To appear in KDD 2009.
http://exp-platform.com

Seven Pitfalls to Avoid when Running Controlled Experiments on the Web

Thomas Crook tcrook@microsoft.com
Brian Frasca brianfra@microsoft.com
Ron Kohavi ronnyk@microsoft.com
Roger Longbotham rogerlon@microsoft.com
Microsoft, Experimentation Platform, One Microsoft Way, Redmond, WA 98052

ABSTRACT

Controlled experiments, also called randomized experiments and A/B tests, have had a profound influence on multiple fields, including medicine, agriculture, manufacturing, and advertising. While the theoretical aspects of offline controlled experiments have been well studied and documented, the practical aspects of running them in online settings, such as web sites and services, are still being developed. As the usage of controlled experiments grows in these online settings, it is becoming more important to understand the opportunities and pitfalls one might face when using them in practice. A survey of online controlled experiments and lessons learned were previously documented in *Controlled Experiments on the Web: Survey and Practical Guide* (Kohavi, et al., 2009). In this follow-on paper, we focus on pitfalls we have seen after running numerous experiments at Microsoft. The pitfalls include a wide range of topics, such as assuming that common statistical formulas used to calculate standard deviation and statistical power can be applied and ignoring robots in analysis (a problem unique to online settings). Online experiments allow for techniques like gradual ramp-up of treatments to avoid the possibility of exposing many customers to a bad (e.g., buggy) Treatment. With that ability, we discovered that it's easy to incorrectly identify the winning Treatment because of Simpson's paradox.

Categories and Subject Descriptors
G.3 **Probability and Statistics/Experimental Design**: controlled experiments, randomized experiments, A/B testing.
I.2.6 **Learning**: automation, causality.

General Terms
Management, Measurement, Design, Experimentation, Human Factors.

Keywords
Controlled experiments, A/B testing, e-commerce, Simpson's paradox, robot detection

KDD'09, June 28– July 1, 2009, Paris, France.

1. INTRODUCTION

> *Almost any questions can be answered, cheaply, quickly and finally, by a test campaign. And that's the way to answer them – not by arguments around a table. Go to the court of last resort – the buyers of your product*
> – Claude Hopkins, Scientific Advertising (1923)

Sir Ronald A. Fisher led the development of statistical experimental design while working at the Rothamsted Agricultural Experimental Station near London, England in the 1920s. His work had "profound influence on the use of statistics, particularly in the agricultural and related life sciences" (Montgomery, 2005). Over 70 years later, the esoteric field has grown mainstream: Forbes published an article on MultiVariable Testing titled "The New Mantra: MVT" (Koselka, 1996). The article begins with the following two sentences: "If you haven't yet applied multivariable testing to your business, get moving. Whether you run a factory, a mail-order house or a hospital, it will probably improve your performance." Montgomery (2005) wrote that "Applications of designed experiments have grown far beyond the agricultural origins. There is not a single area of science and engineering that has not successfully employed statistical designed experiments."

Toyota's famous production system with the principle of ongoing hypothesis testing of improvements often requires reconfiguration of the work area. The fascinating story in *Learning to Lead at Toyota* (Spears, 2004) describes how ideas are continuously tested even though reconfigurations of the work area are expensive: "75 [experiments]…required relocating material stores and moving the light curtains, along with their attendant wiring and computer coding. These changes were made with the help of technical specialists…." With software, testing new hypotheses is much easier; code can be modified and restored much more easily than physical artifacts. The web provides an unprecedented opportunity to evaluate ideas quickly using controlled experiments.

Controlled experiments typically generate large amounts of data, which can be analyzed using statistical and data mining techniques to gain deeper understanding of the factors influencing the outcome of interest, leading to new hypotheses and creating a virtuous cycle of improvements. Multiple lessons learned from deploying controlled experiments online and analyzing them were documented in the *Practical Guide to Controlled Experiments on the Web* (Kohavi, et al., 2007) and its longer version (Kohavi, et al., 2009). In this follow-on paper, we focus on pitfalls learned in the last three years, and especially in our last year, as we ramped up and ran numerous controlled experiments across multiple web sites at Microsoft.

The goal of the KDD industrial track is to "highlight challenges, lessons, and research issues arising from deploying KDD

technology." This paper focuses on important lessons, described as pitfalls, and related challenges we have identified. The pitfalls are all "real" in the sense that we experienced them and spent significant time working around them and documenting them so that you can avoid them.

The paper is organized as follows. Following a brief overview and definitions in Section 2, we review issues with choosing an OEC, the Overall Evaluation Criterion for experiments in Section 3. In Section 4 we highlight that computation of confidence intervals when reporting percent effects is not accurate and show how to compute these for combinations of metrics. In Section 5 we point out that for families of metrics the standard statistical formulas for computing variances fail to give the correct result because the independence assumption is violated. We recommend using Bootstrap, which is compute-intensive. In Section 6 we warn readers about occurrences of Simpson's paradox, a common problem when ramping-up experiments. Sections 7 warns about robots and proposes a novel way to evaluate whether robots that impact experimental results. Sections 8 warns about audits, instrumentation and controlling all differences. We conclude the paper with a short summary.

2. CONTROLLED EXPERIMENTS

In the simplest controlled experiment, often referred to as an A/B test, users are randomly exposed to one of two variants: Control (A), or Treatment (B), shown in Figure 1. This section mirrors the terminology and basic hypothesis testing overview as provided in *Controlled Experiments on the Web: Survey and Practical Guide* (Kohavi, et al., 2009) where additional motivating examples and multiple references to the literature are provided.

The terminology for controlled experiments varies widely in the literature. Below we define key terms used in this paper and note alternative terms that are commonly used.

Overall Evaluation Criterion (OEC) (Roy, 2001). A quantitative measure of the experiment's objective. In statistics this is often called the **Response** or **Dependent Variable** (Mason, et al., 1989; Box, et al., 2005); other synonyms include **Outcome**, **Evaluation metric**, **Performance metric**, or **Fitness Function**. Experiments may have multiple objectives and a scorecard approach might be taken, although selecting a single metric, possibly as a weighted combination of such objectives is highly desired and recommended (Roy, 2001 p. 50). A single metric forces tradeoffs to be made once for multiple experiments and aligns the organization behind a clear objective. A good OEC should not be short-term focused (e.g., clicks); to the contrary, it should include factors that predict long-term goals, such as predicted lifetime value and repeat visits.

Variant. A user experience being tested by being exposed to one of several variants, which include the Control and one or more Treatments.

Experimental Unit. The entity over which metrics are calculated before averaging over the entire experiment for each variant. Sometimes called an **item**. The units are assumed to be independent. On the web, the user is a common experimental unit. It is important that the user receive a consistent experience throughout the experiment, and this is commonly achieved through randomization based on user IDs stored in cookies. Throughout this paper, we will assume that randomization is by user.

Null Hypothesis. The hypothesis, often referred to as H_0, that the OECs for the variants are not different and that any observed differences during the experiment are due to random fluctuations.

Confidence level. The probability of failing to reject (i.e., retaining) the null hypothesis when it is true.

Power. The probability of correctly rejecting the null hypothesis, H_0, when it is false. Power measures our ability to detect a difference when it indeed exists.

A/A Test. Sometimes called a Null Test. Instead of an A/B test, you exercise the experimentation system, assigning users to one of two groups, but expose them to exactly the same experience. An A/A test can be used to (i) collect data and assess its variability for power calculations, and (ii) test the experimentation system (the Null hypothesis should be rejected about 5% of the time when a 95% confidence level is used).

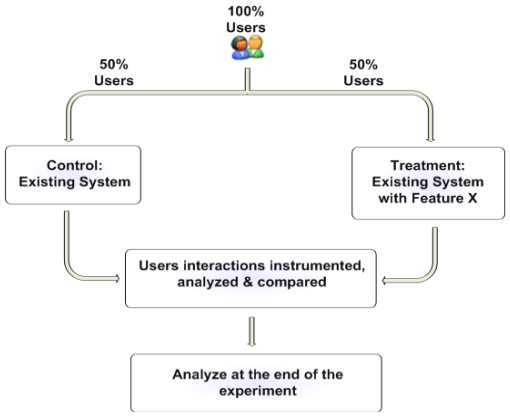

Figure 1: High-level flow for an A/B test

Standard Deviation (Std-Dev). A measure of variability, typically denoted by σ.

Standard Error (Std-Err). For a statistic, it is the standard deviation of the sampling distribution of the sample statistic (Mason, et al., 1989). For a mean of n independent observations, it is $\hat{\sigma}/\sqrt{n}$ where $\hat{\sigma}$ is the estimated standard deviation.

Statistical Significance. To evaluate whether the Overall Evaluation Criterion differs for user groups exposed to Treatment and Control variants, a statistical test can be done. If the test rejects the null hypothesis, which is that the OECs are not different, then we accept a Treatment as being statistically significantly different. We will not review the details of the statistical tests, as they are described very well in many statistical books (Mason, et al., 1989; Box, et al., 2005; Keppel, et al., 1992).

3. The Overall Evaluation Criterion

To run a controlled experiment, one needs to decide on the OEC, or the Overall Evaluation Criterion, the key metric that is going to be compared. For web sites, our recommendation is to tie that metric to a long-term goal, such as using customer lifetime value. For example, a retail site might want to optimize not just short-term revenues, but also for long-term indicators of loyalty and increasing wallet share: increase in repeat visits and purchases, signing up for e-mails, and purchasing from multiple departments.

Sometimes, when getting the true metric is hard, sites will use a surrogate metric as the following example shows.

3.1 Office Online Example

Microsoft's Office Online site (http://office.microsoft.com) had the following design (Control), shown in Figure 2.

The areas with red around them are "revenue generating links," which had a certain probability of leading to a sale of the Office suite. Tracking the actual purchase was hard, so the team settled on a surrogate OEC, which was "clicks on revenue generating links." They ran a controlled experiment, where the new treatment had a new design as shown in Figure 3.

Figure 2: The Control

Figure 3: The Treatment

The team thought that the new design would win on the OEC: clicks on revenue generating clicks, marked in red. However, the new design had 64% fewer clicks on those links. The experiment by itself was useful because the team thought their new design would perform better on OEC, and they now had to adjust their intuition, so it was a good learning experience.

However, there is a serious flaw with the OEC: clicks are a reasonable approximation to sales only if the conversion rate from click to purchase is the same in the old and new designs. The new version had the price shown on the page, and it sent more qualified users who are willing to spend $149.95, thus having a significantly higher conversion rate.

Another common problem with OECs that we have seen is a local focus. For example, measuring the click-through rate on a small area of the page, ignoring the impact on other areas of the page.

A final example is picking an OEC like "time on site." It may initially seem like a good OEC, but we have examples where a new feature was introduced that was so hard to use that it slowed users' effectiveness, growing their time on the site, but for the wrong reason.

The litmus test for an OEC should be: is it possible to do something simple (sometimes clearly dumb) and wrong that will improve the OEC but not meet the real business goal? If that is easy, how do you know that your complicated feature is not improving the OEC because it has a small "dumb" component? Here is why the above OECs do not pass the litmus test.

1. Office online click-throughs on revenue generating links. The OEC assumes that the conversion rate from a click to purchase is fixed. One can create a link labeled "Free download for 60 days" that will do wonders to the OEC, but the conversion will be much lower than a "Buy for $149.95" link. Is this ultimately going to generate more revenues? Unclear.
2. Click-through on a small area of the site (e.g., slot). It's easy to make an area stand out by making it a bold, with a different background, maybe even flashing. More people might click in the short term, but what about the whole-page click-through rate? What about long-term value?
3. Time on site. By making things harder to find or making navigation harder, users might stay longer on the site, but leave frustrated.

Pitfall 1: Picking an OEC for which it is easy to beat the control by doing something clearly "wrong" from a business perspective.

We want to caution against overcorrecting here. Sometimes picking a simple OEC is a good way to start experimenting, without worrying about the perfect OEC. When the MSN home page wanted to display an additional ad, we helped pick a simple OEC that looked at immediate revenue impact due to reduced click-throughs on the page, ignoring long-term effects such as slot blindness. The idea was negative even under this simple and conservative OEC, so it would have been worse under more sophisticated versions (Kohavi, et al., 2009).

3.2 Support Sites are Challenging

Many support sites provide an explicit feedback mechanism in the form of inline and/or pop-up surveys that allow users to rate their experience in terms of factors such as relevance and usability. These ratings are problematical. Such surveys are subject to non-response bias, wherein the sample of respondents is not representative of the total user population. It is well known that users with negative attitudes towards the company or product, or who have had an unsatisfactory experience, are more likely to respond to such surveys.(Hill, et al., 2007). Hill and his co-authors note that the minimum response rate needed to correct for non-response bias is 30%(p. 84). Given that the observed response rates for online support sites we have worked with is in the low

single digits, we assert that online surveys are not a suitable source of input for Overall Evaluation Criteria.

Prior research to infer user interest based on implicit actions used an instrumented browser, such as the Curious Browser (Claypool, et al., 2001). The researchers found that time spent on a page and the amount of scrolling on a page has a strong correlation with explicit interest, while individual scrolling methods and mouse-clicks are ineffective in predicting explicit interest. Later research also noted that how a user exited a result or ended a search session is important (Fox, et al., 2005).

Setting the OEC to time spent on page (dwell time) fails the litmus test noted in pitfall 1. For example, in a Microsoft health related site, a widget was redesigned to make health articles more accessible. Time spent on pages and total session time increased (satisfying the objective), but drilling down to the reasons, the new widget in the Treatment was used less often than the one in the Control. Users may have been more confused, thus taking longer to find what they need.

We also ran an experiment on Microsoft's support site, support.microsoft.com, where dwell time was the OEC. However, it was not clear at all whether the lower times were due to the user experience improving or users giving up.

Finding a good general OEC for support sites is challenging. We do want to mention that limited experiments are still possible. For example, a particularly successful support site experiment we ran involved the test of a rudimentary personalization feature. The support.microsoft.com site contained a top center "Instant Answers" module with links to common support issues selected by the site editors. We tested a new treatment that personalized these links by the browser and operating system versions of the user's HTTP header. The treatment performed over 50% better than the control on the OEC of Click-through rate, without decreasing the clickthrough rate for the whole page.

4. CONFIDENCE INTERVALS

It is useful to give a confidence interval for the difference in the means of the Treatment and Control in addition to the results of the hypothesis test. The confidence interval provides a range of plausible values for the size of the effect, whereas the hypothesis test only determines if there is a statistically significant difference in the means. The formula for the confidence interval for the difference in two means is fairly straightforward (Box, et al., 2005).

For many online metrics, the difference in the means is so small that percent change has much more intuitive meaning than the absolute difference. For example, for a recent experiment we ran, the Treatment effect for clickthrough rate was 0.00014. This translated to a 12.85% increase for the Treatment. The latter number was much more meaningful to decision makers. The percent difference is calculated as the delta between the means of the Treatment and Control divided by the mean for the Control times 100%.

Forming a confidence interval around the percent change is not a straightforward extension of the confidence interval for the absolute effect. The reason is we are now dividing by a random quantity. The initial derivation of this interval is due to Fieller (1940) and the formulas are shown in Kohavi et al (2009). We would not want to use a log or other transformation since business owners may reject results that are not expressed in the same units they are familiar with and percent increase has a natural business interpretation.

These formulas assume the covariance between the Treatment and Control mean is zero, which will be true in a controlled experiment when the randomization is carried out properly.

OECs may be a combination of metrics, or key performance indicators (KPIs). This combination could be either

1) A linear combination of metrics
2) A nonlinear combination of metrics that have the same basis[1] or
3) A nonlinear combination of metrics that do not have the same basis.

In the first case, the mean and variance of the OEC can be calculated from the means and variance of the metrics using the standard formulas and the confidence intervals are the usual symmetric confidence intervals using the normal distribution.

In the second case, one can calculate the OEC for each experimental unit then calculate the mean and variance of the OEC values across experimental units and then the confidence intervals.

The third case is more challenging, but we can use Rao's result: (1973 p. 387). If the OEC is a general function of k primary metrics, i.e. OEC = $g(X_1, X_2, \ldots, X_k)$, and if g(.) is a totally differentiable function of k variables, if $(X_1, X_2, \ldots, X_k)$ asymptotically follow a joint Normal distribution with means μ_1, $\mu_2, \ldots \mu_k$, and covariances σ_{ij}, i, j = 1,…k, then the OEC will asymptotically follow a Normal distribution with mean $g(\mu_1, \mu_2, \ldots \mu_k)$ and variance

$$\sigma^2(OEC) = \sum_{i=1}^{k}\sum_{j=1}^{k} \sigma_{ij} \frac{\partial g}{\partial X_i}\frac{\partial g}{\partial X_j} \qquad (1)$$

Provided $\sigma^2(OEC)$ is not zero and that $g(\mu_1, \mu_2, \ldots \mu_k)$ exists. The totally differentiable requirement leaves out many functions where truncation or discretization is utilized. We also have to assume the sample sizes are large enough for $g(X_1, X_2, \ldots, X_k)$ to have a Normal distribution.

Pitfall 2: Incorrectly computing confidence intervals for percent change and for OECs that involve a nonlinear combination of metrics

5. METRICS, STANDARD DEVIATIONS AND POWER

To compute statistical significance for different metrics of interest, we need to estimate the variance of the OEC. After running thousands of A/A tests, we discovered that variances for some metric families are inaccurately estimated using the standard statistical formulas. Specifically, the variance for click-through rate (CTR), defined as (sum of clicks)/(sum of page views) for the Treatment or Control for the time period of the experiment was significantly underestimated. In these cases, we have found the Bootstrap method (Efron, 1993) to be an excellent way to estimate the variance. The bootstrap is a resampling technique with

[1] Two metrics have the same basis if they are calculated over the same experimental unit. For example, page views per user-day and clickthroughs per user-day have the same basis, user-day.

replacement where the parameter of interest is calculated for each sample drawn and then we calculate the variance of these estimates. We currently take 1000 bootstrap samples. We recommend that you compare the formula variance for any metric with the Bootstrap estimate if you are not sure the formula for the variance is accurate. We now routinely use the bootstrap method to estimate variances whenever the experimental unit used in the calculation of the metric is different from the one used in the random assignment to the variants. For example, our standard method of random assignment is to assign users to Treatment or Control using a user ID stored in the cookie. Then we will use the bootstrap estimate for the variance of any metric that does not have user as the experimental unit (e.g. clicks per user-day or session). Care must be taken in the calculation of variance and power. The metrics may be considered in two categories: those where the experimental unit is the same as the randomization unit (referred to below as *per user* metrics) and those where it is not.

5.1 Per User Metrics

It is difficult to calculate the power for per user metrics because these metrics accumulate over time and most have increasing means and standard deviations, e.g., clicks per user and page views per user. A metric that is a ratio for each user (e.g. clickthrough rate) does not necessarily have an increasing mean and standard deviation, but the standard deviation of the mean does not decrease with the square root of the sample size as normally expected (Kohavi, et al., 2009).

The best way to calculate the power for these metrics is to run an A/A test prior to the A/B test to get the mean and standard deviation for different lengths of test. One can then interpolate or extrapolate to get the approximate power.

5.2 Non-Per User Metrics

Metrics, such as those with an experimental unit of user-day or session, have the complication that the experimental units are not independent, even if the averages and standard deviations are not increasing. Below are three examples of non-per user metrics.

- User-day metrics are those where user's behavior during 24 hour time periods are averaged, e.g. page views per user per day.
- Session metrics are defined during a period of user activity and are separated by periods of inactivity, customarily 30-minutes. We can then look at metrics, such as clicks or page views per session.
- Click-through rate defined for the duration of the experiment. Business users tend to focus on this metric, although we found that it to be very sensitive to robots.

There is usually some positive correlation between experimental units for these metrics and sites that have more loyal customers (higher return rate) have higher correlations. Ignoring the correlations leads to underestimation of the standard deviation. We have been using Bootstrapping to estimate the standard deviation for these metrics and getting good results, validated through A/A tests.

The only class of metrics where the power and standard deviation calculations are straightforward are conversion rates for users. For example, the percent of users who purchase an item or the percent of users who click on a link. These metrics follow the Bernoulli distribution when randomization is by user.

Pitfall 3: Using standard statistical formulas for computations of variance and power.

6. SIMPSON'S PARADOX

One of our recommendations for running online controlled experiments is to start an experiment with a small percentage of users assigned to the Treatment(s) and ramp that percentage (Kohavi, et al., 2007). One of the problems with ramp-up is that an analysis of the Control and Treatment that includes two or more periods with different percentages assigned to the treatment can be incorrect due to Simpson's paradox (Simpson, 1951; Malinas, et al., 2004; Wikipedia: Simpson's Paradox, 2008).

Table 1 shows a simple example, where a website has one million visitors per day, on each of two days: Friday and Saturday. On Friday, the experiment runs with 1% of traffic assigned to the Treatment, and then on Saturday that percentage is raised to 50%. Even though the treatment has a conversion rate that is better on Friday (2.30% vs. 2.02%) and a conversion rate that is better on Saturday (1.2% vs. 1.00%), if the data is simply combined over the two days, it would appear that the Treatment is performing worse (1.20% vs. 1.68%).

Table 1: Conversion Rate for two days. Each day has 1M customers, and the Treatment (T) is better than Control (C) on each day, yet worse overall

	Friday C/T split: 99%/1%	Saturday C/T split: 50%/50%	Total
C	$\frac{20{,}000}{990{,}000} = 2.02\%$	$\frac{5{,}000}{500{,}000} = 1.00\%$	$\frac{25{,}000}{1{,}490{,}000} = 1.68\%$
T	$\frac{230}{10{,}000} = 2.30\%$	$\frac{6{,}000}{500{,}000} = 1.20\%$	$\frac{6{,}230}{510{,}000} = 1.20\%$

There is nothing wrong with the above math. It is mathematically possible that $\frac{a}{b} < \frac{A}{B}$ and that $\frac{c}{d} < \frac{C}{D}$ while $\frac{a+c}{b+d} > \frac{A+C}{B+D}$. The reason this seems unintuitive is that we are dealing with weighted averages, and the impact of Saturday, which was a day with an overall worse conversion rate, impacted the Treatment more.

Here are other examples from controlled experiments where Simpson's paradox may arise:

1. Users are sampled. Because there is concern about getting a representative sample from all browser types, the sampling is not uniform, and users with some browsers (e.g., Opera, Netscape) are sampled at higher rates. It is possible that the overall results will show that the Treatment is better, but once the users are segmented into the browser types, the Treatment is worse for all browser types.
2. An experiment runs on a web site that is implemented in multiple countries, say US and Canada. The proportions assigned to the Control and Treatment vary by country (e.g., the US runs at 1% for the Treatment, while the Canadians do power calculations and determine they need 50% for the Treatment). If the results are combined, the Treatment may seem superior, even though if the results were broken down by country, the Treatment will be inferior. This example directly mirrors the ramp-up example shown previously.

3. An experiment is run at 50/50% for Control/Treatment, but an advocate of the most valuable customers (say top 1% in spending) is concerned and convinces the business that this customer segment should be kept stable and only 1% will participate in the experiment. It is possible that the experiment will be positive overall, yet it will be worse for both the most valuable customers and for the non-valuable customers.
4. An upgrade of the website is done for customers in data center DC1 and customer satisfaction improves. A 2nd upgrade is done for customers in data center DC2, and customer satisfaction there also improves. It is possible that the auditors looking at the combined data from the upgrade will see that overall customer satisfaction decreased.

While occurrences of Simpson's paradox are unintuitive, they are not uncommon, and we have seen them happen multiple times in real life. Possible solutions include: (i) paired t-tests where each pair (Control, Treatment) is chosen from a period where the proportions were stable; and (ii) using weighted combinations. The simplest solution, which we use, is to throw away the data from the ramp-up period, which is usually short relative to the experiment.

Pitfall 4: Combining metrics over periods where the proportions assigned to Control and Treatment vary, or over subpopulations sampled at different rates

7. ROBOTS IMPACT RESULTS

Web sites are accessed not only by human users but also by robots such as search engine crawlers, email harvesters and botnets. The traffic generated by robots is not representative of the human population (e.g., excessive clicks and page views in patterns that differ from human patterns) and can cause misleading results.

Robots should be excluded from experiments focused on improving the human experience whereas humans should be excluded from experiments focused on the robot experience (e.g., for Search Engine Optimization). In practice, however, identifying robots is difficult (Tan, et al., 2002; Kohavi, et al., 2004; Bomhardt, et al., 2005; Bacher, et al., 2005; Wikipedia: Internet bot, 2008; Wikipedia: Botnet, 2008).

For example, in an experiment on the MSN portal, where a small change was done to only one module, we found that the click-through rate on several areas of the page were statistically significantly different. Since the change was small and localized to one area of the page, we were surprised to see significant differences in unrelated areas. Upon deeper investigation, we found that the differences were caused by robots that accept cookies and execute JavaScript. Executing code in JavaScript is one of the most common characteristics that separate humans from robots, and some web analytic vendors even claim that page tagging using JavaScript is so robust that no additional robot detection should be done. Yet in this case these robots were executing JavaScript "onclick" events, which fire on the MSN portal when users click a link on a web page, at extremely high rates of about 100 per minute for durations of 2.5 hours.

Robots implemented by automating browsers such as Internet Explorer or Firefox support all of the functionality of those browsers including cookies and JavaScript. Furthermore, when such a robot runs from a machine also used by a human, both the robot and human will typically share the same cookies. If the user identity is stored in a cookie (very common), then the user appears to be schizophrenic, acting like a human at certain times and like a robot at others.

For experimentation, we are primarily concerned with removing robots that cause a bias. If the traffic from a robot is distributed across the variants of an experiment in an unbiased way, then the presence of the robot adds noise to the data and reduces the power of the experiment but does not invalidate the results. Robots that are seen as multiple unique users due to resetting their cookies or running from multiple machines do not introduce bias. Robots that act like a single user and consistently generate traffic for a single variant, however, can create a significant bias. For example, if a robot consistently assigned to variant A generates an excessive number of clicks, it may cause A to have a statistically significantly higher click-through rate than B even if B is preferred by human users.

Although it is difficult to identify all robots in general and there is no clear way to evaluate how good a robot detection algorithm performs on real data, controlled experiments can provide such a unique evaluation function, at least for the robots most critical for analysis: those that can skew the results by accepting cookies and behave like extreme users. The novel evaluation scheme we propose is to use A/A tests, where users are split into Control and Treatment, but there is no systematic difference between the two versions they are exposed to. The Null hypothesis in an A/A test should be rejected about 5% of the time when a 95% confidence level is used. If this does not hold true, then there is a bias introduced by extreme behavior of users, which are most likely robots being assigned to a particular variant. Multiple A/A tests must be run in order to have confidence whether biased robots exist in the data. However, an interesting observation is that these don't have to be live A/A tests. It is sufficient to run tests post-hoc ("offline") by re-randomizing users and assigning them to Control/Treatment and evaluating the hypothesis that they are the same. We are now developing heuristics to detect robots, but it is a significant challenge.

Pitfall 5: Neglecting to filter robots

8. AUDITING THE ANALYSES

It is critical to validate the collection of user behavior data, the assignment of users to experiment variants, and the calculation of metrics. While running experiments on numerous websites, we have encountered problems in every stage of the analysis pipeline that have led to incorrect results. This section describes the validation steps we developed to detect data quality and analysis problems.

8.1 Logging Test

After instrumenting the application (e.g., website) to send user behavior data to the experimentation system, a logging test should be run to validate that the data is being properly recorded. There are several ways to do this validation and ideally all should be used:

8.1.1 Compare with system of record

Most websites already send user behavior data to a reporting system or other system of record. Data loss or corruption can often be detected by comparing the data received by the experimentation system with the system of record. If possible, it is best to do a detailed record-by-record comparison between the two systems. This allows flagging specific records captured by

only one of the systems which can lead to insights if there is a collection problem. Otherwise, doing comparisons of aggregate values (e.g., received X page views in a particular hour) can still provide a high level sanity check. If the experimentation system uses data directly from the system of record and there is no alternative data collection system, then the other techniques discussed below are still applicable.

It is interesting to point out that in a few cases our audits found serious problems with the Microsoft "system of record." Some of these systems have complicated ETL (Extract-Transform-Load) processes and have evolved over the years. Our relatively simple logging infrastructure has fewer opportunities to lose data.

8.1.2 Compare with generated data

For many applications including websites, end user behavior can be simulated through software. Comparing the simulated user actions with the collected user behavior data is a powerful validation technique. Since you know exactly what data should be received, it is easy to identify missing, extra or corrupted data. This is in contrast to comparing with a system of record which itself may have unreliable data.

One challenge with this technique is mimicking the diversity of end users. In the case of a website, end users may be located around the world, have different internet connections speeds and use different web browsers which may all impact the reliability of data collection. Certain applications may also maintain state for end users (e.g., shopping cart, order history, wish list, contacts, etc.) which can be difficult to mimic.

Nevertheless, this technique has proven quite useful in practice even with very simple simulated data. We have identified several data collection bugs since we started using this technique after a couple of experiments failed due to incorrectly logged data.

8.1.3 Look for unexpected patterns

Typically, there are certain patterns that we expect to find in the data. For example, most websites have more traffic during the day and on weekdays than they do during the night and on weekends. When the patterns observed in the data do not match the expected patterns for the application then it casts doubt on the validity of the data and raises a flag that a deeper investigation may be necessary. Since such patterns are highly application specific, it is important to work with the business owners to understand the expected behavior.

Here are some of the patterns we've found useful to look at:

1. *Volume of data over time.* One of the most useful patterns to look at is the count of observations (e.g., page views) received over time. An outage in either the data collection system or the application itself will appear as a drop in data volume. Also, as noted above, comparing the observed data with the pattern expected by the business can identify potential data collection problems.
2. *Number of new and repeat users over time.* Seeing fewer repeat users than expected may indicate a bug where the user identifier is regenerated causing repeat users to appear as new users.
3. *Ratios of related observations over time.* Observations such as page views and clicks in a website are typically proportional to each other. An abnormal change the ratio of such observations is a likely indication of either a data collection problem or a robot that only generates data for one of the two observations.
4. *Dimensional analysis.* All of the above patterns can be broken down by dimensional attributes for additional insight. For example, breaking down the patterns by the web browser used (e.g., IE6, IE7, Firefox 2, Firefox 3, etc.) may highlight problems that appear in some browsers but not in others.

8.2 A/A Test

Distributing end users across the variants of an experiment both consistently and without bias are critical requirements for running valid controlled experiments. Each user must consistently receive the same variant over the course of the experiment in order to minimize inconsistent experiences and primacy effects. Each variant must be given to an unbiased set of users in order to make the comparison between variants valid. If there is a bias where users of Internet Explorer 7 are more likely to receive variant A than B, for example, the comparison between those variants is impacted not only by the difference between the variants but also the difference between browser versions.

While a logging test helps to validate that data is being properly recorded, it will not detect problems due to end users being incorrectly assigned to variants. An A/A test, however, can be used for that purpose. The application code used to assign users to variants and execute the appropriate variant must be the same as it would if the variants were different. Running an experiment in this configuration allows us to perform a number of sanity checks to validate that the experimentation apparatus itself is functioning properly.

Verifying that each end user consistently received a single variant can be done by injecting variant specific information into the user behavior data. For example, if users in variant A should receive page X but users in variant B should receive page Y then recording the URL (X or Y) in a page view observation allows checking whether any user received the wrong page.

A critical sanity check is to verify that users are divided between the variants in the appropriate ratio. For example, if each variant is configured to be assigned to 50% of users (recommended to maximize the statistical power in A/B tests) then check that the actual percent of users assigned to each variant is not statistically significantly different from 50%. This check can also be done on sub-populations in order to detect an assignment bias. The browser bias described above could be detected by performing this test on browser versions. In addition to looking at the number of distinct users assigned to each variant, we have also found it useful to look at the amount of data generated by those users. This will detect data collection bugs that impact the variants differently (e.g., data collection only being enabled for the Treatments and not for the Control).

Finally, by making the variants identical we know that there should be very little difference in the metrics measured for each variant during the experiment. Specifically, 95% of metrics should have no statistically significantly difference between the variants when a 95% confidence interval is used to determine statistical significance. If too many (or too few) metrics are statistically significantly different between the variants of an A/A test then the results are suspect and further investigation is warranted.

8.3 Offline A/A Test

As mentioned in Section 7, we initially developed the idea of an "offline" A/A test as a mechanism to evaluate robot detection

algorithms. However, we have found this technique to be useful in uncovering other metric calculation problems as well.

When we first attempted to validate our results using offline A/A tests we found that 30% (as opposed to the expected 5%) of metrics where statistically significant. Standard formulas underestimated the standard deviation for many of the metrics we calculate as discussed in Section 4.

It is important to note that offline A/A tests identify very different problems than normal A/A tests. An offline A/A test finds problems with the calculation of metrics whereas a normal A/A test detects variant assignment bugs and biased data collection.

8.4 Rich Instrumentation

Rich server and client side instrumentation is required for comprehensive analysis of online experiments.

8.4.1 Collect data at referrer and destination points

To get a full picture of users' behavior, it is important to collect data at all referrer and destination points in online applications. For example, if you only record the behavior of users once they click through to a secondary page, you will be missing information about users who never clicked through in the first place. The following example illustrates this concept:

A team we worked with wanted to test a new version of a Flash-based navigation component on its homepage. Clickable areas within the existing and experimental versions of the Flash component served to direct users to content pages deeper within the site. The team elected not to instrument the home page or the Flash component but to rely solely on page views on destination pages (with referrers other than the home page filtered out) to measure their OEC of click-throughs from the Flash control to destination pages.

Because we were limited to destination page view data with referrer information, we only knew the performance of the old and new variants conditioned on the event that the user clicked on the Flash control at all. The problem here is that some users may dislike one of the versions of the Flash control so much that they never click at all. Lacking a page view observation on the home page, we could not get a complete record of user behavior.

Rich server and client side instrumentation is required for comprehensive analysis of online experiments.

8.4.2 Over-instrumenting is better than under-instrumenting

Collecting more observations than required for computing your metrics and OEC can help identify implementation bugs that can bias experiment results. For example, by collecting server side page request observations we were able to identify an issue in which FireFox was requesting each page twice due to an IMG tag with an empty SRC attribute on the page.

In contrast to our advice to collect rich observational data, we do not advocate the reporting of long lists of metrics. Providing too many results allows people to cherry pick the ones that support their favored outcome while ignoring the results that do not support it. Remember that when using a 95% confidence level, one out of twenty results will show significance due to random chance.

Pitfall 6: Failing to validate each step of the analysis pipeline and the OEC components

9. Control is Crucial

It is all too easy to allow the variants you are comparing to differ in some way besides the feature you want to test. For example, if you are using client side redirect through JavaScript to show the content of the Treatment and not the Control, you may have an extra delay on in the Treatment. This will likely cause a decrease in click-through rate and other metrics. Of course any experiment where there is a redirect or other delay in one variant and not the others will be biased. Our recommendation is to choose an approach to experimentation that does not require a redirect, but if you need to use that method you should include the redirect in all variants you are testing.

Another common mistake experimenters make is when a site conducting an experiment has frequent updates (e.g. news or other content) and these updates are not made equally to all variants. One experiment we ran involved a test of headline placement on the MSN homepage. The headlines being shown were intended to be same in Treatment and Control, but in a different order. However, one of the headlines was different for a seven hour period. A graph of the hourly clickthrough rate (CTR) for two days of this experiment is given in Figure X with the red box highlighting the seven hour period.

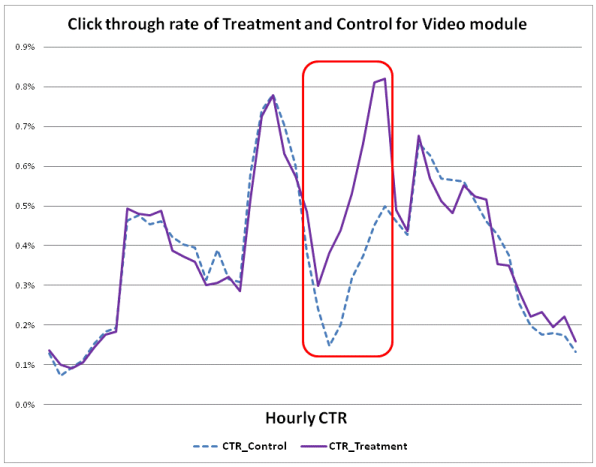

Figure 4: Click-through Rate for Video module

The Treatment was significantly better than the Control before taking this seven hour period out of the analysis but there was no difference once it was removed.

Pitfall 7: Forgetting to control for all differences, and assuming that humans can keep the variants in sync

10. SUMMARY

Good judgment comes from experience, and and a lot of that comes from bad judgment.
-- Will Rogers

Controlled experiments have had profound influence on multiple fields, including medicine, agriculture, manufacturing, and advertising. Their widespread adoption in software development of web sites and services is just beginning. We reviewed pitfalls we have seen in running experiments at Microsoft over the last three years since the Experimentation Platform team was formed.

We started off with pitfall 1 related to the most important decision when running an experiment: the Overall Evaluation Criterion.

Too many OECs that we have seen fail our suggested litmus test. While the statistics can be computed correctly, one needs to ask whether the right metric is being optimized, especially if there are plans to run a series of experiments to optimize the OEC. Pitfall 2 warns about computing confidence intervals for percent effects and how to combine metrics. Pitfall 3 warns about using standard statistical formulas for computing variances; we switched to Bootstrap estimates when we realized the problem. Pitfall 4 warns that without more complicated analyses, it is too easy to reach incorrect conclusions because of Simpson's paradox; other well-intentioned sampling techniques can likewise lead to incorrect conclusions. Pitfall 5 warns about robots, which have dramatic impact on results sometimes. Pitfalls 6 and 7 highlight the importance of audits and controlling for all differences.

Knowing these pitfalls can increase the trust in controlled experiments and help organizations build better software by making data-driven decisions.

ACKNOWLEDGMENTS

We would like to thank members of the Experimentation Platform team at Microsoft, especially Randy Henne, Andrew Hesky, David Messner, and Justin Wang. We thank Jennifer Abdo for her feedback. Special thanks to David Treadwell and Ray Ozzie; without their support the experimentation platform would not have existed.

REFERENCES

Bacher, Paul, et al. 2005. Know your Enemy: Tracking Botnets. *The Honeynet Project.* [Online] March 13, 2005. http://www.honeynet.org/papers/bots/.

Bomhardt, Christian, Gaul, Wolfgang and Schmidt-Thieme, Lars. 2005. Web Robot Detection - Preprocessing Web Logfiles for Robot Detection. [book auth.] Maurizio Vichi, et al. *New Developments in Classification and Data Analysis.* s.l. : Springer, 2005.

Box, George E.P., Hunter, J Stuart and Hunter, William G. 2005. *Statistics for Experimenters: Design, Innovation, and Discovery.* 2nd. s.l. : John Wiley & Sons, Inc, 2005. 0471718130.

Claypool, Mark, et al. 2001. Inferring user interest. *IEEE Internet Computing.* 2001, Vol. 5, pp. 32-39. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.28.5967.

Efron, Bradley and Robert J. Tibshirani. 1993. *An Introduction to the Bootstrap.* New York : Chapman & Hall, 1993. 0-412-04231-2.

Fieller, E C. 1940. The Biological Standardization of Insulin. *Supplement to the Journal of the Royal Statistical Society.* 1940, Vol. 7, 1, pp. 1-64.

Fox, Steve, et al. 2005. Evaluating implicit measures to improve web search. *ACM Transactions on Information Systems (TOIS).* 2005, Vol. 23, 2, pp. 147-168. http://portal.acm.org/citation.cfm?id=1059981.1059982.

Hill, Nigel, Roche, Greg and Allen, Rachel. 2007. *Customer Satisfaction: The Customer Experience Through the Customer 's Eyes.* s.l. : Cogent Publishing, 2007.

Hopkins, Claude. 1923. *Scientific Advertising.* New York City : Crown Publishers Inc., 1923.

Keppel, Geoffrey, Saufley, William H and Tokunaga, Howard. 1992. *Introduction to Design and Analysis.* 2nd. s.l. : W.H. Freeman and Company, 1992.

Kohavi, Ron, et al. 2009. Controlled experiments on the web: survey and practical guide. *Data Mining and Knowledge Discovery.* February 2009, Vol. 18, 1, pp. 140-181. http://exp-platform.com/hippo_long.aspx.

Kohavi, Ron, et al. 2004. Lessons and Challenges from Mining Retail E-Commerce Data. 2004, Vol. 57, 1-2, pp. 83-113. http://ai.stanford.edu/~ronnyk/lessonsInDM.pdf.

Kohavi, Ron, Henne, Randal M and Sommerfield, Dan. 2007. Practical Guide to Controlled Experiments on the Web: Listen to Your Customers not to the HiPPO. *The Thirteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2007).* August 2007, pp. 959-967. http://exp-platform.com/hippo.aspx.

Koselka, Rita. 1996. The New Mantra: MVT. *Forbes.* March 11, 1996, pp. 114-118.

Malinas, Gary and Bigelow, John. 2004. Simpson's Paradox. *Stanford Encyclopedia of Philosophy.* [Online] 2004. [Cited: February 28, 2008.] http://plato.stanford.edu/entries/paradox-simpson/.

Mason, Robert L, Gunst, Richard F and Hess, James L. 1989. *Statistical Design and Analysis of Experiments With Applications to Engineering and Science.* s.l. : John Wiley & Sons, 1989. 047185364X .

Montgomery, Douglas C. 2005. *Design and Analysis of Experiments.* 6th edition. s.l. : John Wiley & Sons, Inc, 2005. 0-471-66159-7.

Rao, C. Radhakrishna. 1973. *Linear Statistical Inference and Its Applications.* 2nd. s.l. : John Wiley & Sons, Inc., 1973.

Roy, Ranjit K. 2001. *Design of Experiments using the Taguchi Approach : 16 Steps to Product and Process Improvement.* s.l. : John Wiley & Sons, Inc, 2001. 0-471-36101-1.

Simpson, Edward H. 1951. The Interpretation of Interaction in Contingency Tables. *Journal of the Royal Statistical Society, Ser. B.* 1951, Vol. 13, pp. 238–241.

Spears, Steven J. 2004. Learning to Lead at Toyota. *Harvard Business Review.* May 2004, pp. 78-86.

Tan, Pang-Ning and Kumar, Vipin. 2002. Discovery of Web Robot Sessions based on their Navigational Patterns. *Data Mining and Knowledge.* 2002, Vol. 6, 1, pp. 9-35. http://citeseer.ist.psu.edu/article/tan02discovery.html.

Wikipedia: Botnet. 2008. Botnet. *Wikipedia.* [Online] 2008. [Cited: February 28, 2008.] http://en.wikipedia.org/wiki/Botnet.

Wikipedia: Internet bot. 2008. Internet Bot. *Wikipedia.* [Online] 2008. [Cited: February 28, 2008.] http://en.wikipedia.org/wiki/Internet_bot.

Wikipedia: Simpson's Paradox. 2008. Simpson's paradox. *Wikipedia.* [Online] 2008. [Cited: February 28, 2008.] http://en.wikipedia.org/wiki/Simpson%27s_paradox.