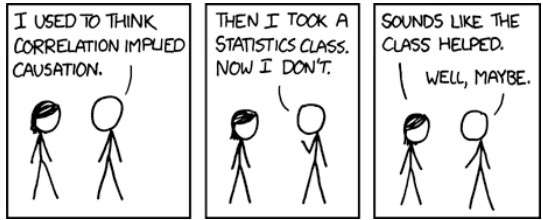
Estimation and Testing Methods for Causal Inference with Interference

Kevin Han

Stanford University

July 15, 2023

Correlation does not imply causation



- Correlation does not imply causation.
- Causation is very important in many areas (economics, education, psychology, medicine, product development).
- Examples: Effect of job training program on long-term employment rates (Riccio et al., 1989; Friedlander and Robins, 1995), effect of information intervention on student absence (Rogers and Feller, 2017), effect of taking Aspirin on headache (Imbens and Rubin, 2015).
- We could argue that someone else taking an aspirin in a different location cannot have an effect on my headache. While in job training programs, the outcomes for one participant may be affected by the number of people trained because of increased competition for certain jobs.

Pipeline of causal reasoning

1. Conduct randomized experiments.
   - May not be able to conduct randomized experiments.
2. Collect data from randomized experiments.
   - Non-compliance.
3. Analyze data (estimation and inference) and draw conclusion.
   - Interference (SUTVA violation).

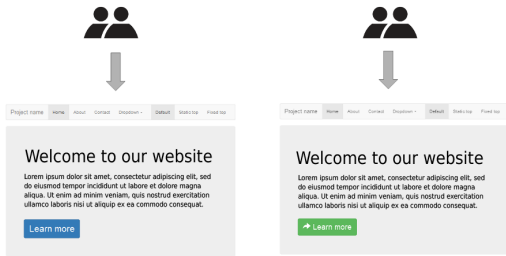
Interference

- In the most general case, each unit in an experiment on n units is associated with 2^n potential outcomes, i.e., every possible assignment vector leads to a different outcome for each individual.
- SUTVA or no interference assumption implies that each unit is associated with only two potential outcomes, i.e., for each unit i, there are only two potential outcomes $Y_i(0)$ and $Y_i(1)$. This assumption significantly reduces the number of potential outcomes.

Interference

- In the most general case, each unit in an experiment on n units is associated with 2^n potential outcomes, i.e., every possible assignment vector leads to a different outcome for each individual.
- SUTVA or no interference assumption implies that each unit is associated with only two potential outcomes, i.e., for each unit i, there are only two potential outcomes $Y_i(0)$ and $Y_i(1)$. This assumption significantly reduces the number of potential outcomes.
- Violation of SUTVA has been found in many applications, including politics (Sinclair et al., 2012), education (Hong and Raudenbush, 2006; Rosenbaum, 2007), economics (Sobel, 2006; Manski, 2013), and public health (Halloran and Struchiner, 1995).
  - In job training programs, as we argued above, large number of participants may create increased competition for certain jobs, and hence affect outcomes of each individual participant.

Interference

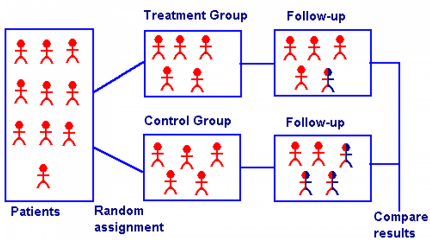


- In this example, a website wants to test which UI design yields higher click rate.
- If the action of click or not for one user does not depend on what other users see on their webpages.
  - No interference.
- Otherwise, there exists interference.

Two questions

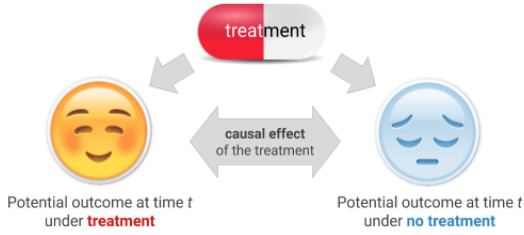
Testing

- Detecting interference in randomized experiments.



Estimation

- Causal effect estimation under interference.



- Solutions to the above two questions provide practitioners tools to do causal inference under interference.

Detecting interference in randomized experiments

Randomized experiments in the technology industry (a.k.a. A/B tests) are often implemented with increasing treatment allocation: the new treatment is gradually released to an increasing number of units through a sequence of randomized experiments. In such a case, a valid testing procedure for interference could provide valuable and timely feedback on the choice of designs and help experimenters update development road-maps accordingly.

Contributions

- Detecting Interference in Online Controlled Experiments with Increasing Allocation. Kevin Han, Shuangning Li, Jialiang Mao, Han Wu. KDD 2023. *arXiv:2211.03262*, 2022.

Causal effect estimation under interference

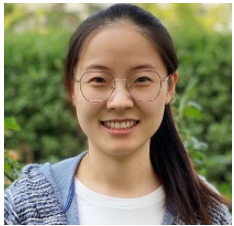
When analyzing data from randomized experiments, researchers commonly assume that one unit's assignment does not affect another unit's response, i.e., SUTVA holds. However, when experimental units interact with each other, SUTVA is often untenable. Estimation under interference is of considerable scientific interest in many settings.

Contributions

- Population Interference in Panel Experiments. Kevin Wu Han, Iavor Bojinov, Guillaume Basse. Under revision at *Journal of Econometrics*. *arXiv:2103.00553*, 2021.
  - Estimation and inference for causal effects in panel experiments.
- Model-Based Regression Adjustment with Model-Free Covariates for Network Interference. Kevin Han, Johan Ugander. Accepted by *Journal of Causal Inference*. *arXiv:2302.04997*, 2023.
  - Estimation and inference of the global average treatment effect under network interference.

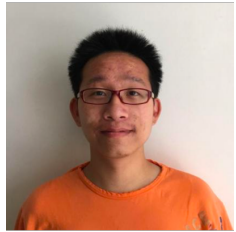
Detecting Interference in Online Controlled Experiments with Increasing Allocation

Joint work with



Shuangning Li

Jialiang Mao (LinkedIn)

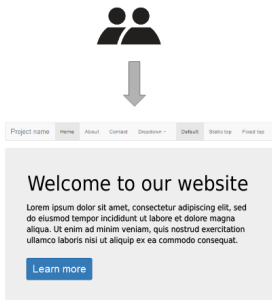


Han Wu

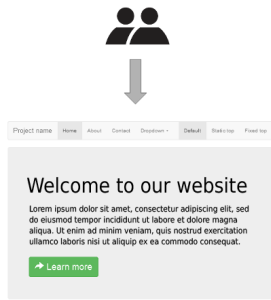
Han, Li, Mao & Wu. **Detecting Interference in Online Controlled Experiments with Increasing Allocation.** KDD 2023. *arXiv:2211.03262*, 2022.

A/B Testing

- A/B testing has been adopted by the technology industry to guide product development and make business decisions.



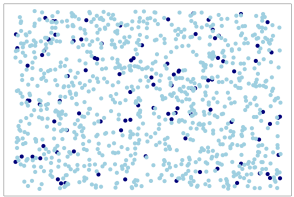
Click rate: 52 %



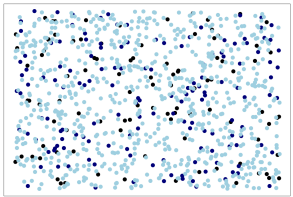
72 %

A/B tests with increasing allocation

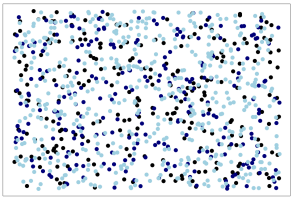
- In practice, A/B tests are often implemented with increasing treatment allocation: the new treatment is gradually released to an increasing number of units through a sequence of randomized experiments.



10%



25%



50%

A/B tests with increasing allocation

- The most straightforward statistical analysis following A/B tests is to compute the difference-in-means estimator, i.e., the difference in the average of outcomes of the treatment group and that of the control group.

A/B tests with increasing allocation

- The most straightforward statistical analysis following A/B tests is to compute the difference-in-means estimator, i.e., the difference in the average of outcomes of the treatment group and that of the control group.
- Under the classical Stable Unit Treatment Value Assumption (SUTVA), when we compute the difference-in-means estimator for any single randomized experiment in an A/B test with increasing allocation, the value of the estimator should not change by much.

A motivating example

- However, in some real-world scenarios, we observe drastic change in the difference-in-means estimators throughout the experiments.

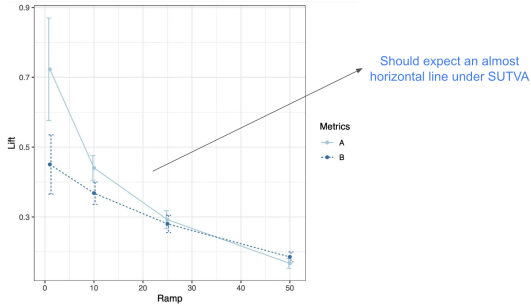


Figure: An A/B test at LinkedIn with increasing allocation. A and B are different outcome metrics.

- On the x-axis, we show the percentage of units that are in the treatment group; on the y-axis, we show the value of the difference-in-means estimator.

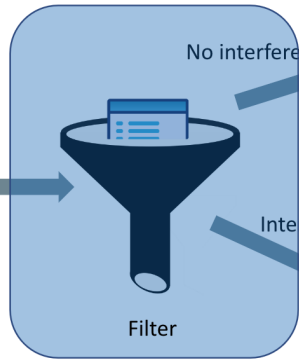
A motivating example

- The difference-in-means estimator decreases as the treatment is released to more units.
  - What causes this phenomenon?
  - Could it be purely due to randomness?
  - Is the SUTVA violated in this case?

Detecting interference from A/B tests with increasing allocation

Our contribution

A/B Tests with Increasing Allocation



Standard statistical analysis

Cluster randomized trials
Budget-split platform
…

- Scalable, parallelable, "almost for free!"
- Agnostic to interference mechanism

Platforms that carefully deal with interference (Ugander et al., 2013; Liu et al., 2021; Bojinov et al., 2022)

Setup

- Suppose that there are K experiments on a population of n units.
- Let π_k be the marginal treatment probability of the k^{th} experiment. The treatment probabilities satisfy $\pi_1 < \pi_2 < \cdots < \pi_K$.
- For the k^{th} experiment, each unit is randomly assigned to treatment group with probability π_k.
- Once being assigned into the treatment group, a unit will stay in the treatment group in subsequent experiments.

Setup

Specifically, the experiments are implemented in the following way.

- In the first experiment, each unit i is randomly assigned a treatment $W_{i,1}$, where

$$W_{i,1} \sim \text{Bernoulli}(\pi_1) \text{ independently.}$$

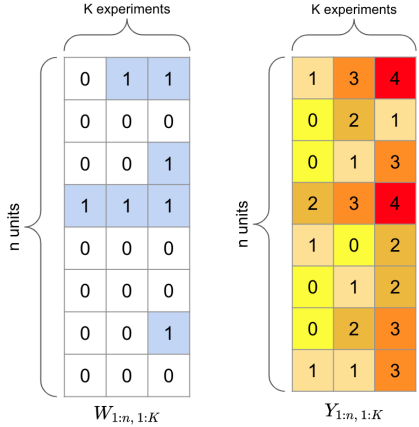
- In the subsequent experiments, each $W_{i,k}$ is sampled from the following distribution independently:

$$\begin{cases} W_{i,k} \sim \text{Bernoulli}\left((\pi_k - \pi_{k-1})/(1-\pi_{k-1})\right), & \text{if } W_{i,k-1} = 0; \\ W_{i,k} = 1, & \text{if } W_{i,k-1} = 1. \end{cases}$$

  This formulation guarantees that if we look at the k^{th} experiment alone, then the treatments $W_{i,k}$'s are i.i.d. Bernoulli(π_k).

Setup

- Assignment matrix $W_{1:n,1:K} \in \mathbb{R}^{n \times K}$ that records the assignments for all n units in K experiments.
- Outcome matrix $Y_{1:n,1:K} \in \mathbb{R}^{n \times K}$ that records the outcomes for all n units in K experiments.



Two sources of interference

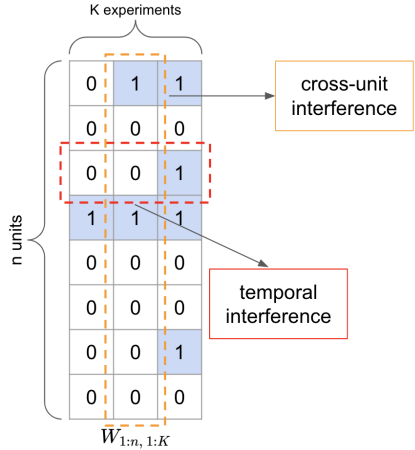


Figure: An illustration of two sources of interference - cross-unit interference and temporal interference.

No cross-unit interference

Hypothesis (No cross-unit interference)

For each unit i and $k \in \{1, \cdots, K\}$, $Y_{i,k}(w_{1:n,1:K}) = Y_{i,k}(\tilde{w}_{1:n,1:K})$ if $w_{i,1:K} = \tilde{w}_{i,1:K}$.

- The hypothesis states that the outcomes of unit i depend only on the treatments of unit i and not on the treatments of others.
- We develop methods that test against the no cross-unit interference inspired by Athey et al. (2018).
- There has been many methods for testing interference with a single experiment (Athey et al., 2018; Pouget-Abadie et al., 2019; Basse et al., 2019; Puelz et al., 2022).
- However, none of these works addresses the problem of multiple experiments, and their methods tend to have lower power when directly applied in our setup.

Candidate exposures

- For each experiment k and each unit i, we use $H_{i,k} = h_i(W_{-i,k}) \in \mathbb{R}^m$ to denote the candidate exposure. $W_{-i,k}$ is the treatments given to all other units except unit i in the k-th experiment.
- We use the form $h_i(W_{-i,k})$ to emphasize that the candidate exposure depends on other units' treatments.
- We will use the candidate exposures explicitly to construct test statistics later.
- We do not require the candidate exposure to be correctly specified.

Candidate exposures

Candidate exposures capture the potential form of interference.

Example (Network experiments (Cai et al., 2015; Basse and Airoldi, 2018))

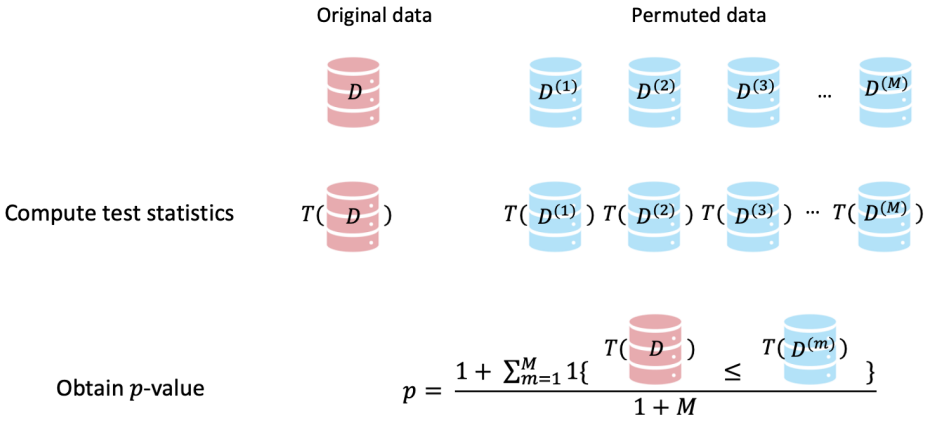
Experimenters may suspect that a user's outcome is influenced by treatments of "friends", i.e., users connected through the social network. Thus in this example, some plausible choices of candidate exposures include the fraction of friends who are treated, and the number of friends who are treated.

Example (Marketplace experiments (Holtz et al., 2020; Johari et al., 2022))

When we consider marketplace competition, advertisers are the subjects of treatment. Here the sales of an advertiser may be impacted by the treatments of competitors, i.e., advertisers that sell similar products. In this application, experimenters can choose candidate exposures to be the number of treated advertisers that sell products of the same category, or an average of treatments given to other advertisers weighted by some product similarity metric.

Permutation test

- We consider permutation tests:



Testing with two experiments

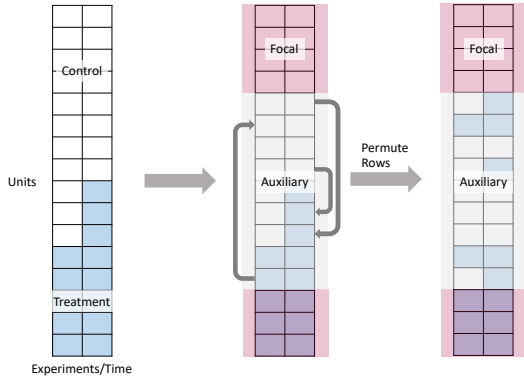


Figure: An illustration of Algorithm 1. After selecting the set of focal units and auxiliary units, we randomly permute rows of the treatment matrix and compute test statistics and p-values based on the permuted data.

Testing with two experiments

Algorithm 1 Testing for interference effect (two experiments).

Input: Datasets $\mathcal{D}_1 = (W_{1:n,1}, X_{1:n}, Y_{1:n,1}, H_{1:n,1})$, $\mathcal{D}_2 = (W_{1:n,2}, X_{1:n}, Y_{1:n,2}, H_{1:n,2})$, exposure function h, test statistic T.

1. Let $\mathcal{I}_{\text{nc}} = \{i : W_{i,1} = W_{i,2}\}$ be the set of units whose treatment didn't change over the experiments.
   Randomly sample a subset of $\mathcal{I}_{\text{nc}}$ of size $n/2$. We call the subset $\mathcal{I}_{\text{foc}}$. Let $\mathcal{I}_{\text{aux}} = [n] \setminus \mathcal{I}_{\text{foc}}$.
2. Take the difference of $Y_{\text{foc},2}$ and $Y_{\text{foc},1}$: let $Y^{\text{diff}}_{\text{foc}} = Y_{\text{foc},2} - Y_{\text{foc},1}$. Compute a test statistic $T^{(0)} = T(W_{\text{foc},1:2}, X_{\text{foc}}, Y^{\text{diff}}_{\text{foc}}, H_{\text{foc},1:2})$ that captures the importance of H in predicting Y^{diff}.
3. **For** $b = 1, \dots B$:
   Randomly permute treatments for the auxiliary units of the data: $\widetilde{W}^{(b)}_{i,1:2} = W_{\sigma^{(b)}(i),1:2}$ for $i \in \mathcal{I}_{\text{aux}}$, for some permutation $\sigma^{(b)}$ of $\mathcal{I}_{\text{aux}}$.
   Recompute the candidate exposure for the focal units: $\widetilde{H}^{(b)}_{i,k} = h_i(W_{\text{foc}\setminus\{i\},k}, \widetilde{W}^{(b)}_{\text{aux},k})$ for $i \in \mathcal{I}_{\text{foc}}$ and $k \in \{1, 2\}$.
   Recompute the test statistic: $T^{(b)} = T(W_{\text{foc},1:2}, X_{\text{foc}}, Y^{\text{diff}}_{\text{foc}}, \widetilde{H}^{(b)}_{\text{foc},1:2})$.
   End For

Output: The p-value

$$p = \frac{1}{B+1}\left(1 + \sum_{b=1}^{B} \mathbb{1}\left\{T^{(0)} \leq T^{(b)}\right\}\right).$$

Testing with two experiments

- The idea of choosing focal units is from Athey et al. (2018).
- In Athey et al. (2018), the choice of focal units cannot depend on the treatment assignments $W_{1:n}$, whereas in Algorithm 1, the focal units are randomly chosen from those whose treatments didn't change.
- Instead of regenerating treatments as in Athey et al. (2018), Algorithm 1 permutes the treatments of the auxiliary units. This change is necessary to guarantee the procedure's validity; the choice of focal units depends on the treatment vector, and thus naively regenerating treatments will not give a valid procedure anymore.
- Y^{diff} is used to reduce variance.

Testing with a time fixed effect model

- Up to now, we allow the existence of "arbitrary time effect".
- In particular, no cross-unit interference hypothesis allows the outcome $Y_{i,k}$ to depend on the treatments in other experiments, and does not restrict the relationship among outcomes in different experiments.
- This brings flexibility and generality, but it could reduce the power of the testing procedure.

Testing with a time fixed effect model

Assumption (No temporal interference)

For each unit i and $k \in \{1, \cdots, K\}$, $Y_{i,k}(w_{1:n,1:K}) = Y_{i,k}(\tilde{w}_{1:n,1:K})$ if $w_{1:n,k} = \tilde{w}_{1:n,k}$.

Under no temporal interference assumption, we can write potential outcomes as $Y_{i,k}(w_{1:n,k})$.

Assumption (Time fixed effect)

For any $w_{1:n} \in \{0,1\}^n$, $i \in \{1, \ldots, n\}$ and $k \in \{1, \ldots, K\}$,

$$Y_{i,k}(w_{1:n}) = \alpha_i(w_{1:n}) + u_k + \epsilon_{i,k}(w_{1:n}).$$

The random variables $\epsilon_{i,1}(w_{1:n}), \ldots, \epsilon_{i,K}(w_{1:n})$ are zero mean, and are independently and identically distributed, independently of functions $\alpha_{1:n}$, variables $u_{1:K}$, treatments $W_{1:n,1:K}$, covariates $X_{1:n}$ and other errors $\epsilon_{j,l}$ for $j \neq i$.

Testing with a time fixed effect model

Under no temporal interference and time fixed effect assumption, no cross-unit interference hypothesis becomes

Hypothesis (Two-way fixed effect)

For any $w \in \{0,1\}$, $i \in \{1,\ldots,n\}$ and $k \in \{1,\ldots,K\}$,

$$Y_{i,k}(w) = \alpha_i(w) + u_k + \epsilon_{i,k}(w), \tag{1}$$

such that the vectors $\epsilon_{1:n,1}(w),\ldots,\epsilon_{1:n,K}(w)$ are i.i.d., and independent of functions $\alpha_{1:n}$, vector $u_{1:K}$, treatments $W_{1:n,1:K}$, covariates $X_{1:n}$ and other errors $\epsilon_{j,l}(w)$ for $l \neq k$.

Testing with a time fixed effect model

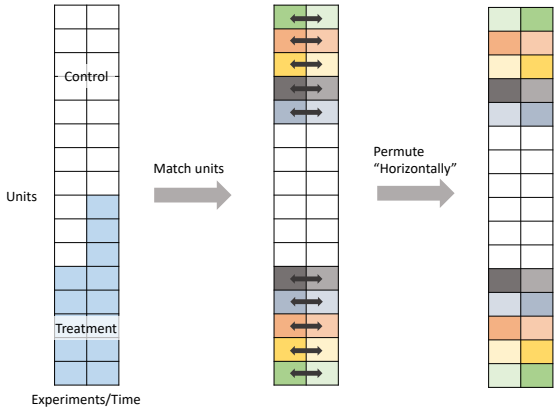


Figure: An illustration of Algorithm 2. Algorithm 2 permutes the outcomes across experiments, whereas Algorithm 1 permutes the treatments across units.

Why we can do horizontal permutations under time fixed effect assumption

To motivate the permutation test, consider two units i and j. Assume that i has been in the treatment group the whole time while j has been in the control group the whole time. Under the null,

- for the first experiment,
  $Y_{i,1} - Y_{j,1} = (\alpha_i(1) + u_1 + \epsilon_{i,1}(1)) - (\alpha_j(0) + u_1 + \epsilon_{j,1}(0)) = \alpha_i(1) - \alpha_j(0) + \epsilon_{i,1}(1) - \epsilon_{j,1}(0)$,
- and for the second experiment,
  $Y_{i,2} - Y_{j,2} = \big(\alpha_i(1) + u_2 + \epsilon_{i,2}(1)\big) - \big(\alpha_j(0) + u_1 + \epsilon_{j,1}(0)\big) = \alpha_i(1) - \alpha_j(0) + \epsilon_{i,2}(1) - \epsilon_{j,2}(0)$.
- Thus,

$$\begin{aligned} Y_{i,1} - Y_{j,1} &= \alpha_i(1) - \alpha_j(0) + \epsilon_{i,1}(1) - \epsilon_{j,1}(0) \\ &\stackrel{d}{=} \alpha_i(1) - \alpha_j(0) + \epsilon_{i,2}(1) - \epsilon_{j,2}(0) = Y_{i,2} - Y_{j,2}. \end{aligned} \tag{2}$$

  - To put it simply, under the null, $Y_{i,1} - Y_{j,1}$ has the same distribution as $Y_{i,2} - Y_{j,2}$.

Why we can do horizontal permutations under time fixed effect assumption

However, the two distributions could be different when there is interference.

- Consider a simple model:

$$Y_{i,k} = W_{i,k} H_{i,k} + \epsilon_{i,k}, \tag{3}$$

  where $H_{i,k}$ is the fraction of neighbors of unit i treated in experiment k, and $\epsilon_{i,k}$'s are some i.i.d. zero mean errors.
- Under this model, $Y_{i,1} - Y_{j,1} = H_{i,1} + \epsilon_{i,1} - \epsilon_{j,1}$ and $Y_{i,2} - Y_{j,2} = H_{i,2} + \epsilon_{i,2} - \epsilon_{j,2}$.
- When the number of neighbors of unit i is large, by law of large numbers, we have $H_{i,1} \approx \pi_1$ and $H_{i,2} \approx \pi_2$. We can then observe that $Y_{i,1} - Y_{j,1}$ and $Y_{i,2} - Y_{j,2}$ have different distributions; in particular, they have different means.

Testing with a time fixed effect model

Algorithm 2 Testing for interference effect (two experiments, time fixed effect model).

Input: Datasets $\mathcal{D}_1 = (W_{1:n,1}, X_{1:n}, Y_{1:n,1}, H_{1:n,1})$, $\mathcal{D}_2 = (W_{1:n,2}, X_{1:n}, Y_{1:n,2}, H_{1:n,2})$, matching algorithm m, test statistic T.

1. Let $\mathcal{I}_0 = \{i : W_{i,1} = W_{i,2} = 0\}$ and $\mathcal{I}_1 = \{i : W_{i,1} = W_{i,2} = 1\}$.
2. For each i in $\mathcal{I}_1$, match an index $j \in \mathcal{I}_0$ to i (with no repeat): let $m(i)$ be the matched index of i. Let $\mathcal{I}_m = \{m(i) : i \in \mathcal{I}_1\}$ be the set of matched indices.
3. For each $k \in \{1, 2\}$, compute $Y^{\text{diff}}_{\mathcal{I}_1,k} = \left(Y_{i,k} - Y_{m(i),k}\right)_{i \in \mathcal{I}_1}$, which is the vector of differences between the outcomes of the treated units and those of the matched units.
   Compute a test statistic $T^{(0)} = T(Y^{\text{diff}}_{\mathcal{I}_1,1:2}, X_{\mathcal{I}_m}, H_{\mathcal{I}_m,1:2}, X_{\mathcal{I}_1}, H_{\mathcal{I}_1,1:2})$.
4. **For** $b = 1, \ldots B$:
   For each $i \in \mathcal{I}_1$:
   Randomly permute outcomes across experiments: $\widetilde{Y}^{(b)}_{i,k} = Y_{i,\sigma_{i,b}(k)}$ and $\widetilde{Y}^{(b)}_{m(i),k} = Y_{m(i),\sigma_{i,b}(k)}$ for some permutation $\sigma_{i,b}$ of $\{1, 2\}$.
   End For
   Recompute $\widetilde{Y}^{\text{diff},(b)}_{\mathcal{I}_1,k} = (\widetilde{Y}^{(b)}_{i,k} - \widetilde{Y}^{(b)}_{m(i),k})_{i \in \mathcal{I}_1}$.
   Recompute the test statistic: $T^{(b)} = T(\widetilde{Y}^{\text{diff}(b)}_{\mathcal{I}_1,1:2}, X_{\mathcal{I}_m}, H_{\mathcal{I}_m,1:2}, X_{\mathcal{I}_1}, H_{\mathcal{I}_1,1:2})$.
   End For

Output: The p-value

$$p = \frac{1}{B+1}\left(1 + \sum_{b=1}^{B} \mathbb{1}\left\{T^{(0)} \leq T^{(b)}\right\}\right).$$

Remarks on algorithms

- Algorithm 1 and 2 can be easily extended to more than two experiments with slight modifications on the choice of focal units.
- For Algorithm 1, test statistics can be any statistics that reflect the predictive power of candidate exposures on outcomes.
- For Algorithm 2, one simple choice of test statistic is the difference-in-differences statistic:

$$T(Y^{\text{diff}}_{\mathcal{I}_1,1:2}, X_{\mathcal{I}_m}, H_{\mathcal{I}_m,1:2}, X_{\mathcal{I}_1}, H_{\mathcal{I}_1,1:2}) = \left|\text{mean}(Y^{\text{diff}}_{\mathcal{I}_1,2}) - \text{mean}(Y^{\text{diff}}_{\mathcal{I}_1,1})\right|,$$

where $\mathcal{I}_1$ and $\mathcal{I}_m$ are defined in the first step of Algorithm 2.

  - No graph needed!

We can also bring the candidate exposures into the picture by using

$$\left|\text{Corr}\left[Y^{\text{diff}}_{\mathcal{I}_1,2} - Y^{\text{diff}}_{\mathcal{I}_1,1}, H^{\text{diff}}_{\mathcal{I}_1,2} - H^{\text{diff}}_{\mathcal{I}_1,1}\right]\right|.$$

Validity of the testing procedures

Theorem (General assumptions)

Assume that the treatments are assigned according to rules defined in the setup. Under no cross-unit interference hypothesis, the p-value produced by Algorithm 1 is valid in the following sense: for any $\alpha \in (0, 1)$,

$$\mathbb{P}\left[p \leq \alpha\right] \leq \alpha.$$

Theorem (Time fixed effect model)

Assume that the treatments are assigned according to rules defined in setup. Under Assumptions 1- 2 and no cross-unit interference hypothesis, the p-value produced by Algorithm 2 is valid in the following sense: for any $\alpha \in (0, 1)$,

$$\mathbb{P}\left[p \leq \alpha\right] \leq \alpha.$$

Application - an experiment at LinkedIn

- As an illustration, we consider an online controlled experiment implemented by LinkedIn.
- The treatment in this experiment corresponds to a new feature that improves the quality of LinkedIn members' attribute for ads targeting.
- We run a series of experiments with increasing allocation with the members as the randomization units.

Application - an experiment at LinkedIn

- Interference effect is expected in these experiments.
  - When the allocation percentage is small, only a small set of members have the updated attributes, making them easier to be targeted by ad campaigns.
    - Thus when comparing metrics such as total ad impressions, these members tend to have larger average results than members in the control group.
  - When the treatment allocation increases, more members get the improved attributes.
    - Since the total ad budget does not increase much, the average difference between treated and control units becomes smaller.

Application - an experiment at LinkedIn

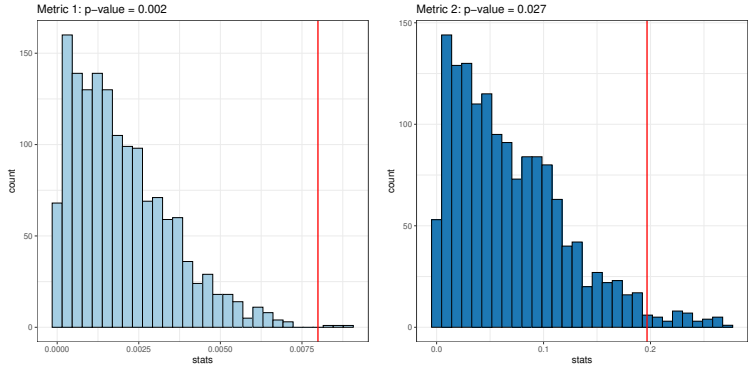


Figure: Example experiment: Test statistics and p-values from permutation. Results on two metrics are shown.

- There is clear interference.

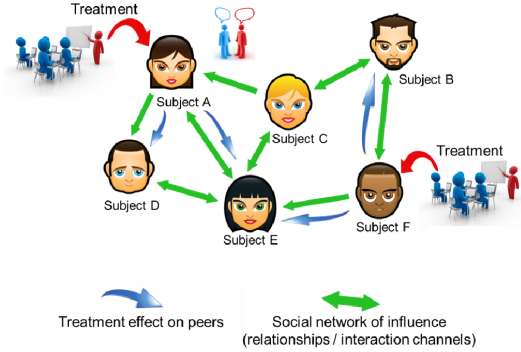
Model-Based Regression Adjustment with Model-Free Covariates for Network Interference

Joint work with

Johan Ugander

Han & Ugander. **Model-Based Regression Adjustment with Model-Free Covariates for Network Interference.** Accepted by *Journal of Causal Inference*. *arXiv:2302.04997*, 2023.

Network Interference

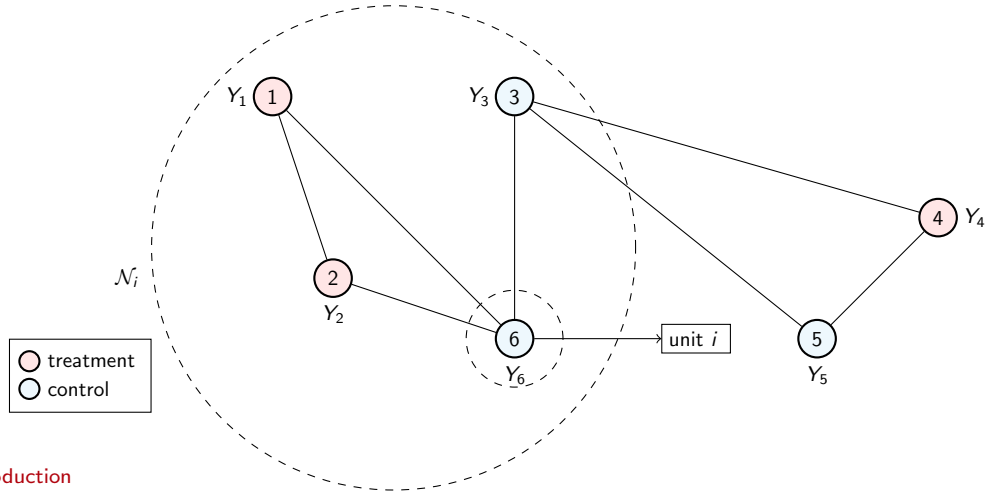


Network interference

- A social network/graph G that describes the social interactions among n subjects indexed by $i = 1, ..., n$ with an edge set $\mathcal{E}$.
- Each unit is assigned to treatment independently, $W_i \in \{0, 1\} \sim \text{Bern}(p_i)$ with $0 < p_i < 1$.
- The graph G is associated with a symmetric matrix $A \in \mathbb{R}^{n \times n}$ so that $A_{ij} = 1$ if $(i, j) \in \mathcal{E}$ and zero otherwise.
- $\mathcal{N}_i^{(k)}$ denotes the k-hop neighborhood around node $i \in V$ (the superscript k will be dropped when $k = 1$).

Network interference

- Potential outcomes $Y_i(w)$ for each assignment vector $w \in \{0,1\}^n$ and each unit i.
- Observed outcomes $Y_i = Y_i(w)$.



Global Average Treatment Effect

The causal estimand of interest is

$$\tau = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}[Y_i(\mathbf{1}) - Y_i(\mathbf{0})].$$

This estimand

- measures the overall effect of the intervention on the experimental units.
- is simply the average treatment effect (ATE) under SUTVA.

A Regression Perspective

- A typical approach to estimate the GATE is by modeling how the assignment vector affects the outcomes (Toulis and Kao, 2013; Cai et al., 2015; Chin, 2019).
- We assume two functions f_0 and f_1 such that for each unit i and each assignment vector $w \in \{0,1\}^n$,

$$Y_i(w) = w_i f_1(i, w, x_i, G) + (1 - w_i) f_0(i, w, x_i, G) + \epsilon_i, \tag{4}$$

with ϵ_i's being exogenous, i.e. $\mathbb{E}[\epsilon_i | w] = 0$.

Linear interference

- Given (4), we can use the treated units to estimate f_1 and control units to estimate f_0. Suppose $\hat{f}_0$ and $\hat{f}_1$ are two estimates of f_0 and f_1 respectively, then a natural estimator of the GATE would be

$$\hat{\tau} = \frac{1}{n} \sum_{i=1}^{n} [\hat{f}_1(i, \mathbf{1}, x_i, G) - \hat{f}_0(i, \mathbf{0}, x_i, G)]. \tag{5}$$

- Unfortunately, estimation of the GATE will be impossible (there exists no consistent estimators) without any further assumptions on the structure of the functions f_0 and f_1 (Basse and Airoldi, 2018).

Linear interference

We consider a specific interference structure - linear interference.

Definition (Linear interference)

We say that the model $\mathcal{Y} = \{Y_i(w) : w \in \{0,1\}^n, i \in [n]\}$ exhibits *linear interference* if there exists a function

$$g : [n] \times \{0,1\}^n \times \mathcal{X} \times \mathcal{G} \to \mathbb{R}^K \text{ and } \theta_0 \in \mathbb{R}^K, \theta_1 \in \mathbb{R}^K$$

such that

$$f_0(i, w, x_i, G) = \theta_0^T g(i, w, x_i, G) \text{ and } f_1(i, w, x_i, G) = \theta_1^T g(i, w, x_i, G). \tag{6}$$

We call each coordinate function g_j of g a *feature* of the interference.

- Though the functional form is linear, g can be nonlinear. Hence this definition also includes nonlinear functions.
- Untestable from data.

Example (Linear-in-means model (Bramoullé et al., 2009))

Consider the structural model

$$\mathbf{y} = \alpha\mathbf{1} + \beta\tilde{A}\mathbf{y} + \gamma\mathbf{w} + \delta\tilde{A}\mathbf{w} + \boldsymbol{\epsilon}, \quad \mathbb{E}[\boldsymbol{\epsilon}|\mathbf{w}] = 0, \tag{7}$$

where $\mathbf{y}$ is the $n \times 1$ outcome vector, $\tilde{A}$ is the degree-normalized adjacency matrix, i.e., $\tilde{A}_{ij} = A_{ij}/d_i$, $\mathbf{w}$ is the assignment vector, and $(\alpha, \beta, \gamma, \delta)$ are parameters. Under some mild conditions on the coefficients and the graph G, we can rewrite the above model as

$$\mathbf{y} = \alpha/(1-\beta)\mathbf{1} + \gamma\mathbf{w} + (\gamma\beta + \delta)\sum_{j=0}^{\infty} \beta^j \tilde{A}^{j+1}\mathbf{w} + \sum_{j=0}^{\infty} \beta^j \tilde{A}^{j+1}\boldsymbol{\epsilon}. \tag{8}$$

Note that now the outcome is linear in the assignment vector $\mathbf{w}$ as well as $\{\tilde{A}^{j+1}\mathbf{w}\}_{j=0}^{\infty}$. Let $f_0(i, w, x_i, G) = f_1(i, w, x_i, G) = \alpha/(1-\beta) + \gamma w_i + (\gamma\beta + \delta)\sum_{j=0}^{\infty} \beta^j \tilde{A}^{j+1} w$ and notice that $\mathbb{E}[\sum_{j=0}^{\infty} \beta^j \tilde{A}^{j+1}\boldsymbol{\epsilon}|w] = 0$. Thus, the linear-in-means model (7) can be written in the form of (4).

Estimation of the GATE under linear interference

- If we know the function g a priori, Chin (2019) provides a complete solution.
- If we don't know the function g, then there are three significant challenges.
  - First, how should we *construct* g so that the one we construct approximates the true one?
  - Second, suppose we have many candidate functions then how should we *select* among them?
  - Third, even if we have satisfactory answers to the first two questions, how should we do inference?

Model-free covariates

- To answer the first two questions, we introduce a sequential procedure to generate and select features of interference.
  - We generate rich candidate features based solely on the graph structure as well as the assignment vector and select among these features based on the observed outcomes.
- These model-free covariates will be used to estimate the GATE.

Model-free covariates

- We call the procedure ReFeX-LASSO as it builds on the graph mining technique ReFeX (Henderson et al., 2011) to generate candidate features while using LASSO (Tibshirani, 1996) to select features.
- ReFeX (Recursive Feature Extraction) was originally designed to generate features for graph mining tasks. It can be viewed as
  - a recursive algorithm that starts with base features of each node in the graph and iteratively (i) adds and (ii) prunes features based on aggregations over features from neighboring nodes.
  - a simple early precursor to recent methods for graph representation learning based on graph convolution networks (GCNs) (Hamilton et al., 2017; Kipf and Welling, 2017).

Ingredients of ReFeX

- **Base features**.
  - Base features are those features that can be constructed by only looking at each node's 1-hop neighborhood.
  - Examples: graph features, features constructed by using both the graph and the assignment vector, features constructed by using both the graph and the pre-treatment covariates.
- **Aggregation functions**.
  - Aggregation functions are functions that take features from neighboring nodes as inputs and output a single value.
  - One aggregation function essentially computes a statistic based on the sample of feature values from neighbors.
  - Examples: min, max, sum, mean and variance.

Illustration of ReFeX

ReFeX

Regional

Base

Nodes	Local		Egonet		Recursive		
1	1	1	1	2	3	0	1
2	3	3	1	1	1	3	0
3	1	3	3	0	2	1	1
4	1	1	3	1	1	0	3

ReFeX-LASSO

Algorithm 3 ReFeX-LASSO

Input: Graph $G = (V, \mathcal{E})$, assignment vector $w \in \{0,1\}^n$, maximum number of iterations T.
Output: A set of covariates S.

1: Initialize $S = \{\}$, active feature set $A = \{\}$.
2: For each node/unit i, construct m base features and add m base features to A.
3: **for** $t = 1$ to T **do**
4: Regress y on w and features from S and A using LASSO with no penalty on features from S.
5: If no feature in A is selected, return S. Otherwise, add selected features from A to S.
6: Recursively construct features by performing aggregations of features in A over neighbors in 1-hop neighborhood.
7: Delete old features in A and add those new features to A.
8: **end for**
9: Return S.

ReFeX and multi-hop information

Example

Suppose one of the base features we use in ReFeX-LASSO is the fraction of treated neighbors,

$$\rho_i = \frac{1}{d_i} \sum_{j \in \mathcal{N}_i} w_j,$$

and we limit ourselves to mean aggregation. We call the two-hop aggregated feature $\tilde{\rho}_i$. Then

$$\begin{aligned}\tilde{\rho}_i &= \frac{1}{d_i} \sum_{j \in \mathcal{N}_i} \rho_j = \frac{1}{d_i} \sum_{j \in \mathcal{N}_i} \frac{1}{d_j} \sum_{k \in \mathcal{N}_j} w_k \\ &= \sum_{j=1}^{n} \frac{A_{ij}}{d_i} \sum_{k=1}^{n} \frac{A_{jk}}{d_j} w_k = \sum_{j=1}^{n} \tilde{A}_{ij} \sum_{k=1}^{n} \tilde{A}_{jk} w_k = [\tilde{A}^2 w]_i,\end{aligned}$$

where A and $\tilde{A}$ are the same as defined in the linear-in-means model example from (7). Clearly, this feature is informative for unit i's 2-hop neighborhood.

Estimation of the GATE with model-free covariates

- Suppose ReFeX-LASSO returns K features, we then would think that g maps i, w, x_i, G to a $K-$dimensional vector that consists of these feature values.
- Recall under linear interference,

$$f_0(i, w, x_i, G) = \theta_0^T g(i, w, x_i, G) \text{ and } f_1(i, w, x_i, G) = \theta_1^T g(i, w, x_i, G).$$

Therefore we could estimate the GATE as follows. We

1. run ordinary least squares with observations from the control group only and covariates returned by ReFeX-LASSO to obtain $\hat{\beta}_0$.
2. run ordinary least squares again, now with observations from treatment group only and covariates returned by ReFeX-LASSO to obtain $\hat{\beta}_1$.
3. get covariate values u_i^{gc} and u_i^{gt} under $w = \mathbf{0}$ and $w = \mathbf{1}$.
4. output

$$\hat{\tau} = \frac{1}{n} \sum_{i=1}^{n} (\hat{\beta}_1^T u_i^{gt} - \hat{\beta}_0^T u_i^{gc}).$$

Selection properties of ReFeX-LASSO

For each iteration t, let $\{u_1^t, u_2^t, \cdots, u_{i_t}^t\}$ be the set of features generated in the ReFeX step of ReFeX-LASSO and s_{*t} be the selected features at the t-th iteration (note that s_{*t} may contain features that were selected in previous iterations and thus are not in the set $\{u_1^t, u_2^t, \cdots, u_{i_t}^t\}$). Moreover, we let $\mathcal{R}(s)$ to denote the space spanned by features in s.

Proposition (No highly correlated features)

For $t \geq 1$ and any $j \in \{1, \cdots, i_{t+1}\}$, if $u_j^{t+1} \in \mathcal{R}(s_{*t})$ then $j \notin s_{*(t+1)}$.

Proposition (Nested feature sets)

Our selection is nested in the sense that $s_{*1} \subseteq s_{*2} \subseteq \cdots \subseteq s_{*T}$.

Additional remarks of ReFeX-LASSO

- We also derive consistency results for the estimator under standard assumptions of LASSO in the paper.
- There is also a closely related method to ReFeX-LASSO in the paper where selection is done after generating all the features.
- We use LASSO instead of the original feature pruning step in ReFeX. The reasons are two-fold.
  1. First, we would use selected features to do regression adjustment and LASSO does feature selection based on a linear model.
  2. Second, the feature pruning step in ReFeX requires choosing a hyperparameter p and unlike LASSO where we can choose λ by cross validation, there is no natural procedure to choose p according to data.

Confidence interval for the GATE

To do inference, we would like to construct confidence intervals for the GATE.

Difficulties:

- The true model is unknown.
- Features are constructed (specifically, the selection step) using the observed outcomes.
- Therefore, ReFeX-LASSO leads to an estimator with no clear variance expression.

Solution: block bootstrap.

Confidence interval via a block bootstrap

- The bootstrap sample is only used in the feature selection step of ReFeX-LASSO.
  - That being said, for each iteration, we still use the same graph G to generate features but then we use the bootstrap sample of these features to do selection.
  - The intuition behind using the original graph for feature generation is that we view the graph as fixed and the correlation structure of all features are then induced by this graph.
- Kojevnikov (2021) develops consistency results of block bootstrap for a class of network processes under strong technical assumptions on both the network process itself and the graph structure.
  - As a simple example, if the graph consists of disjoint clusters of same size, then the assumptions on the graph structure in Kojevnikov (2021) would be satisfied.

Confidence interval via a block bootstrap

Algorithm 4 Block bootstrap for ReFeX-LASSO

Input: Graph $G = (V, \mathcal{E})$, assignment vector $w \in \{0,1\}^n$, number of bootstrap samples B.

Output: Confidence interval for τ.

1: Collect the assignment w_i and outcome y_i for each unit i. Record the stopping time for ReFeX-LASSO T^*.
2: Use k-hop max clustering with $k = T^* + 1$ to divide n units into C clusters $\mathcal{C}_1, \cdots, \mathcal{C}_C$.
3: **for** $b = 1$ to B **do**
4: Sample C clusters with replacement from $\mathcal{C}_1, \cdots, \mathcal{C}_C$.
5: Construct the b-th bootstrap sample with units from sampled clusters.
6: Rerun ReFeX-LASSO with the original sample for feature generation and the bootstrap sample for feature selection.
7: Use the covariates returned from last step as well as the bootstrap sample to get estimate of τ, $\hat{\tau}^b$.
8: **end for**
9: Repeat line 2-8 for ℓ times and obtain $\ell \cdot B$ bootstrap estimates in total.
10: Compute the $\alpha/2$-th quantile $q^*_{\alpha/2}$ and the $(1-\alpha/2)$-th quantile $q^*_{1-\alpha/2}$ of the sample of all bootstrap estimates $\hat{\tau}^1, \cdots, \hat{\tau}^{\ell B}$.
11: Return $\left[q^*_{\alpha/2}, q^*_{1-\alpha/2}\right]$ as the $(1-\alpha) \times 100\%$ confidence interval for τ.

Insurance adoption data analysis

The data were collected from a field experiment conducted in rural China (Cai et al., 2015).

- A random subset of farmers were provided with intensive information sessions about an insurance product.
- Cai et al. (2015) find that the diffusion of insurance knowledge drove network effects in product adoption.
- Though we know that network effects do exist, defining an exact exposure model is difficult.

Insurance adoption data analysis

Estimator	Estimate	Standard Error
DM	0.078	——
Hájek_1hop ($q = 0.75$)	0.163	——
$\hat{\tau}_{\text{chin}}$	0.122	0.056
$\hat{\tau}_{\text{num}}$	0.178	0.027
$\hat{\tau}_{\text{refex-lasso}}$	0.178	0.043

Table: Estimates and standard errors of different estimators for the global average treatment effect on insurance adoption Cai et al. (2015).

- $\hat{\tau}_{\text{chin}}$ is the estimator in Chin (2019) that adjusts for four covariates: the fraction of treated neighbors, the number of treated neighbors, the fraction of treated neighbors in 2-hop neighborhoods and the number of treated neighbors in 2-hop neighborhoods.
- $\hat{\tau}_{\text{num}}$ only adjusts for the number of treated neighbors.
- $\hat{\tau}_{\text{refex-lasso}}$ is the ReFeX-LASSO based regression adjustment estimator.
- The standard errors for $\hat{\tau}_{\text{num}}$ and $\hat{\tau}_{\text{chin}}$ were calculated assuming that the linear model is the true model while the standard error for $\hat{\tau}_{\text{refex-lasso}}$ was calculated from block bootstrap.

Summary

Interference brings challenges to causal inference. In this talk,

- we presented valid testing procedures for interference in A/B tests with increasing allocation.
- we presented a method that estimates the global average treatment effect under network interference.

References I

Athey, S., Eckles, D., and Imbens, G. W. (2018). Exact p-values for network interference. *Journal of the American Statistical Association*, 113(521):230–240.

Basse, G. W. and Airoldi, E. M. (2018). Limitations of design-based causal inference and a/b testing under arbitrary and network interference. *Sociological Methodology*, 48(1):136–151.

Basse, G. W., Feller, A., and Toulis, P. (2019). Randomization tests of causal effects under interference. *Biometrika*.

Bramoullé, Y., Djebbari, H., and Fortin, B. (2009). Identification of peer effects through social networks. *Journal of econometrics*, 150(1):41–55.

Cai, J., De Janvry, A., and Sadoulet, E. (2015). Social networks and the decision to insure. *American Economic Journal: Applied Economics*, 7(2):81–108.

Chin, A. (2019). Regression adjustments for estimating the global treatment effect in experiments with interference. *Journal of Causal Inference*, 7(2).

Fisher, R. A. (1925). *Statistical Methods for Research Workers*. Number 3. Oliver and Boyd.

References II

Friedlander, D. and Robins, P. K. (1995). Evaluating program evaluations: New evidence on commonly used nonexperimental methods. *The American Economic Review*, pages 923–937.

Halloran, M. E. and Struchiner, C. J. (1995). Causal inference in infectious diseases. *Epidemiology*, pages 142–151.

Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Hemerik, J. and Goeman, J. (2018a). Exact testing with random permutations. *Test*, 27(4):811–825.

Hemerik, J. and Goeman, J. J. (2018b). False discovery proportion estimation by permutations: confidence for significance analysis of microarrays. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(1):137–155.

References III

Henderson, K., Gallagher, B., Li, L., Akoglu, L., Eliassi-Rad, T., Tong, H., and Faloutsos, C. (2011). It's who you know: graph mining using recursive structural features. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 663–671.

Holtz, D., Lobel, R., Liskovich, I., and Aral, S. (2020). Reducing interference bias in online marketplace pricing experiments. *arXiv preprint arXiv:2004.12489*.

Hong, G. and Raudenbush, S. W. (2006). Evaluating kindergarten retention policy: A case study of causal inference for multilevel observational data. *Journal of the American Statistical Association*, 101(475):901–910.

Imbens, G. W. and Rubin, D. B. (2015). *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press.

Johari, R., Li, H., Liskovich, I., and Weintraub, G. Y. (2022). Experimental design in two-sided platforms: An analysis of bias. *Management Science*, 68(10):7069–7089.

References IV

Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Kojevnikov, D. (2021). The bootstrap for network dependent processes. *arXiv preprint arXiv:2101.12312*.

Manski, C. F. (2013). Identification of treatment response with social interactions. *The Econometrics Journal*, 16(1):S1–S23.

Pouget-Abadie, J., Saint-Jacques, G., Saveski, M., Duan, W., Ghosh, S., Xu, Y., and Airoldi, E. M. (2019). Testing for arbitrary interference on experimentation platforms. *Biometrika*, 106(4):929–940.

Puelz, D., Basse, G., Feller, A., and Toulis, P. (2022). A graph-theoretic approach to randomization tests of causal effects under general interference. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(1):174–204.

Riccio, J. et al. (1989). Gain: Early implementation experiences and lessons. california's greater avenues for independence program.

References V

Rogers, T. and Feller, A. (2017). Intervening through influential third parties: Reducing student absences at scale via parents. *Work. Pap., John F. Kennedy Sch. Gov., Harvard Univ., Cambridge, MA*.

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

Sinclair, B., McConnell, M., and Green, D. P. (2012). Detecting spillover effects: Design and analysis of multilevel experiments. *American Journal of Political Science*, 56(4):1055–1069.

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

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, pages 267–288.

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

References VI

Ugander, J. and Yin, H. (2020). Randomized graph cluster randomization. *arXiv preprint arXiv:2009.02297*.

Vovk, V. and Wang, R. (2020). Combining p-values via averaging. *Biometrika*, 107(4):791–808.

References

Sources of Figures

https://blog.ml.cmu.edu/2020/08/31/7-causality/
https://www.raybeam.com/focus/inferring-the-effect-of-an-event-using-causal-infere
https://www.fomatmedical.com/randomized-clinical-trials/
Edward K. Kao's PhD thesis
https://en.wikipedia.org/wiki/A/B_testing
https://sites.bu.edu/causal/
https://towardsdatascience.com/ab-testing-challenges-in-social-networks-e67611c929

Thank you!

Graph clustering algorithm

To run block bootstrap, a graph clustering algorithm is necessary. In our block bootstrap algorithm, we utilize the following random graph clustering algorithm (Ugander and Yin, 2020):

Algorithm 5 k-hop-max graph clustering

Input: Graph $G = (V, E)$.

Output: Graph clustering $\mathcal{C}_1, \cdots, \mathcal{C}_c$.

1: **for** $i \in V$ **do**
2: $\quad X_i \leftarrow \mathcal{U}(0, 1)$;
3: **end for**
4: **for** $i \in V$ **do**
5: $\quad i \leftarrow \text{argmax}([X_j \text{ for } j \in B_k(i)])$;
6: **end for**
7: Return $\mathcal{C}_1, \cdots, \mathcal{C}_c$.

About p-values

- One issue with the algorithms proposed is that randomly splitting the data (Algorithm 1) or the random matching step (Algorithm 2) can inject randomness into the p-value.
- In order to de-randomize the procedure, we can run the algorithms many times and aggregate the p-values.
  - Since the p-values can be arbitrarily dependent on each other, we cannot use Fisher's method to aggregate the p-values, which requires independence (Fisher, 1925).
  - Some possible ways include, e.g., setting $p = 2\sum p_i/n$ (See (Vovk and Wang, 2020) for more details).

Testing for interference with more than two experiments

We provide details on how we could generalize our methods for two experiments to more than two experiments. For testing under general assumption, we

- let $\mathcal{I}_{\text{nc}} = \{i : W_{i,1} = \cdots = W_{i,K}\}$ be the set of units whose treatment didn't change over the experiments. Randomly sample a subset of $\mathcal{I}_{\text{nc}}$ of size $n/2$. We call the subset $\mathcal{I}_{\text{foc}}$. Let $\mathcal{I}_{\text{aux}} = [n] \setminus \mathcal{I}_{\text{foc}}$.
- permute the treaments of auxiliary units as in Algorithm 1.

Testing for interference with more than two experiments

For testing under time fixed effect assumption, we first provide an illustration here.

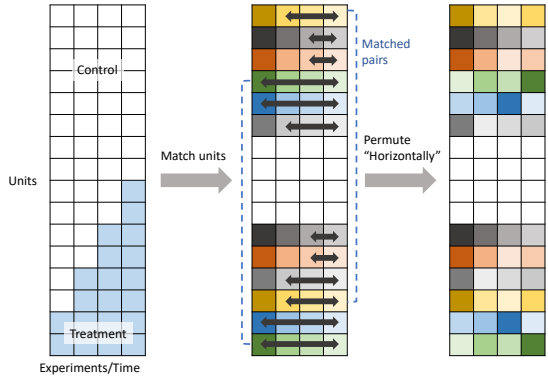


Figure: An illustration of extension of Algorithm 2 to more than two experiments. Pairs of units are matched and the outcomes of paired units are permuted together across experiments.

Testing for interference with multiple experiments

Some differences between Algorithm 2 and its extension to multiple experiments.

- The choice of $\mathcal{I}_0$ and $\mathcal{I}_1$.
  - $\mathcal{I}_0 = \{i : W_{i,1} = \cdots = W_{i,k} = 0\}$ is the set of units that are in the control group in all experiments.
  - $\mathcal{I}_1 = \{i : W_{i,K-1} = W_{i,K} = 1\}$ is the set of units that are in the treatment group in the last two experiments (i.e. units that are treated in at least two experiments).
- The way we permute the assignment matrix.
  - Let $S_i = \{k : W_{i,k} = 1\}$ be the set of experiments in which unit i is treated.
  - We randomly permute outcomes across S_i: $\widetilde{Y}^{(b)}_{i,k} = Y_{i,\sigma_{i,b}(k)}$ and $\widetilde{Y}^{(b)}_{m(i),k} = Y_{m(i),\sigma_{i,b}(k)}$ for all $k \in S_i$, where $\sigma_{i,b}$ is a random permutation of S_i.

The theorem used to prove the validity results

We make use of the following theorem in (Hemerik and Goeman, 2018a,b, Theorem 2).

Theorem (Random permutations)

Let $A_1, A_2, \ldots, A_n \in \mathcal{A}$ be n random variables. Let $\mathcal{S}_n$ denote the set of all permutations on $[n]$. Assume that

1. *$G \subset \mathcal{S}_n$ is a subgroup;*
2. *For any $\sigma \in G$, $A = (A_1, \ldots, A_n) \stackrel{d}{=} (A_{\sigma(1)}, \ldots, A_{\sigma(n)}) = A_\sigma$.*

If $\sigma_1, \ldots, \sigma_B$ are drawn independently uniformly from G, then for any test statistic T, the p-value

$$p = \frac{1}{B+1}\left(1 + \sum_{b=1}^{B} \mathbb{1}\left\{T(A) \leq T(A_\sigma)\right\}\right) \tag{9}$$

satisfies

$$\mathbb{P}\left[p \leq \alpha\right] \leq \alpha. \tag{10}$$

for any $\alpha \in (0, 1)$.