Statistics at Kameleoon

August 25, 2025

Contents

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. After that we explain how we chose to implement both CUPED and our way to run sequential tests. 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 differ 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 different than that of version A.

- $H_0 : x_B = x_A$
- $H_A : x_B \neq x_A$

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 5%, this means that five 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 $i \in \{B, C, D\}$:

- $H_0 : x_i = x_A$
- $H_A : x_i \neq x_A$

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

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

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 CUPED

Controlled-experiment Using Pre-Experiment Data (CUPED) was first introduced in a paper from Microsoft [4]. It is a variance reduction technique based on the **control variates method** [5]. The idea is to leverage information about a known variable to help reduce the error done when estimating an unknown one. We will quickly cover the theory before detailing how we apply this technique at Kameleoon.

Let's say our variable of interest is Y and we want to estimate $\mathbb{E}(Y) = \mu$. The idea is to introduce a new random variable X with known expectation $\mathbb{E}(X)$ in order to build

$$\hat{Y}_{control} = Y + \theta * (X - \mathbb{E}(X)) \tag{2.5.1}$$

We can show that $\hat{Y}_{control}$ is an unbiased estimator of μ independently of the choice of θ because

$$\mathbb{E}(\hat{Y}_{control}) = \mathbb{E}(Y) + \theta * (\mathbb{E}(X) - \mathbb{E}(\mathbb{E}(X)) = \mathbb{E}(Y) = \mu \tag{2.5.2}$$

And that its variance is

$$Var(\hat{Y}_{control}) = Var(Y + \theta * (X - \mathbb{E}(X))) = Var(Y) + \theta^2 Var(X) + 2\theta * Cov(Y, X) \tag{2.5.3}$$

One can then show that this variance is minimal when $\theta = -\frac{Cov(Y,X)}{Var(X)}$ and with this θ we get

$$Var(\hat{Y}_{control}) = Var(Y) - \frac{Cov(Y, X)^2}{Var(X)} = (1 - \rho^2)Var(X) \tag{2.5.4}$$

Where $\rho = Corr(Y, X)$ is the correlation between Y and X. Hence the greater this coefficient is in absolute value, the more the final variance will be reduced.

So the difficulty lies in choosing the right control variate for which we know its expectation, and which is highly correlated (positively or negatively) to the outcome. And in a test framework, we have to know the expectation of the variable of interest for the visitors within both the control and the variation group. This is where pre-experiment data come in handy, because before the experiment start they are actually equal (i.e. there is no difference between the groups prior to the start of the experiment).

The natural choice for a variable correlated with the outcome variable, which will be computed on pre-experiment data is actually the outcome variable itself. **Empirical results**[4] have shown this to be the best choice for control variates. In this great paper from Microsoft, researchers also warn about the following points:

- In general, the optimal pre-experiment window is 1–2 weeks. A shorter window doesn't capture enough variance and a longer window captures noise.
- For longer experiments, a longer pre-experiment window is needed to ensure that the same users are observed both during and prior to the experiment.
- The covariate X has to be evenly distributed between treatment and control. If it is impacted by the treatment, CUPED becomes invalid.
- The CUPED method only removes variance that can be accounted for linearly.

Having all that in mind we chose as covariate the same objective on a time window of two weeks prior to start of the experiment. After using pre-experiment data to build our new CUPED corrected metric, we run our statistical test on this new metric.

2.6 Sequential Testing

Sequential testing is a statistical approach used to monitor the progress of A/B tests over time and make data-driven decisions on when to stop a test while keeping the risk of a type I error under control. Sequential testing is the main answer to **"peeking"**[6] which is the most common source of unwanted risk inflation in AB testing.

Indeed the classical frequentist test is built on the assumption that the test is run once we have collected all our samples and not before. But if we take a sneak peek at results during the data collection phase and we stop the test if we find a reliable result, then the resulting false positive rate is way higher than the one we expected.

Kameleoon has chosen to use a Gaussian mixture asymptotic confidence sequence for its sequential tests, for a great review of the different framework of sequential tests see this **blog**[7] made by Spotify. This sequence was proposed in **Waudby-Smith et al. (2023)** [8]. It is based on always valid inference which was proposed by **Howard et al. (2021)** [9], as stated in their paper, it provides a confidence sequence which has the following properties all very necessary to our field of work:

- The coverage guarantee holds for all sample sizes and even without distributional assumption
- No previous knowledge of the stopping rule is required
- No bound on sample size
- The width of the confidence sequence converges towards zero

Those properties are perfect for our use case. Indeed the customer running its AB test is able to choose to keep his experiment running to gather more data or to stop it at any point in time according to any rule without inflating his risk. Of course there is always a price to pay, the resulting test will have less power than the corresponding fixed sample test.

The sequential confidence interval for the improvement rate is given by

$$[IR \pm \hat{\sigma}_n * \sqrt{\frac{\phi + n}{n} * \log(\frac{\phi + n}{\phi * \alpha^2})}] \tag{2.6.1}$$

with n being the total number of visitors targeted by the test, for example for an AB test we have $n = n_A + n_B$, $\hat{\sigma}_n$ the mean standard error after collecting n samples and α a given level of significance. We use 5% by convention.

The trick of this method is that this sequence is obtained by injecting a mixing distribution inside the Sequential Probability Ratio Test (SPRT) which was introduced by **Wald, A. (1945)** [10]. Specifically here we use a gaussian mixture but we have a lot of liberty in the choice of the parameters of this distribution, as shown by the parameter ϕ in the equation above. Howard et al. (2021) include in their article a very insightful and extensive discussion on the choice of this parameter (section 3.5). As they state "All uniform boundaries involve a tradeoff of tightness at different intrinsic times: making a bound tighter for some range of times requires making it looser at other times". They show that it is possible to optimize the interval boundaries so they will be tightest for a given number of observation. They also show that this parameter doesn't need to be very precise as long as it is conservative. Indeed the price to pay is way steeper in case of overestimation than under estimation.

We chose to optimize for the time $n_{optimal} = 20000$ as it is a number of visitors which makes sense for our wide range of customers. In order to compute the corresponding value of ϕ we use the following equality (corresponding to equation (21) in [9]):

$$\frac{n_{optimal}}{\phi} = -W_{-1}(-\frac{\alpha^2}{e}) - 1 \tag{2.6.2}$$

$W_{-1}(x)$ being the lower branch of the Lambert W function. We can then use the following inequality to approximate the Lambert function:

$$\log(x) - \log(\log(x)) < W(x) < \log(x) - \log(\log(x)) - \log(1 - \frac{\log(\log(x))}{\log(x)}) \tag{2.6.3}$$

This allows us to express ϕ according to $n_{optimal}$ the number of observations for which we optimize as follow:

$$\frac{n_{optimal}}{\phi} \sim \log(\log(\frac{e}{\alpha^2})) - 2 * \log(\alpha) \tag{2.6.4}$$

This yields a value of $\phi = 2520$.

2.7 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.

2.8 Ratio metric

Kameleoon also allows you to build and test metrics which are defined as a ratio of two other metrics. To formalize it, we define a ratio metric *ratio* like this: $ratio = \frac{X}{Y}$. We now want to build a test to be able to tell if the average of this ratio is improved in a variation compared to another one. In order to do so we have to be able to estimate the mean and the variance of this ratio. Working with a ratio of random variable is tricky and we need to assume some of the properties of the denominator Y to be able to approximate the ratio distribution.[11] The main assumption to fulfill is that the distribution of Y should be far enough from 0. Hence its mean should be strictly positive and its standard deviation should be small enough relatively to its mean.[12]

If the variance of Y, σ_Y^2 is large then the following happens:

- The variance of the ratio increases significantly
- The normal approximation of the ratio may not hold
- The ratio distribution could be skewed or heavy-tailed instead of normal

In our case, we are working with sample means, hence as the sample grow we usually fall under the necessary assumption to be able to apply the delta method, namely that the asymptotic distribution of the sample mean of the ratio is gaussian. Then we can apply the delta method to our ratio as we have $ratio = \frac{X}{Y} = g(X, Y)$ with $g : x, y \to \frac{x}{y}$ being differentiable in a neighborhood of $(\bar{x}, \bar{y})$ since the average of the sample mean of Y is strictly positive.

Computing the first-order Taylor expansion around (μ_X, μ_Y):

$$g(\bar{X},\bar{Y}) \approx g(\mu_X,\mu_Y) + \left(\frac{\partial g}{\partial x}\Big|_{(\mu_X,\mu_Y)}\right)(\bar{X}-\mu_X) + \left(\frac{\partial g}{\partial y}\Big|_{(\mu_X,\mu_Y)}\right)(\bar{Y}-\mu_Y).$$

Computing derivatives:

$$\frac{\partial g}{\partial x} = \frac{1}{y}, \quad \frac{\partial g}{\partial y} = -\frac{x}{y^2}.$$

Evaluating at (μ_X, μ_Y):

$$\frac{\partial g}{\partial x}\Big|_{(\mu_X,\mu_Y)} = \frac{1}{\mu_Y}, \quad \frac{\partial g}{\partial y}\Big|_{(\mu_X,\mu_Y)} = -\frac{\mu_X}{\mu_Y^2}.$$

Thus, the linear approximation is:

$$\bar{ratio} \approx \frac{\mu_X}{\mu_Y} + \frac{1}{\mu_Y}(\bar{X}-\mu_X) - \frac{\mu_X}{\mu_Y^2}(\bar{Y}-\mu_Y).$$

Thus, for large n:

$$\bar{ratio} \approx \mathcal{N}\left(\frac{\mu_X}{\mu_Y}, \frac{\sigma_X^2}{n\mu_Y^2} + \frac{\mu_X^2\sigma_Y^2}{n\mu_Y^4} - 2\frac{\mu_X}{\mu_Y^3}\frac{\rho\sigma_X\sigma_Y}{n}\right). \tag{2.8.1}$$

With this we can then follow the same path to test the improvement rate on the ratio of variation A against variation B.

2.9 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$_\text{A}$, version$_\text{i}$), version$_\text{i}\in${version$_\text{B}$, 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:

$$x_i = x_A(1+IR) \tag{2.9.1}$$

The sample size for version A (n_A) and for the version i (n_i) then can be approximated by [13]:

$$n_A = k_i n_i \tag{2.9.2}$$

$$n_i = \frac{(z_{\alpha/2} + z_\beta)^2}{(x_i - x_A)^2}[x_A(1 - x_A)/k + x_i(1 - x_i)] \tag{2.9.3}$$

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

$$d_i = \lceil \frac{n_0 + n_i}{n_{\text{daily}}} \rceil \tag{2.9.4}$$

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.

$$d = max\{d_1, ..., d_v\} \tag{2.9.5}$$

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

$$d_{\text{left}} = d - d_{\text{passed}} \tag{2.9.6}$$

2.10 Continuous metrics

In the general case where the variable on which we are testing is not a Bernoulli variable but a continuous one, we do not have a closed-form expression for the resulting variance. Hence we have to estimate it from the data we observed using the following unbiased estimator:

$$\sigma^2 = \frac{\sum_i^n (x_i - \bar{x})^2}{n - 1} \tag{2.10.1}$$

This allows us to estimate the variance inside each variation A and B. For continuous metrics, we directly compute a test statistic for the improvement rate introduced in equation 2.4.1. We have to use the delta method in order to estimate the variance of the improvement rate since it is a function of two variables. This variance σ^2 is expressed as:

$$\sigma^2 = \frac{1}{n_A + n_B} * \frac{x_B^2}{x_A^2} * (\frac{\sigma_A}{x_A^2} + \frac{\sigma_B}{x_B^2}) \tag{2.10.2}$$

We can then inject the variance of the improvement rate to compute a confidence interval and the test p-value as described in the previous sections.

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 $x_B > x_A$ [14]. 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 [15]. 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.

3.3 Continuous metrics

In order to perform a bayesian statistical hypothesis test in the case of a continuous metric, we still have to start from a prior probability distribution that we will update with the data we observe. When the metric is continuous, instead of building a distribution on the conversion rate of each variant, we focus directly on the improvement rate and we start with a prior probability distribution on the improvement rate that we update to reach the posterior probability distribution.

We chose to use a normal prior for the improvement rate, centered around zero and with a standard deviation of 0.1. This means that in average we expect that the variation will not show any improvement over the reference on the mean of the observed metric, and that ninety-nine percents of the improvements observed range between plus or minus thirty percents. This value was derived after studying more than a year of empirical data. One of the advantage of using a normal prior is that it is a conjugate prior [16], hence we can derive a close-form expression for the posterior which will also be normal.

Indeed, for a prior distribution $N(\mu_0, \sigma_0^2)$, after collecting data with mean μ and variance σ^2, the posterior parameters are the following:

$$\mu_{posterior} = \frac{(\frac{\mu_0}{\sigma_0^2} + \frac{\mu}{\sigma^2})}{\frac{1}{\sigma_0^2} + \frac{1}{\sigma^2}}, \sigma^2_{posterior} = (\frac{1}{\sigma_0^2} + \frac{1}{\sigma^2})^{-1} \tag{3.3.1}$$

This actually simplifies a lot after we plug in $\mu_0 = 0$ and $\sigma_0 = 0.1$. Once we have a closed-from expression for the posterior distribution, we can use it to compute a credible interval at a given level α as well as the probability for the improvement rate to be greater than 0. The improvement rate mean was introduced in 2.4.1, in order to compute the variance of the improvement rate we have to leverage the delta method as described in 2.10.2.

4 Outlier Handling

We chose to use Winsorization in order to handle outlier values at Kameleoon. Winsorization is a statistical technique used to limit extreme values in data to reduce the impact of outliers, by using percentiles of your data. Outliers are data points that significantly differ from other observations and can skew the results of your AB tests. By Winsorizing your data, you can ensure that your results are more robust and reliable.

In AB testing, we compare two or more variations to determine which performs better. Outliers can distort the true performance of these variations, leading to misleading conclusions, mainly due to "whale users". By applying Winsorization, we mitigate the effect of these extreme values, providing you with more accurate and actionable insights. Winsorization is particularly useful when:

- Your data contains extreme values that are not errors but are still significantly different from other observations.
- You are looking for a simple and effective method to handle outliers without resorting to more complex techniques.
- You need to maintain a balance between data integrity and managing outliers effectively.

The Winsorization algorithm is fairly simple, it takes as input two quantiles: q_{lower} and q_{upper}. Then it will compute the values corresponding to this quantiles. At Kameleoon we compute those values on the two past weeks of collected data and update them daily to avoid data drifts. Then when checking your experiment results, all the values below q_{lower} are replaced by q_{lower} and all the values greater than q_{upper} are replaced by q_{upper}.

5 Bandits Algorithms

5.1 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 hour 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:

$$\varepsilon_t = min\{0.1 + e^{-1.3|Z_{winning}|}, 1\} \tag{5.1.1}$$

$|Z_{winning}|$ 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:

$$p_{loosing,t} = \varepsilon_t \frac{1 - p_{A,t}}{n} \tag{5.1.2}$$

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

$$\begin{aligned} p_{i,t} &= \varepsilon_t p_{i,t-1} \\ p_{A,t} &= p_{A,t-1} + (1 - \varepsilon_t)(1 - p_{A,t}) \end{aligned} \tag{5.1.3}$$

Otherwise:

$$\begin{aligned} p_{i,t} &= \varepsilon_t p_{i,t-1} + (1 - \varepsilon_t)\frac{1 - p_{A,t}}{n} \\ p_{A,t} &= p_{A,t-1} \end{aligned} \tag{5.1.4}$$

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.

5.2 Contextual Bandits

The Contextual Bandit (C-MAB usually) is a personalized version of the MAB where the assignation of a visitor is no longer assigned randomly to a variation based on the deviations valid at time T. With the contextual bandit a visitor is assigned to the variation which maximises his probability to convert the objective based on his characteristics.

Indeed, at each time step t, a user arrives with a context vector $x_t \in \mathbb{R}^d$. The algorithm must choose an action $a_t \in \{1, \ldots, K\}$, corresponding to one of K possible variations. After making this choice, the algorithm observes a reward $r_t(a_t)$, and uses it to update its decision model.

At Kameleoon we implemented an approach based on neural bandits and on the following algorithm which was introduced in the paper "A Contextual-Bandit Approach to Personalized News Article Recommendation", where they use disjoint Linear Models. As they put it, *This model is called disjoint since the parameters are not shared among different arms.* [19].

Algorithm Steps

1. **Observe context:** x_t
2. **Estimate expected rewards:**

   $$\hat{r}_t(a) = f_a(x_t)$$

   where f_a is a model predicting the expected reward for action a given context x_t.
3. **Choose action using uncertainty-aware exploration:**

   $$a_t = \arg\max_a \left(f_a(x_t) + \alpha \cdot UCB_a(x_t) \right)$$

   where $UCB_a(x_t)$ represents a model-dependent uncertainty measure, and $\alpha \geq 0$ controls the exploration-exploitation balance.
4. **Observe reward:** $r_t(a_t)$
5. **Update model:** Incorporate $(x_t, a_t, r_t(a_t))$ into the learning dataset.

The goal is to minimize cumulative regret over time:

$$R_T = \sum_{t=1}^{T} \left(r_t(a^*(x_t)) - r_t(a_t) \right)$$

where $a^*(x_t)$ is the optimal action for context x_t.
Two main things to note here, we have decided to use neural networks as models predicting the expected reward for action a given context x_t. And regarding the parameter α which serves to define the balance between exploration and exploitation. We have implemented the contextual bandit as a two step process, with a learning / training phase and an exploitation phase. Our neural networks first train then we uses them to assign variations so we have set $\alpha = 0$. This is something we monitor and it may evolve in the future.
The main use cases for contextual bandits are the following

- Product recommendations based on user attributes
- Real-time UX adaptations across traffic segments
- Dynamic pricing or offer strategies

Its advantages is the personalization in real-time and the ability to learn from user interactions, at the cost of a possible sub-optimal performance during the learning phase of the models.

Context Used

Inside our neural network we use a mix a static features and dynamic ones in order to estimate the expected reward for each variation. Indeed our model is based on three main categories of features, the ones based on previous sessions done by the visitor, the ones based on static information collected when the visitor lands on the site and finally the ones built from the user browsing behavior up to the moment when we assign him a variation.
Here is a non exhaustive catalogue of our features:

1. **Previous Sessions Features**
    - Number of sessions
    - Average time between sessions
    - Number of visited pages
    - Number of clicks
    - ...
2. **Static Features**
    - Browser
    - OS
    - Landing page
    - Referrer

- Time since last session
- ...

3. **Dynamic Features**
    - Number of pages visited
    - Number of clicks
    - Time spent on the website
    - ...

Note that you can also enrich our models with your own data by setting up custom data and marking them learnable. Then our machine learning models will use them to improve their estimations.

References

[1] *Wikipedia*. SHA-2. 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] *Alex Deng, Ya Xu, Ron Kohavi, Toby Walker*. Improving the Sensitivity of Online Controlled Experiments by Utilizing Pre-Experiment Data. https://exp-platform.com/Documents/2013-02-CUPED-ImprovingSensitivityOfControlledExperiments.pdf

[5] *Wikipedia*. Control Variates. https://en.wikipedia.org/wiki/Control_variates

[6] *Analytics Toolkit*. What does "Peeking" mean? https://www.analytics-toolkit.com/glossary/peeking/

[7] *Spotify R&D*. Choosing a Sequential Testing Framework - Comparisons and Discussions https://engineering.atspotify.com/2023/03/choosing-sequential-testing-framework-comparisons-and-discussions/

[8] *Ian Waudby-Smith, David Arbour, Ritwik Sinha, Edward H. Kennedy, and Aaditya Ramdas*. Time-uniform central limit theory and asymptotic confidence sequences. https://arxiv.org/pdf/2103.06476v7.pdf

[9] *Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, Jasjeet Sekhon*. (2022). Time-uniform, nonparametric, nonasymptotic confidence sequences. https://arxiv.org/pdf/1810.08240.pdf

[10] *Wald, A. (1945)*, 'Sequential tests of statistical hypotheses. https://doi.org/10.1214/aoms/1177731118.

[11] *Wikipedia* Ratio distribution. https://en.wikipedia.org/wiki/Ratio_distribution#Exact_correlated_noncentral_normal_ratio

[12] *Díaz-Francés, Eloisa and Rubio, F.* (2013). On the existence of a normal approximation to the distribution of the ratio of two independent normal random variables. https://www.researchgate.net/publication/257406150_On_the_existence_of_a_normal_approximation_to_the_distribution_of_the_ratio_of_two_independent_normal_random_variables

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

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

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

[16] *Wikipedia*. Conjugate prior. https://en.wikipedia.org/wiki/Conjugate_prior

[17] (2013). NIST/SEMATECH e-Handbook of Statistical Methods. https://doi.org/10.18434/M32189

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

[19] *Lihong Li, Wei Chu, John Langford, Robert E. Schapire*. (2012). A Contextual-Bandit Approach to Personalized News Article Recommendation. https://arxiv.org/pdf/1003.0146

[20] *Dongruo Zhou, Lihong Li, Quanquan Gu* Neural Contextual Bandits with UCB-based Exploration. https://proceedings.mlr.press/v119/zhou20a/zhou20a.pdf