# How A/B Tests Could Go Wrong: Automatic Diagnosis of Invalid Online Experiments*

Nanyu Chen
LinkedIn Corporation
Sunnyvale, CA
nchen@linkedin.com

Min Liu
LinkedIn Corporation
Sunnyvale, CA
mliu@linkedin.com

Ya Xu
LinkedIn Corporation
Sunnyvale, CA
yaxu@linkedin.com

## ABSTRACT

We have seen a massive growth of online experiments at Internet companies. Although conceptually simple, A/B tests can easily go wrong in the hands of inexperienced users and on an A/B testing platform with little governance. An invalid A/B test hurts the business by leading to non-optimal decisions. Therefore, it is now more important than ever to create an intelligent A/B platform that democratizes A/B testing and allows everyone to make quality decisions through built-in detection and diagnosis of invalid tests. In this paper, we share how we mined through historical A/B tests and identified the most common causes for invalid tests, ranging from biased design, self-selection bias to attempting to generalize A/B test result beyond the experiment population and time frame. Furthermore, we also developed scalable algorithms to automatically detect invalid A/B tests and diagnose the root cause of invalidity. Surfacing up invalidity not only improved decision quality, but also served as a user education and reduced problematic experiment designs in the long run.

## CCS CONCEPTS

• **Mathematics of computing**; • **Probabilistic inference problems**; • **Computing methodologies**; • **Causal reasoning and diagnostics**;

## KEYWORDS

A/B testing, experimentation, controlled experiment, causal inference, algorithms, automatic decision making

**ACM Reference Format:**
Nanyu Chen, Min Liu, and Ya Xu. 2019. How A/B Tests Could Go Wrong: Automatic Diagnosis of Invalid Online Experiments. In *The Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19), February 11–15, 2019, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3289600.3291000

*Produces the permission block, and copyright information


*WSDM '19, February 11–15, 2019, Melbourne, VIC, Australia*

ACM ISBN 978-1-4503-5940-5/19/02...$15.00
https://doi.org/10.1145/3289600.3291000

## 1 INTRODUCTION

A/B testing, also known as controlled experiment [4, 8], has gained popularity as the gold standard for evaluating new product ideas across the industry. Many companies have built in-house A/B testing platforms to meet their complex experimentation needs. Some have been discussed at length in past papers [3, 11, 19, 21], including best practices and pitfalls [7, 10].

Over the past few years we have seen a massive growth of online experiments at Internet companies. As a manifest of our 'test everything' mindset, not only has the portfolio of experiments grown to be more diverse, the number of metrics generated for each experiment has also grown into the thousands. While our experimentation platform is powerful and flexible enough to enable automated design and analysis, it has become apparent that not everyone is able to design valid A/B tests and correctly digest results on thousands of metrics without guidance. Invalid A/B tests and misinterpretation of test results lead to non-optimal decisions as well as inaccurate expectations of company growth in the long run. The goal of our work is to fill such gap by automatically detecting and diagnosing experiments that are invalid, both internally and externally.

When an experiment is *internally invalid*, the observed difference between treatment and control may not be the outcome of the experiment treatment. We might incorrectly conclude a treatment effect when there isn't one, or even reverse the direction of the impact. In this paper, we focus on the most prevalent internal invalidity – incomparable samples. In our A/B testing platform, about 10% of experiments suffer from this type of internal invalidity. Needless to say, it is a nightmare to get biased results as it renders the whole A/B test in vain. Without proper diagnosis algorithms, finding the root cause and fixing it can often take longer than running the A/B test itself. After analyzing a large number of biased experiments, we are able to summarize and categorize various types of internal invalidity into an automated toolkit, which reduced root cause investigation time from weeks to within hours. The details of this topic will be discussed in section 3.

*External invalidity*, on the other hand, happens when an experiment itself is unbiased, but the result gets generalized beyond the settings of the experiment and no longer holds valid. For example, we could incorrectly generalize the results beyond the experiment population by experimenting on a subpopulation and assuming impact on the whole population is the same. An equally dangerous but more insidious generalization is generalizing beyond the experiment time horizon without ensuring long-term and short-term impacts are the same. Typically an experiment runs for no longer than a week due to the opportunity cost of A/B testing [22]. When running an A/B test, apart from picking the winning treatment, we

also want to measure the impact of the winner. In particular, we wish to state "if we launch the treatment, metrics will grow by $\Delta\%$ over the next quarter". This statement implicitly assumes impact measured in a week long experiment holds for a quarter, which is clearly not true when the treatment effect is time dependent. To complicate things even more, not all metrics show time dependent effects together in an experiment. Therefore, without an automated detection algorithm, it is impractical for even the most experienced experimenter to sift through thousands of metrics, looking out for time dependent treatment effect. In Sections 4.1 and 4.2 we define, formulate and introduce detection algorithms for two most common time dependent treatment effects, *novelty effect* [13] and *trigger-day effect*. What is very intriguing is that these two effects are entirely different in root cause but very similar in presentation. In Section 4, we will discuss how to differentially diagnose them and give different recommendations on estimating the long-term impact. Even though some past papers [10, 12] have given practical examples of novelty effect, they were based on heuristics and did not discuss how to automatically detect such effect.

Here is a summary of our **main contributions** in this paper:

- As far as we know, we are the first to productionize automated algorithms to detect and diagnose invalid A/B tests in large scale experimentation systems
- We categorize a few common causes of internal invalidity, and share many real experiments as examples, together with learnings from studying such examples.
- We investigate a few root causes for time dependent treatment effect, and for the first time show that some experiments can appear to have time dependent impact purely because of triggering. We name this new phenomenon *trigger-day effect*.
- The algorithms we share are generally applicable. They leverage the most generic summary statistics of an experiment and hence are inexpensive to implement on any A/B testing platform.

## 2 TRIGGERED ANALYSIS

Much of the discussion in this paper requires understanding of triggered analysis, an analysis that includes only users who could have experienced the change introduced by the experiment. We devote this section to discussing concepts related to triggered analysis and introduce notation that will be used throughout the paper.

Triggered analysis benefits an experiment through improved sensitivity compared to all-up analysis [6], or targeted analysis, where all active users are included in analysis. As an example, a website could have 20 million daily active users, but only 2 million of them visited the "jobs" page where the experiment is actually run [21]. Without identifying who those 2 million users are, it is hard to isolate the real experiment impact from noise, especially in experiments where only a small fraction of active users actually experience the change.

Multiple types of triggered analysis are discussed in [6], including the most commonly used session-triggering and user-triggering. In this paper, we focus on user-triggered analysis, which includes all activities of a user after she triggers. User-triggering is more popular as it does not assume treatment effect only exists in the sessions that users trigger [6].

In practice, triggered users are identified through tracking fired during code calls. An example shared in [21] identifies triggered users through events fired during the evaluation of an experiment in the code:

$$String\ color = client.getTreatment(memberID, "buttonColor")$$

In fact, getting the code to work is not equivalent to being able to run a successful A/B test. Where this line is placed in the code is crucial from analysis perspective as we will see in examples in later sections.

Before reaching the main body of this paper, we explain a few key concepts related to triggering.

### 2.1 Cross-day VS Single-day Impact

For an experiment that runs for a total of $k$ days, one typically computes both the

- *cross-day impact* from day 1 to day $t$ for $1 \le t \le k$, which is calculated based on all users who trigger on any day between 1 and $t$ and their metrics in the same period.
- *single-day impact* on day $t$ for $1 \le t \le k$ which is calculated from users who trigger on day $t$ and their metrics on the day

We use superscripts to denote analysis date range. $n^{[1,t]}$, $X_i^{[1,t]}$, $\Delta\%^{[1,t]}$ represent the sample size, total metric values of user $i$, and percentage impact between day 1 and day $t$ inclusive. Similarly, single-day stats are indexed as $n^{[t,t]}$, $X_i^{[t,t]}$, $\Delta\%^{[t,t]}$.

### 2.2 Fully-covered VS Partially-covered Metrics

As described previously, once a user triggers in an experiment, all activities in the analysis period are counted towards the final A/B test result, even on days she does not trigger. This leads to a classification of metrics that turned out to have unique properties in experimental analysis. Each metric belongs to one of the following two categories relative to the trigger condition:

- *fully-covered metrics* when all the channels through which users can contribute to the metric are covered by the trigger condition. In other words, a user will not have any value in this metric unless she triggers experiment.
- *partially-covered metrics* when a strict subset of the channels are covered by the trigger condition.

For example, for the metric total page views, users can generate page views through many channels such as home page, search page, etc. For an experiment that triggers on visiting the site, total page views would be a fully-covered metric, as a user cannot have page views unless she visits. On the other hand, for an experiment that triggers on search page, total page views would be partially-covered, because the channel home page is not covered by trigger condition.

### 2.3 In-trigger VS. Off-trigger Impact

For a partially-covered metric, we can define two types of metric impacts:

- *in-trigger impact*, the treatment impact that happens on days users trigger

- *off-trigger impact*, the treatment impact that happens on on days users do **not** trigger

By definition the off-trigger impact of a fully-covered metric is always 0. The concepts of fully vs. partially-covered and in vs. off-trigger are central to the discussion of trigger-day effect in Section 4.2.

# 3 INTERNAL INVALIDITY

Internal Invalidity refers to a situation when something other than the treatment affects the metric. If the marginal distribution of each user being part of treatment or control is independent $Bernoulli(p)$ ($p$ is the desired % of users in treatment) , all confounders are balanced and one can estimate treatment effect under Rubin's causal model [16] with $\Delta$ defined as

$$\Delta = \frac{\sum_{i \in T} X_i}{\sum_{i \in T} 1} - \frac{\sum_{i \in C} X_i}{\sum_{i \in C} 1} \tag{1}$$

where $X_i$ denotes the metric value of the $i^{th}$ user. However, $\Delta$ may not be a consistent estimator of treatment effect for two major reasons, resulting in internal invalidity. First, the Stable Unit Treatment Value Assumption (SUTVA) [17] is not satisfied. SUTVA states that every user's behavior is affected only by their own treatment and not by the treatment of any other users. This assumption does not hold in social networks when users inter-connect with each other and the treatment 'spills over' to the control side. Second, the samples used for analysis are not random (the assignment scheme for a subset or all of the users are not $Bernoulli(p)$). In this case, the unconfoundedness assumption [15] does not hold and the intrinsic differences in the samples is a plausible alternative explanation for the $\Delta$ observed. In this paper, we focus on cases when unconfoundedness assumption breaks in controlled experiments. Procedures to deal with treatment spill-over and confounding in natural experiments can be found in [18, 20].

One might assume that the unconfoundedness naturally holds in a controlled experiment. However, in a practical setting of online A/B testing system, true randomness is not trivially guaranteed. Unlikely traditional experiments where samples are fixed ahead of time, samples in online experiments are usually collected sequentially over time as users trigger into the experiment by visiting the affected part of the site. There are many practical reasons where a user is only going to trigger when they are in treatment and not in control, or vice versa. As a result, the assignment scheme for the user is not a Bernoulli random variable, leading to internal invalidity.

The rest of this section is organized as follows. We first describe a simple mechanism that can efficiently detect biased experiments. We then devote section 3.2 to discuss five most common causes based on our experience of diagnosing a large number of real experiments, ranging from design imposed reasons to user self-selection. Lastly, we discuss the logic of a toolkit we built to automatically root-cause a biased experiment. The toolkit is deployed and reduces diagnosis time from weeks to within hours.

## 3.1 Detecting Internal Invalidity

It is important to use a robust statistical test to identify experiments that truly suffer from internal invalidity. Detecting a biased experiment is very similar to testing whether a coin being tossed is biased based on the outcome of tossing it many times. This hypothesis can be efficiently tested with the chi-squared test of goodness of fit [14]. Suppose we observe sample sizes $N_t$ and $N_c$ in treatment and control in an experiment with traffic allocation $r_t$ and $r_c$. Without loss of generosity, we assume there are only two variants in the experiment hence $r_c = 1 - r_t$. Also let $E_t = (N_t + N_c) * r_t$ and $E_t = (N_t + N_c) * r_c$ be the expected sample sizes. One can test whether the observed distribution matches expected distribution using the chi-squared statistic $\frac{(N_t - E_t)^2}{E_t} + \frac{(N_c - E_c)^2}{E_c}$ , which has a $\chi_1^2$ distribution under the null hypothesis. We will refer to this test as 'Sample Size Ratio Test' from now on. Experiments failing the Sample Size Ratio Test are likely to have internal invalidity in results and are referred to have *Sample Size Ratio Mismatch* (SSRM). Variations of this method is also discussed in [2].

## 3.2 Common Causes

In this section we describe the characteristics of five most common root causes that results in SSRM.

*3.2.1 Residual Effect.* Residual effect refers to contamination of a former experiment to a subsequent one with the same user split. The process to release a feature through an A/B test usually involves multiple iterations that ramps up site traffic gradually to balance speed, quality and risk [22]. To avoid reverting users back to control once exposed to the treatment, one usually keeps the same user split across iterations. This creates potential residual effect if the treatment changes the probability a user re-triggers the experiment. As long as one counts unique users from the very beginning of the experiment, the sample sizes should match expectation. However, to avoid Simspon's paradox [5], one usually needs to restart analysis when ramping up the experiment. In this case the sample size would not match expectation when user re-trigger rate is changed as users are counted from a time point after the experiment initially started.

We observed such an effect in an experiment to evaluate a new People You May Know algorithm. The experiment showed promising results on engagement across the site but started showing SSRM after the first iteration. After careful investigation, we found that this was due to the fact that the algorithm was so good that it made users come back more often! In particular, we separated users who triggered the experiment for the first time and returned users, and there was no mismatch on the new users. Second, we counted the sample sizes using a second source of data that tracks users visiting the site and observed similar mismatch. This rules out the possible reason that the mismatch was due to experiment tracking issues.

It is worth pointing out that organic residual effect can rarely be so severe that it causes an SSRM. There are other causes that create 'residual-alike' symptoms in an experiment and one needs to carefully collect evidence to rule out other causes.

When residual effect is the root cause, the sample size ratio usually converge to expected ratio as the analysis period gets longer. The bias can be corrected by re-randomization and counting unique users from the very beginning of the experiment.

*3.2.2 Pre-trigger Condition Bias.* Even if there is no organic user residual effect, experiments can be implemented in a way where users' re-trigger rates differ significantly between treated and control users. While triggering an experiment in the code may be a simple one liner, the order at which logics are applied may determine whether or not the experiment will result in SSRM. To illustrate this with a real example, a feature was launched to allow users to rebuild their own news feed by following content they care about. To improve awareness of the feature, a cross-promotional [1] widget was shown to users at the top of the feed page (as shown in figure 1 right). The widget was launched via an A/B test so that we can measure its two-fold impact. On one hand, the widget is expected to improve the adoption of the follow-feed product. On the other, it might hurt member engagement on feed as members are distracted away by the widget. Moreover, to avoid having users repeatedly seeing the widget every time they come back, a user becomes a "cool-off" state when they have seen the widget for more than twice or clicked on it at least once. Initially, this cross promotion experiment was implemented with the logic in figure 1 left. While the logic of the code seems to serve the experiment with the correct logic, we saw severe SSRM after the experiment was launched.

As described in section 2, triggered analysis relies on runtime tracking events that are fired during variant evaluation (line (3)). The logic above does not fire tracking events when a user is cooled off. Since only users in treatment group will ever be in cool-off period after conditions are met, there will be proportionally less unique users in treatment group unless we count members from the very beginning of the experiment. Similar to what is observed in a residual effect, one would start to see SSRM as soon as the analysis restarts.

Note that the analysis based on mismatched users from logic above are usually negatively biased since users who are more active are more likely to trigger into experiments early on, and hence are more likely to be excluded from the analysis report. The solution to this problem is to exchange lines (1) and (3).

```
If (member in cool-off):        (1)
       Do not show widget       (2)
Else if (member in control):    (3)
       Do not show widget       (4)
Else:                           (5)
       Show widget              (6)
```

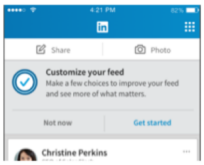


**Figure 1: Left: An implementation of cross-promotion widget experiment that results in biased samples. Right: A visualization of the widget**

It is worth pointing out that this sample size ratio shares common characteristics as that in a residual effect. However, we are sure the root cause of the bias in this experiment is not organic residual effect because we did not observe a mismatch if we looked at sample counts from alternative tracking events, such as an event tracking users' page views.

*3.2.3 Dynamic Targeting.* Another source of internal invalidity comes from targeting. Targeting [7] refers to running experiments on specific user sets based on their properties and activities to deliver personalized experience. The type of attributes used for targeting can range from more static primitives, such as country, industry and locale, to user attributes that capture user engagement level or status at the time, such as connection count, job seeking status, profile completeness. If the targeting attribute is static, the set of users who are targeted is fixed and there is little chance for bias. However, the latter categories are prone to frequent changes. If one needs to target based on these dynamic attributes, one needs to make a trade-off between the recency of the attributes and potential biases, as the treatment can change the number of users targeted. If target attribute is changed after the experiment is started, one might end up with samples that are not comparable.

In most cases, bias from targeting also surfaces as a residual-alike effect, as the previous iteration often leads to biases in the next. We have seen experiments that targets dormant users, where the treatment attempts to re-engage them with emails. As some users engage through emails, they are no longer classified as dormant users, leading to biases in the next iteration. Another example is an experiment that was ran on users with incomplete profile information and the treatment attempts to make users complete profile.

However, in more complex experiments, such bias can surface even in the first iteration. We share one such example through an experiment that was run to test the relevance of the Jobs You May Be Interested In (JYMBII) algorithm [16]. The experiment was set up so that active job seekers were randomly served algorithms A and B, and the remaining users were randomly served algorithms C and D. The same random user split was applied on active job seekers and other users. Whether a user is an active job seeker is a targeting attribute generated by an independent machine learning algorithm that takes into account user interaction with jobs. This experiment started to fail sample size ratio test after the first few days. More intriguingly, the test only failed in the active job seeker segment (the comparison between A and B) and not on the remaining users.

It turned out that model C was performing so well that it lifted page views and clicks on jobs pages. These activities are features used in the classification model to identify job seekers. Some users were converted to active job seeker and fall into the model A variant, resulting in more than expected user count in model A. Even though the remaining user segments do not fail sample size ratio test, the comparison between model C and D was polluted as some users who were included in C were in fact getting algorithm A.

*3.2.4 Biased implementation.* Getting an experiment to serve as expected does not guarantee quality analysis. There are many scenarios where engineers implement an experiment without considering validity in analysis. We share such pitfalls through an experiments to evaluate a re-vamped desktop homepage. Users can trigger the experiment in two scenarios during the experiment. First, a user is triggered when she hits the top-level traffic router by entering through the main site URL. Second, there are designated URLs that leads to the new homepage. To make sure users cannot access those pages if the experiment was turned off, engineers decided to check the variant assignment (and hence triggered members) when users hit those URLs. On the other hand, such check was determined unnecessary on URLs leading to old homepage.

As the experiment was running, we observed more than expected members in the treatment group. It turned out that this was merely due to the fact that some users enter the site through direct URLs (without going through the router) and there was disproportionately more members from treatment that entered through the new homepage URLs, leading to additional triggered user count. Again, unless one explicitly think through the consequences on analysis, optimizing for engineering functionality does not guarantee quality experiment.

*3.2.5 Dependent Experiments.* Another common cause of invalidity results from running multiple dependent experiments. By using the same user split, one effectively creates a layer in which dependent experiments can be tested in fully or fractionally factorial fashion [7]. The danger for bias comes in when one experiment affects the trigger rate of another, either explicitly or implicitly. Without exposing the details, we ran two experiments on two different pages. The same user split was used as the treatment of the second experiment cannot work with the control of the first experiment. To observe the effect of the treatment of the first experiment with and without the treatment of the second experiment, the treatment of the first experiment was ramped to 50 percent of users while the second experiment was ramped to 25 percent (see figure 2 for details). During analysis of this experiment, we found severe SSRM in the first experiment while the second experiment matched expectation. After a series of investigation, we identified the fact that in certain scenarios the second experiment would result in the users landing on the page of the first experiment while the control would not.

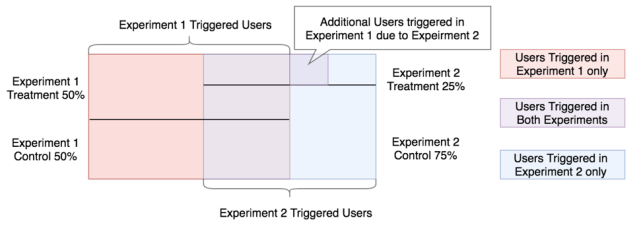


**Figure 2: Example of Sample Size Ratio Mismatch resulted from Dependent Experiments**

## 3.3 Diagnosis

We shared five most common root causes of invalid A/B tests due to biased samples. In this section, we describe our procedure to diagnose the root cause of biased experiments once they are deemed problematic.

In fact, diagnosis comes down to running tests that uniquely identifies each of the root causes. A good analogy is when a patient goes to a doctor to diagnosis a potential disease. The doctor would run a sequence of tests to come up with a conclusion. We name our tool 'SSRM-Clinic' to resemble this analogy. Here are the tests that are automatically run inside the clinic:

- **Targeting Ratio Test** Run sample size ratio test for all targeted users, regardless of whether they trigger or not. Note that if the system updates the targeted users during the experiment, one needs to compute the targeted sample sizes for the entire experimental period being considered.
- **New User Ratio Test** Compute sample size ratios separately for users who trigger the experiment for the first-time and users who are returned users. This will allow us to separate causes that result in bias only in the feedback loop, such as cool-off, residual effect and causes that affect all users, such as biased implementation and dependent experiments. New users being unbiased also suggests that rehashing and counting users from the beginning will result in matched samples. With the default single-day and cross-day sample sizes on a daily basis, one can derive new and returned users without explicit computation. As mentioned earlier, on day k of an experiment, we have sample sizes $n^{[1,k-1]}$, $n^{[1,k]}$, $n^{[k,k]}$ by default and one can simply conclude

$$n^k_{returned} = n^{[1,k-1]} + n^{[k,k]} - n^{[1,k]}$$
$$n^k_{new} = n^{[1,k]} - n^{[1,k-1]}$$

The output of this test is whether the experiment has a SSRM on new user only.

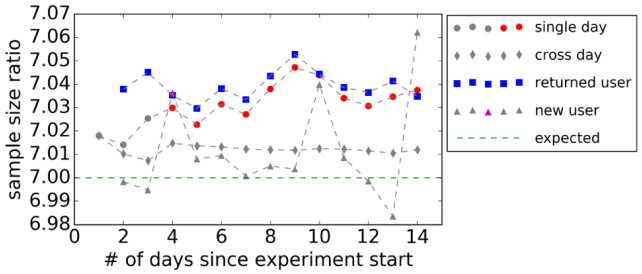


**Figure 3: Sample Size Ratio for New and Returned Users of the People You May Know experiment**

- **Alternative Tracking Test** Whenever possible, identify trigger condition through another tracking event. For most tests triggering on visiting particular pages, one may be able to use tracking of page views. This test can distinguish between *Pre-trigger Condition Bias* and the organic *Residual Effect* as the former should not have mismatched sample sizes based on such alternative events.
- **Meta-data Breakdown Test** This test is motivated by the biased implementation example shared in the previous section. One might have meta-data about each of the possible piece of code that triggers users in an experiment. In our case, the service name is used as there does not need to more than one place to trigger users within each service. A biased implementation often involves triggering in multiple services and testing sample sizes triggered per service can effectively identify the culprit.
- **Experiment Overlap Test** When there are multiple experiments sharing user split, these experiments can affect each other and result in potential SSRM. As motivated by the example in the previous section, one can establish evidence of whether two specific experiments are affecting each other

by looking at the overlapping, hence union-ed, sample sizes. Precisely, in an ideal setting, one breaks down the sample of each variant into the following:
- $B_1$: number of users triggering the first experiment only.
- $B_2$: number of users triggering the second experiment only.
- $O_1$ number of users triggering both experiments but triggered the first experiment first.
- $O_2$ number of users triggering both experiments but triggered the second experiment first.

When the second experiment causes SSRM on the first, one would observe no SSRM on $B_1 + O_1$ while SSRM on $O_2$, as the former is the portion of samples unaffected by the second experiment.

In practice, breaking down $O_1$ and $O_2$ is challenging as one would need to rely on there being trustworthy fields to identify chronological order of triggering. In fact, a necessary condition for the observed SSRM in experiment being caused by the second experiment is that the union-ed sample sizes ($B_1 + B_2 + O_1 + O_2$) does not have SSRM. This is a cheaper test that we refer to as "Experiment Overlap Test" which is effective in practice.

With the outcome of the tests above collectively, one can identify a unique root-cause from the five categories in the previous section (Each row is distinct in Figure 4). In practice, the clinic has been helpful in identifying the potential areas of problems but human judgement is still critical to fully nail down the root cause. In particular, we have seen experiments with hybrid causes that makes automatic decision flow extremely challenging.

| Test Outcome | Targeting Ratio Test fails? | New User Ratio Test fails? | Alternative Tracking Test fails? | Meta-data Breakdown Test fails? | Experiment Overlap Test fails? |
|---|---|---|---|---|---|
| Residual Effect | × | × | √ | × | √/NA |
| Pre-trigger Condition Bias | × | × | × | × | √/NA |
| Dynamic Targeting | √ | √/× | × | × | √/NA |
| Biased Implementation | × | √ | √ | √ | √/NA |
| Dependent Experiments | × | √ | √ | × | × |

**Figure 4: Diagnosis outcome for different root causes**

## 4 EXTERNAL INVALIDITY

*External validity* refers to the extent to which the results of an A/B test can be generalized to other settings, other population and over time.

When running an A/B test, apart from picking the best treatment, we also want to conclude what is the impact of it, in particular, we wish to make statements like "if we launch the treatment to all users, revenue will grow by Δ% over the next quarter". This statement, however, implicitly generalizes the A/B test result to other populations and over time. Even when the test itself is internally valid, the final impact statement could still be incorrect if we did not pay close attention to the generalization process. In our practice, the most common types of generalizations are the following:

1. **Generalizing beyond the experiment population** It is very important to watch out for attempts to infer treatment impact on a population that is different from the A/B test population. We usually start A/B tests with a pilot group of users. After a winner is identified and *local* impact measured on the pilot group, we cannot simply assume the treatment impact on users outside the pilot group is the same. If we want to understand the impact on a new population, we should do so by running an A/B test on that population. After A/B testing on multiple non-overlapping subpopulations and getting *local* impacts, we can convert the local impacts to *site wide impacts* and then add them up to get the global impact. The definition and computation of site-wide impact are discussed in detail in [21].
2. **Generalizing over time** Sometimes the treatment effect is time dependent, and in such scenarios extra care needs to be taken when stating the treatment effect. Because running an A/B test is costly, we usually limit the experiment duration to within a week [22]. Although it is completely valid to claim the treatment made an impact of Δ% during the experiment, we can break external validity by stating "if we launch the treatment, metric is going to grow by Δ%", and baking the growth numbers into the quarterly or annual financial forecasts. Such statements implicitly assumes the impact measured in a short-term week-long experiment can be generalized to long-term, which is clearly not valid when treatment effect itself is time dependent.

In this paper, we focus on addressing the external invalidity entailed by time dependent treatment effect since the population generalization problem was thoroughly covered in [21]. There are mainly two root causes of time variant treatment effect, novelty effect and trigger-day effect. The first effect is a result of change in user behavior over time, while the second effect arises from the nature of triggered analysis, hence the name "trigger-day effect". They are similar in presentation but the root causes are very different and inform us of different insights about the experiment. Therefore, it is very important that we are able to automatically detect these effects, as well as distinguish them through automatic "differential diagnosis".

### 4.1 Novelty Effect

Novelty effect stems from change in user behavior or ecosystem. Sometimes a user 's reaction to a treatment can be different the first time she experiences the it, vs. when she has experienced it many times. For example, users might be very responsive to notifications at first, but over time learn to ignore or disable them [9], thus making the treatment less effective.

The fact that the treatment effect increases/decreases as users trigger it more means we would see a strong and consistent trend in both single-day and cross-day impact when novelty effect exists(see Figure 5 left for an example). This is because each single-day analysis population on average have triggered the experiment more than previous day 's single-day population (see Figure 5 right for the average number of days triggered in the experiment in Figure 5 left).

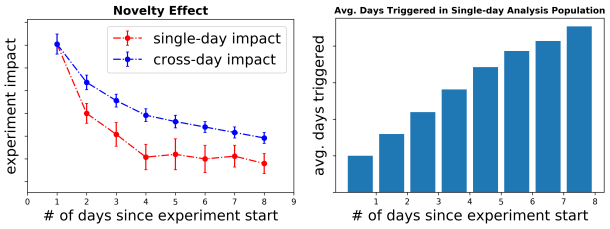


**Figure 5: Example of Novelty Effect**

With the observations above, we propose the following novelty effect detection algorithm:

1. Run a regression $\Delta\%^{[t,t]} = \beta_0 + \beta_1 \frac{1}{t^\alpha} + \beta_2 \frac{1}{t^\gamma}$ with one week's single-day impacts. $\alpha$ and $\gamma$ are chosen such that $\frac{1}{t^\alpha}$ is a slow-decay term with respect to $t$ while $\frac{1}{t^\gamma}$ is a fast-decay term (for example, $\alpha = 0.35$, $\gamma = 2$). These two terms were chosen to represent the two typical types of trend we have identified in thousands of real A/B tests, as illustrated in Figure 6. The slow-decay term fits the gradual type of trend on the left, while the fast-decay term fits well with the 'elbow' type on the right. A linear combination of the two captures trends in between.

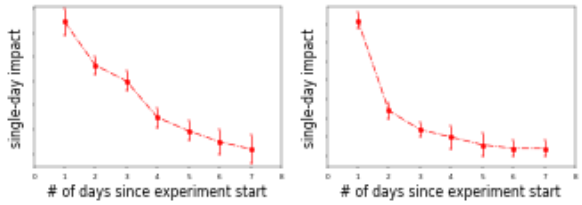


**Figure 6: Two Typical Types of Single-day Impact Trend**

2. Novelty effect is flagged when all following conditions are met: (a) The linear model captures the single-day effect trend well, in practice we require $R^2 \geq 0.8$; (b) The fitted line is monotonic in $t$; (c) The largest single-day impact is statistically significantly different from smallest single-day impact. A smaller significance level (e.g. at 0.005) is helpful in limiting the number of false positives. This condition has proven to be very effective in teasing out spurious novelty effects [10] from the final results.
3. Step 1 and 2 guarantees the A/B single-day impact exhibits strong trend, it does not however guarantee the trend is caused by novelty effect. Other possible causes for a trend in single-day impacts are: (a) day-of-week effect, where users triggering the experiment on weekends are fundamentally different from those triggering on weekdays, so an experiment starting on a Saturday or Sunday would also show a trend in single-day impact resembling that of the 'elbow' type; (b) seasonality in treatment effect[9], which we have not seen in practice, but is possible theoretically. The **definitive test** for novelty effect is a cohort-analysis comparing users who have been exposed to the treatment for a while v.s. those newly exposed to the treatment. We randomly select two groups of users, expose the first group to treatment for some time, and then expose the second group, and immediately compare the metrics between the two groups. Because the only difference between these two groups is how long they have been exposed to the treatment, any metric difference can be attributed to novelty effect.

We want to point out that it is impossible to estimate the magnitude of novelty effect based on a sequence of single-day impacts alone, because on any given day, the single-day analysis population is a mix of users triggering the experiment for the first time, as well as users who have triggered it many times. Furthermore, we cannot predict how users behavior changes just based on the current trend. When measuring novelty effect itself is important, we can again leverage the cohort analysis, exposing the first group to treatment and waiting till single-day impact stabilizes, then expose the second group and the difference in metrics between the two groups is the strength of novelty effect. We also want to point out, attempting to estimate novelty effect by segmenting users based on days triggered would inevitably introduce a selection bias when treatment effect confounds with trigger rate.

This detection algorithm has been in production and based on all the novelty effects identified, we have noticed three types of experiments that are most prone to novelty effect: (1) Big changes where users tend to explore the new features in the initial days; (2) Notification or badging experiments; (3) Recommendation algorithms with limited candidate pool.

## 4.2 Trigger-day Effect

Trigger-day effect is another type of time dependent treatment effect. In fact, it is a lot more prevalent than novelty effect. A major distinction between trigger-day effect and novelty effect is that in trigger-day effect, the cross-day impact has a strong trend while single-day impact stays completely flat apart from the random noise, see Figure 7. for a comparison.

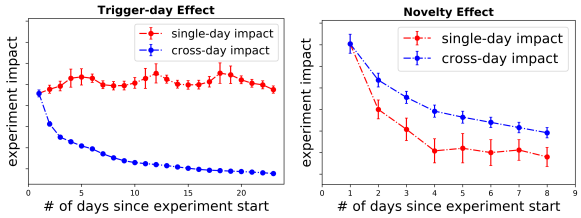


**Figure 7: Contrasting trigger-day effect with novelty effect**

It is quite puzzling how constant single-day impacts give rise to time dependent cross-day impacts. We will explain the mechanism first through some intuitions, and then with mathematical derivations. Trigger-day effect arises when there are multiple channels through which the user can contribute to a metric, and the experiment triggers on a subset of the channels and has a big impact, while having a very smaller or even opposite effect on other channels. In the terminology of Section 2, trigger-day effect happens when the off-trigger impact is very different from in-trigger impact. As an example, suppose users can visit our site either organically or through clicking on a notification, and the baseline organic page

view is one per day. Now a treatment sends out notifications once every week, and the trigger condition is sending notification. On the day of trigger, users view two pages instead, one organically and one through the notification. This is a 100% impact on metric total page views. However, the impact on organic page views is zero, and user's behavior is unaffected on days she does not trigger. Therefore over the duration of a week, the treatment only lifts total page views by $\frac{8}{7} - 1 \approx 14.3\%$. Had we run the experiment for one day only, we would have inferred a 100% impact, when in fact the true impact is only 14.3%. In other words, there is 100% *in-trigger impact*, 0% *off-trigger impact*, and a combined 14.3% *cross-day impact*. With this example, we can see that even though the single-day impacts are constant, the cross-day impact is actually a function of the analysis duration due to a change in mix of in-trigger and off-trigger impacts.

We now formulate trigger-day effect mathematically and understand it beyond heuristics. In the discussion we assume the test does not experience novelty effect at the same time, given it is extremely rare for one test to show both novelty and trigger-day effect. Let $X_i$ be the total metric value of user $i$ in the experiment (e.g. total page views by user $i$ over entire experiment period), $I_i$ be the metric value from triggered days and $O_i$ from non-triggered days. Hence $X_i = I_i + O_i$. For an experiment running for $k$ days,

$$\Delta\%_X^{[1,k]} = \frac{\bar{X_t}^{[1,k]}}{\bar{X_c}^{[1,k]}} - 1 = w_k \Delta\%_I^{[1,k]} + (1 - w_k)\Delta\%_O^{[1,k]} \quad (2)$$

where

$\Delta\%_I^{[1,k]} = \frac{\bar{I_t}^{[1,k]}}{\bar{I_c}^{[1,k]}} - 1$ is the in-trigger impact,

$\Delta\%_O^{[1,k]} = \frac{\bar{O_t}^{[1,k]}}{\bar{O_c}^{[1,k]}} - 1$ is the off-trigger impact,

$w_k = \frac{\bar{I_c}^{[1,k]}}{\bar{X_c}^{[1,k]}}$ is the weight of in-trigger impact in the mix of cross-day impact ($0 \le w_k \le 1$).

We would like to point out a few important observations on the above formulation:

1. The in-trigger impact $\Delta\%_I^{[1,k]}$ is time independent and denoted by $\Delta\%_I$. $\Delta\%_I$ is by expectation equal to single-day impact $\Delta\%_X^{[t,t]}$ for $t = 1, \ldots, k$. This is because on any given day, the single-day analysis population is simply users who trigger on that day, hence single-day impact is an in-trigger impact by definition. Similarly, we write the off-trigger impact $\Delta\%_O^{[1,k]}$ as $\Delta_O$.
2. Cross-day impact $\Delta\%_X^{[1,k]}$ is a weighted average of in-trigger and off-trigger impacts. It differs from in-trigger impact, hence single-day impact, if (a) in-trigger and off-trigger impacts are different and (b) off-trigger impact has a non-zero weight in cross-day impact, i.e. $w_k < 1, \forall k$.
3. Cross-day impact $\Delta\%_X^{[1,k]}$ is time dependent because $w_k$ is. Suppose on any given day a user triggers with probability $p$. Let $r = \frac{\mu_I}{\mu_O}$ with $\mu_I = \mathbb{E}(I)$ and $\mu_O = \mathbb{E}(O)$. For an experiment that ran for $k$ days with a total triggered population of $n$ users, $n\binom{k}{t}p^t(1-p)^{k-t}$ are expected to have triggered experiment on $t$ days. Therefore,

$$\sum_i I_i = \sum_{t=1}^{k} n\binom{k}{t}p^t(1-p)^{k-t}t\mu_I = nkp\mu_I \quad (3)$$

$$\begin{aligned}\sum_i O_i &= \sum_{t=1}^{k} n\binom{k}{t}p^t(1-p)^{k-t}(k-t)\mu_O \\ &= n\mu_O[k\sum_{t=1}^{k}\binom{k}{t}p^t(1-p)^{k-t} - \sum_{t=1}^{k}\binom{k}{t}p^t(1-p)^{k-t}t] \\ &= n\mu_O\{k[1-(1-p)^k] - kp\} = nk[(1-p)-(1-p)^k]\mu_O\end{aligned} \quad (4)$$

Hence $w_k = \frac{\sum I_i}{\sum I_i + \sum O_i} = \frac{pr}{(1-p)-(1-p)^k+pr}$. Intuitively, if the A/B test ran for one day only, then there is no off-trigger impact, i.e. $w_k = 1$ with $k = 1$. On the other hand, as $k \to \infty$, $w_k$ stabilizes to $w_\infty = \frac{1}{1+\frac{1-p}{pr}}$, and cross-day impact $\Delta_X\%$ stabilizes to $w_\infty\Delta_I\% + (1-w_\infty)\Delta_O\%$. With trigger-day effect, it is actually possible to predict how the impact evolves over time and what it approaches to in the long-term, while it is in general impossible to predict how novelty effect evolves. In Figure 8., we show $w_k$ predicted using first week's impacts and real $w_k$ for the A/B test shown in Figure 7. left.

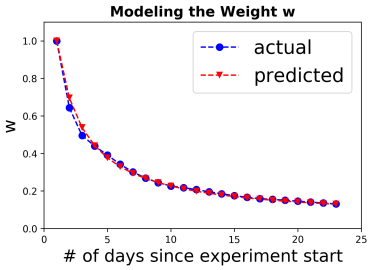


**Figure 8: w evolving over time**

With the above mathematical formulation, we can now design an algorithm to automatically flag trigger-day effect when the following conditions are met:

1. $w$ is reasonably small, otherwise the cross-day impact is predominantly single-day impact and stays constant over time.
2. $\Delta\%_I$ and $\Delta\%_O$, as defined in Equation 2., are statistically significantly different. $\Delta\%_I^{[1,k]}$ and $\Delta\%_O^{[1,k]}$ can be computed from cross-day and single-day summary statistics. In particular, $\bar{I_c} = \frac{\sum_i I_{c,i}}{n_c}$ where $\sum_i I_{c,i}^{[1,k]} = \sum_{t=1}^{k}\sum_i X_i^{[t,t]}$, similarly for $\bar{I_t}$. We also have $\bar{O_t} = \bar{X_t} - \bar{I_t}$ and $\bar{O_c} = \bar{X_c} - \bar{I_c}$. $var(I)$ and $var(O)$ cannot be derived from cross-day and single-day summary stats, but we can bound them. Note that if $cov(I, O) \ge 0$, which is in general true since users who are more engaged on triggered days are on average also more engaged on non-triggered days, then $var(X) = var(I + O) = var(I) + var(O) + 2cov(I, O) \ge var(I)$ and $var(O)$. Because $var(\Delta_I\%) = \frac{var(I_t)}{\bar{I_c}^2 n_t} + \frac{var(I_c)\bar{I_c}^2}{\bar{I_c}^4 n_c}$ based on Delta method, plugging in the upper bound of $var(I)$ into above equation results

in upper bound on $var(\Delta_I\%)$, and similarly for $var(\Delta_O\%)$. Then we can do a t-test between $\Delta_I\%$ and $\Delta_O\%$, whose actual false positive rate is guaranteed to not exceed the nominal test level.

## 5 SUMMARY AND FUTURE WORK

In this paper, we discussed causes and diagnostic algorithms for internally and externally invalid A/B tests. Even though our work has made diagnosis of an internally invalid experiments faster, fixing these experiments still involves a lengthy process. Automatic correction for bias such as computing impact on a targeted population in place of residual effect, or measuring impact on newly triggered users will greatly reduce the time to getting internally valid test results. For external invalidity, we discussed two root causes and detection method of time-varying effect, including the novel concept of trigger-day effect. A useful future development would be triggering an automated cohort analysis as a validation step once novelty effect has been detected.

## 6 ACKNOWLEDGEMENT

The authors would like to thank Iavor Bojinov, Guillaume Saint-Jacques and Ye Tu for thoughtful reviews of the paper. We would also like to thank the experimentation team at LinkedIn, especially Jiada Liu, Amir Sepheri and Weitao Duan for discussions on some of the examples shared in the paper.

## REFERENCES

[1] [n. d.]. Cross Promotion. https://en.wikipedia.org/wiki/Cross-promotion
[2] [n. d.]. Detecting and Avoiding Bucket Imbalance in A/B Tests.
[3] Eytan Bakshy, Dean Eckles, and Michael S Bernstein. 2014. Designing and deploying online field experiments. In *Proceedings of the 23rd international conference on World wide web*. ACM, 283–292.
[4] George EP Box, J Stuart Hunter, and William Gordon Hunter. 2005. *Statistics for experimenters: design, innovation, and discovery*. Vol. 2. Wiley-Interscience New York.
[5] Thomas Crook, Brian Frasca, Ron Kohavi, and Roger Longbotham. 2009. Seven pitfalls to avoid when running controlled experiments on the web. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1105–1114.
[6] pages = 349–358 title = Diluted Treatment Effect Estimation for Trigger Analysis in Online Controlled Experiments year = 2015 Deng, Alex; Hu, Victor, booktitle = Proceedings of the Eighth ACM International Conference on Web Search and Data Mining - WSDM '15. [n. d.].
[7] Pavel Dmitriev, Somit Gupta, Dong Woo Kim, and Garnet Vaz. 2017. A Dirty Dozen: Twelve Common Metric Interpretation Pitfalls in Online Controlled Experiments. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1427–1436.
[8] Alan S Gerber and Donald P Green. 2012. *Field experiments: Design, analysis, and interpretation*. WW Norton.
[9] Henning Hohnhold, Deirdre O'Brien, and Diane Tang. 2015. Focusing on the Long-term: It's Good for Users and Business. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1849–1858.
[10] Ron Kohavi, Alex Deng, Brian Frasca, Roger Longbotham, Toby Walker, and Ya Xu. 2012. Trustworthy online controlled experiments: Five puzzling outcomes explained. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 786–794.
[11] Ron Kohavi, Alex Deng, Brian Frasca, Toby Walker, Ya Xu, and Nils Pohlmann. 2013. Online controlled experiments at large scale. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1168–1176.
[12] Ron Kohavi, Alex Deng, Roger Longbotham, and Ya Xu. 2014. Seven rules of thumb for web site experimenters. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14*. https://doi.org/10.1145/2623330.2623341
[13] Ron Kohavi and Roger Longbotham. 2017. Online controlled experiments and a/b testing. In *Encyclopedia of machine learning and data mining*. Springer, 922–929.
[14] Jon NK Rao and Alastair J Scott. 1981. The analysis of categorical data from complex sample surveys: chi-squared tests for goodness of fit and independence in two-way tables. *Journal of the American statistical association* 76, 374 (1981), 221–230.
[15] Paul R Rosenbaum and Donald B Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70, 1 (1983), 41–55.
[16] Paul R. Rosenbaum and Donald B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* (1983). https://doi.org/10.1093/biomet/70.1.41 arXiv:http://www.jstor.org/stable/2335942
[17] Donald B Rubin. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology* 66, 5 (1974), 688.
[18] Martin Saveski, Jean Pouget-Abadie, Guillaume Saint-Jacques, Weitao Duan, Souvik Ghosh, Ya Xu, and Edoardo M Airoldi. 2017. Detecting network effects: Randomizing over randomized experiments. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 1027–1035.
[19] Diane Tang, Ashish Agarwal, Deirdre O'Brien, and Mike Meyer. 2010. Overlapping experiment infrastructure: More, better, faster experimentation. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 17–26.
[20] Ya Xu and Nanyu Chen. 2016. Evaluating mobile apps with a/b and quasi a/b tests. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 313–322.
[21] Ya Xu, Nanyu Chen, Addrian Fernandez, Omar Sinno, and Anmol Bhasin. 2015. From Infrastructure to Culture. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15*. https://doi.org/10.1145/2783258.2788602
[22] Ya Xu, Weitao Duan, and Shaochen Huang. 2018. SQR: Balancing Speed, Quality and Risk in Online Experiments. *arXiv preprint arXiv:1801.08532* (2018).