Bias Correction for A/B Testing in Social Network

Jian CHEN†, Junpei KOMIYAMA††, and Masashi TOYODA††

† Graduate School of Information Science and Technology, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 1138654 Japan
†† Institute of Industrial Science, The University of Tokyo,
4-6-1 Komaba, Meguro-ku, Tokyo, 1538505 Japan
E-mail: †{kenn-chen,jkomiyama,toyoda}@tkl.iis.u-tokyo.ac.jp

Abstract A/B testing is a widely used method to estimate the effect of an intervention on the outcome (like the effect of changing the position of a button on the click-through rate). Traditional A/B testing methods assume SUTVA (Stable Unit Treatment Value Assumption) holds, which states that any experiment unit's outcome will not be interfered by other experiment units. However, this assumption does not hold in social network in many occasions due to the network effect among users. Recent work on A/B testing proposed many new sampling methods and estimators to solve this problem. However, if the outcome is a real value, most of those methods still underestimate the effect when the network effect is large. So in this paper, we propose new bias correction method to further improve the estimation accuracy of A/B testing in social network. Compared with existing bias correction method, our proposed method makes use of the information of more neighbors of a unit, making it less likely to underestimate.

Key words bias correction, A/B testing, SNS, graph partitioning

1. INTRODUCTION

There are many occasions that we need to know the *effect* of something. For example, pharmacists need to know the effect of their new drugs on a certain disease, politicians need to know the effect of their new policies on a certain social problem, and web designers need to know the effect of their new designs on the click-through rate.

A/B testing, which includes the Rubin causal model [2], is the most widely used method to estimate the effect. In A/B testing, *Treatment* indicates making an intervention (such as applying a new drug), *control* indicates not making an intervention, and *assignment* is either treatment or control. We denote assignment as Z. $Z = 1$ indicates treatment and $Z = 0$ indicates control. *Outcome* is the result we are interested in (such as the condition of the patient), and we denote it as Y. We summarize some notations used in this paper in Table 1.

Representation	Example	Meaning
Uppercase normal letter	X	Random variable
Uppercase bold letter	$\mathbf{X}$	Random vector or matrix
Lowercase bold letter	$\mathbf{x}$	Vector
Lowercase normal letter	x	Scalar

Table 1: Notations of Different Types of Symbols

The effect of an individual is called *Individual Treatment Effect* (ITE), which is represented as:

$$\delta_i = Y_i(Z_i = 1) - Y_i(Z_i = 0) \tag{1}$$

where δ_i is the ITE of unit i. A/B testing aims to estimate the *Average Treatment Effect* (ATE), which is the average ITE over all units, and is represented as:

$$\begin{aligned}\delta &= \frac{1}{N}\sum_{i=1}^{N}\delta_i \\ &= \frac{1}{N}\sum_{i=1}^{N}\big[Y_i(Z_i = 1) - Y_i(Z_i = 0)\big] \\ &= \frac{1}{N}\sum_{i=1}^{N}Y_i(Z_i = 1) - \frac{1}{N}\sum_{i=1}^{N}Y_i(Z_i = 0) \\ &= \frac{1}{N}\sum_{i=1}^{N}Y_i(\mathbf{Z} = \mathbf{1}) - \frac{1}{N}\sum_{i=1}^{N}Y_i(\mathbf{Z} = \mathbf{0})\end{aligned} \tag{2}$$

where δ is ATE and N is the number of units, $\mathbf{Z} = \mathbf{1}$ indicates all units are treated and $\mathbf{Z} = \mathbf{0}$ indicates all units are controlled. ATE can be interpreted as the difference between the average outcomes of two "parallel universes", in one of which all units are treated, and in the other all units are controlled.

However, we are not able to both treat and control a unit at the same time, that is, Z_i cannot be both 1 and 0. So in fact, ATE is a value that is impossible to be obtained.

What A/B testing does is to estimate the ATE, by making use of randomization. Randomization means the assignment of a unit is independent of the assignment of other units. Based on randomization, the following *difference-in-means estimator* is usually used to estimate the ATE.

$$\hat{\delta} = \frac{1}{N_1} \sum_{\{i;\mathbf{z}_i=1\}} \mathbf{Y}_i(\mathbf{Z}=\mathbf{z}) - \frac{1}{N_0} \sum_{\{i;\mathbf{z}_i=0\}} \mathbf{Y}_i(\mathbf{Z}=\mathbf{z}) \quad (3)$$

where N_1 and N_0 is the number of treated units and controlled units respectively, $\mathbf{Y}$ is a random vector and $\mathbf{Y}_i$ is the outcome of unit i, $\mathbf{z}$ is the assignment vector. For all i, if $P(\mathbf{z}_i = 1) = P(\mathbf{z}_i = 0) = 0.5$, where P indicates the probability, the difference-in-means estimator is an unbiased estimator [3].

This unbiased estimator is based on the *Stable Unit Treatment Value Assumption* (SUTVA), which states that one unit's outcome cannot be influenced by other units. This assumption is quite reasonable in many occasions, such as the case that testing the effect of a new drug, because the condition of a patient often will not be influenced by other patients.

However, SUTVA is hard to hold for A/B testing in social network. Users in social network interact with each other intensively and there are many occasions that a unit's outcome can be influenced by other users. For example, if we developed a recommendation algorithm that recommends interesting tweets to each user, and the outcome we are interested in is the number of retweets of each user, and we also assume the recommendation algorithm is indeed effective so treated users would do more retweets, then the users who follow the treated users can also see more interesting tweets in their timelines, and as a result, their number of retweets will also increase. Therefore, in this case, a unit's outcome can be influenced by other units. The effect that a unit received from other units is often called *network effect*.

Bias is of great importance to evaluate the performance of an A/B testing method. The bias of ATE is expressed as $\mathbb{E}(\hat{\delta}) - \delta$. There are mainly two ways to reduce the bias introduced by the interference among units. The first way is to improve the sampling method. When SUTVA holds, uniform sampling is enough. When it does not hold, *cluster randomized sampling* is often used [4] [5], which first partitions the network into clusters, and then samples on cluster level. In this way, a treated user will have more treated neighbors and a controlled user will also have more controlled neighbors. In other words, the treatment group and control group will be more similar to the two 'parallel universes'. Since cluster randomized sampling cannot solve the problem completely, *bias correction* is also needed, which is the second way to reduce bias and what we are going to introduce in this paper.

2. RELATED WORK

To estimate the average outcome when adding a new feature, which only takes effect when a user and at least d of its 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 is introduced [4] [5] [8], and some unbiased estimators are also proposed [3] based on cluster randomized sampling and SUTVA. Since they are also based on SUTVA, they are not truly unbiased when there exist interferences among clusters. [5] uses 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 the network effect or test the existence of the network effect [9] [10] [11].

3. Synthetic Outcome Model

In A/B testing, although outcomes such as the condition of patients or the number of retweets of each users, are observable, the ATE is impossible to be obtained as we explained in Section 1. To obtain the ground truth of ATE to evaluate the proposed methods, a synthetic outcome model is necessary.

In this paper, we use synthetic outcome model proposed in [8]. It is written as:

$$\begin{aligned} \mathbf{Y}^*_{i,t} &= \alpha + \lambda_1 \mathbf{Z}_i + \lambda_2 \frac{1}{\mathbf{d}_i} \sum_{j \in \eta(i)} \mathbf{Y}_{j,t-1} + \mathbf{U}_{i,t} \\ \mathbf{Y}_{i,t} &= g(\mathbf{Y}^*_{i,t}) \end{aligned} \quad (4)$$

where α is a constant, λ_1 is the *direct treatment effect*, λ_2 is the network effect, $\mathbf{A}$ is the adjacency matrix (a binary matrix), $\mathbf{d}$ is the degree vector, $\mathbf{U}$ is a random vector representing user specific traits and for all i, $\mathbf{U}_i \sim \mathcal{N}(0,1)$, the subscript '$t$' is the iteration step, and g is a function.

The outcomes are computed iteratively until the mean of $\mathbf{Y}$ converges, and $\mathbf{Y}$ is initialized as $\mathbf{0}$. $\mathbf{Y}$ is summed up by the following four components.

- α: α is the baseline value which is a constant. It simply indicates 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 unit i is treated, and $\mathbf{Z}_i = 0$ if it is controlled. Therefore, the outcome of a user will increase by

λ_1 if it is treated, and will not increase if it is controlled. So we call λ_1 direct treatment effect.

- $\lambda_2 \frac{1}{\mathbf{d}_i} \sum_{j \in \eta(i)} \mathbf{Y}_{j,t-1}$: this component is the average outcome of user i's neighbors [(1)] at the previous iteration step multiplied by a coefficient λ_2, which is the network effect. A large λ_2 indicates the outcome of a user is influenced more by the neighbors, while a small λ_2 indicates the outcome of a user is influenced less by the neighbors, and in particular, when $\lambda_2 = 0$ the outcome of a user does not depend on other users, which is equivalent 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., a Gaussian random variable is used to capture these traits.

The function g is applied to the outcomes at each iteration step. When $g(x) = \mathbb{1}(x)$, where $\mathbb{1}$ is a indicator function, the synthetic outcome model is a probit model. In this case, the outcome is either 0 or 1, and it can represent the kind of outcomes such as like/dislike. When $g(x) = x$, it is a linear-in-means model, which is a model usually used to capture the interaction of social and economic phenomenon [12] [13]. In this case, the outcome is a real value, and it can represent the kind of outcomes such as the number of retweets or the number of clicks.

4. Existing Methods

To estimate the ATE when the interferences among units present, recent work mainly tries to propose new sampling methods and estimators.

4.1 Uniform Sampling And Cluster Randomized Sampling

Uniform sampling is extensively used in traditional A/B testing, which assumes SUTVA. In uniform sampling, $Z_i \sim$ Bernoulli(0.5), and every unit has the same probability to be either treated or controlled.

When all units are treated, a treated unit is surrounded by only treated units, and when all units are controlled, a controlled unit is surrounded by only controlled users. So we should make treated units closer to treated units and controlled units closer to controlled units. To achieve this, we can first partition the network into clusters, and then sample on cluster level [4] [5]. If we partition the network into M clusters, $C_1, C_2, \ldots, C_M$, and we denote the assignment of cluster j as W_{C_j}, then $W_{C_j} \sim$ Bernoulli(0.5), and $Z_i = W_{C_j}$ if unit i is in cluster j. According to the analysis in [3], when we use the difference-in-means estimator as expressed in Equation 3, the clusters should be balanced in size to reduce the bias. In this case, some balanced partitioning algorithm are readily to use [14] [15] [16].

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

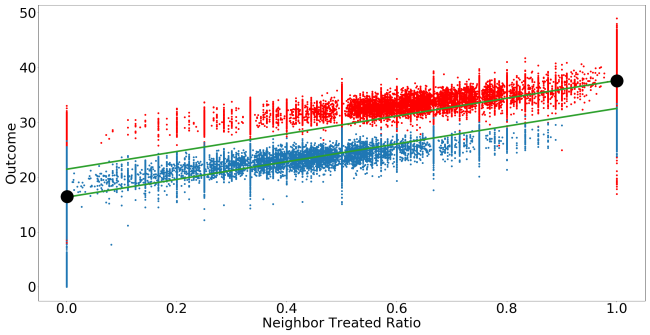


Figure 1: Linear regression using the linear model estimator. Green lines are the fitted lines, the left and right black points are the estimated average outcomes when all units are controlled and treated respectively.

4.2 Exposure Condition

Even using cluster randomized sampling, some treated units may still have few treated neighbors, and some controlled units may still have few controlled units, which indicates they are not "effectively" treated or controlled. Therefore, we may only use the data of the units who are "effectively" treated or controlled to estimate the ATE, and they are *network exposed to treatment* and *network exposed to control* respectively based on the definition of *exposure condition* [4].

4.3 Unbiased Estimators Based on Cluster Randomized Sampling And SUTVA

When making use of cluster randomized sampling, the difference-in-means estimator in Equation 3 is no longer unbiased even assuming SUTVA. Several unbiased estimator, such as Horvitz-Thompson estimator and Raj estimator, are proposed [3]. But those estimators for cluster randomized sampling are only unbiased based on SUTVA.

4.4 Linear Model Estimator

The linear model estimator proposed in [5] differs with other estimators in that its estimated ATE is not a pure statistic obtained from the observed outcome data, but the predicted value based on the assumption of outcome model. It assumes the outcome is a linear function of the assignment Z and the *neighbor treated ratio* σ. It is expressed as

$$\mathbf{Y}_i = \alpha + \beta \mathbf{Z}_i + \gamma \boldsymbol{\sigma}_i \tag{5}$$

$$\boldsymbol{\sigma}_i = \frac{1}{\mathbf{d}_i} \sum_{j \in \eta(i)} \mathbf{Z}_j \tag{6}$$

where $\boldsymbol{\sigma}$ is the neighbor treated ratio. The parameters α, β and γ can be estimated using linear regression as shown in Figure 1.

If all users are treated, $\mathbf{Z} = \mathbf{1}$, and also $\boldsymbol{\sigma} = \mathbf{1}$. If all

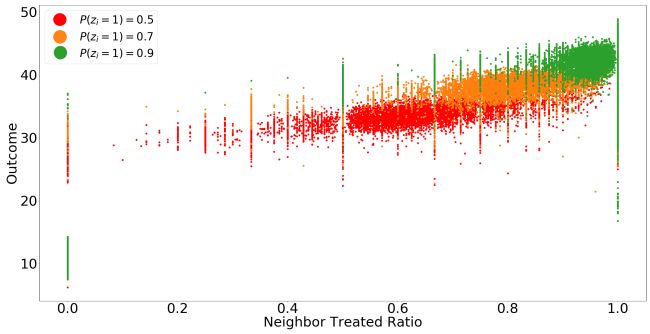


Figure 2: Different outcomes obtained by setting different treatment probability.

users are controlled, $\mathbf{Z} = \mathbf{0}$, and also $\boldsymbol{\sigma} = \mathbf{0}$. So the average outcome when all users are treated can be predicted by setting both $\mathbf{Z}$ and $\boldsymbol{\sigma}$ to $\mathbf{1}$, which is the right black point in Figure 1. And likewise, the average outcome when all users are controlled can be predicted by setting both $\mathbf{Z}$ and $\boldsymbol{\sigma}$ to $\mathbf{0}$, which is the left black point in Figure 1. Finally, the ATE can be estimated as the vertical distance between those two points.

5. BIAS CORRECTION FOR ATE ESTIMATION

In this section, we first explain the reason why bias correction is necessary, and then introduce our proposed bias correction method.

5.1 Why Bias Correction Is Necessary

In Figure 2, we show different outcomes obtained by setting different treatment probability (0.5, 0.7, 0.9). When we conduct an A/B testing experiment, usually 50% units are treated and the outcomes are plotted by red points. When we increase the probability of treatment, the average outcome is going to be larger, as the yellow and green points show. If all users are treated, the probability of treatment is 1, and therefore the average outcome will be larger than the average outcome of treated users when the treatment probability is 0.5. For the same reason, the average outcome in the case that all users are controlled will be smaller than the average outcome of controlled users when the treatment probability is 0.5. So we have

$$\begin{aligned}\hat{\delta} &= \frac{1}{N_1}\sum_{\{i;\mathbf{z}_i=1\}}\mathbf{Y}_i(\mathbf{Z}=\mathbf{z}) - \frac{1}{N_0}\sum_{\{i;\mathbf{z}_i=0\}}\mathbf{Y}_i(\mathbf{Z}=\mathbf{z}) \\ &\leqq \frac{1}{N}\sum_{i=1}^{N}\mathbf{Y}_i(\mathbf{Z}=\mathbf{1}) - \frac{1}{N}\sum_{i=0}^{N}\mathbf{Y}_i(\mathbf{Z}=\mathbf{0}) \\ &= \delta\end{aligned} \tag{7}$$

Thus, common estimators such as the linear-in-means estimator usually tend to underestimate. This is the reason why we need bias correction.

5.2 Proposed Bias Correction Method

To correct the bias, we need make an assumption of the outcome model. [5] assumes the outcome model is a linear function depending on the assignment and the neighbor treated ratio, as expressed in Equation 5, and they achieved better estimation accuracy than the estimators which only use pure statistic from the data. Since the linear model estimator only uses the information of 1-hop neighbors, we can further increase the estimation accuracy by making use of the information of more neighbors in the social network.

We define the *treated strength* as:

$$\boldsymbol{\tau}_i = (1-p)\mathbf{Z}_i + p\frac{1}{\mathbf{d}_i}\sum_{j\in\eta(i)}\boldsymbol{\tau}_j \tag{8}$$

where p is a hyper-parameter and $0 \leqq p \leqq 1$. If we expand Equation 8, it can be written as:

$$\begin{aligned}\boldsymbol{\tau}_i &= (1-p)\mathbf{Z}_i + \frac{p}{\mathbf{d}_i}\sum_{j\in\eta(i)}\boldsymbol{\tau}_j \\ &= (1-p)\mathbf{Z}_i + \frac{p}{\mathbf{d}_i}\sum_{j\in\eta(i)}(1-p)\mathbf{Z}_j + \frac{p}{\mathbf{d}_i}\sum_{j\in\eta(i)}\frac{p}{\mathbf{d}_j}\sum_{k\in\eta(j)}\boldsymbol{\tau}_k\end{aligned} \tag{9}$$

The treated strength $\boldsymbol{\tau}$ can be further expanded, so the treated strength of a single unit uses the information of all connected units(2). It is also easy to show that the weight of $(k+1)$-hop neighbors is less than that of k-hop neighbors. For example, the weight of 0-hop neighbor (the unit it self) is $1-p$, the weight of 1-hop neighbors is $\frac{p(1-p)}{\mathbf{d}_i}$, and the weight of 2-hop neighbors is $\frac{p^2(1-p)}{\mathbf{d}_i\mathbf{d}_j}$. Therefore, close neighbors contribute more to a unit's treated strength than faraway neighbors do.

The treated strength $\boldsymbol{\tau}$ can be computed iteratively until convergence as expressed in the following equations:

$$\boldsymbol{\tau}_{i,t} = (1-p)\mathbf{Z}_i + p\frac{1}{\mathbf{d}_i}\sum_{j\in\eta(i)}\boldsymbol{\tau}_{j,t-1} \tag{10}$$

$$\boldsymbol{\tau}_{i,0} = (1-p)\mathbf{Z}_i \tag{11}$$

We then show some lemmas for our defined treated strength.

Lemma 1. *For all i, $0 \leqq \boldsymbol{\tau}_i \leqq 1$*

Proof. This can be easily proved by induction. When $t = 0$, since $0 \leqq 1-p \leqq 1$ and $\mathbf{Z}_i \in \{0,1\}$, we have $0 \leqq \boldsymbol{\tau}_{i,0} \leqq 1$.

Assume, for $t = k$, $0 \leqq \boldsymbol{\tau}_{i,k} \leqq 1$. Let $t = k+1$,

$$\begin{aligned}\boldsymbol{\tau}_{i,k+1} &= (1-p)\mathbf{Z}_i + p\frac{1}{\mathbf{d}_i}\sum_{j\in\eta(i)}\boldsymbol{\tau}_{j,k} \\ &\leqq (1-p) + p \\ &\leqq 1\end{aligned}$$

(2) : Unit a and unit b is connected if there is a path from unit a to unit b.

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 2: Graph dataset information

and $\boldsymbol{\tau}_{i,k+1} \geqq 0$ because all summands in Equation 10 are greater than or equal to 0 when $t = k+1$. So $0 \leqq \boldsymbol{\tau}_{i,k+1} \leqq 1$.

Therefore, $0 \leqq \boldsymbol{\tau}_{i,t} \leqq 1$ holds for all t. ■

Lemma 2. *If $p \neq 1$, for all i, $\boldsymbol{\tau}_i = 0$ when $\mathbf{Z} = \mathbf{0}$, and $\boldsymbol{\tau}_i = 1$ when $\mathbf{Z} = \mathbf{1}$.*

Proof. This can also be proved by induction. When $\mathbf{Z} = \mathbf{0}$, $\boldsymbol{\tau}_{i,0} = 0$. Assume for $t = k$, $\boldsymbol{\tau}_{i,k} = 0$. Let $t = k+1$, since for any unit i, both $\mathbf{Z}_i = 0$ and $\sum_{j \in \eta(i)} \boldsymbol{\tau}_{j,k} = 0$, we have $\boldsymbol{\tau}_{i,k+1} = 0$. Therefore, when $\mathbf{Z} = \mathbf{0}$, in any iteration round t, $\boldsymbol{\tau} = \mathbf{0}$.

When $\mathbf{Z} = \mathbf{1}$, we have

$$\boldsymbol{\tau}_{i,0} = (1-p)$$
$$\boldsymbol{\tau}_{i,1} = (1-p) + p(1-p) = (1-p)(1+p)$$
$$\boldsymbol{\tau}_{i,2} = (1-p) + p(1-p)(1+p) = (1-p)(1+p+p^2)$$

Assume for $t = k$, $\boldsymbol{\tau}_{i,k} = (1-p)\sum_{a=0}^{k} p^a$. Let $t = k+1$,

$$\begin{aligned}\boldsymbol{\tau}_{i,k+1} &= (1-p) + p(1-p)\sum_{a=0}^{k} p^a \\ &= (1-p) + (1-p)\sum_{a=1}^{k+1} p^a \\ &= (1-p)\sum_{a=0}^{k+1} p^a\end{aligned}$$

So for all $t \geqq 0$, $\boldsymbol{\tau}_{i,t} = (1-p)\sum_{a=0}^{t} p^a = 1 - p^{t+1}$, and then $\lim_{t\to\infty} \boldsymbol{\tau}_{i,t} = 1$. Therefore $\boldsymbol{\tau}_i = 1$ when Equation 10 converges. ■

Based on the treated strength we defined, we propose a new linear model estimator:

$$\mathbf{Y}_i = \alpha + \beta \boldsymbol{\tau}_i \tag{12}$$

The parameters α and β can be estimated using linear regression. Then the outcome of unit i is estimated as:

$$\hat{\mathbf{Y}}_i = \hat{\alpha} + \hat{\beta} \boldsymbol{\tau}_i \tag{13}$$

Since $\boldsymbol{\tau}$ varies with the choice of the hyper-parameter p, we choose the p that minimizes the regression loss when estimating the parameters α and β.

When all units are treated, $\mathbf{Z} = \mathbf{1}$, and by Lemma 2, $\boldsymbol{\tau} = \mathbf{1}$. Likewise, when all units are controlled, $\boldsymbol{\tau} = \mathbf{0}$. So the ATE is estimated as:

$$\begin{aligned}\hat{\delta} &= \sum_{i=1}^{N} \hat{\mathbf{Y}}_i(\mathbf{Z} = \mathbf{1}) - \sum_{i=1}^{N} \hat{\mathbf{Y}}_i(\mathbf{Z} = \mathbf{0}) \\ &= \sum_{i=1}^{N} \hat{\mathbf{Y}}_i(\boldsymbol{\tau} = \mathbf{1}) - \sum_{i=1}^{N} \hat{\mathbf{Y}}_i(\boldsymbol{\tau} = \mathbf{0}) \\ &= (\hat{\alpha} + \hat{\beta}) - \hat{\alpha} \\ &= \hat{\beta}\end{aligned} \tag{14}$$

Therefore, the ATE is estimated as $\hat{\beta}$.

6. EXPERIMENT AND RESULTS

In this section, we evaluate our proposed bias correction method in terms of the estimation bias.

6.1 Experiment Settings

a) Data Sets

For the network data, we use "wiki-Vote", "soc-Epinions1" and "soc-Slashdot0811" from [1]. The information of these data sets is listed in Table 2. In the experiment, we converted all these graphs to undirected graphs, and dangling nodes, whose degree is 0, are removed.

b) Outcomes

As we mentioned before, in A/B testing, the true ATE as expressed in Equation 2 is unobservable. So to evaluate our proposed method, we have to generate the outcome by a synthetic outcome model. And the true ATE can also be obtain making use of the synthetic outcome model. In our experiment, we use the synthetic outcome model expressed in Equation 4, by setting $g(x) = \mathbb{1}(x)$ and $g(x) = x$. Without loss of generality, when $g(x) = \mathbb{1}(x)$, α in the model is set to -1.5, and when $g(x) = x$, α is set to 3. For both cases, λ_1 is set to $0.0 \sim 1.0$, and λ_2 is set to $0.0, 0.2, 0.4, 0.6, 0.8, 0.9$.

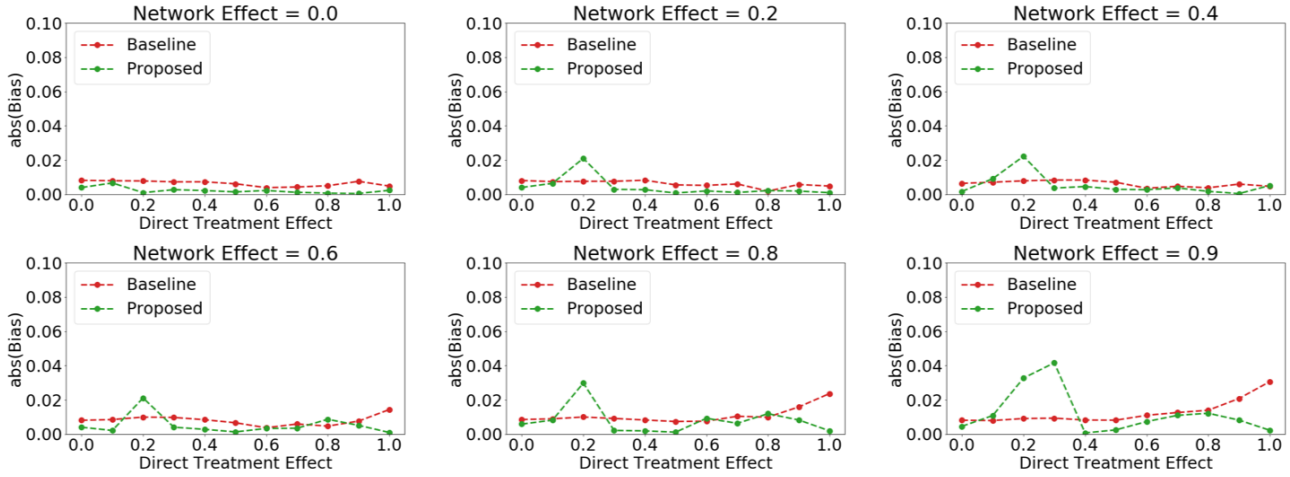
c) Baseline Method

We use the linear model estimator proposed in [5] and discussed in Section 4.4 as our baseline method because it achieved the best estimation accuracy among existing methods. Both our proposed method and the baseline method use cluster randomized sampling.

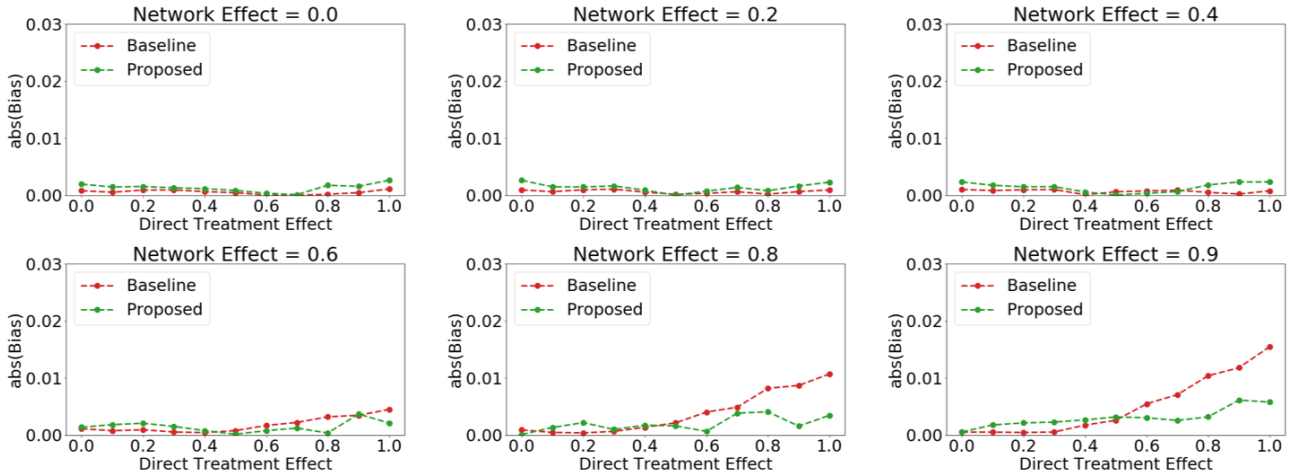
6.2 Results

Since the variances of both our proposed bias correction method and the baseline method are relatively small compared with the bias, we evaluate the proposed method in terms of the absolute value of bias $|bias|$. The smaller the $|bias|$, the better the performance.

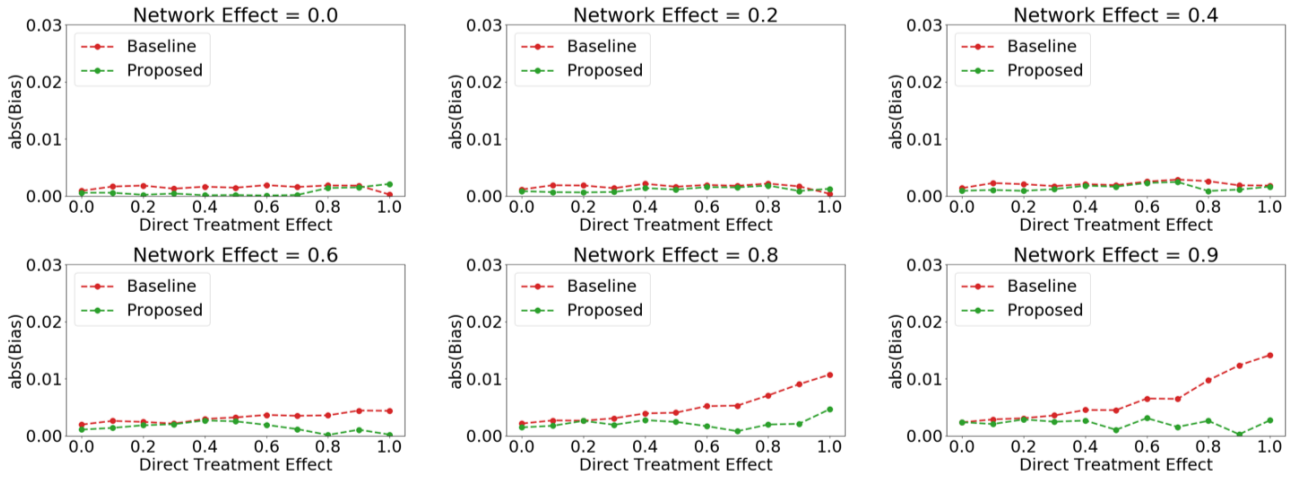
When $g(x) = \mathbb{1}(x)$, the results are shown in Figure 3. Since



(a) wiki-Vote



(b) soc-Epinions1

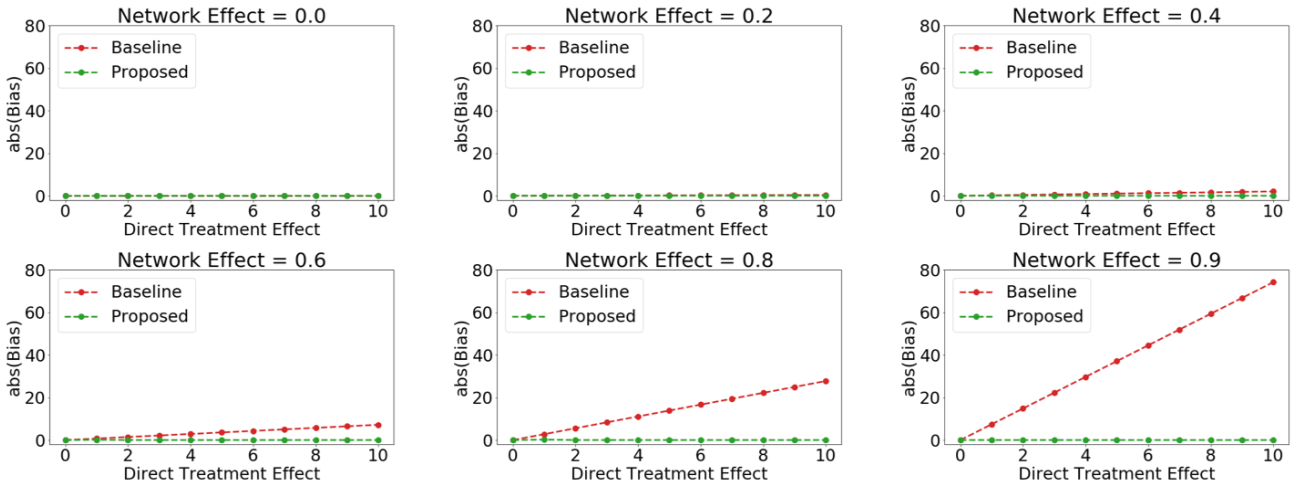


(c) soc-Slashdot0811

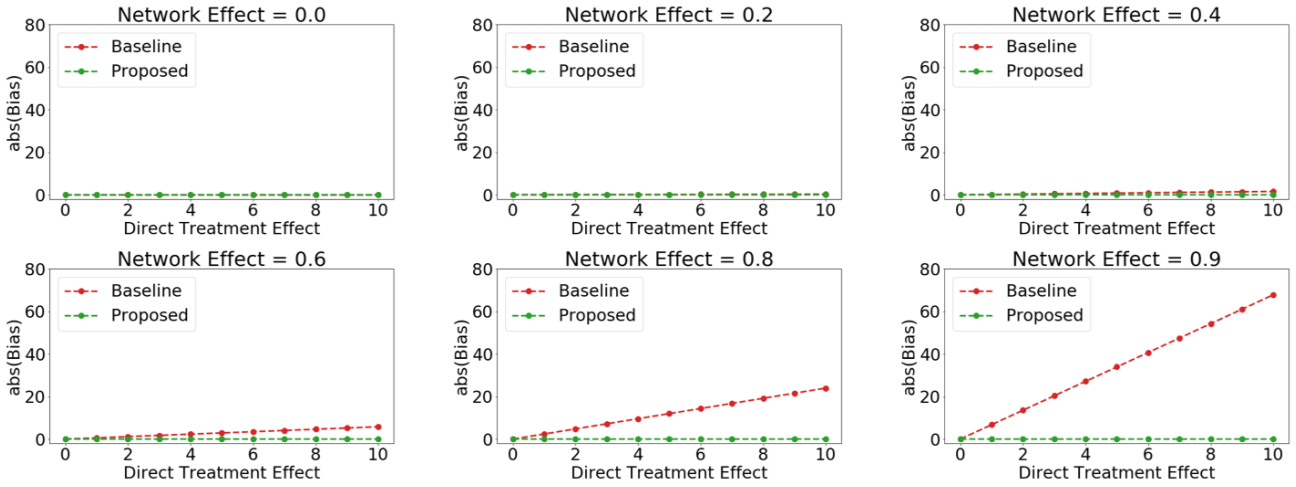
Figure 3: Results on different data sets when $g(x) = \mathbb{1}(x)$.

$\lambda_1 \in [0,1]$, $\lambda_2 \in [0,1]$, and $|bias| < 0.02$, the estimation bias of both baseline method and our proposed method is small. When λ_1 and λ_2 are both large, our proposed method perform slightly better, while the proposed method may be unstable and produces slightly large bias when λ_1 is small.

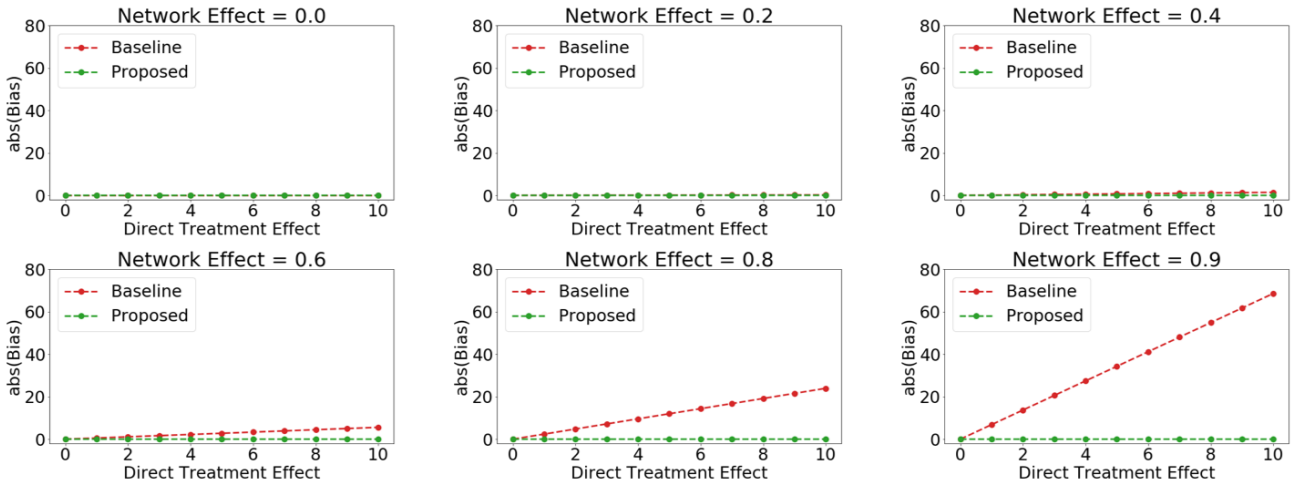
When $g(x) = x$, the results are shown in Figure 4. It is easy to observe that when λ_2 is large, the baseline method produces significant bias while our proposed method still produces small bias. So our proposed method is less likely to underestimate the ATE.



(a) wiki-Vote



(b) soc-Epinions1



(c) soc-Slashdot0811

Figure 4: Results on different data sets when $g(x) = x$.

7. CONCLUSION

In this paper, we argued that without bias correction, existing estimation methods usually tend to underestimate the ATE, and to correct the bias, we proposed a new bias correction method. In our proposed bias correction method, we first defined the treated strength, which makes use of the assignment information of all connected units, then we proposed a new linear model estimator based on the treated strength. Since our proposed method incorporates the infor-

mation of not only 1-hop neighbors, but also 2-hop, 3-hop, ..., n-hop neighbors, it is less likely to be underestimate the ATE. As shown in the experiment results, when $g(x) = x$, in which case the outcome tends to get large due to large network effect, our proposed bias correction method still estimate the ATE accurately while the baseline method underestimates significantly.

Since we use a synthetic outcome model to generate the outcomes and evaluate our proposed method, the performance may depend on the synthetic outcome model we choose. For future research, we plan to evaluate our proposed method based on more kinds of synthetic outcome models.

Acknowledgement

This work was partially supported by JSPS KAKENHI Grant Number 16H02905.

References

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

[2] Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.

[3] 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.

[4] 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.

[5] 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.

[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] 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.

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

[10] 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.

[11] 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.

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

[13] Brendan Kline and Elie Tamer. Some interpretation of the linear-in-means model of social interactions. 2012.

[14] 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.

[15] 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.

[16] 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.