Quality for Online Experimentation

Ji Chen, Roger Longbotham, Justin Wang
Alex Deng, Dave Debarr
Bing Data Mining, Microsoft Corporation, One Microsoft Way, Redmond, WA 98052

Abstract

Controlled experimentation has been proven to be an effective way to test ideas and evaluate changes in websites and web services. While the basic theoretical foundation for controlled experiments has been well established, in reality more often than not, we are faced with data quality issues that could easily bias the results of the experiments and confound the decision making process. This talk will discuss several data quality concerns specific to online experimentation and provide best practices to address them. We will cover challenges such as web robot detection, traffic anomaly alerts, user session identification, page instrumentation issues and web data cleansing. Most of the techniques discussed are also applicable to web analytics in general. Some research questions will be presented at the end.

Key Words: online experimentation, data quality, web analytics, anomaly detection

1. Introduction

Online experimentation is a popular practice in website development (Kohavi, R. 2007, Thomke, S. 2003). In its simplest form, live users are randomly assigned to the Control (usually the current version of the website) and the Treatment (usually the new version of the website). Metrics are then calculated and statistical tests conducted to test if there is a statistically significant difference between the Control and the Treatment. Fundamentally speaking, the theory for online experimentation, or A/B testing as it is sometimes called, has been well established, and this method has the advantage over other methods in terms of testing causal relationship. However, in practice, good analysis can only be based on good quality data. Data quality is especially important for online experiments due to the bug prone, fast evolving web environment.

In this paper, we will discuss how to improve data quality from five aspects: web robot detection, outlier handling, traffic anomaly alerts, user/session identification, and instrumentation.

2. Web Robot Detection

How to identify traffic generated by robots is one of the most challenging topics in online data analysis. There are several types of robots and the levels of damage if including them into the data analysis are very different.

One type of robots which will cause very serious consequence if not being filtered out is the ones that generates a very large amount of observations e.g., clicks or page views or both. We have found robots can contribute as many as half of the traffic for some

websites. The analysis conclusions would be totally skewed by those robots. Running an AA experiment , i.e. there is no difference between Treatment and Control (Peterson, E. 2004), is not only a good practice to test if the experimentation platform is well instrumented; it also provides a good opportunity to understand the robot activities on the website. Figure 1 gives an example of an AA experiment. We would expect them both to have the same daily pattern; however, the graph shows that there are many hours where Treatment and Control deviate by a large amount. Further investigation reveals that all of the large deviations were caused by robots. Although each hour has thousands of users, each of these deviations was caused by a single user (robot) with a very large number of clicks. This type of robots that skew data by many actions can be easily removed by applying behavioral heuristics (Tan, P-N & Kumar V. 2002) like excluding users exceeding a certain number of page views/clicks in a certain period of time (e.g., more than 100 clicks in an hour) due to their prominent outlier behavior. Another type of robots can be identifies by behavioral heuristics are those visit the website with a regular pattern (e.g. click every 10 minutes). Removing those robots can potentially significantly increase the power of the statistics tests.

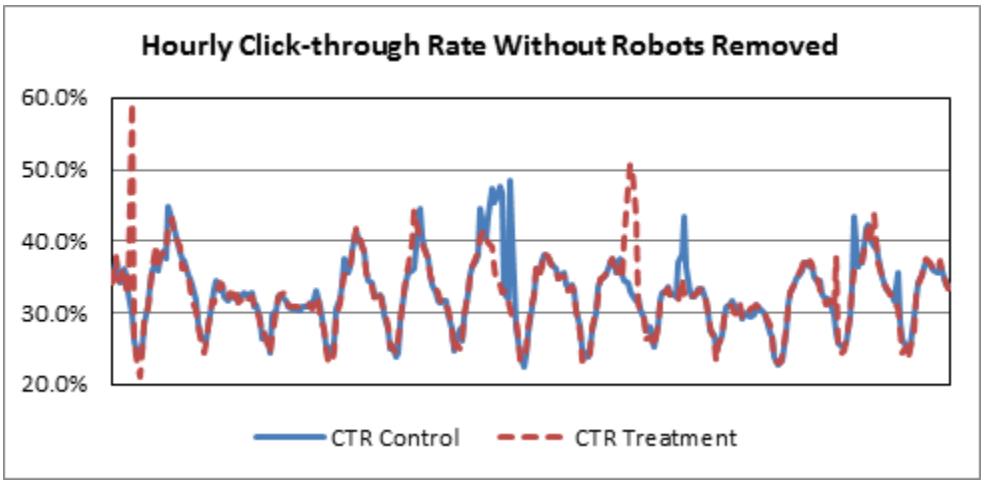

Figure 1: Graph for hourly click-through rate (robots not removed)

An extreme example we encountered in practice was a robot written by the website maintenance to monitor if the Buy button on the page works properly. This robot executed once a day and the code was embedded within a series of maintenance programs. Since the cookie was cleared every time and the time the code was executed was different every day depending on the workload and status of the server, it cannot be detected with behaviour heuristic algorithms from an AA experiment. We finally detect this robot from a very suspicious outcome. Since the way the robot is written makes it only clicks on Buy button on the current page but not on the tested page, without excluding this robot, we would have incorrectly concluded the tested page has few clicks on the Buy button.

Another reason why detecting robots is very challenging is that, in internet data analysis, users are mainly identified by their cookies and users can clean cookies any time they want. Some robots clear their cookies and come as a new user (first time visitor) every time. Cookie along with behavior heuristics won't be able to identify this type of robots. Internet data provide very limited information to identifying users, but besides cookie, IP

address and user agent are the other two pieces of information that could be helpful to characterize users. It is worth noticing that some prior processes on IP address and user agent are required before using them to classify users. For example, some internet providers apply dynamic IP, therefore, a user can have a different IP address every time he/she visits. User agent can be tricky also. Some user agents include the visitor date/time as part of the string. Simply classifying users with IP address and user agent usually won't produce the desired results. We will get to this user identification problem in more detail later in this paper.

3. Distributional Issues

Online metrics may be categorized as one of three types: Bernoulli (often called Conversion Rate, e.g. did a user purchase or not), count (number of page views per user) or measurements (e.g. number of milliseconds to load a page). Most online metrics are count metrics. Distributions for most count metrics and many measurement metrics, e.g. dollar value of an order, are quite skewed. Even after log transformation, the histograms of the log of counts and the log of Order Total per user (Figure 2) are quite skewed. Besides count metrics, measurement metrics such as session time can also be quite skewed (Nicholas et al. 2001).

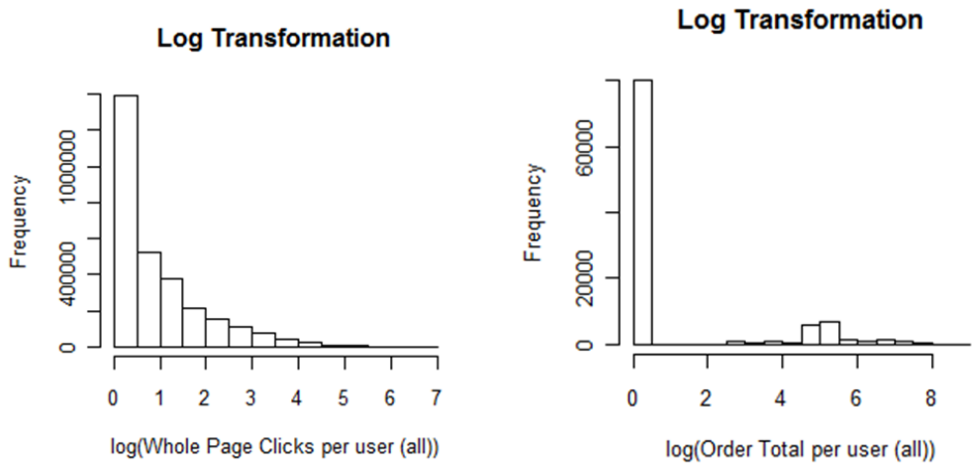

Figure 2: Histogram of two metrics, whole page clicks per user (all) and order total per user (all) after log transformation

For many sites the largest proportion of users has minimal engagement with the site having zero or one clicks per user or zero purchases. Most of the other users generally have a moderate amount of engagement. However, there are usually some who have a large amount of engagement. Some of these may be robots (see other section) and others will be real users who are different from the norm. For most experiments/metrics we don't want to measure the activity of robots, so, as much as possible we want to remove the robots so that our metrics are reflecting real users. Having done that we are still left with some who we believe to be real users but who have a large amount of activity. The skewness of the data often reduces the statistical power to detect small changes, and thus affect our analysis.

We conducted simulation studies with actual online data for four methods to compare Treatment to Control to compare the power of the methods against different alternatives.

1. First we considered the standard two-sample t-test. Since the sample sizes we deal with for online experiments are quite large the means of the two groups would have a normal distribution (except perhaps in rare cases.) However, the power of the comparison may be degraded by much skewed distributions.
2. The second method we employed was a truncated t-test, where values larger than some percentile, say P, were given the value of the Pth percentile.
3. A non-parametric comparison, the Wilcoxon Rank-Sum test.
4. A two-sample t-test using the log of the data.

We had hoped to find that one of these methods would have the best power or close to the best power for most or all metrics. Unfortunately, that was not the case. The t-test and truncated t-test tended to perform the best for certain metrics and the Wilcoxon and log transform tended to perform the best for the others. The t-tests performed the best for metrics where a large percentage of the values were the same. For example, Revenue per user or Whole page clicks per user. Both of these metrics had at least 70% of users have zero revenue or zero clicks. For metrics that were continuous or count metrics with less than half having a single value the Wilcoxon and log transform (t-test) performed best. In addition, the last three (truncated t-test, Wilcoxon, log transform) did well in the presence of robots or large outliers.

4. Traffic Anomaly Alerts

Traffic anomaly alerts are always highly desired and once properly implemented very useful (Teng et al. 1990, Hajji, H. 2005, Lakhina et al. 2004). It helps with real-time detection of outage, buggy deployment, robots etc., and makes sure that the low quality data is identified before analysis.

While conducting online experimentation, besides recording data for control and treatment(s), it is also helpful to maintain an ongoing monitoring system in case of need for reference point. Some bugs could impact the experiment so lower than normal traffic could be logged for the treatment, or even control. These bugs could be biasing the experimental results and even when they are not biasing between control and treatment, it is possible that it could lead to a sample size (power) issue. In such cases, by comparing the logging from the experiment with a long term monitoring system, problems in implementation could be easily identified.

For both ongoing monitoring and experiment logging systems, automatic alerts are important and most often these alerts are set up using approaches that combine statistics and heuristics. The simplest alerts involve just a heuristic threshold for traffic, such as number of unique users per day. Statistical approach such as control charts (Chambers, D. 1992) can be used to identify outliers in traffic (e.g., unusually high/low traffic time point) by calculating mean and standard error etc. Figure 3 shows the control chart type of graph for a relatively stable metric, click-through rate. However, simple control charts work best on static time series data and that is usually not true for web data. The existence of a trend is very common in web data, and so various methods can be used to accommodate that, the simplest including using a moving window when calculating the control chart. It is also important to incorporate cyclicality since web data usually displays strong weekly pattern and some seasonality.

Most common alerts are fired based on a single metric, for example, pageviews per day, pageviews per users, etc. However, it is also possible that some relationship between metrics exists and we could generate alerts based on the cross-metric relationship. For example, for a search engine site, in most cases, there is a negative correlation between the number of users visiting the site and the average queries issued per user that day. The reason for that is that when number of users visiting increases, more often than not, we would see higher percentage of less loyal users who only issue a small number of queries, which lowers the average. If both of these numbers drop significantly on the same day, we should check the recent deployment to see if it is due to the treatment being tested or some bugs. More complicated methods could be used to generate such cross-metric alerts, such as principal component analysis (PCA), and other multivariate analysis (Ringberg et al. 2007, Lakhina, A. 2005).

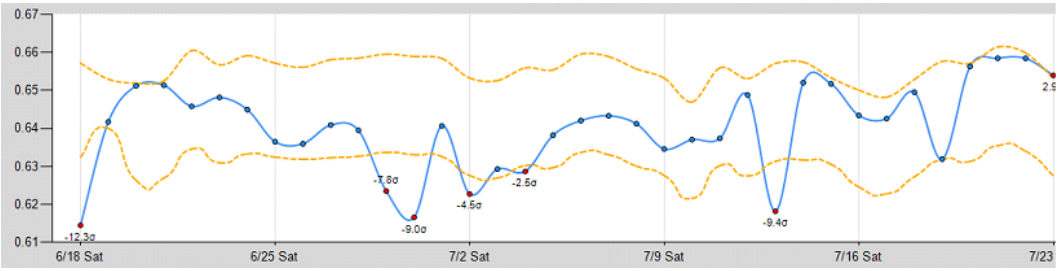

Figure 3: Time series for click-through rate. Blue, actual click-through data; yellow, band for anomaly detection.

False alarms are common in automatic alerts, and they are sometimes hard to avoid. Usually we could choose some very important metrics to impose a stricter threshold and for other metrics, loosen the threshold to reduce the number of false alarms. False alarms are associated with the multiple hypothesis testing problem. Since we monitored a whole bunch of metrics, statistically the threshold should be corrected to accommodate multiple testing. Usually corrections like Bonferroni are too stringent in such cases, and what is more widely used is to control False Discovery Rate (Benjamin et al. 1995).

5. User Identification

User identification is a vital part in web analytics. For one reason, many important metrics are by nature on user level and user identification is a presumption for these metrics to be available. To name a few, sessions per user is a heartbeat signal for a web site. If a web site is being improved users get better and better experience, we should be able to see sessions per user for a given time period increases steadily. Other commonly used metrics includes page views per user, clicks per user, etc. The second reason is closely related to experimentation. User is widely used as the randomization unit of controlled experiment on web. Therefore without user identification, user based controlled experiment is impossible.

If a website requires user login, then user identification is generally very simple and with high fidelity. For privacy concern, a user login id is encrypted into anonymous id (Ackerman, M. 1999). This mapping should be 1 to 1. The only concern for this user identification scheme is that a real user might have multiple accounts. But we believe for most practical cases the impact of this is negligible. Overall, using login information for user identification has the least data quality concern.

The second approach is to use cookie. A cookie (refer to wikipedia) is a piece of text stored on a user's computer by the web browser. Browser can use the text for various tasks such as authentication, preference storing and user/session identification (Eirinaki, M. & Vazirgiannis, M. 2003). Cookie based user identification can be used in almost all practical cases as long as the browser accepts cookie. Its weakness is also obvious:

1. User could disable cookies;
2. Cookies could be expired automatically by system;
3. User could clear cookies from time to time;
4. User using different browsers that are not sharing cookies would appear to be different users since they have different cookies. Users using different machines would always appear to be different users.

For users who disable cookies, we could choose to exclude those users in our analysis if the percentage is not high. In recent years cookies are widely used in all kinds of websites and it is not very common that a user would completely disable cookies. Users using different browsers and have different cookies could be an issue if multi-browser user population keep growing. But the impact of this is limited, since one user can only install a few browsers and most people would stick to one browser most of the time. Cookie churn, caused by automatic expiration or user deletion, poses the biggest challenge to analysis. Automatic expiration by system could be avoided if we set expiration date to be infinity or a large number. However, there are many constraints that prevent having really long lived cookies including some policy regulations. User deletion is also hard to avoid. For example, many modern browsers allow user to automatically delete cookie as well as temporary files and caches every time the browser closes. Many privacy savvy users might also use some browser extension to manage automatic cookie deletion.

For the purpose of experimentation, we need to make sure the cookie churn in control and treatment are the same, otherwise a serious bias could occur. If the root cause of the cookie churn is in one of the four cases above, then it is uncommon that the treatment could somehow interfere with the way cookie expires or is deleted. But we should always be aware of the limitation of cookie based user identification and question the unbiasedness assumption when we get unexpected results. An effective check is to test the sample size ratio between control and treatment. If the cookie churn is unbiased, the number of users of control or treatment will follow binomial distribution with the probability depending on the corresponding weights. A binomial test can therefore be used. If the test is significant, then the data quality alarm is fired. All user based metrics would be impacted.

To alleviate cookie churn, cookie gluing can be used if there is high confidence that two cookie are actually used by the same user. One example is if two different cookies are linked with the same login user name, then it is almost sure we usually assume that they are used by the same user. In light of this, people can reprocess the data to glue cookies together. However, although this method sounds like a good idea, it could introduce unexpected problem for experimentation. Because of the cookie churn, there are two types of cookies in each variant group. The first type is "full user", that is, cookies that are not churned during the experiment, or cookie churned but the reassigned cookie falls in the same variant group, causing it being glued together with the original cookie before cookie churn. The second type of cookie is "half user", i.e., cookies that churned during the experiment or cookies that born during the experiment as a result of the reassignment of cookie due to a previous cookie churn. It can be shown (with careful probabilistic

argument) that if two variant group are of different sizes, when cookie gluing is applied, the smaller group will end up with a larger "half users/ full user" ratio comparing to the larger variant group. As a result, per uu engagement metrics such as clicks per uu will favor the larger variant group because more "full user" leads to larger clicks/uu. Note that this delta is not a real treatment effect, but merely an unexpected result of applying cookie gluing.

In general, a clear trend is that users are more and more aware of privacy issues (Tavani, H. 1999, Vogelsang, I. & Compaine, B. 2000). Modern browsers try to follow this trend and address this issue in almost every updates by providing better and better privacy service. It could be in the near future that we need to redesign the user identification method. Or a new standard of user tracking will be accepted by public. Another interesting trend is that the percentage of user blocking JavaScript is getting lower. This might be due to more and more JavaScript usage in popular sites such as Facebook. Research is needed in this area to find out better technology to facilitate more accurate analysis as well as protecting user privacy.

6. Instrumentation

Proper instrumentation is critical for ensuring we are able to effectively capture metrics of interest for an experiment while minimizing impact on the user experience. Common metrics of interest for a web page include the number of page views, the page load time, and the number of clicks. To ensure high quality data, we need to ensure 1) any instrumentation that is added has a similar impact for both control and treatment groups, 2) we capture user actions of interest for our experiment, 3) we are able to able to isolate effects caused by browser differences, and 4) our instrumentation is easy to maintain as our web site evolves. Next we are going to talk about several instrumentation issues to consider:

1. Larger propagation delay for treatment or control due to use of different servers and/or redirects: It's important that both the control and treatment variants be hosted in the same environment as they would be deployed. Additional propagation delays caused by lower bandwidth connections, or HTTP redirects, or the use of older servers, or the use of misconfigured servers, will provide misleading results. Prior to a first experiment, or after any significant configuration changes, it is advisable to perform an AA test, to ensure there are no significant performance/engagement differences caused by differences other than the treatment.
2. Advantages of JavaScript instrumentation: While image (e.g. one pixel ".gif") files can be useful for recording page view event information, JavaScript provides the ability to record enhanced information, such as click events, hover events, scroll events, page load completion events, and browser window dimensions. The drawback is, of course, that your instrumentation now requires the user to have JavaScript enabled to work. Fortunately, many popular web sites have this requirement.
3. Observed differences in effect due to differences in how browsers render the page or process the instrumentation: It's useful to track key metrics for experiments by browser, to identify whether performance/engagement differences may be related to differences in browsers. Possible differences between browsers include both how the page is rendered and how events are processed. For example, an observed difference between Internet Explorer and FireFox meant that we sometimes would not see

JavaScript event beacons from one of them. Having the metrics broken down by browser allowed us to easily identify this source of experimental bias.

4. Additional clicks caused by a user clicking a link more than once while waiting for the page to start loading: Some users will double click links, rather than clicking only once. It's important to take steps to either ensure the browser will only generate one event beacon, or to ensure that only one of the beacons is processed on the backend if multiple beacons are generated.
5. Hovers, flyouts, and tab/slide navigation counted as clicks: It is often the case that we want to measure overall engagement on a page in terms of whole page clicks per user; however, this should likely exclude some events, such as hovers. We recommend tracking hovers, flyouts, and tab/slide navigation events separately, as part of secondary metrics.
6. Tracking source of the referral to the page, as well as location on the page and destination for links: It's useful to track these additional sources of bias. For referral sources, it is typically sufficient to track the few most frequent sources, along with an "other" category. For location on the page, it can be beneficial to whether a link is above/below the fold (below the fold requires scrolling the view), in the header/footer, in the left/right rail, or in some particular module. Tracking links by location allows locations to be easily compared between control and treatment variants, often showing how enlarging a module leads to more clicks on the module by cannibalizing clicks from other modules. Tracking links by destination, allows destination engagement to be easily compared between control and treatment variants, again often showing how increasing exposure to one destination impacts other destinations.
7. Using cookies to cache treatment assignments: Treatment assignments are more efficiently handled on the server side, instead of requiring the client to make a call for the treatment assignment prior to loading the page. Care should be taken, however, to ensure that the number of treatments assignment calls per user is similar for both treatment and control groups.

Acknowledgements

We would like to thank members of the Experimentation Platform team at Microsoft, especially Ronny Kohavi and Randy Henne.

References

Kohavi, R.., 2007. Practical guide to controlled experiments on the web. In: *Knowledge Discovery and Data Mining (KDD 2007)*, August 12-15, San Jose, CA.

Thomke, S.H., 2003. *Experimentation matters: unlocking the potential of new technologies for innovation*, 4th ed., Harvard Business Press.

Peterson, E.T., 2004. *Web analytics demystified: a marketer's guide to understanding how your web site*, Celilo Group Media.

Tan, P-N & Kumar, V., 2002. Discovery of web robot sessions based on their navigational patterns, *Data Mining and Knowledge Discovery* 6 (1), pp.9-35.

Nicholas, D., Huntington, P. & Willianms, P., 2001. Establishing metrics for the evaluation of touch screen kiosks, *Journal of Information Science* April 2001 27, pp. 61-71.

Teng, H., Chen, K. & Lu, S., 1990. Adaptive Real-time Anomaly Detection Using Inductively Generated Sequential Patterns. In: *IEEE Comp. Soc. Symp.* May 1990, Oakland, CA.

Hajji, H., 2005. Statistical Analysis of Network Traffic for Adaptive Faults Detection, *IEEE Trans. Neural Networks* 16 (5), pp. 1053-1063.

Lakhina, A., Papagiannaki, K., Crovella, M. & Diot, C., 2004. Structural Analysis of Network Traffic Flows. In: *ACM SIGMETRICS*, Jun. 2004, Columbia University, NY.

Chambers, D., 1992. Understanding Statistical Process Control, 2nd ed., SPC PRESS.

Ringberg, H., Soule, A., Rexford, J. & Diot, C., 2007. Sensitivity of PCA for traffic anomaly detection. In: *ACM SIGMETRICS*, Jun. 2007, San Diego, CA.

Lakhina, A., Crovella, M. & Diot, C., 2005. Mining anomalies using traffic feature distributions. In: *ACM SIGCOMM*, Oct. 2005, Philadelphia, PA.

Benjamin, Y. & Hochberg, Y., 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing, *Journal of the Royal Statistical Society*, *Series B (Methodological)* 57 (1), pp.289-300.

Ackerman, M., 1999. Privacy in e-commerce: examining user scenarios and privacy preferences. In: *ACM conference on Electronic commerce*, November 3-5, 1999, Denver, Colorado.

Eirinaki, M. & Vazirgiannis, M., 2003. Web mining for web personalization, *ACM Transactions on Internet Technology (TOIT)* 3 (1), pp.1-27.

Tavani, H., 1999. Privacy online, *ACM SIGCAS Computers and Society* 29 (4), pp. 11-19.

Vogelsang, I. & Compaine, B., 2000. *The Internet Upheaval: Raising Questions, Seeking Answers in Communications Policy (Telecommunications Policy Research Conference)*, The MIT Press.