19

The A/A Test

If everything seems under control,
you're just not going fast enough
– *Mario Andretti*

If everything is under Control,
then you're running an A/A test
– *Ronny Kohavi*[1]

Why you care: *Running A/A tests is a critical part of establishing trust in an experimentation platform. The idea is so useful because the tests fail many times in practice, which leads to re-evaluating assumptions and identifying bugs.*

The idea of an A/A test is simple: Split the users into two groups as in a regular A/B test but make B identical to A (hence the name A/A test). If the system is operating correctly, then in repeated trials about 5% of the time a given metric should be statistically significant with p-value less than 0.05. When conducting t-tests to compute p-values, the distribution of p-values from repeated trials should be close to a uniform distribution.

Why A/A Tests?

The theory of controlled experiments is well understood, but practical implementations expose multiple pitfalls. The A/A test (Kohavi, Longbotham et al. 2009), sometimes called a Null test (Peterson 2004), is highly useful for establishing trust in your experimentation platform.

[1] https://twitter.com/ronnyk/status/794357535302029312

A/A tests are the same as A/B tests, but Treatment and Control users receive identical experiences. You can use A/A tests for several purposes, such as to:

- Ensure that Type I errors are controlled (e.g., at 5%) as expected. For example, as will be shown in Example 1 later in this chapter, standard variance calculations may be incorrect for some metrics; or the normality assumption may not hold. A/A tests failing at an unexpected rate will point to issues that must be addressed.
- Assessing metrics' variability. We can examine data from an A/A test to establish how a metric's variance changes over time as more users are admitted into the experiment, and the expected reduction in variance of the mean may not materialize (Kohavi et al. 2012).
- Ensure that no bias exists between Treatment and Control users, especially if reusing populations from prior experiments. A/A tests are very effective at identifying biases, especially those introduced at the platform level. For example, Bing uses continuous A/A testing to identify a carry-over effect (or residual effect), where previous experiments would impact subsequent experiments run on the same users (Kohavi et al. 2012).
- Compare data to the system of record. It is common for the A/A test to be used as the first step before starting to use controlled experiment in an organization. If the data is collected using a separate logging system, a good validation step is to make sure key metrics (e.g., number of users, revenue, click-through rate (CTR)) match the system of record.
- If the system of records shows X users visited the website during the experiment and you ran Control and Treatment at 20% each, do you see around 20% X users in each? Are you leaking users?
- Estimate variances for statistical power calculations. A/A tests provide variances of metrics that can help determine how long to run your A/B tests for a given minimal detectable effect.

We highly recommend running continuous A/A tests in parallel with other experiments to uncover problems, including distribution mismatches and platform anomalies.

The following examples highlight the why and how of running A/A tests.

Example 1: Analysis Unit Differs from Randomization Unit

As discussed in Chapter 14, randomizing by user and analyzing by pages is something that may be desired. For example, alerting systems typically look at page-load-time (PLT and CTR by aggregating every page in

near-real-time. Estimating the Treatment effect by page is therefore often needed.

We now look at CTR and discuss the two common ways to compute it, each with different analysis units. The first is to count the clicks and divide by the number of page views; the second is to average each user's CTR and then average all the CTRs. If randomization is done by user, then the first mechanism uses a different analysis unit than the randomization unit, which violates the independence assumption and makes the variance computation more complex.

We analyze both and compare them in this example.

Here, n is the number of users and K_i the number of pageviews for user i. N is the total number of pageviews: $N = \sum_{i=1}^{n} K_i$. $X_{i,j}$ is the number of clicks for user i on their jth page.

Now we look at our two reasonable definitions for CTR:

1. Count all the clicks and divide by the total number of pageviews as shown in Equation 19.1:

$$CTR_1 = \sum_{i=1}^{n} \sum_{j=1}^{K_i} X_{i,j} \Big/ N \tag{19.1}$$

If we have two users, one with no clicks and a single pageview and the other with two clicks, one on each of their two pageviews, then (see Equation 19.2):

$$CTR_1 = \frac{0+2}{1+2} = \frac{2}{3} \tag{19.2}$$

2. Average each user's CTR and then average all CTRs, essentially getting a double average (see Equation 19.3):

$$CTR_2 = \sum_{i=1}^{n} \frac{\sum_{j=1}^{K_i} X_{i,j}}{K_i} \Big/ n \tag{19.3}$$

To apply the example in definition 1 (see Equation 19.4):

$$CTR_2 = \frac{0}{1} + \frac{2}{2} \Big/ 2 = \frac{1}{2} \tag{19.4}$$

There is no right or wrong in these definitions, both are useful definitions for CTR, but using different user averages yields different results. In practice, it

is common to expose both metrics in scorecards, although we generally recommend definition 2 as we find it more robust to outliers, such as bots having many pageviews or clicking often.

It's easy to make mistakes when computing the variance. If the A/B test is randomized by user, then we get this when computing the variance of the first definition (see Equation 19.5):

$$\mathrm{VAR}(CTR_1) = \left. \sum_{i=1}^{n} \sum_{j=1}^{K_i} \left(X_{ij} - CTR_1\right)^2 \right/ N^2 \tag{19.5}$$

This is incorrect, as it assumes that the X_{ij}s are independent (see Chapter 14 and Chapter 18). To compute an unbiased estimate of the variance, use the delta method or bootstrapping (Tang et al. 2010, Deng et al. 2011, Deng, Lu and Litz 2017).

We initially made this observation not because it was an obvious violation of the independence assumption, but because in our A/A tests, CTR_1 was statistically significant far more often than the expected 5%.

Example 2: Optimizely Encouraged Stopping When Results Were Statistically Significant

The book *A/B Testing: The Most Powerful Way to Turn Clicks into Customers* (Siroker and Koomen 2013) suggests an incorrect procedure for ending experiments: "Once the test reaches statistical significance, you'll have your answer," and "When the test has reached a statistically significant conclusion …" (Kohavi 2014). The statistics commonly used assume that a single test will be made at the end of the experiment and "peeking" violates that assumption, leading to many more false positives than expected using classical hypothesis testing.

Early versions of Optimizely encouraged peeking and thus early stopping, leading to many false successes. When some experimenters started to run A/A tests, they realized this, leading to articles such as "How Optimizely (Almost) Got Me Fired" (Borden 2014). To their credit, Optimizely worked with experts in the field, such as Ramesh Johari, Leo Pekelis, and David Walsh, and updated their evaluations, dubbing it "Optimizely's New Stats Engine" (Pekelis 2015, Pekelis, Walsh and Johari 2015). They address A/A testing in their glossary (Optimizely 2018a).

Example 3: Browser Redirects

Suppose you are building a new version of your website and would like to run an A/B test of the old versus the new. Users in variant B are redirected to your new website. Spoiler alert: B will lose with high probability. Like many ideas, it is simple and elegant, but flawed.

There are three problems with this approach (Kohavi and Longbotham 2010, section 2):

1. Performance differences. Users who are redirected suffer that extra redirect. This may seem fast in the lab, but users in other regions may see wait times of 1–2 seconds.
2. Bots. Robots handle redirects differently: some may not redirect; some may see this as a new unseen area and crawl deeply, creating a lot of non-human traffic that could impact your key metrics. Normally, it is not critical to remove all small-activity bots, as they are distributed uniformly in all variants, but a new site or updated site is likely to trigger different behavior.
3. Bookmarks and shared links cause contamination. Users that go deep into a website (e.g., to a product detail page) using a bookmark or from a shared link must still be redirected. Those redirects must be symmetric, so you must redirect users in Control to site A.

Our experience is that redirects usually fail A/A tests. Either build things so that there are no redirects (e.g., server-side returns one of two home pages) or execute a redirect for both Control and Treatment (which degrades the Control group).

Example 4: Unequal Percentages

Uneven splits (e.g., 10%/90%) may suffer from shared resources providing a clear benefit to the larger variant (Kohavi and Longbotham 2010, section 4). Specifically, least recently used (LRU) caches shared between Control and Treatment have more cache entries for the larger variant (note that experiment IDs must always be part of any caching system that could be impacted by the experiment, as the experiments may cache different values for the same hash key). See also Chapter 18.

In some cases, it is easier to run a 10%/10% experiment (not utilizing 80% of the data so useful in theory) to avoid LRU caching issues, but this must be done at runtime; you cannot run 10%/90% and throw away data. Your 50/50% A/A test may pass, but if you run experiments at 90%/10%, run these A/A tests in practice.

Another problem with unequal percentages is that the rate of convergence to a Normal Distribution is different. If you have a highly skewed distribution for a metric, the Central Limit Theorem states that the average will converge to Normal, but when the percentages are unequal, the rate will be different. In an A/B test, it's the delta of the metric for Control and Treatment that matters, and the delta may be more Normal if the two constituents have the same distribution (even if not Normal). See Chapter 17 for details.

Example 5: Hardware Differences

Facebook had a service running on a fleet of machines. They built a new V2 of the service and wanted to A/B test it. They ran an A/A test between the new and old fleet, and even though they thought the hardware was identical, it failed the A/A test. Small hardware differences can lead to unexpected differences (Bakshy and Frachtenberg 2015).

How to Run A/A Tests

Always run a series of A/A tests before utilizing an A/B testing system. Ideally, simulate a thousand A/A tests and plot the distribution of p-values. If the distribution is far from uniform, you have a problem. Do not trust your A/B testing system before resolving this issue.

When the metric of interest is continuous and you have a simple Null hypothesis, such as equal means in our A/A test example, then the distribution of p-values under the Null should be uniform (Dickhaus 2014, Blocker et al. 2006).

Figure 19.1 is a real histogram showing far from uniform distribution.

Figure 19.2 shows that after applying the delta method, distribution was much more uniform.

Running a thousand A/A tests may be expensive, but here's a little trick you can use: replay the last week. This of course, assumes that you stored the relevant raw data. This is an example of why we say to store your data for running future tests and applying newly developed metrics. There are limits to this approach, of course: you will not catch performance issues or shared resources such as the LRU cache mentioned above, but it is a highly valuable exercise that leads to identifying many issues.

Because you are not really making a change to your product and the two variants being tested are identical, you can just simulate the A/A test. For each iteration, pick a new randomization hash seed for user assignment and replay

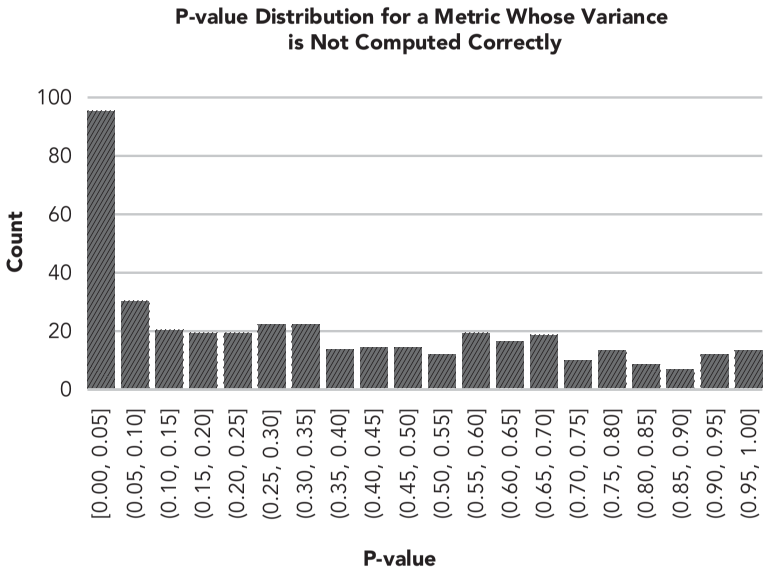

Figure 19.1 Non-uniform p-value distribution from A/A tests for a metric whose variance is not computed correctly because the analysis unit is not equal to the randomization unit

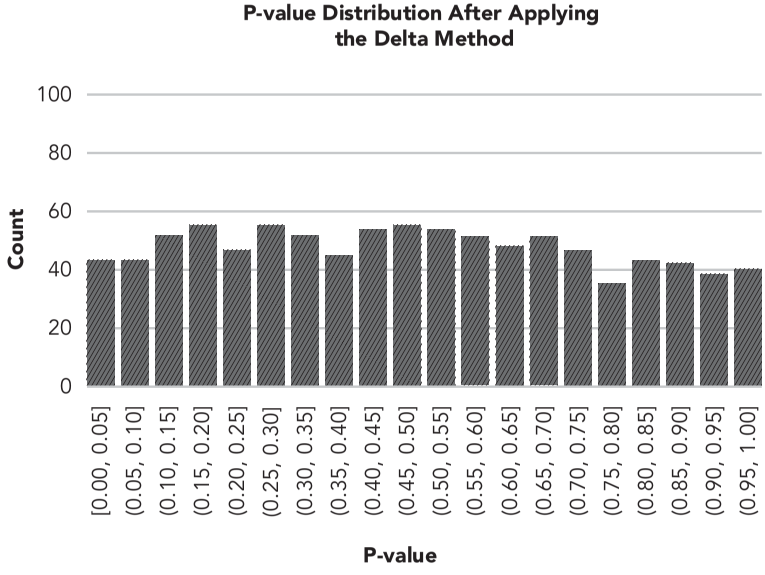

Figure 19.2 Distribution is close to uniform after applying the delta method to compute variance

the last week of data, splitting users into the two groups. Then generate the p-value for each metric of interest (usually tens to hundreds of metrics) and accumulate them into histograms, one for each metric.

Now run a *goodness-of-fit* test, such as Anderson-Darling or Kolmogorov-Smirnoff (Wikipedia contributors, Perverse Incentive 2019, Goodness of fit) to assess whether the distributions are close to uniform.

When the A/A Test Fails

There are several common p-value scenarios where the test fails the goodness-of-fit for a uniform distribution (Mitchell et al. 2018):

1. The distribution is skewed and clearly not close to uniform. A common problem is a problem with variance estimation of metrics (see Chapter 18). Check for the following:
   a. Is the independence assumption violated (as in the CTR example) because the randomization unit differs from the analysis unit? If so, deploy the delta method or bootstrapping (see Chapter 15).
   b. Does the metric have a highly skewed distribution? Normal approximation may fail for a small number of users. In some cases, the minimum sample size may need to be over 100,000 users (Kohavi et al. 2014). Capped metrics or setting minimum sample sizes may be necessary (see Chapter 17).
2. There is a large mass around p-value of 0.32, indicating a problem with outliers. For example, assume a single very large outlier o in the data.

   When computing the t-statistics (see Equation 19.6):

$$T = \frac{\Delta}{\sqrt{var}}(\Delta) \tag{19.6}$$

   the outlier will fall into one of the two variants and the delta of the means will be close to o/n (or its negation), as all the other numbers will be swamped by this outlier. The variance of the mean for that variant will also be close to o^2/n^2, so the T value will be close to 1 or close to -1, which maps to a p-value of about 0.32.

   If you see this, then the reason for the outlier needs to be investigated or the data should be capped. With such large outliers, the t-test will rarely lead to statistically significant results (see Chapter 18).
3. The distribution has a few point masses with large gaps. This happens when the data is single-valued (e.g., 0) with a few rare instances of non-zero

values. The delta of the means can only take a few discrete values in such scenarios, and hence the p-value can only take a few values. Here again, the t-test is not accurate, but this is not as serious as the prior scenario, because if a new Treatment causes the rare event to happen often, the Treatment effect will be large and statistically significant.

Even after an A/A test passes, we recommend regularly running A/A tests concurrently with your A/B tests to identify regressions in the system or a new metric that is failing because its distribution has changed or because outliers started showing up.