Website Monitoring and Improvement

Roger Longbotham, Ji Chen, Dave DeBarr, Alex Deng, Justin Wang

Bing Data Mining

Microsoft Corporation, One Microsoft Way, Redmond, WA 98052

There are specific statistical issues and methodologies for monitoring and improvement of websites. We will discuss metrics, limitations and pitfalls to ongoing website monitoring and experimentation to improve website performance. The authors have many years of experience with a number of websites and will share best practices and lessons learned. Specific topics include online experimentation principles, feedback analysis and monitoring, visitor segmentation and content optimization.

Keywords: website monitoring, website improvement, website experimentation, web metrics

Statistical methods play a big role in many areas of website monitoring and improvement. We will highlight some of those areas that we have had experience with in many years in the online world. We all currently work at Microsoft and have worked on many Microsoft websites, but we also bring experience from Amazon and Zillow. We have chosen to focus our comments on the areas of website metric monitoring, experimentation, feedback monitoring and feature optimization.

Website Metric Monitoring

In order to track the long term performance of the website, many websites have developed their own system of ongoing monitoring of the key metrics of their sites. There are several aspects to the ongoing monitoring that we want to discuss:

1. Choosing the right metrics. There are some commonly tracked key metrics which most people agree on as follows:

Unique Users per Day/Week/Month/Year
Pageviews per Day/Week/Month/Year
Clicks per Day/Week/Month/Year
Clicks per User (day)
Click-through Rate
Sessions per User (over a certain period, such as day, week and month)

However, there are several pitfalls for even choosing the most agreed upon metrics. An example is Click-through rate, which is calculated as clicks/pageviews. Normally an increase of the click-through rate is considered a healthy sign, unless the increase comes at the expense of overall lower clicks and pageviews. Something also worth noting is the cannibalization effect. Suppose a new feature is launched on a certain module of the page, and a natural idea would be to monitor the clicks to that specific module to gain insight into the feature, however, we also have to note if there is change in other modules. We could just be moving clicks from one module to another without attracting more clicks overall. There, in this case, besides clicks to this module, whole page clicks is also a reasonable metric to look at.

2. Taking into account non-stationarity

Most times web monitoring data would have a trend line (including flat) and also strong cyclicality. Take the number of unique users for example, the cyclicality could include:

- Weekly pattern. Most websites would see a strong weekly pattern in traffic where weekends are lower and weekdays are higher.
- Seasonality. An example for that would be skiing resort websites.
- Holiday dip/rally. Most websites see significantly lower traffic than usual during major holidays, such as Thanksgiving, Christmas, etc.

These would have to be accounted for when trying to evaluate the trend for the site.

3. Alerts for monitoring system

It is beneficial to have automatized alerts for the monitoring system so as to catch anomaly quickly and without much human effort. Most alert systems are based on heuristic algorithm, such as when traffic drops to a certain level. To build more refined systems, the non-stationarity of the traffic could be considered.

4. Competitive research

When monitoring the website long term, it is usually good to do some competitive research to get a good reference point. Competitive research could be especially useful in the three areas:

- Impact of feature release. For some websites, such as search engine, an important success metric of a feature release could be an increase in market share against other competitors.
- Seasonality. To confirm the seasonality observed for your own site, an approach to take could be to observe the other sites in the same sector.
- Macro trend for this sector. When reading/interpreting the trend line of your own website, it is always good to put it into context with other similar websites to see if your trend line is aligned with/outperformed/underperformed other sites in the same sector.

Experimentation

Issues of Power

Using a large % of population is a direct way to increase sample sizes, and therefore to increase power. For a fixed duration experiment, the sample size is proportional to the % of traffic that is assigned to the experiment. It is therefore fairly straightforward to do power calculation.

Sample size can also be increased by running experiment longer, but the power calculation is more subtle for this case. Firstly, as experiment runs, the mean and variance for a metric may also change. A crucial value for power calculation is the coefficient of variation, which is defined as the standard deviation divided by the mean. Using empirical data, we found that coefficient of variation for different metrics behaves differently. Some increases and some decreases as experiment run longer. However, for those metrics with coefficient of variation relatively stable, we can set an upper bound of the value and use this upper bound in the power calculation to get a conservative estimation of the power.

Secondly, sample size does not increase linearly as the function of experiment running length. We found with our empirical data that sample size can be well fitted as proportional to d^r where d is the number of days the experiment run and $0 < r < 1$. r is determined by the daily return rate. In one extreme, if the return rate is 0, i.e., all users only appear once, then $r = 1$ and sample size grows linearly. On the other extreme, if return rate is 1,i.e., all users are extremely loyal and return every day, then $r = 0$ and sample size will not grow at all.

- Ways to increase power

If the low power issue is due to triggering rate of certain feature:
Triggered analysis /Filtered analysis.
Pitfall: logging counterfactual in real time may add additional performance hit. (Badabing example, in which logging whether Badabing would have fire would require the server go through extra Badabing work flow). Also, it is very hard to make the real time counterfactual logging right. We recommend using also a standard control so that by comparing the counterfactual logging control with the standard control, we can quantify the potential performance hit and also detect any unexpected effect due to counterfactual logging.

An alternative to real time counterfactual logging is to log counterfactual offline. For example, if we know exact the rule that determines the trigger of certain feature, we can then determine whether a feature would have been triggered offline. But when the triggering rule is very complicated, offline counterfactual logging can be formidable or resource consuming.

If the low power issue is due to the fact that there is only a small expected treatment effect, then we have to question whether it is worth running the experiment in the first place.
If the low power issue is due to generally low traffic, then the only solution is to run the experiment longer with all traffic assigned to the experiment.

Cost of experimentation

The purpose of doing experimentation is to reduce the risk of exposing users to bad experience had we launch a feature without carefully assessing its impact. But, at a price to pay, we still need to expose a percentage of users to the bad experience to learn the very fact that it is bad. Every experiment conductor should keep this in mind because unlike traditional control experiment such as clinical trial, there are no direct interaction between experiment conductor and the end users. It is very easy for us to overlook the fact that it is real users that are experiencing any changes we make and they may response to a disappointing experience by leaving and giving up the web site. Therefore, we should pay extra attention to those experiments having high risk. (example: adult content experiment?) Also, a systematic approach to assess the “cost” of experimentation is important. For example, we need to keep tracking on how many users ever experienced a bad treatment. When monetization is available, it is even better to have the idea of how much it cost to run a bad treatment. Statistical method can be used to help us defined a bad treatment. One choice is that we can simply use OEC. If OEC is significantly worse, than we say a treatment is bad. If we have a few key metrics, principle component analysis can effectively reduce these metrics to a linear combination of these metrics, leaving only one signal to help us determine whether a treatment is statistically good or bad.

Experimentation philosophy

- Test everything vs only test "substantial" features

If we strictly follow the data driving decision making philosophy, we would like to test everything! However, in practice, there are limitations. If a feature has only a tiny impact, then it is highly likely that the corresponding statistical test cannot reach enough power, which means the experiment will end up with no statistically significant result and hence this experiment is futile. On the other hand, packing all small features into a big “substantial” feature for testing is generally a bad idea. First, the effects of all these individual features are combined together and also potentially can interact with each other. Since the result of the test only tells us how the combined feature performs, it is impossible for us to have an idea of how each individual feature contributes to the combined feature. Secondly, a feature that is too “substantial” is likely to impose “newness” effect on user. For example, user needs time to familiar herself with a big UI change. This “newness” effect could easily confound the analysis and even change the direction of the treatment effect. We believe that experiment design is an art and picking the right feature to test is a vital part of the design. The feature to be test needs to at least pass the power analysis. If a feature is too “substantial” that it is likely to have “newness” effect, then try to divide the feature into several features and conduct a series of tests. If the feature cannot be divided, the “newness” effect needs to be aware of. We need to carefully monitor the trend of the treatment effect to identify how significant and how lasting the “newness” effect is. After we find that the “newness” effect is gone or can be negligible, then we can do analysis based on the data collected after this time spot.

- Test raw ideas vs make sure the feature is polished

Many people get the idea of using experimentation to quantify the performance of a feature. But experimentation is equally useful in helping us polishing raw ideas into full-fledged features.
The availability of doing experiment quickly and with low cost encourages us to take advantage of it into design phase. Testing a raw idea and knowing it is bad prevent us from investing more resources on this idea. More importantly, testing several variations of raw ideas and find out which works and which doesn’t provides us with a better understanding of the user of the web site. This knowledge can also tell us how to further polish these raw ideas into production.

- Only roll out features that show positive statistical significance

One argument is that we should only ship features that show positive statistically significant change. If a feature has flat or even negative change and we still want to ship it, we should at least try to refine it or combine it with other features. But eventually there has to be an experiment that clearly shows positive statistically significant change for us to ship the feature, either in a refined form or combined with other features.

On the other hand there are definite exceptions to the above rule. One exception is the case when PR already releases the feature to public. Then we may be forced to ship the feature. But in this case a reverse experiment is strongly recommended. Another exception is when the anticipated benefit cannot be measures with a short term experiment. For example, a change to make the site more up-to-date may turn off some users in the short term (as measured by an experiment over the first few weeks) since users generally react negatively to change. But in the long run this may add capabilities that the site can deliver richer content in the future, giving users a better experience. Another situation is when the underlying software that runs the site must be updated. You should run an experiment just to make sure there are no problems with the implementation. What you would hope to see is no change in user behavior.

Randomization and analysis unit implications

There are important distinctions between randomization unit and analysis unit. Randomization unit is the unit that the randomization is applied upon. Two common choices are user (commonly with a cookie) and page view. Analysis unit is related to a metric. If a metric is a per user metric, then the analysis unit is user. Similarly, a per-page view metric uses page view as analysis unit. When analysis unit and randomization unit are different, we need to pay extra attention on the statistical analysis. For example, when randomization unit is user and analysis unit is page view, delta method should be applied in the variance calculation for the t-test. There are also pros and cons in choosing different randomization unit which is out of the scope of this paper. (refer to Deng, Longbotham, Walker and Xu?)

Success Rate of Tested Ideas

The success rate of ideas that are tested in experiments is lower than most people expect. A book published by QualPro (ref) give the success rate of ideas (those that give a statistically significantly improvement) at approximately 25%.

Online experimentation often has an even lower success rate. One experimentation platform we have worked on had an approximate 25% rate, comparable to QualPro. For the Bing search engine we have recently measured it as 11% and, in a recent article (ref) Google gave their success rate as 8%. One reason for the difference in these numbers is undoubtedly how they count an idea as a success, but another is the maturity of the website. It is more difficult to find ideas that will improve a mature website that has gone through many rounds of testing and is more “optimized” than for a less mature site.

The low success rate could be discouraging to someone who is wanting to do experiments, but it also points out the necessity of running experiments to improve websites and web services. Without the help of experimentation the business person would not be able to determine which of their ideas are really helpful in meeting their goals and they would be implementing many ideas that are either not beneficial or actually harmful to their site or customers.

Tracking

Almost all online experiments depend on tracking the activities of users to determine whether a feature is helpful or not. One reason is that randomization is normally done by user. If we randomize by another unit, e.g. page view, and a change is very noticeable, it may be a highly negative customer experience for the page to change every time the user comes to the site (opens the page). We often get comments on the negative effect of large changes to a site and to have the site changing on every page load could be disastrous from a customer retention perspective and it could bias the experiment, as well. Another reason to identify and track users is for the calculation of important metrics. Some of the most important metrics are often ones that reflect the impact of the tested feature on the engagement or longer term behavior of users. These metrics could not be calculated if we could not track users.

Current methods of tracking users are not perfect. Some sites require the user to sign in, so they would be able to identify and track users fairly reliably. However, most sites do not require a sign in and depend on user ids stored in cookies on the user’s machine. These cookies are specific to a browser so if a user comes to the site with Internet Explorer and later with Firefox or Chrome they will have a different cookie and would appear to be a different user. Also, if the user comes to the site from a different computer they would appear to be a different. If more than one person uses a computer, you may have several persons identified as a single user. Finally, many users clear cookies, either regularly or occasionally, which means a new cookie will get set with a new user id.

There is a general trend in the U.S. and even more so in Europe for online privacy which often translates to users opting out of (cookie) tracking and many browsers have an option now that the user can set that will discourage or disable tracking capabilities. We expect this trend to continue and we will likely get to the point that tracking by user is no longer a viable option. We may need to develop alternative methods of experimentation, especially randomization schemes and new metrics that don't depend on identifying users.

Feedback Monitoring

User feedback can also usefully contribute to experimental results and website monitoring. Examples of feedback links for Microsoft include the "Feedback" link at the bottom of the www.msn.com homepage and the "Tell us what you think" link at the bottom of the www.bing.com homepage. The two principal types of data collected include Likert scale ratings (Unsatisfactory, Poor, Satisfactory, Good, Excellent) and text comments. For Likert scale ratings we test the null hypothesis that the proportion of users providing feedback is equal for both the control and treatment groups, as well as the null hypothesis that the mean Likert scale rating is equal for the control and treatment groups. For the text comments, we use clustering to summarize groups of related comments to make it easier for human review and sentiment classification to identify whether comments are positive or negative. We can then test the null hypothesis that the proportion of users providing negative feedback is equal for both the control and treatment groups.

Feature Optimization

Users clicking more on the links on the page is an indication of user satisfaction and engagement in the content of the page. One of the common objectives of a web page is to provide the links which attract more clicks. Pages like http://ww.msn.com, http://www.yahoo.com and www.cnn.com have very rich-content layouts consisting of multiple modules and tabs. Users visit the page might have different preferences or specific interests. For example, some users might be more interested in sports; others might visit the page mainly to check finance news and the others might scan the whole page and only click on the links that seem to be more interesting to them. Users' visit history provides the clue for understanding them better. Setting the default module and tab to the users' most interested category brings convenience to the users and can encourage user loyalty.

Besides individualized module optimizations (MOPs) and tab optimizations (TOPs) , each module and tab usually contain multiple headlines for users to click on for the detailed articles/stories. Editors constantly update articles and stories to reflect the current happenings and only a few of them(usually fewer than 6) can be displayed in a module to the users due to limited module space. The study of how to choose the most popular headlines to display and in which order to display the chosen ones is called headline optimizations (HOPs). This sessions discusses HOPs algorithm.

HOPs can be formulated as a multi-armed bandit problem (Vermorel 2005) which simultaneously attempts to acquire new knowledge and to optimize its decisions based on existing knowledge. The algorithm starts with splitting the users into two groups: the exploration group and the exploitation group. The exploration group is usually a relatively smaller group used to collect information about which headlines are most attractive to users. One trivial way to achieve this is showing users in the exploration group randomly selected headlines from all the headlines provided by editors in a random order. This is not a very efficient way and can hurt user experience dramatically. There are other improved bandit strategies which always show the "best" headlines except when a random action is taken to test other headlines. Those improved algorithms are designed to enhance the efficiency and mitigate the negative impact on the exploration group. After learning the possibility of the headlines will be clicked from the exploration group, the headlines with highest expected probability to be clicked are then displayed to the users in exploitation group by the order of the expected probability that they will be clicked.

Since the users in the exploration group see the headlines selected and ordered in a sub-optimal way, we recommend replacing the exploration group with different users regularly to avoid constantly hurting the experience of a certain group of users. 1) How large the exploration group should be, 2) how often it should be replaced, 3) how long it should be run before applying the results to the exploitation group and 4) how to use the exploration results to estimate the possibility of the headlines will be clicked by the exploitation group depend on the traffic volume of the website and quantity and quality of the headlines and can potentially be analytically studied for the optimal setup. A large volume website with millions visitors a day like MSN homepage, we find choosing 0.5% of users as the exploration group and replacing the group every minute significantly improve the number of clicks comparing with the headlines manually selected and ordered by the editors.

"Multi-Armed Bandit Algorithms and Empirical Evaluation", European Conference of Machine Learning 2005, Joannès Vermorel, Mehryar Mohri