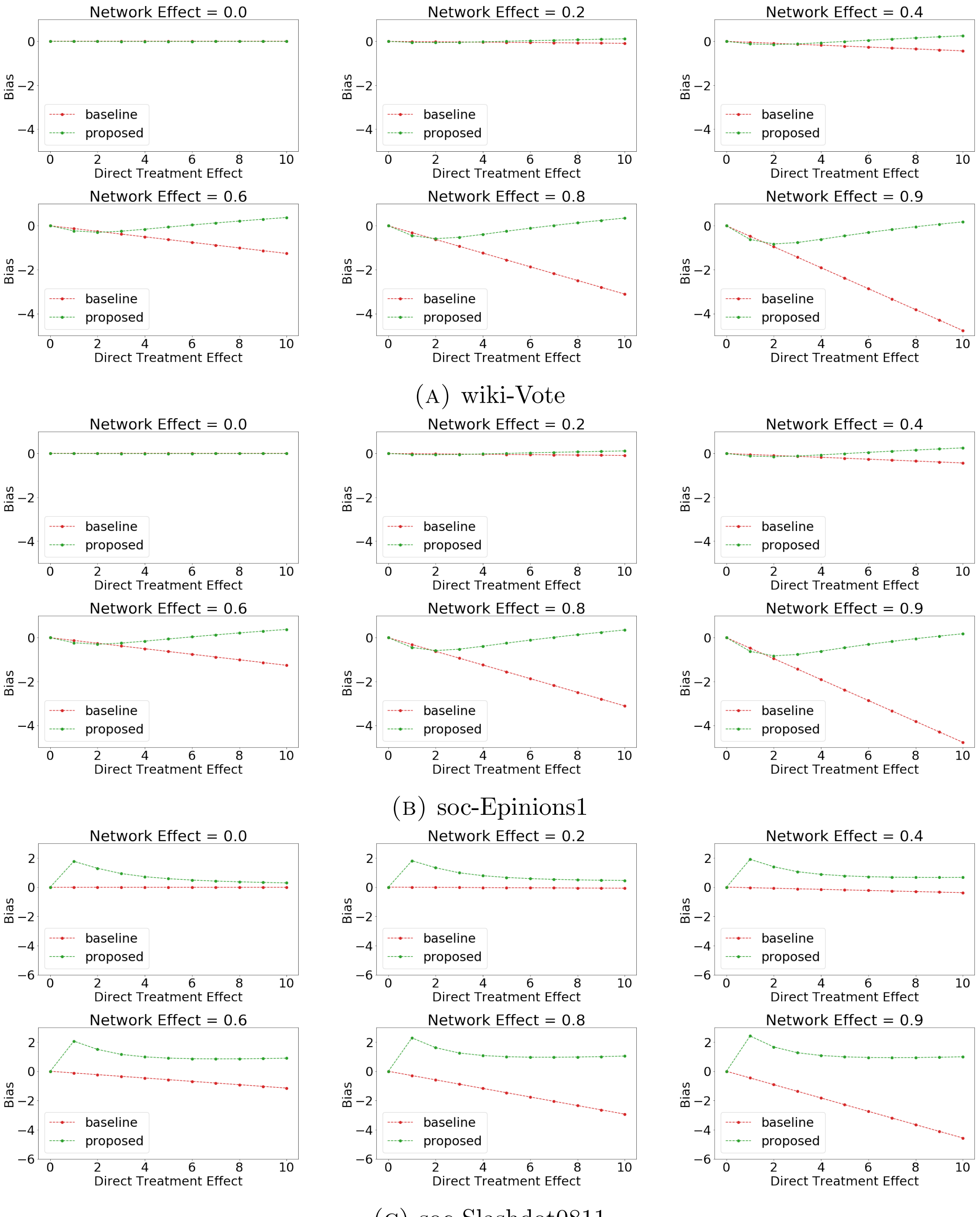


(A) wiki-Vote

(B) soc-Epinions1

(C) soc-Slashdot0811

Figure 4.8: Results of bias corrected estimator based on assumption for outcome function f_2.

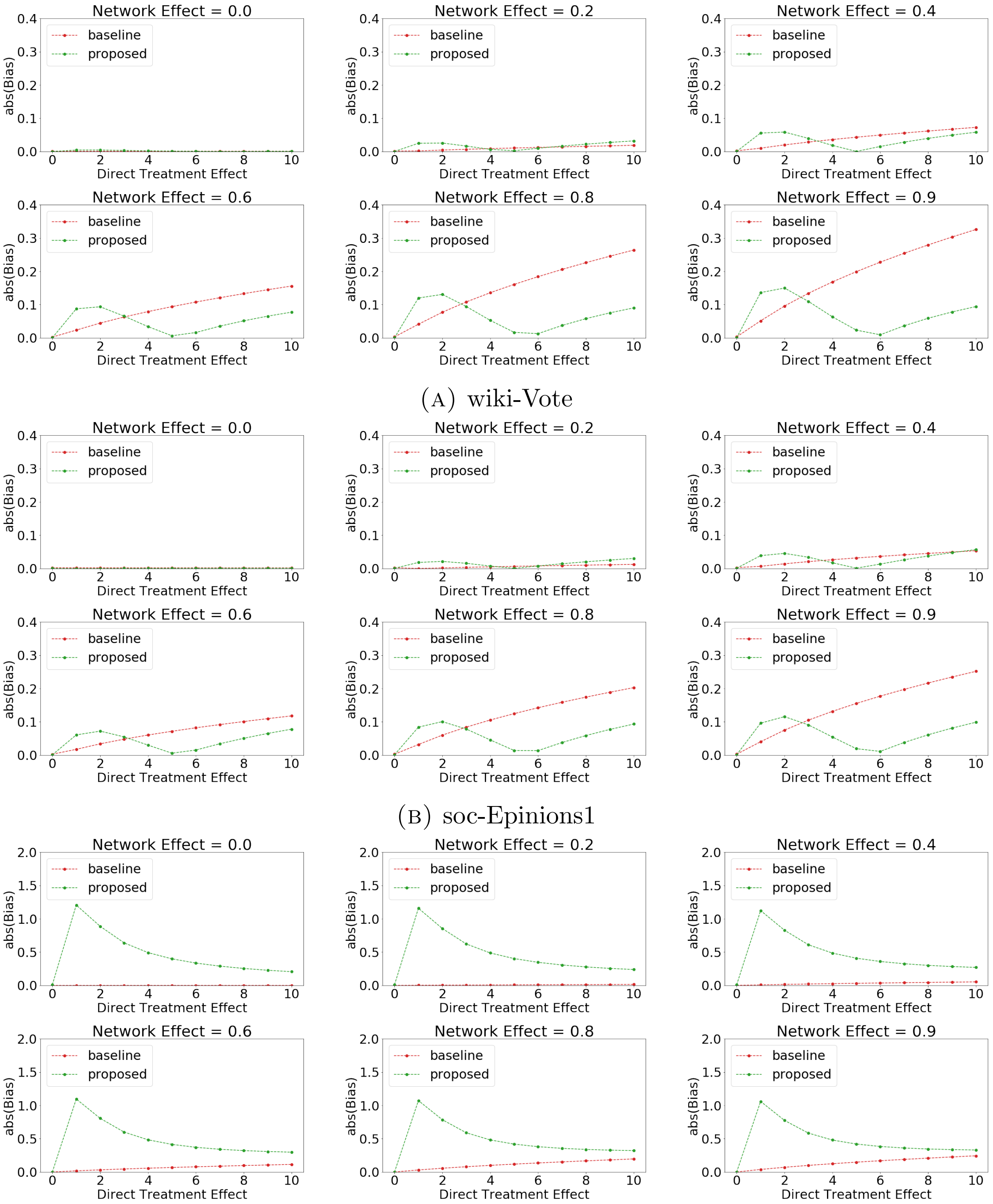


(A) wiki-Vote

(B) soc-Epinions1

(C) soc-Slashdot0811

FIGURE 4.9: Results of bias corrected estimator based on assumption for outcome function f_3.



(A) wiki-Vote

(B) soc-Epinions1

(C) soc-Slashdot0811

FIGURE 4.10: Results of bias corrected estimator based on assumption for outcome function f_4. Note that the vertical axis is the absolute value of bias.

Chapter 5

Conclusion

When network effects present among the experiment units in A/B testing, the tradition method using difference-in-means estimator with uniform sampling tends to underestimated the ATE. In particular, the network effects are very common in the social network services, on which In this thesis, facing with the problem of network effects, two kinds of methods are proposed to reduce the bias, one of which tries to improve the sampling method by minimizing the interference between the treatment group and the control group while keeping the uniformity of sampling as far as possible, the other tries to correct the bias by making the most the network structure or making an assumption of the outcome function.

The proposed sampling methods is only applicable to directed network. They outperform the baseline method when the network is not strongly connected.

The proposed estimators aim to correct the bias by assuming the outcome function is a linear function which depending on the assignment and the neighbor treatment ratio (or average neighbor outcome). The proposed estimator based on network structure can effective correct the bias if the parameter p_2 is properly chosen. And another estimator based on the assumption that the outcome has linear relationship with the average neighbor outcome can correct the bias without the requirement of choosing parameters.

Bibliography

[1] Judea Pearl et al. Causal inference in statistics: An overview. *Statistics surveys*, 3:96–146, 2009.

[2] Joel A Middleton and Peter M Aronow. Unbiased estimation of the average treatment effect in cluster-randomized experiments. *Statistics, Politics and Policy*, 6(1-2):39–75, 2015.

[3] Charles F Manski. Identification of endogenous social effects: The reflection problem. *The review of economic studies*, 60(3):531–542, 1993.

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

[5] Brendan Kline and Elie Tamer. Some interpretation of the linear-in-means model of social interactions. 2014.

[6] Lars Backstrom and Jon Kleinberg. Network bucket testing. In *Proceedings of the 20th international conference on World wide web*, pages 615–624. ACM, 2011.

[7] Liran Katzir, Edo Liberty, and Oren Somekh. Framework and algorithms for network bucket testing. In *Proceedings of the 21st international conference on World Wide Web*, pages 1029–1036. ACM, 2012.

[8] 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. ACM, 2013.

[9] 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. International World Wide Web Conferences Steering Committee, 2015.

[10] Panos Toulis and Edward Kao. Estimation of causal peer influence effects. In *International Conference on Machine Learning*, pages 1489–1497, 2013.

[11] Jean Pouget-Abadie, Martin Saveski, Guillaume Saint-Jacques, Weitao Duan, Ya Xu, Souvik Ghosh, and Edoardo Airoldi. Testing for arbitrary interference on experimentation platforms. *preprint*, 2017.

[12] 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. ACM, 2017.

[13] Symeon Papadopoulos, Yiannis Kompatsiaris, Athena Vakali, and Ploutarchos Spyridonos. Community detection in social media. *Data Mining and Knowledge Discovery*, 24(3):515–554, 2012.

[14] Fragkiskos D Malliaros and Michalis Vazirgiannis. Clustering and community detection in directed networks: A survey. *Physics Reports*, 533(4):95–142, 2013.

[15] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

[16] Isabelle Stanton and Gabriel Kliot. Streaming graph partitioning for large distributed graphs. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1222–1230. ACM, 2012.

[17] Charalampos Tsourakakis, Christos Gkantsidis, Bozidar Radunovic, and Milan Vojnovic. Fennel: Streaming graph partitioning for massive scale graphs. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 333–342. ACM, 2014.

[18] Joel Nishimura and Johan Ugander. Restreaming graph partitioning: simple versatile algorithms for advanced balancing. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1106–1114. ACM, 2013.

[19] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[20] Youngdo Kim, Seung-Woo Son, and Hawoong Jeong. Finding communities in directed networks. *Physical Review E*, 81(1):016103, 2010.

Publications

Domestic Conferences

- Jian Chen, Masashi Toyoda, A/B Testing for Social Network Services with Directed User Graphs, The 9th Forum on Data Engineering and Information Management, 2017.
- Jian Chen, Junpei Komiyama, Masashi Toyoda, Bias Correction for A/B Testing in Social Network, The 10th Forum on Data Engineering and Information Management, 2018 (submitted).