Challenges in Software A/B Testing and How to Overcome Them

Arenbro, Martin
arenbro@kth.se

Ewaldsson, Jakob
jakobew@kth.se

May 8, 2023

We certify that generative AI, incl. ChatGPT, has not been used to write this essay. Using generative AI without permission is considered academic misconduct.

1 Introduction

A/B testing, also called split testing or bucket testing, is a research methodology that allows for evaluating user satisfaction or engagement of an introduced change. It tests a standard (A) version against a modified (B) version by randomly assigning users one of these and then evaluating which one is most successful based on some key metrics [1]. These could, for example, be the number of clicks on a new UI element, or the clickthrough rate (CTR) for an advertisement. The users that are presented with the standard version are in the *control group* and the users that are presented with the variant version are in the *treatment group*. A/B testing is widely used in software development, especially for websites. In fact, according to Jian Chen and Masashi Toyoda it is the "most standard and widely used method for inferring the causal effect [2]. The strength of A/B testing is that the changes are tested "empirically" by observing how users interact with your product. This is in contrast to letting the developers or some other small group of experts decide which version is "better". For platforms with many users, the large sample size makes the results more reliable. This technique is used by many major software companies such as Google, Microsoft, Amazon, Twitter, and others to improve their products [9].

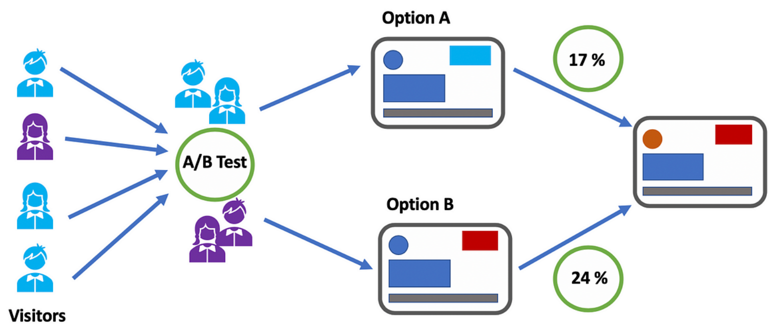

Figure 1: A/B-testing diagram. Source: Towards Data Science

Figure 1 above illustrates how an A/B test can be conducted. A website is altered by changing the color of one of the buttons, and the aim is to ascertain if this leads to that button being clicked more frequently. The visitors are randomly selected for either the control (blue) group or the treatment (purple) group, and the click rate is recorded. As the figure shows the changed version has a higher click rate (24%) than the standard version (17%), and it would therefore be reasonable to adopt the change.

From a DevOps perspective, the task of maintaining, deploying, and delivering multiple versions of software at the same time is a complex task. Especially for large software companies that might want to conduct multiple tests at the same time, there have to be automated procedures in place to make this viable. In DevOps software development, it is also very important to get feedback from a deployed service and to allow data-driven decision-making. A/B testing is very useful in that case. We will discuss some of the challenges with conducting tests at this scale and also introduce some of the suggested solutions.

2 Challenges

On the surface, A/B testing seems like a simple methodology where you test some ideas and implementations against each other and see which of them best satisfies some key metrics. There do however arise several challenges when adopting this methodology, especially when it's done at scale.

At the Practical Online Controlled Experiments Summit in 2019, representatives from thirteen different organizations (including Facebook, Google, LinkedIn, etc.) compiled a study describing the top challenges of performing online controlled experiments (OCEs), also known as A/B tests, at scale. The study was based on one hundred thousand experiments performed by the participating organizations. [4]

2.1 Network Interactions

2.1.1 Problem

One of the challenges raised at the Practical Online Controlled Experiments Summit was in regard to network interactions and their effects on A/B testing. A network interaction is defined, in this case, as occurring when a user's behavior is influenced by the behavior of another user. This is a concern since one of the fundamental assumptions of traditional A/B tests is a stable user treatment value assumption (SUTVA), which means that all user responses are independent. Thus, if network interactions are present in an experiment users in the control group will potentially affect users in the treatment group and vice versa. This means that the SUTVA assumption is invalid and that it's very difficult to draw correct conclusions. [3]

The following is an example of a network effect described by Martin Saveski et al. in their paper "Detecting Network Effects: Randomizing Over Randomized Experiments" [8]. An A/B test is conducted to measure the effects of a new feed ranking algorithm. Given that user A is in the treatment group and user B is in the control group and the two users are connected. If the ranking of the items on user A's feed is altered it will impact their engagement with the feed and it will indirectly change the items that appear on user B's feed. Thus, there is a spillover and the user responses are no longer independent which in turn means that the estimation of the effect of the treatment will be inaccurate.

2.1.2 Solution

In order to account and compensate for the *network effect* introduced by the network interactions one solution is to change how users are sampled. As mentioned in the introduction the most common sampling method for A/B testing is *uniform random sampling*, in which users are randomly selected for either the control group or the treatment group. This method makes it very difficult to eliminate the network effect, as there is little information regarding the structure of the network and there are thus many confounding factors. [10]

Gui et al. describe alternative sampling methods which form clusters of users and all have the goal of eliminating/limiting the interaction between the control and treatment groups [3].

One of the clustering methods is *Graph Cluster Randomization* and it takes into account the graph structure of the network. The partitioning of users (viewed here as nodes) is done by assigning certain users as cluster *center* nodes and then adding the nearest neighbor nodes to that cluster. After this partition, each user is part of a cluster that contains other users, all of which are closely interconnected. The final step of the method is to assign versions (treatment and control) to the users. This is done by randomly assigning one version to each cluster, meaning that all users within a cluster have the same version, thus reducing the spillover effects between the control and treatment groups. Gui et al. state that the advantages of this method include ”the ability to study the interference within the same cluster and the ability to control for ’contamination’ across clusters”. The authors do however assert that this method isn’t sufficient enough at eliminating the network effect since it allows for very large clusters with large social influence. This does, according to them, result in a biased estimation of the effect of the treatment.[3]

Gui et al., therefore, introduce an extension to the *Graph Cluster Randomization* method which aims at creating clusters of equal size. This method is called *Randomized Balanced Graph Partition*, and it uses an iterative process of creating the clusters. First, a partition of the graph is created where all clusters are of equal size. Then the clusters are updated in steps where the total edges in each cluster are maximized. The results of Gui et al.’s study show that this improved method manages to achieve a better estimation of the effect of the change introduced than both the *Graph Cluster Randomization* method and the *uniform random sampling* method. Therefore, it presents a possible solution to the challenge of network interactions, even though it isn’t perfect. [3]

2.2 Estimating Long-term Effects

2.2.1 Problem

One downside of A/B-tests is that they generally favor short-term improvements [4], especially if metrics like profit and user engagement are prioritized over user satisfaction. For example, a news site with a lot of "clickbait" headlines and advertisements could be very profitable and perform very well in an A/B test, which might lead developers to choose this design. However, a version of the site with more accurate headlines might be better at retaining users but that might not be reflected in the test. It is important to strike a balance between profit and user satisfaction because they don't always go hand in hand.

The novelty effect [6] is also important to consider in A/B testing. This is a cognitive bias that in this context means that a new feature might initially engage users simply because they are curious and not really because it is better. In a paper by researchers at Microsoft [7], they present the results of an A/B test of a button on the Microsoft News homepage. While the new button was clicked significantly more often at the start of the experiment the difference in the number of clicks was rapidly decreasing over the duration of the experiment.

This bias only affects users who have used the previous version, for a first-time user both the A and B versions are new so, therefore, the novelty is canceled out.

2.2.2 Solution

The obvious solution is to run experiments for an extended amount of time. The downside of this is mainly increased cost but also increased development which is often not desirable since many software companies today have adopted agile software development. This means that to development cycles are short and new features are shipped as quickly as possible [4]. Instead, the most popular approach is to use observations from shorter experiments and combine them with statistical models to try and estimate user learning, i.e. how users will respond to the new feature over time.

One method of estimating user learning was first proposed by researchers at Google [5] to investigate a user learning effect referred to as *ads blindness and sightedness*. In short, this is a phenomenon where users who are exposed to many advertisements may over time "learn" to ignore them. In order to measure this effect, the experiment in the paper was divided into three phases (Figure 2). During the first phase, or *pre-period* phase, the two groups both received the control version to determine that there were no significant differences between the two. The second phase was like a traditional A/B test. This is where the groups received different treatments, in this case meaning that they were presented with different versions of the site. This is the phase where the short-term impact can be measured and this is also where user learning (might) happen when users adapt to the new version. The third phase, or *post-period*, was the same as the first one, i.e. both groups were yet again exposed to the control version. This is where learning effects can be observed in the group who were exposed to the experimental version during the second phase. Using the observed learning effects during this stage you can try and predict learning effects further into the future using extrapolation. According to the authors, this method has been used successfully at Google and has helped them improve the user experience.

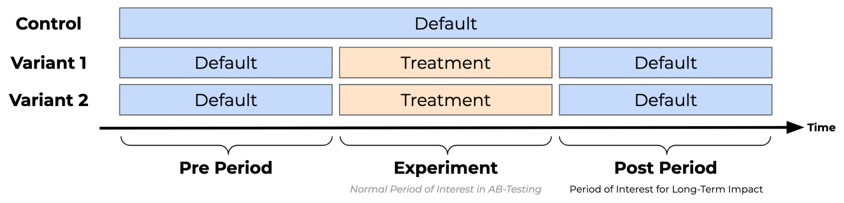

Figure 2: Post-Period (PP) Method. Source: Towards Data Science

3 Conclusion

In this paper, we introduce and describe the widely used research methodology of A/B testing. It provides a way of inferring the causal effect of new changes and thus allows for evidence-based decision-making. Furthermore, this paper explores some of the challenges that exist when conducting A/B tests. Specifically, the network effect well as long-term effects is investigated. The network effect arises when network interactions occur between users (the research subjects), which leads to incorrect results as the user responses are no longer independent. We introduce clustering sampling methods, proposed by Gui et al., as solutions to this challenge, with the *Randomized Balanced Graph Partition* being the preferred method. To address the problem of estimating long-term effects in A/B tests, we present a method from Google to estimate user learning. These are just some of the potential pitfalls in A/B-testing that are important to avoid in order to reap the full benefits of this method. *Because it is such a widely used method in industry, it is important, especially for new software engineers, to be aware of the potential problems and how to overcome them.*

4 Reflection

As described in section 2 A/B testing seemed rather simple at first. However, as we explored more of the subject and the potential problem you have to be aware of in order to correctly implement the methodology we changed our minds and learned quite a bit. Take the challenges that come with network interactions as an example. The papers that explore this topic contain a fair bit of mathematical and statistical content, which always provides opportunities for learning. Being able to read and understand such papers is a very beneficial skill for engineers. Furthermore, A/B testing is utilized especially frequently for online websites, including large companies such as Google, Amazon, and Twitter. The reliance and widespread use of A/B testing, within software development especially, makes it very relevant and it is beneficial knowledge for us in our future professional careers.

References

[1] A/B testing. *A/B testing — Wikipedia, The Free Encyclopedia.* [Online; accessed 03-May-2023]. 2023. URL: `https://en.wikipedia.org/wiki/A/B_testing`.

[2] Jian CHEN and Masashi TOYODA. "A/B Testing for Social Network Services with Directed User Graphs". In: ().

[3] Huan Gui et al. "Network A/B Testing: From Sampling to Estimation". In: *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015.* Ed. by Aldo Gangemi, Stefano Leonardi, and Alessandro Panconesi. ACM, 2015, pp. 399–409. DOI: `10.1145/2736277.2741081`. URL: `https://doi.org/10.1145/2736277.2741081`.

[4] Somit Gupta et al. "Top Challenges from the first Practical Online Controlled Experiments Summit". In: *SIGKDD Explorations* 21.1 (2019). URL: `https://bit.ly/ControlledExperimentsSummit1`.

[5] Henning Hohnhold, Deirdre O'Brien, and Diane Tang. "Focus on the Long-Term: It's better for Users and Business". In: *Proceedings 21st Conference on Knowledge Discovery and Data Mining.* Sydney, Australia, 2015. URL: `http://dl.acm.org/citation.cfm?doid=2783258.2788583`.

[6] Novelty effect. *Novelty effect — Wikipedia, The Free Encyclopedia.* [Online; accessed 07-May-2023]. 2023. URL: `https://en.wikipedia.org/wiki/Novelty_effect`.

[7] Soheil Sadeghi et al. *Novelty and Primacy: A Long-Term Estimator for Online Experiments.* 2021. arXiv: `2102.12893 [cs.HC]`.

[8] Martin Saveski et al. "Detecting Network Effects: Randomizing Over Randomized Experiments". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017.* ACM, 2017, pp. 1027–1035. DOI: `10.1145/3097983.3098192`. URL: `https://doi.org/10.1145/3097983.3098192`.

[9] Ya Xu et al. "From Infrastructure to Culture: A/B Testing Challenges in Large Scale Social Networks". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*. Ed. by Longbing Cao et al. ACM, 2015, pp. 2227–2236. DOI: `10.1145/2783258.2788602`. URL: `https://doi.org/10.1145/2783258.2788602`.

[10] Yifan Zhou et al. "Cluster-adaptive network a/b testing: From randomization to estimation". In: *arXiv preprint arXiv:2008.08648* (2020).