DEIM Forum 2017 D7-3

A/B Testing for Social Network Services with Directed User Graphs

Jian CHEN† and Masashi TOYODA†

† Graduate School of Information Science and Technology, The University of Tokyo
4-6-1 Komaba, Meguro-ku, Tokyo, 153-8505 Japan
E-mail: †{kenn-chen,toyoda}@tkl.iis.u-tokyo.ac.jp

Abstract For social network services (SNSs) like Twitter and Facebook, when a new feature (such as the "poll" in Twitter) is added, it is necessary to evaluate the effect of this new feature. Recent works use network A/B testing for this kind of estimation. However, existing methods regard the social network as undirected graph, in which users always influence each other, while some SNSs such as Twitter should essentially be modeled as directed graph since only followees can influence followers but not vice versa. In this paper, we therefore propose a new A/B testing method for social network with directed user graph, on which we evaluate our algorithm by comparing the result with that of 3 baseline methods.

Key words directed user graph, social network, a/b testing, causal inference

1. INTRODUCTION

A/B testing is the most standard and widely used method for inferring the causal effect. For example, A/B testing is usually adopted to infer the effect of a new medicine. In this case, patients are randomly assigned to either the treatment group or the control group. Patients in the treatment group will take the new medicine (**treated**), while those in the control group will be kept unchanged (**controlled**). Then the Average Treatment Effect (**ATE**) can be obtained by comparing the difference of the average outcomes (such as the severity of the disease) of the two groups.

However, there is a key assumption for traditional A/B testing that is usually called Stable Unit Treatment Value Assumption (**SUTVA**). This assumption states that the outcome of every unit in the experiment only depends on its own assignment (treatment or control) but not the assignments of others. In the case of the above example, which intends to infer the effect of a new medicine, this assumption can be easily satisfied. On the contrary, because of the existence of network effect, users in social network are easily influenced by others, thus making the SUTVA invalid.

To further explain the difference introduced by the network effect, we here consider about the method for obtaining the ATE under SUTVA. Let $\mathcal{U}$ be the set of all units in the experiment with $|\mathcal{U}| = N$, and $\mathbf{Z} \in M^N$ be the random vector representing the assignments of all users, where $M = \{0, 1\}$. $\vec{z}_i = 0$ means user i is under control, and $\vec{z}_i = 1$ means user i is under treatment. We further define $Y_i(\mathbf{Z} = \vec{z})$ as the response (outcome) function of unit i under the assignmnet vector $\vec{z}$. Then the ATE is define as

$$ATE = \mathbb{E}[Y(\vec{1}) - Y(\vec{0})] = \frac{1}{N}\sum_{i=1}^{N}\Big(Y_i(\vec{1}) - Y_i(\vec{0})\Big) \quad (1)$$

where the ATE is the average difference between the outcome under assignment $\vec{z} = \vec{1}$ and the outcome under assignment $\vec{z} = \vec{0}$. However, since it's impossible to both treat and control an experiment unit simultaneously, ATE in equation 1 cannot be computed in an A/B testing experiment. Rather, the ATE is estimated as

$$\hat{ATE} = \frac{1}{N_1}\sum_{\{i|\vec{z}_i=1\}} Y_i(\mathbf{Z} = \vec{z}) - \frac{1}{N_0}\sum_{\{i|\vec{z}_i=0\}} Y_i(\mathbf{Z} = \vec{z}) \quad (2)$$

where N_1 and N_0 are the number of units under treatment and control respectively. Here experiment units are uniformly sampled, which means they are randomly assigned to either the treatment group or the control group. Then the estimated ATE is the difference of outcomes between the two groups. It can be proved that the estimated ATE in equation 2 will converge to the true ATE when $N_1, N_0 \to \infty$ under SUTVA using the Law of Large Numbers [1].

The importance of SUTVA lies in the missing of interference between the treatment group and control group. But in social network, users can influence their friends (such as Facebook) or their followers (such as Twitter) frequently, which poses a problem that by uniform sampling, we can hardly find two groups in which users in one group have no relationship with users in another group. To make this problem clearer, let's consider another example. If we designed a new recommendation algorithm that intends to make users post (or repost) more contents in the social network, using the method we mentioned above, we could randomly select some users to whom we apply the new algorithm, and also

randomly select some users and don't apply the algorithm to them for comparison purpose. Although we expect that the ATE could be estimated by the difference of the outcomes of these two groups, in fact the interference between the two groups can make the estimation inaccurate. Imaging that a user A who is being controlled follows many users who are treated, if the algorithm is actually effective, which means the users followed by user A will post more contents to the social network, it is very likely that user A will also post (repost) more contents. Thus, even though users in control group are not applied the new algorithm, their posted (reposted) contents can increase indirectly due to the application of the new algorithm to users under treatment. So the outcome of control group will be higher, causing the ATE to become smaller.

Several methods have been proposed to deal with the problem of network effect for network A/B testing, and we will introduce some of them in the next section. In this paper, we extended the network A/B testing to directed user graph by proposing a new sampling method, and comparing the efficacy with 3 baseline methods. As far as we know, we are the first to study the A/B testing problem specific to directed user graph.

2. RELATED WORK

A problem that is similar to our network A/B testing problem is formulated by Backstrom et al. [2] named *network bucket testing*. This problem assumes that a certain feature in social network can be normally used by a user only if at least K neighbors of him/her can also use this feature. The set of users satisfying this restriction is called core set, while the set of users used for adding up the K neighbors is called fringe set. Then under this assumption, the task is to estimate the average outcome when the feature is added to the social network. [2] proposed a random walk based method to sample the core set with the tradeoff of the uniformity of sampling and the size of fringe set. [3] also studied this problem and proposed some new algorithms with providing the corresponding variance bound. The main difference between the network bucket testing and network A/B testing we studied here is that network bucket testing doesn't require the control group, and only intends to estimate the overall outcome when all users are treated, while network A/B testing is aimed to estimated the causal effect and requires both treatment group and control group.

For network A/B testing, Ugander et al. [4] proposed a method to first cluster the network and then randomize the assignment on cluster level. They also defined the *network exposure*, with which ATE was obtained as the difference of the outcomes between the users network exposure to treatment and the users network exposure to control. Network exposure can have several definitions. For example, a user i being network exposure to treatment can be defined as user i and qd neighbors of him/her are treated, where d is the number of neighbors of user i and q is a predefined fractional number. They then used the Horviz-Thompson estimator, which is an unbiased estimator, for the estimation of the ATE, and it's variance bound was proved to be linear to the degree of the graph under some constraints. Gui et al. [1] proposed a method which is also based on cluster randomized sampling. But in this method, the graph is first partitioned into equal-sized clusters and the response function Y_i is modeled as a linear function

$$g(\vec{z}_i, \sigma_i) = \alpha + \beta \vec{z}_i + \gamma \sigma_i \tag{3}$$

where $\vec{z}_i$ is the treatment of user i and σ_i is the fraction of treated neighbors. Once the parameters of this linear model are estimated by training on the data of sampled users, the ATE can be obtained as $g(\vec{1}, 1) - g(\vec{0}, 0)$.

Arbour et al. [5] also proposed a method inferring the causal effect making use of the observational data instead of A/B testing.

3. A/B TESTING FOR DIRECTED USER GRAPH

As introduced in the previous section, several methods have been proposed for network A/B testing for undirected user graph. In this section, we discuss about the difference between directed and undirected graph for network A/B testing as well as the clustering methods for directed graph.

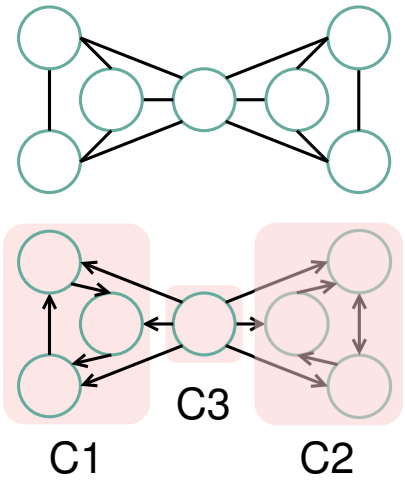


Figure 1: Example of the difference in clustering between undirected and directed graph

3.1 Issues for A/B Testing in Directed Graph

Existing methods of network A/B testing only consider the network as an undirected graph, in which an edge between two users is undirected, enabling them to influence

each other. But many social networks are directed essentially, such as Twitter and Instagram. In these directed user graphs, an edge is directed and represents a follow relationship. A user (called followee) is able to influence the user who follows him/her (called follower) but not vice verse, which gives us an intuition that in order to incorporate the information of unidirectionality, it requires a different clustering method when we use the cluster randomized sampling. As shown in figure 1, the graph on the top is densely connected and it can be regarded as a single cluster, while on the bottom, the structure of this graph is almost the same with the top one, except that the edges become directed. However, the clustering result can be quite different. In this directed graph, the node in cluster $C3$ has many out edges representing the follow relationship. Within cluster $C1$ and $C2$, nodes are strongly connected, and since the node in cluster $C3$ is not followed by other nodes, even if it is influenced by $C1$, it is not able to transmit the influence to $C2$, so $C2$ cannot be influenced by $C1$. And $C1$ will also not be influenced by $C2$ for the same reason.

In addition to the change of edge direction, the structure of the network can also be quite different. In undirected user graphs like Facebook, users connected with an edge are usually friends and the number of friends of a single user cannot be too large, making it easier to partition the graph into well connected clusters. In contrast, a famous person on Twitter can have millions of followers. Imaging a more extreme case that a user is so famous that everyone in the network follows him/her, no matter how we partition the network, this user can always influence users in other clusters. Therefore, the characteristic that some users can have a large number of followers poses another problem for the A/B testing in directed user graph.

3.2 Clustering Directed Graph

In this section, we introduce some of the clustering methods in directed networks. Malliaros et al. [6] made a comprehensive survey about this problem. And based on them, we proposed 3 baseline methods.

The most straightforward and naive method is to transform the directed graph to undirected graph by just ignoring the edge directionality. After the transformation, a large amount of algorithms proposed for undirected graphs can be directly applied. We adopt this method as our baseline-1, in which we first transform our directed user graph to undirected graph and apply the method proposed in [1]. But since this naive method ignores the information of directionality, the accuracy is expected to be low.

An alternative method is to transform directed graph to undirected graph by partially maintaining the directionality. For example, it can be transformed to an weighted undirected graph, in which the information of directionality is maintained as the weight associated with each edge. We adopt this method as our baseline-2, in which we transform the directed graph to undirected graph by ignoring the directionality, but all edges having no corresponding reverse edge will be deleted.

Methods for clustering on undirected graph usually requires an objective criterion such as modularity and normalized cut. So these methods can be readily extended to directed graph by taking the directionality into consideration. We created baseline-3 by extending the clustering algorithm proposed in [1], which is based on label propagation and tries to maximize the internal edges of each cluster in a greedy way.

Other methods for clustering directed graph include information theoretic based methods, probabilistic model based methods, Blockmodeling methods and some others.

3.3 Two-Step Clustering Method for Directed User Graph

As mentioned in section 3.1, A/B testing in directed user graph faces two major problems. The first is that the existence of directionality of edges makes it necessary to design a new sampling method which incorporates this extra information. The second is that users who has too many followers will still have a lot of influence across clusters.

The first way to extend A/B testing from undirected user graph to a directed one may be choosing a suitable clustering method for directed network as discussed in [6]. However, the second problem as we mentioned above is still hard to be avoided by making use of these common clustering methods.

The second way that seems plausible is to ignore those influential users and just sample users who don't have too many followers. In this way, the influence across clusters can indeed be reduced dramatically, and thus the treatment group can have much less influence on control group and vice versa. It seems that this kind of way could estimate the ATE more accurately. However, ignoring influential users can bring significant bias to the estimated result. Because by adopting this sampling method, all tested users are those who have relative small number of followers, but in the real case that a feature is finally added to the social network (which is equivalent to that every one in the social network is treated), influential users actually contribute a lot to the effect. Imaging that if Twitter added a new feature, how about the results between the case that most famous users use it and that no famous user uses it. The former is expected to make this feature more effective. Therefore, sampling by ignoring influential users can underestimate the ATE.

Given the defects of the above methods, we proposed a new sampling methods for directed user graph, called two-

step clustering (TSC) method, which composes the following steps:

(1) We first separate the network into two parts S_1 and S_2. S_1 includes influential users whose in-degrees are above the threshold D, and the rest constitutes S_2.

(2) Then we cluster on S_1 using the same clustering method as in baseline 3.

(3) For users in S_2, they are included in one of the clusters obtained in the previous step by majority vote. More specifically, a user in S_2 will be included to the cluster that most his followees belong to.

(4) Finally, we randomly give the assignment (treatment or control) to each cluster with equal probability.

4. EXPERIMENTS

One of the main problems of A/B testing is that there is no ground truth for the ATE, which means it is impossible for us to evaluate our estimation result in a real A/B testing experiment. Therefore, we instead use an outcome model to simulate the outcome process.

4. 1 Datasets

Our experiments are conducted on three real directed networks from [7] and one synthetic directed network. These networks are shown as below.

(1) Wikipedia vote network

This dataset contains the voting data of Wikipedia, in which administrators vote for promoting users to adminship and it has 7115 nodes and 103689 edges. An directed edge represents a vote relationship.

(2) Epinions social network

This dataset is from a consumer review site Epinions.com. An edge in this network represents a user trusts another user. And this data set contains 75879 nodes and 508837 edges.

(3) Slashdot social network, November 2008

This dataset is from a technology-related news website slashdot.org and an edge from user i to user j means user i tagged user j as a friend or foe. This data set contains 77360 nodes and 905468 edges.

(4) Growing Network

This is a synthetic growing network with 20,000 nodes.

4. 2 Outcome Model and Estimator

Our outcome model is similar to that in [1] except that we extended it to directed network and changed the outcome from binary value to non-negative real number. And the model is expressed as

$$Y_{i,t}(\mathbf{Z} = \vec{z}) = \lambda_0 + \lambda_1 \vec{z}_i + \lambda_2 \frac{\mathbf{A}_{i.} Y_{t-1}}{d_i} + U_{i,t} \qquad (4)$$

where $Y_{i,t}$ is the response function that returns the outcome of user i at time t. This response function is modeled as a linear function of the assignment of the user themselves and the average outcome of their neighbors, as well as some user specific traits (expressed as Gaussian noise). For every users, even if the assignment and the average outcome of neighbors are the same, the outcome can still be different due to some other factors, such as the personality, age, etc. So the term (user specific traits) is add to capture this kind of noise. In this model, λ_0 is the intercept (set as -1.5 in this experiment). $\vec{z}_i \in \{0, 1\}$ is the assignment of user i and λ_1 is the strength of the treatment effect; $\mathbf{A}$ is the adjacency matrix of the directed network and d_i is the out-degree of user i, so $\frac{\mathbf{A}_{i.} Y_{t-1}}{d_i}$ is the average outcome of the followees of user i, and λ_2 is therefore the strength of the network effect. $U_{i,t} \sim \mathcal{N}(0, 1)$ captures user specific traits.

The final outcome can be obtained by setting $Y_{i,0} = 0$ and then iteratively running equation 4. In our experiment, we iteratively run it 5 times.

We use the same estimator as in equation 3, but the σ_i now means the proportion of the treated followees.

4. 3 Experiment Settings

In order to evaluate the efficacy of our proposed method for A/B testing in directed user graph, we compared it with 3 baselines we setup in section 3. 2. For clarification purpose, we again list the baselines here:

- Baseline-1: transforming directed graph into undirected one by ignoring the directionality of the edge and then applying existing method for undirected graph.
- Baseline-2: transforming directed graph by first deleting edges without a reverse edge and then ignoring the directionality, and finally applying existing method for undirected graph.
- Baseline-3: extending the clustering algorithm in [1] to partition directed graph and then directly estimating on this directed graph.

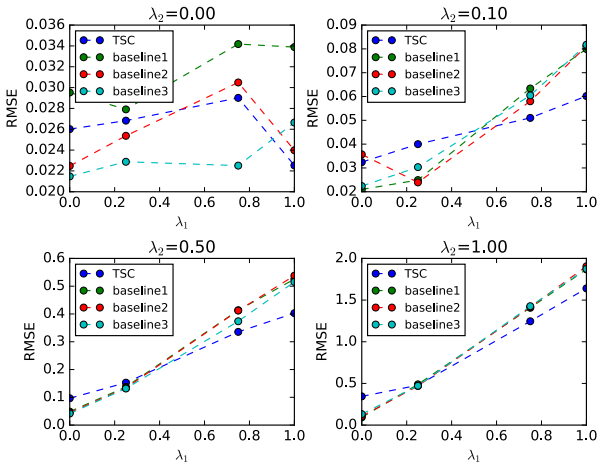
Since there is no ground truth for real A/B testing experiments, we here use the outcome model expressed as equation 4 to generate synthetic outcome data based on the structure of the network and the assignment. The true ATE can be computed as

$$ATE = \frac{1}{N} \sum_{i=1}^{N} \Big(Y_{i,5}(\mathbf{Z} = \vec{1}) - Y_{i,5}(\mathbf{Z} = \vec{0}) \Big) \qquad (5)$$

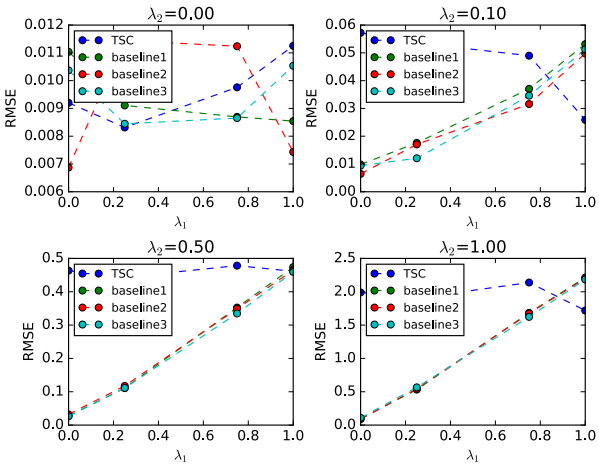
where N is the number of users in the network and the outcome will be generated by iteratively running the outcome model 5 times. Then we will use our proposed method (TSC) and 3 baseline methods to estimate the ATE, and finally we will comparex them based on Rooted Mean Square Error (RMSE) of the estimated ATE.

4. 4 Experiment Result

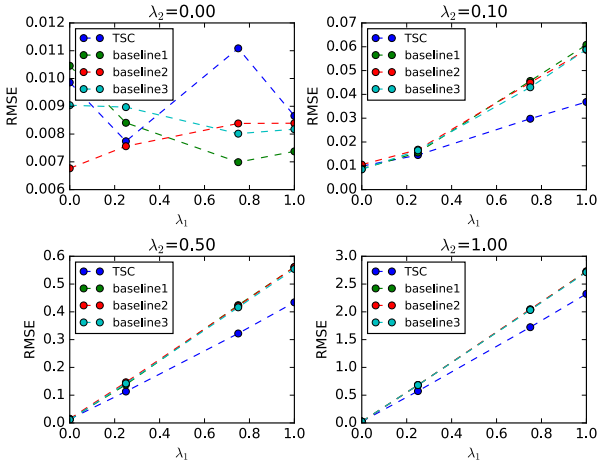
The results are shown in figure 2. In our experiment, λ_1 was set to $0, 0.25, 0.75, 1$, and λ_2 was set to $0, 0.1, 0.5, 1$. From



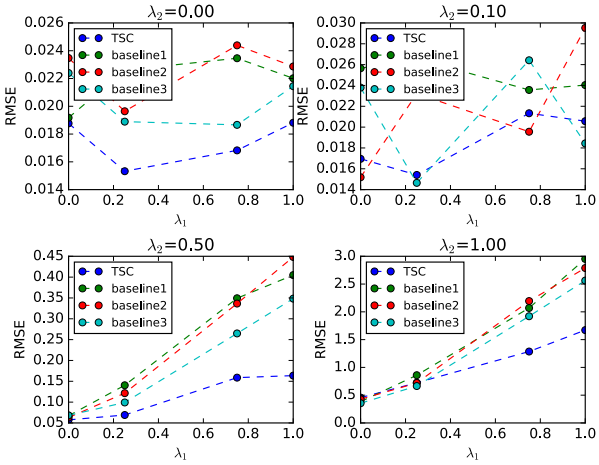
(a) Wikipedia vote network



(b) Epinions social network



(c) Slashdot social network, November 2008



(d) Growing Network

Figure 2: The results for comparing TSC with baseline methods on different directed graphs

the results, we have the following observations.

For all directed user graphs, the RMSE grows when we increase λ_2 (the strength of network effect) or λ_1 (the strength of treatment effect).

The performance of the 3 baseline methods is almost the same, among which baseline 3 is slightly better than the other two mthods. Since baseline 3 incorporates more directional information in the clustering algorithm, it's reasonable that it outperforms baseline 1 and baseline 2.

Except for the 'Epinions social network', the TSC method outperforms all the baseline methods. This shows the efficacy of our proposed method.

However, the TSC method performs poorly for 'Epinions social network', and the RMSE doesn't change a lot when we increase the strength of network effect (λ_1). This is caused by the significant differences among the sizes of the clusters obtained by the TSC. The sizes of the clusters are shown in figure 3, in which we sort the clusters by their sizes and only the top 20 clusters are drawn in this figure. It's easy to see that the differences among clusters in the other three graphs are not so significant as that in the 'Epinions social network', in which the biggest cluster is 6 times bigger than the second biggest one. As a result, most nodes are in the same cluster, causing the error of the estimation being greater.

5. CONCLUSION AND FUTURE WORK

Our proposed method, two-step clustering method (TSC), not only takes the directionality of edge into consideration, but also tries to reduce the cross influence of influential users. And as confirmed by our experiment, the TSC method produces the best results compared with the other three baseline methods in most occasions.

However, for some graphs such as 'Epinions social network', the performance of the TSC can be poor. Therefore,

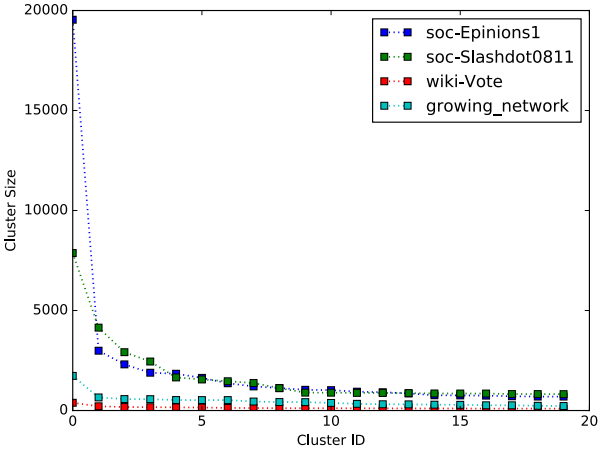


Figure 3: The cluster sizes of each cluster partitioned by the TSC method

we plan to improve the clustering algorithm to solve this problem and make it more robust.

References

[1] 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. ACM, 2015.

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

[3] 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.

[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] David Arbour, Dan Garant, and David Jensen. Inferring network effects from observational data. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 715–724, New York, NY, USA, 2016. ACM.

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

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