22

Leakage and Interference between Variants

> It doesn't matter how beautiful your theory is, it doesn't matter how smart you are. If it doesn't agree with experiment, it's wrong
> – *Richard Feynman*

***Why you care**: In most experiment analyses, we assume that the behavior of each unit in the experiment is unaffected by variant assignment to other units. This is a plausible assumption in most practical applications. However, there are also many cases where this assumption fails.*

In most of the discussions in the book, we assume the Rubin causal model (Imbens and Rubin 2015), a standard framework for analyzing controlled experiments. In this chapter, we discuss these assumptions, scenarios where they fail, and approaches to address them.

A key assumption made in the Rubin causal model is the Stable Unit Treatment Value Assumption (SUTVA), which states that the behavior of each unit in the experiment is unaffected by variant assignment to other units (Rubin 1990, Cox 1958, Imbens and Rubin 2015) as shown in Equation 22.1:

$$Y_i(z) = Y_i(z_i) \tag{22.1}$$

with variant assignment vector $z = (z_1, z_2, \ldots, z_n)$ for all n units.

This is a plausible assumption in most practical applications. For example, a user who likes the new checkout flow described in Chapter 2 is more likely to purchase, and that behavior is independent of others who are using the same eCommerce site. However, if the SUTVA assumption does not hold (see examples later in this chapter), the analysis results in potentially incorrect conclusions. We define *interference* as a violation of SUTVA, sometimes called *spillover* or *leakage* between the variants.

There are two ways interference may arise: through *direct* or *indirect* connections. As an example, two units can be directly connected if they are friends on a

social network or if they visited the same physical space at the same time. Indirect connections are connections that exist because of certain latent variables or shared resources, such as units in Treatment and Control that share the same ad campaign budget. These two categories are similar in that in both cases there is a medium that connects the Treatment and Control groups and allows them to interact. The medium can either be a materialized friendship connection on a social network or a shared advertising budget that clicks from both Treatment and Control users are billed against. It is important to understand the mechanism through which interference can manifest, as the best solution to address them can differ.

To make the problems more concrete, here are examples with more detailed discussions.

Examples

Direct Connections

Two units can be directly connected if they are friends on a social network or if they visited the same physical space at the same time. Two units that are directly connected can be separated into Treatment and Control groups and hence cause interference between variants.

Facebook/LinkedIn. In social networks, such as Facebook or LinkedIn, user behavior is likely impacted by that of their social neighborhood (Eckles, Karrer and Ugander 2017, Gui et al. 2015). Users find a new social-engagement feature more valuable as more of their neighbors use it and are thus more likely to use it themselves. For example, from the user perspective:

- I am more likely to use video chat on Facebook if my friends use it.
- I am more likely to message my friends on LinkedIn if they message me.
- I am more likely to post on LinkedIn if friends in my network post on it.

In an A/B experiment, this implies that if the Treatment has a significant impact on a user, the effect could spill over to their social circles, regardless whether the neighbors are in Treatment or Control. For example, a better recommendation algorithm in Treatment for the "People You May Know" algorithm on LinkedIn encourages users to send more connect invitations. However, users who receive these invitations may be in the Control variant and when they visit LinkedIn to accept the invitation, they may discover more people to connect with. If the primary metric of interest is the total number of invitations sent, both Treatment and Control invitations are likely to increase,

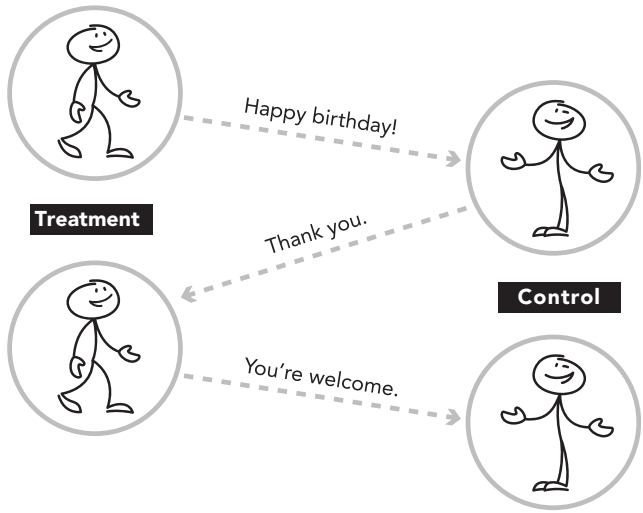

Figure 22.1 As users in Treatment send more messages to their network, users in Control will also send more messages as they reply to those messages.

so the delta is biased downwards and does not fully capture the benefit of the new algorithm. Similarly, if the Treatment encourages users to send more messages, the Control group will also see an increase in messages sent as users reply. See Figure 22.1.

Skype calls. As a communication tool, every call on Skype involves at least two parties. Clearly, if a user decides to call a friend on Skype, the friend ends up using Skype more, at least to answer this call. It is likely that the friend will also use Skype to call their friends. In an A/B setting, assume that Skype improved call quality for Treatment, increasing calls made from the Treatment group. Those calls can go to users who are in Treatment or Control. The result is that users in Control also increase using Skype for calls, so the delta between Treatment and Control is underestimated.

Indirect Connections

Two units can have an indirect connection because of certain latent variables or shared resources. Like direct connections, these indirect connections can also cause interference and biased estimates of Treatment effect.

- **Airbnb** If the AirBnB marketplace site of homes for rent improved conversion flow for Treatment users, resulting in more booking, it would naturally lead to less inventory for Control users. This means that revenue generated

from the Control group is less than what it would have been if there were no Treatment group. Comparing Treatment and Control leads to an overestimation of the Treatment effect (Holtz 2018).

- **Uber/Lyft**. Imagine that Uber wants to test a different "surge price" algorithm that is so awesome that riders in Treatment are more likely to opt-in for a ride. Now there are fewer drivers available on the road, the price for the Control group goes up leading to fewer riders. As a result, the delta comparing the Treatment and Control groups is overestimated (Chamandy 2016).
- **eBay.** Assume that the Treatment for buyers encourages up bidding, like a rebate or promotion. Because the Treatment and Control users are competing for the same items, the higher bid price from Treatment certainly makes Control users less likely to win the auction. If the metric of interest is the total number of transactions, the delta between Treatment and Control is overestimated (Blake and Coey 2014, Kohavi, Longbotham et al. 2009).
- **Ad Campaigns**. Consider an experiment that shows users different rankings for the same ads. If the Treatment encourages more clicks on the ads, it uses up campaign budget faster. Because the budget on a given campaign is shared between Treatment and Control groups, the Control group ends up with a smaller budget. As a result, the delta between Treatment and Control is overestimated. Also, because of the budget constraints, experiments impacting ad revenue tend to yield a different result at the beginning of the month (or quarter) than the end of the month (or quarter) (Blake and Coey 2014, Kohavi, Longbotham et al. 2009).
- **Relevance Model Training**. Relevance models usually rely heavily on user engagement data to learn what is relevant and what is not. Just to explain the concept, imagine a search engine that uses a simple click-based relevance model for ranking. In this case, if more users click target.com when searching for "shoes," the search engine would learn and rank target.com higher for the keyword "shoes." This learning process is *model training*, and it happens continuously as fresh data flows in. Consider a Treatment relevance model that can better predict what users like to click. If we use data collected from all users to train both Treatment and Control models, the longer the experiment runs, the more the "good" clicks from Treatment benefit Control.
- **CPU**. When a user performs an action on a website, such as adding an item to the shopping cart or clicking a search result, it usually results in a request to the website server. Simplified, the request is processed by the server machines and information returned to the user. In an A/B setting, requests from both Treatment and Control are usually processed by the same

machines. We have experienced examples where a bug in Treatment unexpectedly held up CPUs and memory on the machines, and as a result, requests from both Treatment and Control took longer to process. This negative Treatment effect on latency is underestimated if we do the usual Treatment and Control comparison.

- **Sub-user experiment unit**. As we discussed in Chapter 14, even though *user* is a more commonly used experiment unit, there are cases where an experiment randomizes on different units, such as a page visit, or session. Experiment units smaller than a user, such as a pageview, can cause potential leakage among units from the same user if there is a strong learning effect from the Treatment. In this case, "user" is the latent connection. As an example, assume we have a Treatment that dramatically improves latency and we randomize on a pageview, so the same user experiences pageviews in both Treatment and Control. Faster page-load-times usually lead to more clicks and more revenue (see Chapter 5); however, because the user experiences a mixed experience, their behavior on the fast pages may be influenced by that of the slow pages, and vice versa. Again, the Treatment effect is underestimated.

Some Practical Solutions

While the interference in these examples may be caused by different reasons, they can all lead to biased results. In an ad campaign, for example, you might see a positive delta on revenue during the experiment, but when the Treatment launches to all users, the impact could be neutral because of budget constraints. When we run an experiment, we want to estimate the delta between two parallel universes: the universe where every unit is in Treatment, and the universe where every unit is in Control. Leakage between the Treatment and Control units biases the estimate. How can we prevent or correct for it?

There are a few categories of practical approaches to address interference in controlled experiments. Gupta et al. (2019) also have good discussions around some of these methods.

Rule-of-Thumb: Ecosystem Value of an Action

Not all user actions spill over from Treatment to Control. You can identify actions that can potentially spill over, and only have concern about interference if these actions are materially impacted in an experiment. This usually means

not only the first-order action, but also the potential responses to actions. For example, consider the following metrics for an experiment on a social network:

- Total number of messages sent, and messages responded to
- Total number of posts created, and total number of likes/comments they received
- Total number of likes and comments, and total number of creators receiving these likes and comments.

These metrics can indicate the downstream impact. Measuring responses gives an estimate of the depth and breadth of the potential ecosystem impact from first-order actions (Barrilleaux and Wang 2018). An experiment with positive impact on the first-order action and no impact on downstream metrics is not likely to have a measurable spillover effect.

Once you identify metrics indicative of a downstream impact, you can establish *general* guidance for how each action translates to values or engagement for the whole ecosystem. For example, how much does a message from user A translate to visit sessions from both A and their neighbors? One approach for establishing this rule-of-thumb is using historical experiments shown to have downstream impact and extrapolating this impact to the downstream impact of actions X/Y/Z using the Instrumental Variable approach (Tutterow and Saint-Jacques 2019).

This rule-of-thumb approach is relatively easy to implement, because it takes a one-time effort to establish the ecosystem value and you then apply it on any Bernoulli randomized experiment. It is more sensitive than other approaches because it relies on Bernoulli randomization to measure significant impact on downstream metrics. The approach does have limitations, however. By nature, rule-of-thumb is only an approximation and may not work for all scenarios. For example, additional messages resulting from a certain Treatment may have an ecosystem impact larger than average.

Isolation

Interference happens through a medium connecting the Treatment and Control groups. You can remove potential interference by identifying the medium and isolating each variant. The rule-of-thumb approach uses the Bernoulli randomization design to allow you to estimate the ecosystem impact during analysis. To create isolation, you must consider other experimental designs to ensure that your Treatment and Control units are well separated. Here are a few practical isolation approaches.

- **Splitting shared resources.** If a shared resource is causing interference, splitting it between Treatment and Control is the obvious first choice. For example, you can split the ad budget according to variant allocation and only allow 20% budget to be consumed by a variant allocated 20% of traffic. Similarly, in a relevance algorithm training case, you can split the training data according to the variants they are collected from.

  There are two things to watch out for when applying this approach:

  1. Can your interfering resources be split exactly according to the traffic allocation for your variants? While this is easily achieved for budget or training data, it often isn't possible. For example, with shared machines there is heterogeneity among individual machines and simply serving Treatment and Control traffic different machines introduces too much other confounding factors difficult to correct for.
  2. Does your traffic allocation (the resource split size) introduce bias? For training data, model performance increases as it gets more training data. If the Treatment model only gets 5% data to train on and the Control model gets 95% data, this introduces bias against the Control model. This is one of the reasons we recommend a 50/50 split for traffic allocations.
- **Geo-based randomization**. There are many examples where interference happens when two units are geographically close, for example, two hotels competing for the same tourists or two taxis competing for the same riders. It's reasonable to assume that units (hotels, taxis, riders, etc.) from different regions are isolated from each other. This lets you randomize at the region level to isolate the interference between Treatment and Control (Vaver and Koehler 2011, 2012). One caveat: randomizing at the geo level limits the sample size by the number of available geo locations. This leads to a bigger variance and less power for A/B tests. See Chapter 18 for discussions on reducing variance and achieving better power.
- **Time-based randomization**. You can create isolation using time. At any time *t*, you could flip a coin and decide whether to give all users Treatment or all users Control (Bojinov and Shephard 2017, Hohnhold, O'Brien and Tang 2015). Of course, for this to work, the interference that can be caused by the same user over time is not a concern (see the sub-user experiment unit discussion above). The time unit can be short (seconds) or long (weeks), depending on what is practical and how many sample points you need. For example, if "day" is your unit, you can only collect seven sample points in a week, which is probably not fast enough. One thing to keep in mind is that there is usually strong temporal variation, like the day-of-week or hour-of-day effects. This usually helps reduce variance by utilizing this

information in paired t-tests or covariate adjustments. See Chapter 18 for more details. A similar technique called interrupted time series (ITS) is discussed in Chapter 11.

- **Network-cluster randomization**. Similar to geo-based randomization, on social networks, we construct "clusters" of nodes that are close to each other based on their likelihood to interfere. We use each cluster as "mega" units and randomize them independently into Treatment or Control groups (Gui et al. 2015, Backstrom and Kleinberg 2011, Cox 1958, Katzir, Liberty and Somekh 2012).

  There are two limitations to this approach:

  1. It is rare to have perfect isolation in practice. With most social networks, the connection graphs are usually too dense to be cut into perfectly isolated clusters. For example, when attempting to create 10,000 isolated and balanced clusters from the entire LinkedIn graph, there were still more than 80% of the connections between clusters (Saint-Jacques et al. 2018).
  2. Like other mega-unit randomization approaches, the effective sample size (number of clusters) is usually small, which leads to a variance-bias tradeoff when we built the clusters. The larger number of clusters leads to a smaller variance, but also gave us a larger bias with less isolation.

- **Network ego-centric randomization**. In the network-cluster randomization approach, clusters are constructed by minimizing the edge cut between the clusters and each cluster does not employ a specific structure. During experiment assignment, every node in the cluster is also treated the same. Ego-centric randomization addresses similar interference problems on social networks but has fewer limitations. It achieves better isolation and smaller variance by creating clusters each comprised of an "ego" (a focal individual) and its "alters" (the individuals it is immediately connected to). This allows you to decide variant assignment for egos and alters separately. For example, give all alters Treatment and only half of your egos Treatment. By comparing the egos in Treatment to the egos in Control, you can measure first-order impact and downstream impact. You can find a good discussion of this in Saint-Jacques et. al. (2018b).

Whenever applicable, always combine isolation methods to get a larger sample size. For example, while applying network-cluster randomization, you can expand the sample size by leveraging time t as a sampling dimension. If most of interference has a short time span and the Treatment effect itself is transactional, you can use a coin flip each day to determine variant assignment for each cluster. Sometimes, you can create a better isolation by predicting where interference could happen. For example, a user does not message every neighbor in their social network. Knowing that the connection network itself

is usually too dense to create isolated clusters, identifying a subgraph where messages are likely to be exchanged can create better clusters.

Edge-Level Analysis

Some leakage happens in a clearly defined interaction between two users. These interactions (edges) are easy to identify. You can use Bernoulli randomization on users and then label the edges based on the experiment assignment of the users (nodes) as one of these four types: Treatment-to-Treatment, Treatment-to-Control, Control-to-Control, and Control-to-Treatment. Contrasting interactions (e.g., messages, likes) that happen on different edges, allows you to understand important network effects. For example, use the contrast between the Treatment-to-Treatment and Control-to-Control edges to estimate unbiased delta, or identify whether units in Treatment prefer to message other Treatment units over Control units (Treatment affinity), and whether new actions created by Treatment get a higher response rate. You can read up on edge-level analysis in Saint-Jacques et. al. (2018b).

Detecting and Monitoring Interference

Understanding the mechanism of the interference is the key to identifying a good solution. But while getting a precise measurement may not be practical for every experiment, it is important to have a strong monitoring and alert system for detecting interference. For example, if all ad revenue during the experiment comes from budget-constrained vs. budget-non-constrained advertisers, the experiment result could not generalize after launch. Ramp phases can also detect really bad interference (e.g., a Treatment consumes all CPU), such as first ramping to employees or a small datacenter. See Chapter 15 for details.