



HAL
open science

Regression tree for bandits models in A/B testing

Emmanuelle Claeys, Pierre Gancarski, Myriam Maumy-Bertrand, Hubert Wassner

► **To cite this version:**

Emmanuelle Claeys, Pierre Gancarski, Myriam Maumy-Bertrand, Hubert Wassner. Regression tree for bandits models in A/B testing. The Sixteenth International Symposium on Intelligent Data Analysis (IDA 2017), 2017. hal-02572444

HAL Id: hal-02572444

<https://hal.science/hal-02572444>

Submitted on 13 May 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Regression tree for bandits models in A/B testing

Emmanuelle Claeys^{1,2,3}, Pierre Gançarski¹ Myriam Maumy-Bertrand², and
Hubert Wassner³

¹ Strasbourg University, CNRS, ICUBE
300 Bd Sébastien Brant, 67400 Illkirch-Graffenstaden, FRANCE

² Strasbourg University, CNRS, IRMA
7 Rue René Descartes, 67000 Strasbourg, FRANCE

³ AB TASTY
3 Impasse de la Planchette, 75003 Paris, FRANCE
`claeys@unistra.fr`

Abstract. In the context of Web A/B testing, dynamic assignment of traffic aims to promote the best variation (A or B) as quickly as possible. However, dynamic assignment is difficult to use when the difference between A and B affects the visitor differently according to his / her personal characteristics and his / her history (number of visits, navigation on the website ...). In this paper, we propose a dynamic assignment strategy based on a visitor segmentation determined automatically from the visitors navigation and characteristics.

Keywords: A/B Test, best arm identification, bandit models, regression tree, time clustering, topic modeling

1 Introduction

The quality of a website (usability, iconography, customer experience or visitor's navigation) is fundamental for an e-merchant (i.e., corresponding to business based on the web). Developing and adapting the website to improve it is often essential to maintain or increase sales. However, changes can negatively affect sales. To avoid this negative effect, e-merchants must carefully observe the visitor responses to the website changes to evaluate the relevance of these changes. On a sample of e-visitors, an A/B test consists of comparing several variations of the same element. Usually two variants are available, which are denoted by A (e.g., the original web-page) and B (a variation to be tested). During an A/B test, a ratio of visitors only sees a variation while the other visitors only see the other one. The effectiveness of each variation is evaluated according to this e-merchant's objective which is generally based on the conversion rate of the variation (number of clicks on an element, add of items to basket, number of the transactions...). After the test, for the *exploitation phase*, the e-merchant can choose to implement the variation B , to keep the original A or to implement a custom strategy. To implement the observation phase, it is necessary to define the strategy of assignment of a variation to a new visitor, i.e., the page the visitor

will exclusively see. Indeed, during a given A/B test, a variant is definitively affected to a visitor even if he/she comes back again. Consequently, a visitor only sees one of the variants.

As a consequence of this assignment, it is impossible to know the visitor behaviour if he/she have been assigned to the other variant. Thus, comparing the effectiveness of each of the two pages according to the visitors' behaviours is very hard because the tested populations are totally disjointed.

Strategies based on statistical tests to improve the assignment exist but they are static: the assignment ratio between A and B can not change during all the A/B test. However, if identification of the best variation is crucial for the e-merchant, the priority remains the profitability of the website. Thus if, during the test, the new variation leads to very positive effect or, at the opposite, to very negative effect, the e-merchant would want to reduce the observation phase and increase the exploitation phase (for example, by using the best variation immediately). A way to circumvent this problem, is to change the ratio of assignment during the test: the better a variation, the higher the ratio of assignment to this variation, and reciprocally. During the exploration phase used to establish statistical confidences, a dynamic assignment strategy increases/decreases the ratio of visitors affected to the variations according to their performances until identifying the best variation. Then all visitors are assigned to the best variation and the test stops.

Almost all approaches which implement such strategies are based on the bandit model and give good results. Bandit methods automatically adapt the ratio of assignment and try to find the best variation as quickly as possible (exploration phase) in order to assign all the visitors to this best solution as soon as it is determined (exploitation phase). This approach tries to solve the exploration-exploitation dilemma introduced by Robbins et al. [11]. It can also be viewed as a reinforcement learning method [2]. Nevertheless, when the population is not homogeneous [14], the use of only one bandit is often not appropriate (see Section 2). In this paper, we present a new approach consisting of two steps (Section 3). The first one searches the most homogeneous subgroups into the visitors according to their navigation on the website (navigation, interest ...) and their own characteristics (e.g., localisation, navigator used...) using clustering algorithms and non-parametric regression trees. The second step uses a specific assignment strategy to each of them (i.e., a bandit algorithm for each group) to actually make the test. We also present the first experiments carried out on an existing fashion website (Section 4). Then, we conclude and present some perspectives (Section 5).

2 Dynamic assignement using bandit model

The problem is to assign a new visitor to a page variation. But at the beginning of the A/B test, the a priori information about the conversion rate associated with each page is unknown. Thus, the conversion rates (namely, the expected reward) of the two pages must be determined empirically. To do that, the varia-

tion A (resp. B) is associated to the arm Arm_1 (resp. Arm_2) of a bandit. Then, an iteration i of the bandit algorithm consists of assigning the i -th visitor to a variation according to the respective expected conversion rate of the two arms. This expectation is then updated according to the visitor behaviour (conversion or not, amount of the transaction, ...). Initially, both expectations are unknown but it is supposed that the gain of each of the arms follows a probability distribution according to the normal law [1]. The aim of a bandit algorithm is then to find empirically these two distributions and to decide the best one while limiting regret (i.e., the cost of a bad choice) due to the discovery of the best arm (*exploration phase*). If one page is better than the other, it is better to assign visitors to it as quickly as possible (*exploitation phase*) while avoiding premature convergence. This problem is known as the "exploration vs exploitation dilemma" [11].

Let ν_i^j be the distribution of probability associated with Arm^j at the iteration $i, i \in \{1, \dots, n\}$ and μ_i^j the a priori expectation associated with arm j ($j \in \{1, 2\}$). Let Z_i^j be the reward (conversion or not, amount of the transaction, ...) associated with the i -th visit obtained by Arm_j . We assume that $Z_i^j \in \mathcal{L}^1$.

Let Arm^* be the best arm to be found and μ^* its expectation. By definition $\mu^* = \max(\mu_n^1, \mu_n^2)$.

To identify Arm^* , the algorithm has to explore different options (exploration step).

The goal of bandits models is to minimize cumulative regret during testing. Cumulative regret grows each time the algorithm chooses a sub-optimal arm:

$$\mathcal{R}_i(\nu) = i\mu^* - \mathbb{E}_\nu\left[\sum_{k=1}^i Z_k\right]. \quad (1)$$

An effectiveness strategy limits regret as [7]:

$$\inf \lim_{t \rightarrow +\infty} \frac{\mathcal{R}_i(\nu)}{\log(t)} \geq \sum_{Arm: \mu^{Arm} < \mu^*} \frac{(\mu^* - \mu^{Arm})}{\mathbf{KL}(\nu^{Arm}, \nu^*)} \quad (2)$$

where μ^{Arm} is a sub-optimal arm expectation, ν^{Arm} is a sub-optimal arm distribution of probability and \mathbf{KL} is the Kullback-Leibler divergence between two distributions: the bigger the difference between the optimal and sub-optimal distribution is, the quicker the process converges. Conversely, the distance \mathbf{KL} is unusable if the two distributions are very similar.

In summary, bandits models can handle the "exploration-exploitation dilemma" thought dynamic assignment. Nevertheless, if too many visitors are not impacted by the test (i.e., impacted persons are too few to influence initial distribution), the bandit model will not be able to find the best variation.

To solve this problem, some variants of bandit model exists. For instance, the contextual bandits method uses a context vector with d dimensions (visitor's context) to make the best decision.

For instance :

- **LinUCB** (Upper Confidence Bound): assumes a linear dependency (to be identified) between the expected reward and the context vector [8]
- **UCBogram algorithm**: assumes a non-linear dependency (to be identified) between the expected reward and the context vector.
- **NeuralBandit algorithm**: a neural network approach where network learns the context and the associated reward.
- **KernelUCB algorithm**: a nonlinear version of **LinUCB** using a kernel matrix. Used for on-line learning.

(Note that this list is not exhaustive; the reader can find more detailed information in Galichet et al. [4])

In fact, these methods require a prior knowledge on the context, on the relevance of the attributes and on the correlation between them. Unfortunately, in most cases, the e-merchant can not provide such knowledge making them difficult to use.

Our idea, and the main purpose of our work, is to find groups (also called segments) in which the difference between the two distributions is significant (i.e., $\mathbf{KL}(\nu^{Arm}, \nu^*) > 0$) and in which people have the same behaviour (i.e., the individual conversion rate is compatible with the both distributions). So, for each group, a dynamic assignment can be produced by a bandit as the conditions to use it are verified.

3 A new A/B test procedure

To solve issues introduced in the previous section, we propose a new A/B test approach which combines the two steps:

1. a *preliminary analysis* (offline) to identify visitors subgroups (or segments) according to their behaviours during their visits, and to determinate if these subgroups have been differently impacted by the modification of the page (see Section 3.1).
2. a *analysis step* (online) corresponding to the actual test which independently uses a bandit algorithm to each identified subgroup (see Section 3.2).

3.1 Step 1: Preliminary analysis

To create subgroup of visitor which have been similarly impacted by the modification, we assume that visitors of a such subgroup had have similar navigation between the arrival of the visitor on the website and a potential conversion. As we dispose of historical data describing all the pages browsed by the visitors before the test during a given period (of generally two or three weeks), we can use this temporal information to extract such subgroups of visitor. The preliminary analysis we propose, is composed of three steps

1. creation of two clusterings of visitor using historic navigation data.
2. extraction of the topics of interest to improve the visitor profile.
3. patterns searching in visitor profiles to highlight common behaviour.

Creation of different clusterings of visitors To categorize the visitors, two clustering are created using browsing history. The first one is based on the vector $V_{presence}$ which indicates if a visitor visits the website within time intervals (usually, each day), the second on the vector V_{pages} which gives the number of pages browsed by a visitor in each time interval.

Generally, a website is built according to the customer experience. So, it can integrate an implicit expected navigation from the home page to the tested page. Nevertheless, the visitor can start browsing from any of the pages on this path, or even from the tested one, for instance using a search engine. In fact, the browsing history can have different lengths, and can possibly be empty. Thus, in distance-based clustering algorithm, the use of Euclidean distance are not available.

As these two vectors can be seen as time series assigned to each visitor, we can use a classical time series clustering method based on the well-known partitioning algorithm K-MEANS and on the Dynamic Time Warping (D.T.W.) [10]. Centroids calculation are made by the D.T.W. Barycenter Averaging (D.B.A.) method [10], which is an iterative method to calculate clusters' average. This approach is known to be very effective although sensitive to noise and to extreme values. In our experiment, we use the a implementation provided by the **R** language [12].

Taking topic of interest into consideration In order to improve the quality of our visitors profiling, a topic of interest(s) is assigned to each visitor according to the historic navigation (1) by looking for the different key words of browsed pages, (2) by extracting topics from these keywords, (3) by associating a such topic to all the browsed page and (3) by associating all the topics associated to the pages seen by the visitor during his/her navigation. Note that a topic can be associated to several pages. The most used topic extraction method for discrete or symbolic data where topics are not correlated is L.D.A. (Latent Dirichlet Allocation) [5] which is a combination of Bayesian statistical models.

Patterns searching in visitor profiles The purpose of this step is to extract subgroups (or segments) of visitors, before the test (only related to the original variation, the variation B not yet being created) with similar conversion rates. Indeed it is assumed that (1) the test can only affect one or more subgroups and (2) visitors from a same subgroup have same behaviour on the both variations. So we can really compare the two distributions associated to the both variations because the homogeneously of the population.

We apply a non-parametric regression tree algorithm (e.g., conditional inference trees, *CTREE* algorithm, [6]) for discover these subgroups. This regression tree tests statistically significant difference in the conversion rate (Pearson chi-square test, with a 95% confidence level) for each visitor profile (composed of the cluster the visitor belongs to and the and his/her topics series). The built tree can be used to predict the conversion rate for a visitors from his/her profile (see Section 4).

The tree is recursively generated by dividing the population of a node into two subsets at each iteration. The node to divide is chosen through a test of co-variances [9] and the most discriminant variable is selected for splitting [13].

To identify the attributes (clusters vs topics) that better characterise the groups, the tree performs different tests such as the Spearman test, the Wilcoxon-Mann-Whitney test or the Kruskal-Wallis test but also permutation tests based on ANOVA statistics or correlation coefficients (in our experiments, with discrete explanatory classification variables, we use a Kruskal-Wallis test). We compared, using the Kruskal-Wallis non-parametric method, all the combinations of clusters to find the more discriminant ones (i.e., the combination which give the most different ratios of conversion). Note that in the **R** language implementation, the *CTREE* algorithm calculate multiple comparisons (with a Bonferroni correction due to multiple comparisons) [3] to choose the most appropriate test.

Recall that our objective is to use this subgroup to better variation assignment by associating a bandit to each subgroup corresponding to a leaf of this tree.

3.2 Step 2: Analysis (A/B test)

To assign a new visitor during the analysis step (online), all information about his/her navigation from the website homepage to the page to be tested are registered. As presented in Section 3.1, the two vectors corresponding to the visit are then calculated and the visitor is affected to the nearest cluster using D.T.W.

Efficient clustering incremental methods do not exist. So, it is difficult to improve the clusters during the test phase. We decide to build them before through a training period.

The topic series is also affected to the profile. Then the visitor is also affected to a subgroup using the regression tree. As introduced previously, each group extracted by the regression tree contains visitors having similar behaviour. Then we can associate to each leaf of the tree, a bandit occurrence. Then the A/B test itself can start.

4 Experiments

4.1 The data

We have applied our method on a dataset based on an A/B^4 test without dynamic assignment (60% of global traffic sent to A and 40% to B) realised from 11/07/16 to 25/07/16 for a fashion e-commerce websites. We selected visitors arriving before 15/07/16 (midnight) to build our pre-analysis (*train_set*), and visitors arrived after 15/07/16 to test A/B itself (*exp_set*).

The reward was a binary value ⁵. The reward function follows a Bernoulli distribution whose parameters are estimated over time.

⁴ Variation B consisted of changing the link of a "return" button.

⁵ 0: no purchase or 1: purchase

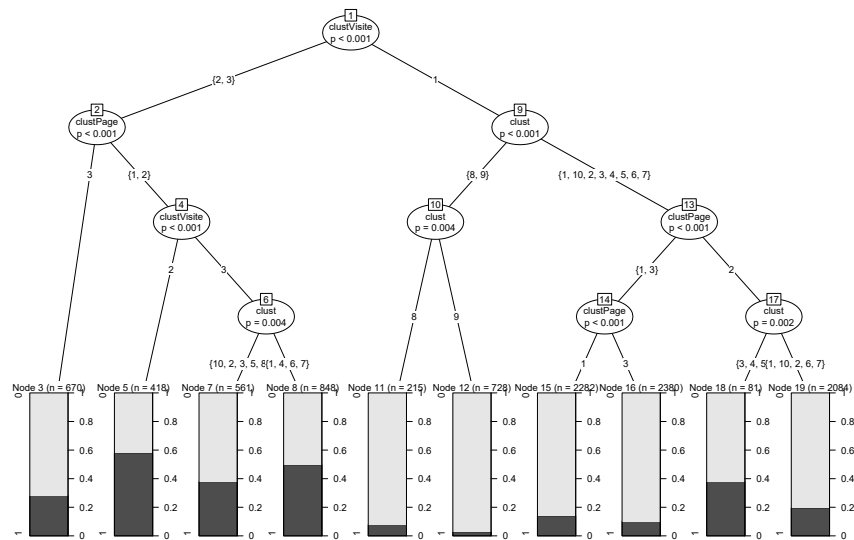
4.2 The test

According to the two step described in Section 2, we produced the clustering $clustPage$ (resp. $clustVisite$) based on the vector $V_{presence}$ (resp. V_{pages}). For technical reasons, we had only used the topic (called $clust$ in the following) corresponding at the last page browsed by the visitor rather all topic encountered during the navigation.

Then, we generated a regression tree (Figure 1) using exclusively the visitors from $train_set$ who have seen the original version A (Step 1). The preliminary analysis identified 10 distinct groups

For instance, "Node3" corresponds to visitors belonging to $clustPage\#3$ and to $clustVisite\#2$ or $clustVisite\#3$.

For Step 2 corresponding to the A/B test itself, we affected a bandit at each group corresponding to a leaf in the regression tree. Then we apply our dynamic assignment strategy for all the "new" visitors from exp_set .



The attributes used to split the node and the confidence are given in the node itself
A box represents the average conversion rate of the population

Fig. 1. Regression tree of $train_set$

4.3 Discussion

To evaluate performances of our method, we compare it with

- a same bandit model, but applied to all the visitors (without group) referred here as **Binomial Bandit**.
- a popular contextual bandit algorithm refereed here as **LinUCB**

Two aspects are observed in particular:

1. Cumulative reward and ratio of cumulative rewards to visitors.
2. Probability to identify the best variation.

Cumulative rewards With our approach, cumulative rewards is 1457 conversions while Binomial Bandit gets only 1252 ones. Table 1 details the best variation retained for each group, the cumulative reward obtained, the number of visitors associated to the group and finally the confidence on the choice of the optimal variation. Note that no visitor has been assigned to bandit #15.

Table 1. Best variation and cumulative reward by group

Bandit	Node 3	Node 5	Node 7	Node 8	Node 11	Node 12	Node 15	Node 16	Node 18	Node 19	TOTAL
Winner	A	B	A	B	A	A	X	A	A	A	X
Rewards	573	50	39	51	10	9	0	628	6	91	1457
Visitors	1878	99	131	181	310	475	0	7380	105	609	11168
Proba	0.66	0.74	0.88	0.63	0.52	0.72	0	0.96	0.73	0.91	X

Confidence on the best variation Our method finds 6 optimal variations with a probability greater than 70% and generally before the 1000-th visitors while the Binomial Bandit requires a least 4000 visitors for the same confidence. Five bandits from them went into the exploitation phase before only one thousand of visitors.

To perform our experiment, we iteratively select a visitor. We apply the algorithm and if it decides to assign at this visitor to variation (V_u) which differs to that given by the dataset, we search in the database, a visitor who presents similar characteristics but who is assigned to the variation (V_u) in the dataset. If it fails, we randomly choose a visitor assigned to the variation (V_u).

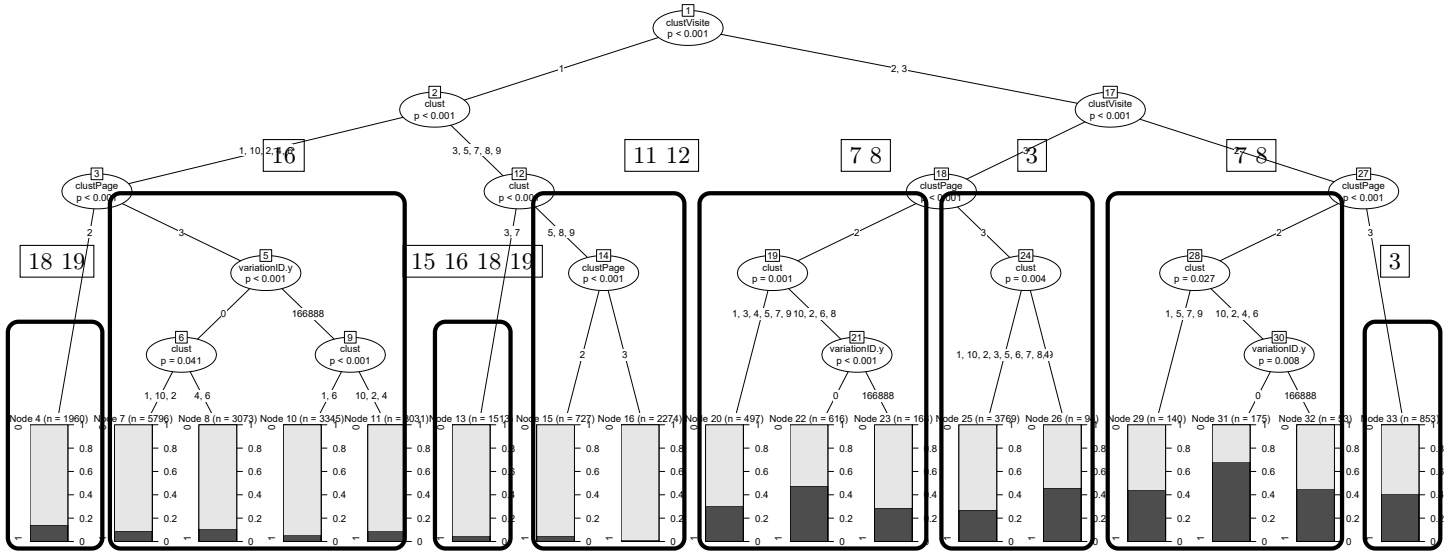
Moreover, as LinUCB is not totally deterministic, we have carried out an hundred runs on the algorithm. In the following we only report the mean of the cumulative rewards obtained by these runs.

Table 2 gives the performances of the three algorithms. It shows that LinUCB gets around 1519 (standard deviation of 49) cumulative rewards. Over 100 runs, it outperforms a Binomial Bandit approach but not ours, with which it compares well while being more complex.

Note that Binomial Bandit converges to the best variation (namely, B) but with a low confidence equals to 0.69.

Table 2. Comparison of the three methods

Approach	Our approach	Binomial Bandit	LinUCB
Rewards/Visitors	0.13	0.11	0.13
Visitors	11168	11168	11168



Rectangles gives the bandits numbers from the previous regression tree

Fig. 2. Regression tree of *exp_set* with associated bandits

Detailed analysis of our approach Figure 2 shows the relation between the groups previously defined and a *new* regression tree obtained from the *exp_set* dataset. It also includes the variations *A* and *B* as explanatory variables.

The idea is to evaluate the overlap between a tree built from the training set and the tree obtained with the validation set. For each visitor belonging to a leaf of the new tree, we search the subgroup which he/she belongs to in the first tree and thus, the bandit that would have been used to assign it to a variation. In the Figure 2, each rectangle denotes the bandit's names for each subgroup of *exp_set* dataset. For instance, the leaf on the right only contains visitors associated to bandit #3 while the leaf at the left, contains visitors associated to bandit #18 and bandit #19. The second subgroup on the right contains to bandit #7 and bandit #8, and so on.

The idea is to compare if we can find visitors impacted by the test on the groups previously constructed for bandit assignment. We can observe if a vis-

itor have been impacted by the modification of the page in the reality with discriminant variable of regression tree obtained from the *exp_set* dataset⁶.

In reality, *A* was better than variation *B* only for small subgroups. According to the tree of figure 2, there are 3 profiles impacted by the test:

- Visitors belonging to *clust#1*, *clust#2*, *clust#4*, *clust#6* or *clust#10* and *clusPage#3* and *clusVisit#1*. These visitors have been assigned using the bandit #16. Our method chooses variation *A*.
- Visitors belonging to *clust#2*, *clust#6*, *clust#8* or *clust#10* and *clusPage#2* and *clusVisit#3*. These visitors have been assigned using the bandit #7 and #8. Our method chooses *A* (select bandit with highest probability).
- Visitors belonging to *clust#2*, *clust#4*, *clust#6* or *clust#10* and *clusPage#2* and *clusVisit#2*. These visitors have been assigned using the bandit #7 and #8. Our method chooses *A*.

The Table 1 shows the detailed results of our method.

Our method adds the following benefits:

- Identification of the optimal variation for a particular group of visitors.
- Faster convergence towards the operational phase.
- Limitation of convergence towards the wrong variation.

Our approach had a algorithmic complexity comparable to U.C.B one for experimental period. Moreover, the computation can be done in a parallel way. Nevertheless, we suppose that the actual reward probability distribution does not change over the time (hypothesis of stationary).

5 Conclusion and future work

This paper presents a new approach of dynamic assignments strategy based on a visitor segmentation determined automatically from the visitors navigation and characteristics. To produce this segmentation we have applied temporal and semantic clustering methods on visits database. To implement the dynamic allocation strategy we have used so-called "bandits" algorithm. To validate our approach, we have compared it with a dynamic allocation algorithm without a prior segmentation. Results are very promising and the concept has strong growth potential.

Currently we automatically determine the optimal number topic number and test our approach with other e-merchants. We also work on the quality of the clusters.

For future work, we plan to use more parameters related to the global traffic of the website in order to anticipate the "peaks" and "dips" of traffic interfere with analysis (differences between the variations can be reduced for a low traffic).

⁶ For a technical reason, variation *A* is called 0 and variation *B* is called 166888

References

1. Agrawal, R., Teneketzis, D., Anantharam, V.: Asymptotically efficient adaptive allocation schemes for controlled markov chains: finite parameter space. *IEEE Transactions on Automatic Control* 34(12), 1249–1259 (Dec 1989)
2. Auer, P., Cesa-Bianchi, N., Freund, Y., Schapire, R.E.: The nonstochastic multi-armed bandit problem. *SIAM J. Comput.* 32(1), 48–77 (Jan 2003), <http://dx.doi.org/10.1137/S0097539701398375>
3. Bonferroni, C.E.: Teoria statistica delle classi e calcolo delle probabilità. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze* 8, 3–62 (1936)
4. Galichet, N.: Contributions to Multi-Armed Bandits : Risk-Awareness and Sub-Sampling for Linear Contextual Bandits. Theses, Université Paris Sud - Paris XI (Sep 2015), <https://tel.archives-ouvertes.fr/tel-01277170>
5. Grün, B., Hornik, K.: topicmodels: An r package for fitting topic models. *Journal of Statistical Software, Articles* 40(13), 1–30 (2011), <https://www.jstatsoft.org/v040/i13>
6. Hothorn, T., Hornik, K., Zeileis, A.: Unbiased recursive partitioning: A conditional inference framework. *Journal of computational and graphical statistics* 15(3), 651–674 (2006)
7. Lai, T., Robbins, H.: Asymptotically efficient adaptive allocation rules. *Adv. Appl. Math.* 6(1), 4–22 (Mar 1985), [http://dx.doi.org/10.1016/0196-8858\(85\)90002-8](http://dx.doi.org/10.1016/0196-8858(85)90002-8)
8. Li, L., Chu, W., Langford, J., Schapire, R.E.: A contextual-bandit approach to personalized news article recommendation. In: *Proceedings of the 19th International Conference on World Wide Web*. pp. 661–670. WWW '10, ACM, New York, NY, USA (2010), <http://doi.acm.org/10.1145/1772690.1772758>
9. Molinaro, A.M., Dudoit, S., van der Laan, M.J.: Tree-based multivariate regression and density estimation with right-censored data. *Journal of Multivariate Analysis* 90(1), 154 – 177 (2004), <http://www.sciencedirect.com/science/article/pii/S0047259X04000296>, special Issue on Multivariate Methods in Genomic Data Analysis
10. Petitjean, F., Ketterlin, A., Gançarski, P.: A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recogn.* 44(3), 678–693 (Mar 2011), <http://dx.doi.org/10.1016/j.patcog.2010.09.013>
11. Robbins, H.: Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society* 58(5), 527–535 (1952), http://www.projecteuclid.org/DPubS/Repository/1.0/Disseminate?view=body&id=pdf_1&handle=euclid.bams/1183517370
12. Sarda-Espinosa, A.: Time series clustering along with optimizations for the dynamic time warping distance. dtwclust R package version 3.1.0 (2017), <https://github.com/asardaes/dtwclust>
13. Strasser, H., Weber, C.: On the asymptotic theory of permutation statistics. Report, Vienna University of Economics and Business Administration (1999), <https://books.google.fr/books?id=pieBNAECAAJ>
14. Wald, A.: Sequential tests of statistical hypotheses. *Ann. Math. Statist.* 16(2), 117–186 (06 1945), <http://dx.doi.org/10.1214/aoms/1177731118>