Statistics at Kameleoon

January 8, 2023

Contents

1 Kameleoon's assignation algorithm **3**

2 Frequentist tests **4**
- 2.1 Simple A/B test . . . 4
- 2.2 A/B/C/n test . . . 5
- 2.3 Multiple testing correction . . . 6
- 2.4 Confidence interval . . . 6
- 2.5 Visit-based tests . . . 7
- 2.6 All conversions tests . . . 7
- 2.7 Minimal test duration to obtain significant result . . . 8

3 Bayesian A/B Testing **10**
- 3.1 Bayes probability . . . 10
- 3.2 Bayesian decision rule . . . 11

4 Multi-armed Bandits **12**

Outline

The goal of this document is to introduce and justify from a theoretical point of view, the different statistical tools used at Kameleoon.

We start by presenting how Kameleoon assigns visitors to different variation within an experiment which is the basis for what follow.

The second part is about Frequentist tests, which are run by default on our result page. We first tackle the simple A/B test with only the original and one variation, we explain the modelization, the hypothesis and how do we compute the reliability. Then we extend those computations to A/B/C/n tests and explain how we compute our confidence intervals. To close this first part we present a few studies we conducted regarding the impact of running visit-based tests and explain how we could run a test on the "all conversions" variable. Finally we give the formulas to compute the minimal test duration for a given power.

In a third part we present our Bayesian A/B testing approach, how does it differs from the frequentist one and how we compute the test statistics in this case.

Finally in a fourth and last part we outline the multi-armed bandit framework, how it applies to A/B testing and give details regarding our own multi-armed bandit implementation.

1 Kameleoon's assignation algorithm

To assign a visitor to an experiment variation, Kameleoon first build an identifier made of the user visitor code which is a string identifying the visitor, the experiment ID and a potential additional element in case of respooling. Then we use our own synchrone implementation of the hash function SHA-256 [1] to compute a hash of this identifier. The integer obtained through hashing is then mapped to a floating number between 0 and 1 in order to assign it to an experiment variation given the experiment assignation setup. The SHA-256 function is deterministic, hence the same user (with the same visitor code) will always be assigned to the same variation for a given experiment unless we explicitly ask to recompute the assignation. This is also the case if you use one of our SDK.

We implemented our own synchronous version of the SHA-256 function to be able to run it inside any web browser or application and to be sure to compute the assignation as efficiently as possible. SHA-256 is uniformly distributed which is a required property in order to distribute visitors into variation as specified in the experiment setup.

For example, if we have an experiment which assigns 30% of its experimentee to the original, 30% to the variation and 40% do not take part in the experiment at all. Then if a user get an assignation value from our hash function of 0.33, first he will always get the same value for that experiment unless an explicit respool is asked. Secondly, we assign all users with identifier hashed between 0 and 0.3 (original deviation) to the original, those between 0.3 and 0.6 (original deviation + variation deviation) to the variation and those between 0.6 and 1 do not take part in the experiment. So this specific user which maps to 0.33 will see the variation.

2 Frequentist tests

2.1 Simple A/B test

A/B test is a simple controlled experiment, in which two versions of a single variable are compared. Version A is the currently used version (the control), while version B is modified in some respect (treatment). Every visitor, who is targeted by a test only sees one version. For instance, version A might be the current checkout experience on a given website and version B - a new checkout experience with a recommendation. Conversion (success) in this case can be defined as a click on finalising the purchase. The goal of the test is to discover which of the versions performs better.

A visitor can either convert a goal with probability p or not with probability $q = 1-p$, so conversions and converted visitors can be represented by Bernoulli distribution with parameters p and q. Let n_i be the number of visitors allocated to a version i, c_i be the number of visitors who converted a goal and were allocated to version i. As a sum of independent Bernoulli trials, c_A and c_B follow binomial distributions $B(n_A, p_A)$ and $B(n_B, p_B)$. The conversion rate x_i for a version i can be defined as:

$$x_i = \frac{c_i}{n_i} \tag{2.1.1}$$

x_A is a current observed conversion rate for the control, x_B - current observed conversion rate for the alternative. The null hypothesis is that there's no difference between the conversion rates of two versions. The alternative hypothesis is that the conversion rate of version B is higher than that of version A.

- $H_0 : x_B \leq x_A$
- $H_A : x_B > x_A$

The test statistic T is the difference between two conversion rates x_A and x_B. In classical hypothesis testing, before the test we also select a probability threshold α, below which the null hypothesis will be rejected. By convention, α is commonly set to 1%, this means that one time out of a hundred there's a risk of concluding that a difference exists when there's no actual difference between the conversion rates.

In the most of our applications, the sample size is large: common rule of thumb is to check whether the sample size is bigger than 30. The observations are independent by design. This is why according to the central limit theorem, the distribution of T under H_0 can be approximated by a normal distribution. The appropriate location test is called a Z-test. To perform this test, we first calculate the population standard deviation σ.

$$\sigma = \sqrt{\frac{x_A(1-x_A)}{n_A} + \frac{x_B(1-x_B)}{n_B}} \tag{2.1.2}$$

Since the sample size is large, we can perform a plug-in test by using the population standard deviation σ, instead of a sample standard deviation, to calculate the standardised statistic Z [2].

$$Z = \frac{x_B - x_A}{\sigma} \tag{2.1.3}$$

To decide whether to reject the null hypothesis in favour of the alternative, we calculate the p-value. **p-value** is the probability of obtaining test results at least as extreme as the results actually observed, under the assumption that the null hypothesis is correct. A very small p-value means that such an extreme observed outcome would be very unlikely under the null hypothesis. Since the distribution of the test statistic is symmetric around 0, the two-sided p-value is calculated as

$$p = 2 * (1 - \Phi(|Z|)) = 1 - erf(\frac{|Z|}{\sqrt{2}}) \tag{2.1.4}$$

where Φ is the standard normal cumulative distribution function and erf is the error function, an entire function defined by

$$\text{erf(z)} = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt \tag{2.1.5}$$

The null hypothesis is rejected if the p-value is less than the chosen threshold α. However, us not being the final decision maker, we prefer to show **reliability** instead of the p-value and leave the client to decide whether the result of the test is sufficient to conclude that there's a significant difference between the versions.

$$\text{reliability} = 1 - p \tag{2.1.6}$$

2.2 A/B/C/n test

The procedure described in the previous section can be extended to test multiple versions at the same time. For example, we could test four versions of a page and do an A/B/C/D test, where A is the original version of a page and B, C, D are versions of the same page each modified in its own respect. The purpose of such a test would be to discover which of the modifications (B, C, D) will result in better performance compared to the original A.

To perform this test, we will repeat the procedure described in the previous section for the following three pairs of versions: A/B, A/C and A/D. Each such pair is treated as a separate test: for each such pair we formulate a set of hypotheses H_0 and H_A.

For $i \in \{B, C, D\}$:

- $H_0 : x_i \leq x_A$
- $H_A : x_i > x_A$

We will calculate the population standard deviation σ_i, the standardised statistic Z_i, the corresponding p-value and the reliability.

$$\sigma_i = \sqrt{\frac{x_A(1-x_A)}{n_A} + \frac{x_i(1-x_i)}{n_i}} \tag{2.2.1}$$

$$Z_i = \frac{x_i - x_A}{\sigma} \tag{2.2.2}$$

$$p_i = 2 * (1 - \Phi(|Z_i|)) = 1 - erf(\frac{|Z_i|}{\sqrt{2}}) \tag{2.2.3}$$

$$\text{reliability}_\text{i} = 1 - p_i \tag{2.2.4}$$

Reliability$_\text{i}$ is shown next a conversion rate x_i of every version i, $i \in \{B, C, D\}$. This procedure can be extended to any number of versions, not only four.

2.3 Multiple testing correction

When the number of variation increases, we increase the probability of obtaining a type I error (false positive). Indeed, the more inference are made, the more likely it is that one is erroneous. Several technics were developed to deal with this issue, we chose to implement the Holm–Šidák method as it is more powerful than the Holm-Bonferroni method. To apply this correction we correct the p-values using the following formula. This formula is applied recursively to the multiples p-values sorted in ascending order.

$$\tilde{p}_{(i)} = max\{\tilde{p}_{(i-1)}, 1 - (1 - p_{(i)})^{m-i+1}\}, where\ \tilde{p}_{(1)} = 1 - (1 - p_{(1)})^m \tag{2.3.1}$$

Then the testing procedure can resume, and a hypothesis is rejected at level α if and only if its adjusted p-value is less than α. This allow us to control the family wise error rate at the given level α.

2.4 Confidence interval

The test can answer the question whether version B performs better than A, but it doesn't quantify how much better it performs or the level of uncertainty associated. To quantify the performance of version B compared to version A, we introduce **improvement rate** [3]:

$$IR = \frac{x_B}{x_A} - 1 \tag{2.4.1}$$

To present the uncertainty associated with this parameter, confidence intervals are shown. **Confidence interval** gives a range of values for improvement rate and has an associated confidence level that gives the probability with which the estimated interval will contain the true value of the parameter. The confidence level represents the theoretical long-run proportion of confidence intervals that contain the true value of the improvement rate. By convention, we use a confidence level of 95%, which means that 95% of confidence intervals computed at this level contain the parameter.

The confidence interval is given by

$$[IR - z_{1-\frac{\alpha}{2}}\sqrt{\frac{1}{c_A} + d\frac{1}{c_0} - \frac{1}{n_A} - \frac{1}{n_0}}, IR + z_{1-\frac{\alpha}{2}}\sqrt{\frac{1}{c_A} + \frac{1}{c_0} - \frac{1}{n_A} - \frac{1}{n_0}}] \tag{2.4.2}$$

where $z_{1-\alpha/2}$ is the $(1-\alpha/2)$-th quantile of a standard normal distribution. For the chosen confidence level $C = 100(1-\alpha)\% = 95\%$, $z_{1-\frac{\alpha}{2}} \approx 1.96$.

Confidence interval is calculated for every pair i of (version_A, version_i), $i \in \{B, C...\}$ to show the uncertainty around improvement rate of every modified version with respect to the original version.

2.5 Visit-based tests

The A/B tests discussed so far were performed on a visitor level. For those tests, the two assumptions were satisfied by design:

- the observations are independent
- the samples are big enough ($n_i \gg 30$)

Thus, the distribution of the test statistic $T = x_A - x_B$ under the null hypothesis is a normal distribution according to the central limit theorem.

Visit-based tests violate the assumption of observations independence, since the same visitor can return many times. To ensure that we can still use the same design for visit-based tests, we compared the empirical CDF to the CDF of the normal distribution and performed a Kolmogorov-Smirnov test. This non-parametric test (of equality of two continuous one-dimensional probability distributions) is used to compare a sample with a reference probability distribution. Our simulations showed that the empirical CDF can be approximated by a normal distribution, thus z-test is appropriate to use.

2.6 All conversions tests

In the case of all conversions test, each visit v_i is no longer a Bernoulli random variable, rather it is a Multinoulli random variable with K outcomes, where K is the maximum number of conversions per visit in the dataset, that is why a different procedure is to be implemented for this type of tests. A Multinoulli random variable takes a value 1 if visit has converted k times and 0 otherwise, the Multinoulli random vector x_i has a length K and zeroes in all positions except for the k-th. Each variation V_j of this test is a sum of Multinoulli random variables, thus it follows a multinomial distribution. Variation can be represented by a vector where for every position k, the vector contains the number of visits converted exactly k times.

The test used to compare two multinomial distributions is a Pearson's Chi-Squared test. [4] To perform the test, each variation $V_j \in \{V_A, V_B\}$ is to be reshaped into a vector $v_j = (v_{j0}, ..., v_{jK})$, where v_{j0} is the number of non-converted visits, v_{j1} - number of visits converted exactly 1 time, ..., v_{jK} - number of visits converted exactly K times. We'll call vectors v_j vectors of frequencies.

To be able to conduct the test, zero values must be treated in the V_A. For every $i \in \{0, ..., K\}$, if $v_{iA} = 0$, it must be summed with the next element until it's not zero. If there's a zero at the K-th position, it is to be summed with the value on the K-1-th position. The corresponding elements of V_B vector should be summed too. Vectors of frequencies v_j-s all have the same dimensionality after this operation.

The test statistic χ^2 can be computed the following way:

$$\chi^2 = \sum_{i=1}^{K} \frac{(v_{iB} - v_{iA})^2}{v_{iA}} \tag{2.6.1}$$

The parameter of a χ^2 distribution is called the degrees of freedom df:

$$df = dim_v - 1 \tag{2.6.2}$$

where dim_v is the dimensionality of a vector of frequencies for any variation.

To obtain the p-value, we calculate the critical value for Chi-squared distribution with df degrees of freedom for the test statistic. The reliability is

$$\text{reliability} = 1 - p \tag{2.6.3}$$

2.7 Minimal test duration to obtain significant result

- α (type I error): probability of rejecting a true null hypothesis
- β (type II error): probability of not rejecting a false null hypothesis
- 1-α: desired reliability
- x_0: current conversion rate
- IR: desired improvement rate
- v: number of versions in the test minus one
- k_i: ratio between traffic allocated to A and traffic allocated to the i-th version
- n_{daily}: average number of daily visitors

For each pair $i = 1...v$ of (version_A, version_i), $\text{version}_\text{i} \in \{\text{version}_\text{B}, \text{version}_\text{C}, ...\}$ the following procedure is repeated: Since the conversion rate for a version is unknown before the test, it can be estimated using the desired improvement rate:

$$x_i = x_A(1 + IR) \tag{2.7.1}$$

The sample size for version A (n_A) and for the version i (n_i) then can be approximated by [5]:

$$n_A = k_i n_i \tag{2.7.2}$$

$$n_i = \frac{(z_{\alpha/2} + z_\beta)^2}{(x_i - x_A)^2}[x_A(1 - x_A)/k + x_i(1 - x_i)] \tag{2.7.3}$$

where $z_{\alpha/2}$ is the $(1 - \alpha/2)$-th lower quantile of a standard normal distribution, z_β is the $(1 - \beta)$ lower quantile of a standard normal distribution.

Required test duration in days for the pair (version_A, version_i) is

$$d_i = \lceil \frac{n_0 + n_i}{n_{\text{daily}}} \rceil \tag{2.7.4}$$

where $\lceil \rceil$ denotes the ceiling operator. This operator maps x to the least integer greater than or equal to x, ceil(x)=$\lceil x \rceil$.

The maximum of all d_i-s is the duration (in days) of the test.

$$d = max\{d_1, ..., d_v\} \tag{2.7.5}$$

During the test, the conversion rates x_i for every version i are known, so the estimation (1.6.1) should be omitted and observed x_i-s should be plugged directly into equation (1.6.3). We repeat the procedure described above to obtain d. The number of days left to achieve a significant result is

$$d_{\text{left}} = d - d_{\text{passed}} \tag{2.7.6}$$

3 Bayesian A/B Testing

3.1 Bayes probability

In the A/B testing framework, the goal of the Bayesian formulation is to calculate the probability that the version A is different from the version B. By convention, if the probability exceeds 95%, we can conclude that the result is statistically significant. This is a necessary but not sufficient condition, the second condition will be described in the next subsection.

- s_A: number of visitors who converted a goal in the version A
- s_B: number of visitors who converted a goal in the version B
- f_A: number of visitors who didn't convert in the version A
- f_B: number of visitors who didn't convert in the version B

The conversion rate is defined as:

$$\begin{aligned} x_A &= \frac{s_A}{s_A + f_A} \\ x_B &= \frac{s_B}{s_B + f_B} \end{aligned} \tag{3.1.1}$$

The biggest distinction between the frequentist approach and the Bayesian one is the concept of prior probability distribution. Prior is the probability distribution that would express one's beliefs about the parameter (the conversion rate in our case) before new evidence is taken into account. We'll model the conversion rate distribution using Beta family of distributions as they provide a family of conjugate prior probability distributions for binomial (hence Bernoulli) distributions.

$$\begin{aligned} x_A &\sim B(s_A + 1, f_A + 1) \\ x_B &\sim B(s_A + 1, f_B + 1) \end{aligned}$$

Using the probability density function of the beta distribution and the Bayes' theorem, we can get **the total probability that x_B is greater than x_A** by integrating the joint distribution over all values for which $x_B > x_A$ [6]. Let's denote this posterior probability as $H(s_A, f_A, s_B, f_B)$.

$$H(s_A, f_A, s_B, f_B) = Pr(x_B > x_A) = \int_0^1 \int_{x_A}^1 \frac{x_A^{s_A}(1 - x_A)^{f_A}}{B(s_A, f_A)} \frac{x_B^{s_B}(1 - x_B)^{f_B}}{B(s_B, f_B)} dx_B dx_A \tag{3.1.2}$$

Or equivalently:

$$Pr(x_B > x_A) = \sum_{i=0}^{s_A - 1} \frac{B(s_A + i, f_A + f_B)}{(f_B + i)B(1 + i, f_B)B(s_A, f_A)} \tag{3.1.3}$$

3.2 Bayesian decision rule

In addition we need to calculate the Bayesian cost function in order to conclude that a result is statistically significant and the test can be stopped. First we'll set a threshold ε. If the probability of conversion rate of A being better than that of B is less than ε, we would be indifferent between choosing either of the versions. **The cost function** is used to estimate whether the expected losses made by choosing A (over B) are below the threshold ε. This cost function comes from the estimation of the Bayes probability, and we are looking for a decision which will minimise this cost function [7]. It can be expressed as follows:

$$\int_0^1 \int_y^1 (y-x) \frac{x^{s_A}(1-x)^{f_A} y^{s_A}(1-y)^{f_B}}{B(s_A, f_A) B(s_A, f_B)} dxdy \leq \varepsilon \tag{3.2.1}$$

The cost function can be simplified as:

$$\frac{B(s_A+1, f_A)}{B(s_A, f_A)} H(s_A+1, f_A, s_B, f_B) - \frac{B(s_A+1, f_B)}{B(s_A, f_B)} H(s_A, f_A, s_B+1, f_B) \leq \varepsilon \tag{3.2.2}$$

ε is similar to the concept of probability threshold α, discussed in section 1. While performing A/B tests, to conclude that the difference between the conversion rates is significant, the p-value has to be less than α. Similarly, in the case of the Bayesian decision rule, the test can be stopped when the expected loss is less than ε. As with threshold α, $\varepsilon = 0.01$ by convention.

4 Multi-armed Bandits

The multi-armed bandit (MAB in short) is an alternative to the traditional A/B testing approach, it uses adaptive learning to choose the best version among many options. The name comes from imagining a gambler at a row of a slot machines (known as "one-armed" bandits), who wants to maximise his winnings. Every machine gives a random reward drawn from a this machine's probability distribution. There are two phases in the game: exploration and exploitation, since the gambler has to choose the right machines to play (**exploration phase**) and then concentrate on them (**exploitation phase**). He has to decide which machines to play, how many times top play each machine, in which order, and whether to continue with the current machine or try a new one.

- n - number of versions, not including A
- $p_{i,t}$ - proportion of traffic to be allocated to a version i at time t, $i \in \{B/C/n\}$
- $p_{A,t}$ - proportion of traffic to be allocated to A at time t

The A/B/C/n test is launched with the original version A and n versions. The traffic is split equally among all the variations. Once the test is launched, every 8 hours we check if there are more than 100 visits and 20 conversions on the winning variation. The allocation doesn't change in the case of low traffic.

To define the new allocation at moment t, we first perform an A/B/C/n test described in the first section. We obtain the standardised statistic Z_i for every pair ($version_A$, $version_i$), $i \in \{B/C/n\}$ and find the winning version. Let ε be a parameter that dictates the fraction of traffic to be allocated to a loosing version. We use an epsilon-decreasing strategy, which means that the MAB prioritises exploration at the beginning of the test and exploitation at the end. ε_t is dependent on the Z of the winning version:

$$\varepsilon_t = min\{0.1 + e^{-1.3|Z_{winning}|}, 1\} \tag{4.0.1}$$

$|Z_{winning}|$ is likely to increase with time as more data is gathered, as it increases, the ε_t decreases and less traffic is allocated to a loosing version.

For the loosing version, the proportion of traffic can be calculated the following way:

$$p_{loosing,t} = \varepsilon_t \frac{1 - p_{A,t}}{n} \tag{4.0.2}$$

For A and any other version i (except for the loosing one), if version A is winning:

$$\begin{aligned} p_{i,t} &= \varepsilon_t p_{i,t-1} \\ p_{A,t} &= p_{A,t-1} + (1 - \varepsilon_t)(1 - p_{A,t}) \end{aligned} \tag{4.0.3}$$

Otherwise:

$$\begin{aligned} p_{i,t} &= \varepsilon_t p_{i,t-1} + (1 - \varepsilon_t)\frac{1 - p_{A,t}}{n} \\ p_{A,t} &= p_{A,t-1} \end{aligned} \tag{4.0.4}$$

At $t = 0$, $\varepsilon_0 = 1$, which leads to highly explorative choices at the beginning of the test. The more the reliability of the winning version increases, the more the value of ε decreases, resulting in highly exploitative behaviour at the end of the test.

References

[1] *SHA-2 Wikipedia* https://en.wikipedia.org/wiki/SHA-2

[2] *Douglas C.Montgomery, George C.Runger*. (2014). Applied Statistics And Probability For Engineers.(6th ed.). John Wiley & Sons, inc.

[3] *Gary Shute*. (2016). https://www.d.umn.edu/~gshute/arch/improvements.xhtml

[4] *Karl Pearson*. (1900). On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. Philosophical Magazine. Series 5. 50 (302): 157–175

[5] *Hansheng Wang, Shein-Chung Chow*. (2007). Sample Size Calculation for Comparing Proportions. Wiley Encyclopaedia of clinical trials.

[6] *Evan Miller*. (2015). https://www.evanmiller.org/bayesian-ab-testing.html

[7] *Chris Stuccio*. (2014). https://www.chrisstucchio.com/blog/2014/bayesian_ab_decision_rule.html

[8] (2013). NIST/SEMATECH e-Handbook of Statistical Methods. https://doi.org/10.18434/M32189

[9] *Peter Auer, Nicola Cesa-Bianchi, Paul Fischer*. (2002). Finite-time Analysis of the Multiarmed Bandit Problem. Machine Learning, 47, 235–256, 2002