Evolving data as a context to improve A/B-Tests

Emmanuelle Claeys[1], Pierre Gançarski[2], Myriam Maumy-Bertrand[2], and Hubert Wassner[3]

[1] Charles Delaunay Institute, University of Technology of Troyes
[2] Strasbourg University, Strasbourg, France
[3] AB Tasty Company, France
{claeys, gancarski, mmaumy}@unistra.fr,
hubert@abtasty.fr

Abstract. Recently promising new methods have been developed to optimize e-commerce A/B testing using dynamic allocation. They provide faster results to determine the best variation and reduce test costs. However, dynamic allocation by traditional methods of reinforcement learning is restrictive on the type of data used : time series cannot be considered as a context. In this paper, we present two new methods, based on a common approach, that allow time series to describe visitors. Our numerical results on data from real tests lead to an improvement on dynamic allocation apply to A/B testing.

Keywords: A/B-Test · Dynamic allocation · Bandit · Time series · D.B.A. · Clustering

1 Introduction

Evaluating the relevance of a modification to an existing entity (drug, web page, etc.) according to one or more objectives (for example, increasing the number of survivals, the number of clicks, the value of purchases, etc.) is a well-known problem in many economic, industrial or even social fields. It can be done by directly and concretely comparing the different variations resulting from the modification. Such a task requires a way to evaluate each variation in order to make the optimal choice according to a defined objective in the given context. An *A/B-Test* consists in concretely evaluating these different alternatives according with an objective pre-defined. *A/B-Test*s have recently been the subject of renewed interest, particularly through their use in e-marketing(22) to improve a webpage in production. They make it possible to know for example, which version of a webpage increases the rate of click and more precisely for which kind of visitor. Unfortunately, current A/B-Test techniques are very expensive in terms of visitors to test or they drastically limit their descriptive (evolving data are excluded). Our method described here in this paper offers to realize a low cost A/B-Test with an evolving visitor description.

An *A/B-Test* consists of affecting *visitors* to the different variations in order to evaluate the relative performance of each. During this *exploration phase*, it is assumed that the result, called *reward*, of each *affectation* can be observed and used by the *A/B-Test* algorithm to evaluate each variation's performance. At the end of this exploration phase the *user* can better decide which variation will replace the current entity to be used in the future (i.e., in *production phase*) according to their relative performance.

An important characteristic of such methods is that the decision to assign a visitor to a variation is irrevocable. For instance, for the entire duration of the test, a visitor will always see the same web page on each of their visits, regardless of the number of visits. Thus, it is impossible to know what the visitor would have done if they had been assigned to another variation. Consequently, the population that has been affected to a variation is distinct from those affected to any other. Finally, it is assumed that visitors are unaware of their participation in a test and thus the existence of different variations.

A classical approach to the exploration phase is referred to as the frequential approach and consists of assigning visitors to the different variations according to explicit predefined ratios (*static allocation*) for a predefined period of time. If the ratios are balanced for each variation, the duration is unfortunately difficult to define *a priori*. Experiments have shown that a user tends to overestimate the duration, causing an inferior variation to have a large detrimental effect on the result for a long period of time. In this case, the obtained cumulative reward will be much lower than that which would have been produced by the allocation of each visitor to the optimal variation. This difference, called *regret*, increases with negative impact. Reducing the exploration phase may reduce regret, but may also lead to a lack of data needed to calculate performance. Therefore, in addition to determining the best option, the challenge of A/B-Test methods is to also minimize regret. Nevertheless, it is important to note that regret cannot be calculated during the observation phase as the optimal variation is obviously unknown *a priori*: the objective of the test is, by definition, to determine it. Finally, in most cases (e.g. time is money, people continue to die, etc.) the sooner the algorithm finds the solution (i.e. the sooner exploration can then be stopped), the better.

To address this problem, new A/B-Tests methods perform dynamic allocation of visitors based on bandit algorithms (14). Bandit *dynamic allocation* consists of adapting the allocation of visitors according to the obtained rewards and thus gradually tipping the visitors towards the optimal variation. Experiments and theoretical studies have shown that dynamic allocation (20) provides better results in terms of cumulative regret, as well as being faster at determining the best variation. In this context, a lot of methods implementing dynamic allocation based on bandit algorithms have been proposed (8; 16; 11) and have proved their ability to find optimal variations in the general case. Nevertheless, experiments also show that these methods often fail when the reward obtained by a visitor depends on both the variation and the visitor itself (15). For instance, in web marketing, visitors naturally tend to click and buy differently according to their own financial resources or their geographical localization. In medical treatment, the efficiency of a drug often depends on the age and/or gender of the patient. To address this problem, bandit algorithms have been extended to form *contextual bandits*, which take into account each visitor's *context*, i.e. their characteristics (age, origin, sex, etc.) when allocating them in order to perform more relevant allocations. Methods such as KERNELUCB (23) and LINUCB (5) have demonstrated not only the benefits of such an approach, but also their limitations, including the type of characteristics considered. Indeed, these bandit algorithms have been extended to take into account the *context* of the visitors, i.e. their characteristics (age, origin, sex, . . .) when allocating them. Unfortunately, these *contextual bandits* (24) limit the contexts to vectors with static numerical or symbolic values. However, an evolutionary characteristic may vary the optimality of a variation. For example, a drug may be more effective according to the evolution of a patient's weight (stable or, on the contrary, highly variable). Marketing researches have also shown that a page display that is customized according to the visitor's evolving behavior can increase the expected reward of a web page(19). By analogy, it could be interesting to integrate temporal data in the descriptive context of a visitor. Thus the use of temporal information (list of sites visited by a new visitor before arriving on the site hosting the test, pages browsed and options clicked by the visitor before arriving at the page under test) should, in our opinion, greatly improve the results from the allocation.

In this paper we propose an original A/B-TEST approach with dynamic allocation that takes into account temporal information about visitors, information that can be updated during navigation. The remainder of this paper is organized as follows. Section 2 presents in more detail the application context, the types of data observed and introduces a comprehensive literature review of existing approaches. Section 3 describes our proposition of two new innovative algorithms DBA-LINUCB and DBA-CTREE-UCB. Section 4 presents the experimental results. Finally the conclusions of the study are drawn in Section 4.

In sake of readability, the remainder of this article focuses on A/B-TESTS on e-commerce webpage with only two variations but all our propositions are directly and easily extended to tests with more applications and number of variations.

2 Application and existing methods

2.1 Evolving behaviors and e-commerce

In (19) the authors propose to distinguish visitors according to their behaviour on the site. Four *typical profiles* are defined from the observation of visitors' behaviour on a large number of different e-commerce sites and their propensity to click and/or buy:

- *Perfect prospects* know exactly what they are looking for, naturally go to the product webpage and look for the best purchasing price/quality . . . .
- *Potential buyers* know more or less what they want, their need is clearly defined but their choice is not completely yet established.
- *Undecided buyers* navigate as a recreational activity. They don't have a clearly defined need, but maybe they will buy something.

- *Accidental arrivals* are those who accessed the site by mistake and have no intention to buy anything.

Experiments have shown that the test may varies from one standard profile to another, e.g. *perfect prospects* buy regardless of the variation displayed, *potential buyers* pay more attention to the design of a webpage, so taking into account the standard profiles during dynamic allocation should improve limit test costs.

In order to determine these behaviours and thus the profile types, the user can collect temporal information such as :

- Historical temporal data collected before the visitor's arrival on the site (data from past navigations on this website or other websites with a history of pages visited, purchases, etc.).
- Evolving temporal data, dynamically generated from visitors actions between the first arrival on the site and the arrival on the tested page (clicks, purchases, etc.). Indeed, in the context of e-merchant websites, based on collected information, the user can collect for each visitor the time spent on the site at each visit, the number and pages visited, or even the purchase history since the first visit to the site. This data describes the *path-to-purchase.*

However, the collection of historical temporal data is strongly limited for legally reasons. For example, sharing informations between different sites without the visitor's agreement can be prohibited. In fact, in practice, A/B-Tests performed on e-commerce only consider evolving temporal data that can be legally and dynamically collected by the tested sites. Before presenting our proposal for dynamic allocation allowing temporal data, we introduce the concept of contextual and non-contextual dynamic allocation.

2.2 Dynamic allocation

To limit the cost due to static allocation, a commonly used solution is based on the use of a *bandit model* (12; 9). Bandits are dynamic allocation algorithms whose purpose is to limit the *regret*, i.e. the difference between the maximum potential rewards and the actual rewards obtained by the user. This maximum reward corresponds to the allocation of all visitors to the optimal variation which is unknown before the test. This regret can therefore only be used as a post-test performance metric[4](11). In practice, a bandit algorithm only seeks to maximize the average reward obtained at the end of the test.

To achieve this, the algorithm chooses a variation from the set of possible variations for the visitor when he/she arrives on the tested page.

The last reward obtained is then used to update the earnings estimates. Indeed, the properties of a variation are not (or only very partially) known at the beginning of the test and can only be re-estimated by allocating visitors to it. The main advantage of such an algorithm is that visitors allocation to the different variations is automatically adjusted over time according to the estimates

[4] on an offline application, with static allocation.

in order to do the necessary (*exploration*) while maximizing profit. However, due to the nature of bandit algorithms, the exploration part tends to decrease with the *exploitation*, which makes it possible to stop exploration when it is no longer necessary.

When choosing the bandit algorithm, two aspects have to be taken into account :

- Are visitor characteristics considered or not? A non-contextual bandit algorithm has to identify the variation with the highest average reward. Contextual approaches assume that there exists sub-groups of visitors, each presenting a different reward distribution. Consequently, contextual bandit algorithms identify the variation with the highest conditional average reward according to different visitor characteristics;
- Is the latency time a constraint for the user or not? In our application context, the bandit algorithm provides a choice that will result in the display of the web page being tested. It is obvious that a slow display for the visitor is not acceptable (more formally, the visitor mustn't wait for the display). Thus, the latency time, corresponding to the time needed by the bandit to make the allocation, must be less than 100 milliseconds.

In this paper we consider the case of a maximization of the final average reward with taking into account the visitor contexts (contextual bandit) on real time (with no latency). Among the existing methods, we select two contextual bandit algorithms that follows these constraints:

- Ctree-Ucb (6; 7) uses a pre-grouping before starting the experiment and does an independent dynamic allocation strategy for each group. This algorithm exploits data from the original variation (in production before starting the test) to produce a classifying function that takes a visitor (with characteristics) as input to assign a group. During the test, each group is associated to a non-contextual bandit. A new visitor is automatically classified in a group and then the bandit associated with the group chooses a variation.
- Lin-Ucb (13) is one of the most popular form of contextual bandits due to its performance and interpretability. This algorithm is based on empirical linear regression, using the contexts and rewards observed on each variation. Lin-Ucb is time greedy due to the inverse of the covariance matrix M (size $d \times d$ where d is the number of characteristics describing a visitor).

2.3 Limits

The algorithms introduced in the previous section have strong restrictions on the format of the context, which must be a vector with real or categorical values. In fact, these restrictions limit the use of bandit algorithms when the characteristics to be considered describes a time series. A solution to overcome this problem is then to transform the time series of each of the temporal characteristics. Two approaches are then possible: choose the dimension of the context such that it can contain all the different elements in all the series or concatenate the size of all the series of all the visitors. There are several problems with these approaches: on the one hand, these contexts may have a lot of empty values if the number of different temporal elements is significant ; on the other hand, and most importantly, the context bandit model relies on empirical statistical modeling that takes a context as an input to provide a probability of success or average reward. However, a large context introduces great variability in model predictions, which requires a lot of visitors to test.

To solve the problem of a very large context vector, (3) propose to empirically estimate parameters from a statistical LASSO model to integrate it into a bandit problem. However, this approach requires the characteristics to be independent of each other, which is obviously not the case when they are comes from the same time series. In fact, we couldn't find any contextual bandit algorithms method that can be adapted to data of variable size. A possible alternative is to reduce each evolving characteristic to a representative categorical or numerical characteristic. It is this alternative that defines our approach.

3 A new method : dynamic allocation based on temporal information

3.1 The approach

Our contribution aims to address the constraints and needs experienced by users in real-world applications. Thus, we first focus on a method that can be handle user's context as time series.

In this context, we propose an approach that consists of apply a time series replacement, which are unusable as they are for a bandit algorithm, by a categorical characteristic. Thus, we first focus on a method that can be applied in a "real word": for instance, in e-marketing, where this characteristic could represent the e-marketing profiles mentioned above. Thus, a first solution to replace each time series by a categorical characteristic requires the user to label each series manually. Unfortunately, this solution cannot be handle when the number of data becomes large and real-time allocation is necessary. Moreover, these standard profiles are difficult to highlight because it's require a manual analysis of data in order to extract characteristic behaviours. The second solution is to replace the series by their average. Experiences have shown that the loss of information decreases results (see Section 4.2). A third solution consists in defining a replacement in pre-processing of the test based on the automatic identification of the *representative categorical characteristics* for each temporal attribute of the visitor context from existing data (i.e. from all the time series present on this characteristic) before the test. During the testing phase, this replacement automatically replaces the characteristics of the tested visitor by categorical characteristics, this modified context is directly use by a classical contextual bandit methods. A *training data* is used to define the replacement function. This past data comes from the *original variation* already in production. In the following section we detail our replacement function (Section 3.2) before describing two new algorithms (Section 3.3).

3.2 From temporal data to categorical data

The replacement function has to summarise temporal information into categorical information. This problem is well known in data mining. In fact, many methods have been suggested, among clustering-based approaches and have shown their effectiveness in this case. Therefore, the replacement we propose is based on two steps:

1. an offline clustering step done in pre-processing which associate for each temporal characteristic of visitors, a set of clusters constructed from characteristics available on past data. We don't consider the rewards obtained because the objective here isn't to predict a reward probability but to identify and represent different behaviours.
2. a online replacement step consisting in replacing, during the test itself, each temporal characteristic of each visitor by a categorical characteristic corresponding to the label of the cluster to which the time series belongs.

The global scheme of our framework is given by Fig. 1.

Since the results of the time series similarity approach have shown good results in data mining, we include it in our experiments. Nevertheless, any other method allowing cluster extraction could be used.

Implementation of the method require to set a measure of similarity (2). The well known Euclidean distance (4) calculate the sum of the squares of the distances of the constituent elements of the sequences considered at each time step. If this distance is widely used, it requires that the series all have the same length and cannot be used in our case. However, in our work, we focus on the case where the observation of the context variables can be irregular. For example, one visitor may visit the site more frequently than another. The measure of similarity *D.T.W. (Dynamic Time Warping)* is a metric between two series of different sizes that is widely considered as relevant for many application (18). D.T.W. constructs a pairing of the elements of the sequences in order to align these sequences. This set of associations respects the total order on the sequencing of the values: in other words, the associations cannot cross each other. The reader can find other measures according to practical cases in (1).

Once a measure of similarity is set, it is then possible to apply different data mining algorithms. Among the existing methods, K-Means is one of the most used. However, it requires a method for calculating the average of observations (17). The averaging method *D.B.A (Dtw Barycentric Averaging)* associated with D.T.W. defined by (18) has shown good results in this context. D.B.A. is a global method, contrasts with pair strategies such as NLAAF (10). The main idea of D.B.A., inspired by K-Means, is based on an iterative refinement of an initial average sequence (arbitrarily set), in order to minimize root mean square distance to average sequences. These advantages and its popular use lead to the natural choice of time series clustering based on D.T.W. method

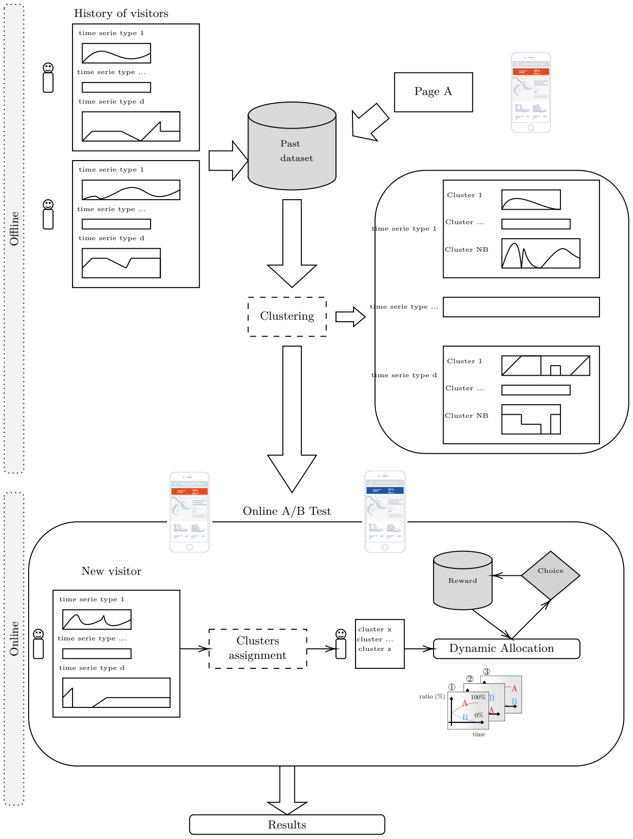


Fig. 1: Offline/online approach for time series integration in A/B-Tests

3.3 Two new methods : DBA-Ctree-Ucb and DBA-LinUcb

To integrate the time series in a contextual bandit strategy, we propose two algorithms DBA-Ctree-Ucb and DBA-LinUcb which are extensions of Ctree-Ucb and Lin-Ucb respectively. These algorithms require an original A variation (for example the existing web page) and evolving temporal data describing the visitors subjected to this original variation on the past. These data will constitute the learning dataset of DBA-LinUcb and DBA-Ctree-Ucb.

A first step in pre-processing of the test, consists in clustering time series via K-Means based on D.T.W./D.B.A in order to determine a set of clusters for each of the temporal characteristics.

If the selected algorithm is DBA-Ctree-Ucb, an intermediate step defines the function of classifying new visitors into homogenous groups. The second step (online) corresponds to the DBA-Ctree-Ucb itself. It consists in replacing, for each visitor submitted to the test, temporal characteristics by categorical characteristics thanks to the replacement function described above. The visitor and a new context are then processed by the selected algorithm.

4 Experimental framework

4.1 A/B Tasty Dataset

We use in our experiments a set of real data (*original database*), collected from a A/B-Test by static allocation made by the company AB Tasty on a webpage of a e-commerce fashion website. The data concerning 11,389 visitors, includes evolutionary temporal data captured during the purchasing paths of the various visitors who browsed the website and saw the test page concerned at least once: from their first visit to their arrival on the test page, their navigation is recorded.

This dataset corresponds to the case where a visitor arrives on the test page, a variation (A or B) and is assigned to it randomly (uniform allocation). For the next visits on the test page, the same variation is displayed to the visitor (irrevocability of the allocation). If a visitor buys after seeing the test page, the reward is 1, otherwise 0. Each visitor is associated with three different time series, of identical size equal to the number of days between his/her first and last visit to the tested page:

- `presence_time_serie` : composed of binary values. For each day the visitor visited the site, a value of 1 is defined, 0 otherwise. Each series starts and ends with the value 1.
- `connexion_time_serie` : composed of integer values between 0 and 24. For each day the visitor visited the site, a value equal to her/his arrival time on the site is defined, 0 otherwise. If the visitor comes several times in the same day, only the first hour is taken into account.
- `time_spend_serie` : composed of values included in $\mathbb{R}$. For each day the visitor visited the site, a value equal to the average activity time during her/his visit (in microseconds) to the site is defined, 0 otherwise.

Each time series is a sequence of daily *logs* associated with the different sessions of a visitor from the first visit to the site to the last, which includes the first arrival on the test page. Its length is therefore equal to the number of days between the first visit and the date of arrival on the page. Consequently, the sizes of the time series can be different for different visitors.

4.2 Comparing methods

In order to validate hypothesis of a improvement of our contextual bandit model by integration of time series, we compare DBA-LinUcb and DBA-Ctree-Ucb with, on the one hand, Lin-Ucb and DBA-Ctree-Ucb by reducing each time series to its mean and, on the other hand, with uniform which corresponds to a uniform static allocation.

Using regret to compare results requires knowing, for each visitor, the reward obtained for all the variations, which is impossible in practice (a visitor sees only one variation). To overcome this, it is possible to replace missing rewards by an average over all visitors, but this could strongly bias the results[5]. We therefore decided to compare only the average gains obtained at the end of the test for each of the configurations. Visitors to the initial database were used to simulate each of the tests. For this, for each visitor, the algorithm chooses a variation. If this variation corresponds to the one made in reality, the visitor, the choice and the associated reward are taken into account in the experience. Otherwise, the visitor is rejected and the next one in the database is subjected to the algorithm (*rejection sampling*).

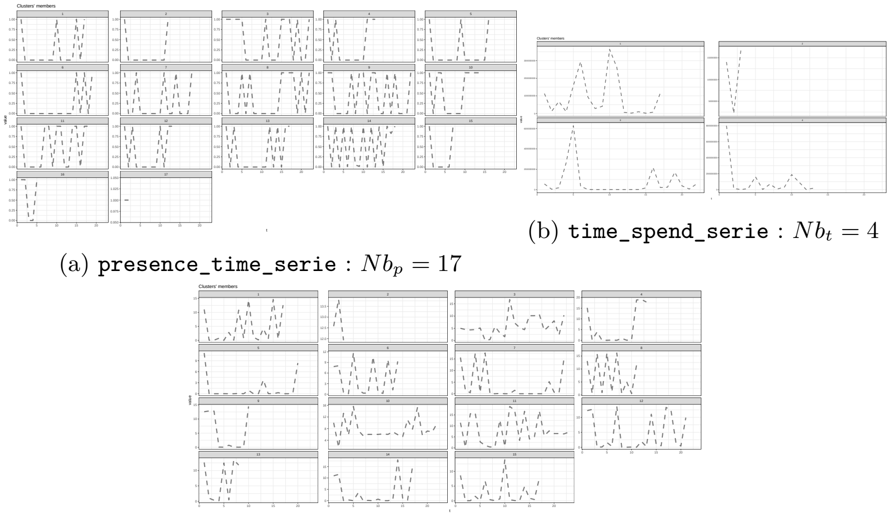
The performance of the approaches is based on average transaction rates (Number of buyers/Number of visitors). These performances are compared according to the size of the learning dataset (necessary to learn the clusters) and the number of clusters set before the test. Since the *rejection sampling* decreases the number of visitors tested (some will be ignored by the algorithm), we propose to replicate the dataset five times. Before observing the performances of the suggested algorithms, we have first carried out a study on the interpretability of the clusters obtained in the first step.

Clusters interpretability Our experiments were run with two configurations: clusters are learned on 30% ($\mathrm{Conf}_{\mathrm{clust30,70}}$) or 100% ($\mathrm{Conf}_{\mathrm{clust100,100}}$) of the database. The choice of learning on 30% of the data is here motivated by (7). In each of these configurations, the choice of the number of clusters is based on quality criteria usually used in clustering. In Fig. 2, we graphically present the centroids of the clusters obtained for the configuration $\mathrm{Conf}_{\mathrm{clust30.70}}$ with $Nb_{\mathtt{p}} = 17$, $Nb_{\mathtt{t}} = 4$ and $Nb_{\mathtt{c}} = 15$ where $Nb_{\mathtt{p}}$ (resp. $Nb_{\mathtt{t}}$ and $Nb_{\mathtt{c}}$) is the number of clusters for characteristic `presence_time_serie` (resp. `time_spend_serie` and `connection_time_serie`).

According to the standard profiles described in section 2.1, with a $\mathrm{Conf}_{\mathrm{clust30.70}}$ configuration, we can identify that the undecided behaviours are for example represented by the $\#3, \#9$ or $\#14$ clusters of the `presence_time_serie` series (see Fig.2a). Indeed, these clusters make visits increasingly. On the other hand, we can assume that with more spaced visits, such as those represented by clusters $\#6, \#12$ or $\#13$ of the `presence_time_serie` series, visitors are potential buyers taking time before make a decision. By contrast, visitors belonging to the #17 cluster from the `presence_time_serie` series are either not interested or are already engaged in their purchasing process. The clusters of the `connection_time_serie` series gives information about connection times habits. Visitors belonging to the #7 or #8 clusters, for example, regularly visit the site around 3pm. Interestingly enough, the cluster combinations for each visitor leads to distinguish a visitor who arrived by mistake from a perfect prospect. Indeed, perfect prospects and those who arrived there by chance on the site are difficult to separate by observing only their paths in the `presence_time_series` series. Their visits are short and they don't necessarily come back to the site. However, by analysing the time spent on the website `connection_time_serie`,

[5] a model allowing the replacement of missing values according to time series of variable size being, as we said in section 2.3, difficult to construct

it's possible to separate these two profiles (a visitor who arrives by mistake will immediately leave the site). In the same way, potential buyers and undecided visitors can return to the site at the same frequency. To distinguish them, the analyst can assume that those who return at the same time regularly can be considered as those visiting the site for entertainment (21). Note that our experiments showed that the centroids obtained with $\text{Conf}_{\text{clust30.70}}$ were globally very similar to those obtained with $\text{Conf}_{\text{clust100.100}}$.



(a) `presence_time_serie` : $Nb_p = 17$

(b) `time_spend_serie` : $Nb_t = 4$

(c) `connexion_time_serie` : $Nb_c = 15$

Fig. 2: Clusters centroïdes

Exploration during the A/B-Tests (Dynamic allocation) For reasons of readability and interpretation of the results, we split the report of our experiments into two parts. The first part compares DBA-Ctree-Ucb and Ctree-Ucb, the second part compares DBA-LinUcb and Lin-Ucb. We compare the average reward (transaction rates) of the previously mentioned algorithms over 200 run-times. We observe the effect of the cluster numbers Nb_p, Nb_t and Nb_c on the results. In order for the clusters to be understandable by the user, we limit the maximum number of clusters for each parameter.

DBA-Ctree-Ucb vs Ctree-Ucb: The transaction rate obtained by uniform (i.e. static allocation) from the original data is 11.74%.

In the tests we change the number of clusters in order to estimate how results of DBA-Ctree-Ucb change according with these parameters (Tab. 1). Tab.1a, presents the results obtained for different configurations (Nb_p, Nb_t, Nb_c). The transaction rates obtained by DBA-Ctree-Ucb systematically exceed those of DBA-Ctree-Ucb based on the average of the series. However, algorithms performance may randomly vary due to rejection sampling[6]. Additional results are therefore presented in Tab.1b where the data were duplicated to represent 5 times the initial data set. With this configuration, the performances decrease but the setting ($Nb_{\mathtt{p}} = 17$, $Nb_{\mathtt{t}} = 3$ and $Nb_{\mathtt{c}} = 3$) remains globally one of the best with or without duplication of the dataset.

Table 1: Influence of parameters on average reward for DBA-Ctree-Ucb and Ctree-Ucb (AB Tasty Dataset)

	$\mathrm{Conf}_{30,70}$ et $\mathrm{Conf}_{\mathrm{clust}30,70}$		$\mathrm{Conf}_{100,100}$ et $\mathrm{Conf}_{\mathrm{clust}100,100}$	
DBA-Ctree-Ucb $Nb_p;Nb_t;Nb_c$	$V_A = P_1$	$V_A = P_2$	$V_A = P_1$	$V_A = P_2$
5;3;3	13,34%	13,56%	12,94%	12,88%
5;4;15	13,96%	14,12%	11,90%	12,23%
11;3;3	14,15%	14,10%	14,62%	14,58%
11;4;15	13,85%	13,35%	13,85%	13,58%
17;3;3	14,23%	**15,46%**	**15,27%**	**15,34%**
17;4;3	**14,26%**	13,60%	14,87%	13,73%
Ctree-Ucb	11,40%	11,71%	12,24%	12,77%
uniform	11,74%	11,74%	11,74%	11,74%

(a) Initial data

	$\mathrm{Conf}_{30,70}$ et $\mathrm{Conf}_{\mathrm{clust}30,70}$		$\mathrm{Conf}_{100,100}$ et $\mathrm{Conf}_{\mathrm{clust}100,100}$	
DBA-Ctree-Ucb $Nb_p;Nb_t;Nb_c$	$V_A = P_1$	$V_A = P_2$	$V_A = P_1$	$V_A = P_2$
5;3;3	12,76%	12,31%	12,58%	**12,59%**
5;4;15	12,53%	12,47%	**12,61%**	**12,59%**
11;3;3	12,49%	12,45%	12,55%	12,57%
11;4;15	12,23%	12,14%	12,49%	12,46%
17;3;3	**12,76%**	**12,49%**	12,53%	12,48%
17;4;15	12,46%	12,43%	12,41%	12,76
Ctree-Ucb	12,28%	12,15%	12,28%	12,14%

(b) Duplicate data

DBA-LinUcb vs Lin-Ucb: By analogy with previous experiments, the Tab. 2 the results of both DBA-LinUcb and Lin-Ucb experiments obtained with initial data (Tab. 2a) and with duplication (Tab. 2b). The transaction rates obtained by DBA-LinUcb are higher than Lin-Ucb if the right number of clusters is properly chosen. However, the worst performances of DBA-LinUcb remain very close to those obtained by Lin-Ucb. As for DBA-Ctree-Ucb the setting ($Nb_{\mathtt{p}} = 17$, $Nb_{\mathtt{t}} = 3$, $Nb_{\mathtt{c}} = 3$) is optimal with $\mathrm{Conf}_{\mathrm{clust}30.70}$ without duplication. For the other configurations, the setting ($Nb_{\mathtt{p}} = 5$, $Nb_{\mathtt{t}} = 4$, $Nb_{\mathtt{c}} = 15$) that maximizes the average gain. Our hypothesis is that DBA-LinUcb is very sensitive to the number of clusters so a large number of clusters implies

[6] In the available code, we propose, in addition to the real data, an experiment on simulated data. The performances are comparable to those obtained in this paper.

a longer learning period due to the modeling of the parameters of the linear function.

Table 2: Influence of parameters on average reward for DBA-LinUcb and Lin-Ucb (AB Tasty Dataset)

DBA-LinUcb Nb_p;Nb_t;Nb_c	Conf$_{\text{clust30,70}}$	Conf$_{\text{clust100,100}}$
5;3;3	12,92%	12,61%
5;4;15	12,77%	**12,76%**
11;3;3	12,70%	12,42%
11;4;15	12,93%	12,44%
17;3;3	**13,07%**	12,65%
17;4;15	12,61%	12,70%
Lin-Ucb	12,43%	12,43%

(a) Test Dataset

DBA-LinUcb Nb_p;Nb_t;Nb_c	Conf$_{\text{clust30,70}}$	Conf$_{\text{clust100,100}}$
5;3;3	11,93%	12,01%
5;4;15	**12,77%**	**12,66%**
11;3;3	12,45%	12,06%
11;4;15	12,51%	12,07%
17;3;3	12,20%	12,59%
17;4;15	12,50%	12,17%
Lin-Ucb	12,57%	12,57%
uniform	11,74%	11,74%

(b) Test Dataset (Duplicated)

5 Conclusion

In this paper, we present two new approaches DBA-Ctree-Ucb and DBA-LinUcb, for A/B-Test based on bandit models. These methods provide a dynamic allocation which including temporal data. It focuses on practical (real) applications. The results are promising in terms of average gain compared to other methods. DBA-Ctree-Ucb is particularly well suited to e-commerce because it can gather in the same group, visitors who have had the same behavior on an existing original variation. For the user, these groups are an interesting for extracting typical visitor profiles. DBA-Ctree-Ucb is also relatively insensitive to the different parameters.

Moreover, a learning of the clusters on 30% of the original database is sufficient to obtain results comparable to a total learning. Additional experiments are currently done to confirm this hypothesis. Our future work will focus on the influence of *rejection sampling* on performance and show very promising results. Finally, since only C.H.A. has been compared (and not retained), further studies will evaluate other clustering algorithms with, possibly, other measures of similarity.

Bibliography

[1] Aghabozorgi, S., Shirkhorshidi, A.S., Wah, T.: Time-series clustering - a decade review. Information Systems **53** (05 2015). https://doi.org/10.1016/j.is.2015.04.007

[2] Agrawal, R., Faloutsos, C., Swami, A.N.: Efficient similarity search in sequence databases. In: Proc. 4th International Conference on Foundations of Data Organization and Algorithms. pp. 69–84. FODO '93, Springer-Verlag, London, UK, UK (1993), `http://dl.acm.org/citation.cfm?id=645415.652239`

[3] Bastani, H., Bayati, M.: Online decision-making with high-dimensional covariates. SSRN Electronic Journal (01 2015). https://doi.org/10.2139/ssrn.2661896

[4] Box, G.E.P., Jenkins, G.M.: Time Series Analysis: Forecasting and Control. Prentice Hall PTR, Upper Saddle River, NJ, USA, 3rd edn. (1994)

[5] Chu, W., Li, L., Reyzin, L., Schapire, R.: Contextual bandits with linear payoff functions. In: Gordon, G., Dunson, D., Dudík, M. (eds.) Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 15, pp. 208–214. PMLR, Fort Lauderdale, FL, USA (11–13 Apr 2011)

[6] Claeys, E., Gancarski, P., Maumy-Bertrand, M., Wassner, H.: Dynamic allocation optimization in A/B tests using classification-based preprocessing (May 2020), `https://hal.archives-ouvertes.fr/hal-01874969`, working paper or preprint

[7] Claeys, E., Gançarski, P., Maumy-Bertrand, M., Wassner, H.: Regression tree for bandits models in a/b testing. In: IDA (2017)

[8] Gittins, A.J.C., Gittins, J.C.: Bandit processes and dynamic allocation indices. Journal of the Royal Statistical Society, Series B pp. 148–177 (1979)

[9] Gittins, J., Jones, D.: A dynamic allocation index for the sequential design of experiments. In: Gani, J. (ed.) Progress in Statistics, pp. 241–266. North-Holland, Amsterdam (1974)

[10] Gupta, L., Molfese, D., Tammana, R., Simos, P.: Nonlinear alignment and averaging for estimating the evoked potential. IEEE transactions on bio-medical engineering **43**, 348–56 (05 1996). https://doi.org/10.1109/10.486255

[11] Kaufmann, E., Cappé, O., Garivier, A.: On the Complexity of A/B Testing. ArXiv e-prints (May 2014)

[12] Lai, T., Robbins, H.: Asymptotically efficient adaptive allocation rules. Advances in Applied Mathematics **6**(1), 4 – 22 (1985). https://doi.org/https://doi.org/10.1016/0196-8858(85)90002-8

[13] Langford, J., Zhang, T.: The epoch-greedy algorithm for multi-armed bandits with side information. In: NIPS (2007)

[14] Lattimore, T., Szepesvári, C.: Bandit Algorithms. Cambridge University Press (2019)

[15] 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). https://doi.org/10.1145/1772690.1772758

[16] Nicol, O., Mary, J., Preux, P.: Icml exploration and exploitation challenge: Keep it simple ! In: Journal of Machine Learning Research (JMLR) (2012)

[17] Niennattrakul, V., Ratanamahatana, C.: On clustering multimedia time series data using k-means and dynamic time warping. pp. 733–738 (01 2007). https://doi.org/10.1109/MUE.2007.165

[18] Petitjean, F., Ketterlin, A., Gancarski, P.: A global averaging method for dynamic time warping with applications to clustering. Pattern Recognition **44**, 678– (03 2011). https://doi.org/10.1016/j.patcog.2010.09.013

[19] Pomirleanu, N., Schibrowsky, J., Peltier, J., Nill, A.: A review of internet marketing research over the past 20 years and future research direction. Journal of Research in Interactive Marketing **7** (08 2013). https://doi.org/10.1108/JRIM-01-2013-0006

[20] Robbins, H.: Some aspects of the sequential design of experiments. Bull. Amer. Math. Soc. **58**(5), 527–535 (09 1952)

[21] Roukine, S.: Améliorer ses taux de conversion web : Vers la performance des sites web au-delà du webmarketing (2011)

[22] Thompson, W.R.: On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. Biometrika **25**(3-4), 285–294 (1933). https://doi.org/10.1093/biomet/25.3-4.285

[23] Valko, M., Korda, N., Munos, R., Flaounas, I., Cristianini, N.: Finite-time analysis of kernelised contextual bandits. In: Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence. pp. 654–663. UAI'13, AUAI Press, Arlington, Virginia, United States (2013)

[24] Zhou, L.: A survey on contextual multi-armed bandits. CoRR **abs/1508.03326** (2015)