

Approche paramétrique de bandits pour le test A/B

Emmanuelle Claeys^{1*} Pierre Gançarski¹ Myriam Maumy-Bertrand² Hubert Wassner³

¹ ICube – Université de Strasbourg – 67412 Illkirch – France

² IRMA – Université de Strasbourg – 67084 Strasbourg – France

³ Entreprise AB Tasty – 3 Impasse de la Planchette – 75003 Paris – France

Résumé

Cet article propose une méthode pour optimiser l'approche de bandits en utilisant un arbre décisionnel dans le contexte du test A/B. Il rappelle les problématiques que le test A/B rencontre et propose d'anticiper le comportement d'un profil de visiteurs, afin d'orienter la stratégie (proposer la meilleure variation au visiteur) du test selon ce comportement.

Mots clés

Méthodes de bandits, personnalisation, clustering temporel, test A/B, allocation dynamique.

Abstract

This paper proposes a new architectural model for A/B testing evaluation in multiple contexts. It exposes the global problematic of A/B testing when we want to manage several types of e-commerces. The goal of this paper is to provide a method for optimize bandits approach using a decision tree.

Keywords

Bandit approach, personalization, time series clustering, A/B testing, exploration/exploitation dilemma.

1 Introduction

Pour un *e-commerçant*, c'est-à-dire un commerçant dont l'activité commerciale dépend du WEB, la qualité du site (ergonomie, iconographies, guide et parcours) est primordiale. Modifier celui-ci afin de le rendre plus efficace est donc tout aussi crucial. Néanmoins, toute modification n'est pas bonne à prendre car elle peut dégrader le site et donc impacter l'activité du e-commerçant. Il n'est donc pas envisageable pour le e-commerçant de mettre en ligne un changement sur son site sans en avoir au préalable tester la pertinence. Pour cela, de plus en plus d'entreprises utilisent des outils d'analyse pour le web marketing, comme, par exemple, les plate-forme de test A/B. Un test A/B (ou A/B testing¹), consiste à créer plusieurs versions d'un

même objet, dans le but de vérifier une hypothèse marketing [1]. Par exemple, il s'agit de tester auprès d'un échantillon d'internautes, nommés *visiteurs*, durant une période donnée, différentes variations (généralement deux) d'une même page web (nommées variation A et variation B) proposée par un e-commerçant. Le but est d'identifier celle qui donne les meilleurs résultats par rapport à un objectif fixé en amont du test (comme par exemple l'augmentation du taux de *conversion*, le nombre de visites ou de clics sur une bannière ou encore la valeur moyenne du panier, etc.). Le test peut alors être évalué sur l'ensemble des visiteurs de la page ou uniquement sur un échantillon de visiteurs ayant une caractéristique spécifique (par exemple les visiteurs parisiens). À l'issue du test, le e-commerçant pourra soit conserver A, soit conserver B, soit, dans une optique de personnalisation, présenter la variation A ou B à des visiteurs en fonction de caractéristiques spécifiques, comme par exemple la géo-localisation.

Derrière cette problématique de marketing, se cachent des verrous scientifiques importants, à cause de la contrainte suivante : lors d'un test A/B, un visiteur est affecté à une variation (en général de façon équiprobable) à sa première venue sur la page, celle-ci lui est affectée jusqu'à la fin du test ainsi que pour toutes ses futures visites. De fait, une caractéristique importante des tests A/B est qu'un visiteur ne voit qu'une seule des deux variations et que de ce fait, l'ensemble des visiteurs qui ont vu la variation A est totalement disjoint de celui des visiteurs ayant vu la variation B. De par cette contrainte, il devient difficile résoudre la problématique par des algorithmes de recommandation classiques qui permettent de réitérer la proposition d'une variation à un visiteur [2].

Du fait de cette irrévocabilité de la décision d'affectation, la question sous-jacente est alors de savoir comment déterminer ce qu'aurait fait un visiteur sur une variation sachant ce qu'il a fait sur l'autre. Enfin, d'autres caractéristiques peuvent intervenir dans l'analyse du test : on ne peut pas, en général déterminer lorsqu'un visiteur ne fait pas de conversion, si cela est dû à la variation présentée ou simplement à l'absence de volonté a priori du visiteur de faire cette conversion (simple visite d'informations sur des offres du site par exemple) ; le nombre de conversions peut être très

*claeys@unistra.fr

1. Le terme A/B étant un terme générique pour qualifier tout type de tests.

largement inférieur au nombre de non-conversions ; le délai entre la conversion et la visite de la page peut être plus ou moins long, etc.

Enfin, les tests statistiques existants [3] pour répondre à ces questions, se heurtent au problème de la *politique d'allocation dynamique*. En effet, l'identification de la meilleure variation et la caractérisation d'un ou plusieurs segments de visiteurs sont importantes pour le e-commerçant, mais sa priorité principale est avant tout de tirer des bénéfices de son site. Si l'une des variations a des conséquences désastreuses, il souhaitera limiter la phase d'*exploration* (c'est-à-dire de test) au profit de la phase d'*exploitation* de la meilleure variation courante : la politique d'allocation changera au fil de temps pour affecter prioritairement les nouveaux visiteurs sur la variation la plus rentable. C'est pour répondre à ce compromis d'exploration-exploitation que les méthodes dites « de bandits » sont très utilisées (cf. Section 2.2)

Notre méthode. Pour répondre à l'ensemble de ces problématiques, nous proposons une méthode combinant une *étape de pré-analyse* (hors-ligne) dans laquelle nous identifions des segments de visiteurs à partir de leurs profils temporels et de leurs centres d'intérêt liés à leurs comportements lors de leur(s) visite(s) suivie d'une *étape d'observation* (en-ligne) correspondant au test proprement dit, durant laquelle nous observons si des segments présentent des taux de conversions différents. Pour cela, pour chaque segment de visiteurs (identifié dans l'étape de pré-analyse), nous utilisons un algorithme de comparaison de variations (méthode de bandit, cf. Section 2.2) de façon totalement indépendante. Ainsi, nous proposons :

- d'identifier des segments des visiteurs en fonction de leur profils temporels et de leurs centres d'intérêts à l'aide de techniques de clustering temporel ;
- d'effectuer les comparaisons, lors de la phase d'analyse, par segment et non uniquement globalement ;
- de limiter les pertes d'exploitation dues à l'expérience en modifiant la politique d'allocation pour chacun de ces segments (donc localement et non plus globalement) en fonction de la convergence plus ou moins rapide du test vers la version optimale pour le segment (si elle existe).

2 Mise en œuvre de l'approche

2.1 Étape 1 : pré-analyse

Cette étape consiste à analyser les visiteurs pendant une période pré-définie (généralement une à deux fois la durée prévue pour le test lui-même) afin d'extraire et de caractériser des segments de visiteurs suivant leurs profils temporels et leurs centres d'intérêt.

Profils temporels. À l'issue de la phase de pré-analyse, à chaque visiteur est associé deux vecteurs, de taille égale à la durée de la pré-analyse. Le premier vecteur $V_{presence}$ indique la visite ou non pour chaque intervalle de temps

(en général chaque jour) : $V_{presence}[i] = 1$ si un visiteur a visité la page à la i -ème date, 0 sinon. Le second vecteur V_{pages} donne le nombre de pages consultées à chaque période par ce visiteur.

Prise en compte des centres d'intérêt. Afin de mieux caractériser les visiteurs, nous avons affecté à chacun d'eux un ensemble de centre d'intérêts en fonction des pages qu'il a traversées avant d'arriver à la page testée. Pour cela nous procédons en deux étapes : (1) recherche des différents mots clés présents pour les pages parcourues par l'ensemble des visiteurs et association à chaque page parcourue d'un ou plusieurs *sujets d'intérêt* (topics) construits à partir de ces mots clés et (2) affectation à chaque visiteur de l'ensemble des sujets associés aux pages qu'il a personnellement parcourues. À noter, qu'un topic peut être associé à plusieurs pages. Cette association a été réalisée grâce à une méthode de *topic modeling* via l'allocation de Dirichlet latente [6]. Le *topic modeling* est une méthode en traitement automatique du langage naturel permettant de modéliser des relations entre différents sujets. L'allocation de Dirichlet latente est basée sur une combinaison de modèles statistiques bayésiens, pour des données discrètes où les sujets ne sont pas corrélés. Le topic modeling est une extension du modèle allocation de Dirichlet latente (L.D.A) lorsque les mots des documents sont corrélés entre les sujets. Nous avons choisi cette méthode pour son utilisation populaire et la facilité de sa mise en œuvre. L.D.A. utilise un nombre de sujets fixé k et travaille itérativement sur chaque document $w = (w_1, w_2, \dots, w_N)$ d'un corpus D contenant N mots d'un vocabulaire composé de V termes différents $w_i \in 1, \dots, V$ pour $i = 1, \dots, N$.

Construction des clusters de visiteurs. Pour construire les clusters de visiteurs (qui seront utilisés pour définir nos segments) nous utilisons les deux vecteurs identifiés précédemment. Ces vecteurs peuvent-être vus comme des séries temporelles, allouées à chaque visiteur. De nombreuses méthodes existent pour réaliser des clusters de séries temporelles. L'approche que nous avons retenue est celle basée sur la métrique D.T.W. (Dynamic Time Warping) car celle-ci présente un certain nombre d'avantages [4] tels que :

- toutes les données sont utilisées ;
- l'alignement des séquences obtenu est optimal ;
- la possibilité d'extraire des comportements décalés ;
- supporte un échantillonnage irrégulier ;
- permet de comparer des séquences de longueurs différentes.

Par contre, elle s'avère assez sensible aux valeurs extrêmes et au bruit.

Nous avons donc choisi l'algorithme de partitionnement (clustering) K-means pour sa simplicité d'utilisation et son efficacité. Néanmoins, en plus d'une mesure de (dis)similarité, celui-ci nécessite un mécanisme de calcul des centres des clusters. Pour cela, nous utilisons la méthode D.B.A. (D.T.W. Barycenter Averaging) [4] sous le langage **R** [5], qui est une méthode itérative de calcul d'une

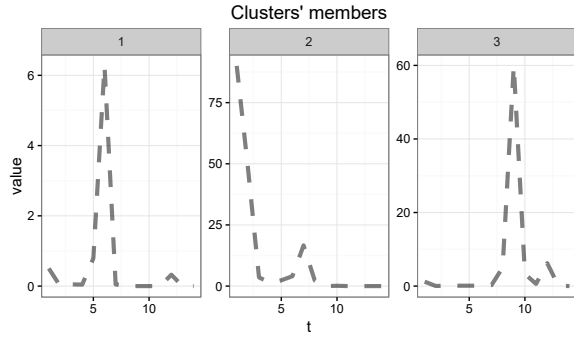


FIGURE 1 – Centres de trois clusters présentant trois évolutions du nombre de pages consultées.

moyenne. La figure 1 montre un exemple de centres pour trois clusters obtenus à partir de pages parcourues à chaque visite par les différents visiteurs et ce sur une période d’observation de 15 jours.

Recherche de régularités dans les profils de visiteurs. L’objectif de cette étape est de voir si certains (sous)segments de visiteurs répondent plus facilement à l’objectif de conversion.

Nous avons choisi d’utiliser un arbre de régression non paramétrique (algorithme CTREE) [8] consistant à étudier tous les profils avec une différence statistiquement significative dans le taux de conversions (test du khi-deux de Pearson, avec un niveau de seuil de confiance fixé à 95 %). Dans notre cas, cet arbre sera utilisé pour identifier les facteurs pronostiques, c’est-à-dire le lien entre les clusters et les topics construits précédemment et les différents taux de conversions.

Un tel arbre se construit de manière itérative en divisant, à chaque étape, la population en deux sous-ensembles. Pour diviser cette population, l’algorithme utilise des tests de permutation [9]. Des procédures d’essais multiples sont appliquées pour déterminer l’association significative entre les variables, via une approche réursive [7].

Pour identifier des variables explicatives discriminantes, l’arbre utilise des procédures de test spéciales comme le test de Spearman, le test de Wilcoxon-Mann-Whitney ou le test de Kruskal-Wallis ainsi que des tests de permutation basés sur des statistiques d’ANOVA ou des coefficients de corrélation. Dans notre expérimentation, pour choisir le test approprié pour vérifier le rapport de différence, nous utilisons une méthode multiple de correction de Bonferroni associée à une méthode de Monte Carlo et à un test de Kruskal-Wallis.

Ceci conclut une pré-analyse, la figure 2 présente un exemple d’arbre obtenu sur un jeu de tests *train_set*.

2.2 Étape 2 : Observation

Remarque : Pour affecter un nouveau visiteur lors de l’étape d’observation (donc du test), nous considérons son

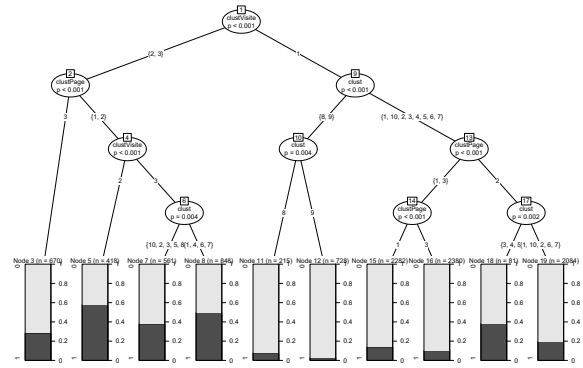


FIGURE 2 – Arbre de régression de *train_set*

parcours depuis sa première arrivée sur le site (sachant qu’il n’a pas encore vu la page test) et nous l’affectons au cluster le plus proche (nous procédons simplement par comparaison du parcours du visiteur aux centres des clusters en utilisant D.T.W.).

Dans la littérature, il existe de nombreuses méthodes pour comparer un test sur deux échantillons, mais elles supposent que les échantillons soient choisis aléatoirement à partir d’une population homogène [3]. Ces méthodes sont capables de gérer de larges volumes de données, avec une distribution paramétrique ou non. Néanmoins, dans le cas du marketing, l’observation d’une variable (par exemple l’achat d’un produit), peut varier selon les différentes populations, cela pouvant être dû aux fluctuations de l’échantillonnage ou au contraire cela révèle des comportements différents selon les catégories. De fait, ces méthodes sont difficiles à utiliser dans le cas des tests A/B :

- elles n’expliquent pas pour quel profil la variation A ou B est la plus optimale.
- les faibles différences peuvent être difficiles à interpréter (convergence trop rapide vers l’exploitation).

De plus, dans le e-commerce, le taux de conversions est particulièrement bas (de l’ordre de 2 à 3 %) et dépend de la position de la page testée dans le parcours de conversion. Enfin, un enjeu important du test A/B est de limiter les pertes dues à l’expérience. Par exemple, si le e-commerçant veut réaliser un test hasardeux comme par exemple changer la photo du produit ou ajouter une fenêtre promotionnelle (pop-up), il veut limiter les pertes qu’engendrait la variation, dès lors qu’il est convaincu que celle-ci est mauvaise pour tous les profils. Cette approche peut être vue comme une minimisation du regret. Dans de nombreux tests A/B, l’échantillonnage est basé sur une proportion égale du trafic (50% du trafic a été alloué à A et 50% à B). Dans cette situation, une variation optimale devra attendre la fin du test pour être choisie.

Il est cependant possible de changer l’allocation au cours du temps via de nombreuses méthodes : les méthodes dites de *bandits* utilisent un échantillonnage adaptatif lorsque la meilleure variation est identifiée. Cette approche vue

comme un compromis exploration-exploitation, est aussi vue comme une méthode d'apprentissage par renforcement.

Un *méthode de bandit* consiste à choisir une solution, de façon itérative, parmi N solutions, chacune donnant une récompense moyenne inconnue a priori, la solution permettant de maximiser le gain [12]. Les algorithmes mis en œuvre peuvent éventuellement être paramétrés par le type ou la forme de la distribution, le seuil de confiance et la perte maximale acceptable. Une variante classique dite de *bandits multi-bras* consiste à identifier la meilleure variation parmi plusieurs variations possibles. Enfin, les *bandits contextuels* sont particulièrement populaires car ils utilisent le contexte rattaché à un visiteur pour prendre la meilleure décision. La méthode utilise pour cela un vecteur de contexte à d dimensions. Le processus d'apprentissage utilise ces vecteurs de contexte ainsi que la récompense de la variation alors affichée. Le lecteur intéressé trouvera dans la littérature un grand nombre de telles méthodes. On pourra citer par exemple :

- L'algorithme `LinUCB` : présume une dépendance linéaire (à identifier) entre le gain attendu et le vecteur de contexte [10].
- L'algorithme `UCBogram` : présume une dépendance non linéaire (à identifier) entre le gain attendu et le vecteur de contexte.
- L'algorithme `NeuralBandit` : approche neuronale où un réseau apprend le contexte et le gain associé.
- L'algorithme `KernelUCB` : version non linéaire de **LinUCB** en utilisant une matrice à noyaux. Utilisée pour le on-line learning.
- L'algorithme `Bandit Forest` : Utilise des forêts aléatoires de bandits basées sur une distribution commune des gains et du contexte. Utile lorsque l'expérience est reproductible.

En pratique ces méthodes sont efficaces mais présentent quelques lacunes ou inconvénients. Par exemple, on considère le non-achat du visiteur comme un échec dès qu'il quitte la page testée. Cependant, un visiteur peut répondre à l'objectif de conversion après plusieurs visites. Si nous considérons le nombre de visites comme un contexte, nous pouvons alors limiter la pénalisation pour le bandit.

Enfin, s'ajoute une contrainte fondamentale pour l'intégration de l'algorithme décisionnel en industrie : le temps alloué à la prise de décision. Il est inenvisageable pour le e-commerçant que le système prenne trop de temps pour décider quelle variation attribuer au visiteur : Quand un visiteur arrive sur un site web, le système doit répondre en moins de 100ms pour éviter d'impacter la visite du visiteur sur le site. Le e-marchand ignore également les informations à utiliser pour définir le contexte, encore plus lorsque les données collectées contiennent des variables potentiellement non pertinentes. Enfin, le trafic et les achats fluctuent au cours du temps, ce qui perturbe parfois l'identification de la variation la mieux adaptée.

Pour notre approche, nous mettons en œuvre une méthode de bandit binomiale [11] pour pour chaque segment de visiteurs identifiés dans notre arbre de pré-analyse. Il s'agit donc d'associer un bandit par segment. Pour chaque nouveau visiteur appartenant au segment considéré, le bandit lui alloue une des variations disponibles (ici deux : A et B), selon la probabilité bayésienne postérieure (estimée de façon itérative) que chaque variation, dite aussi bras, soit optimale. Dans notre cas, le gain est une valeur binaire (1 : conversion ou 0 : non conversion), la fonction de gain suit une distribution de Bernoulli dont les paramètres sont estimés au cours du temps.

3 Expérimentation

3.1 Données et pré-analyse

Le jeu de données de validation est issu d'un test A/B

- où la variation B consistait à changer le lien d'un bouton "retour",
- sans allocation dynamique (60% du trafic a été alloué à A et 40% à B),
- réalisé du 11/07/16 au 25/07/16 pour un e-commerce de vêtements.

Le site de l'e-commerçant reçoit environ 150 000 visiteurs par mois, les données récoltées ici portent essentiellement sur la navigation et le comportement : nous utilisons pour chaque visiteur, l'historique de ses visites, de pages vues, de transactions éventuelles et les dates associées à ces événements. Pour simuler l'étape de pré-analyse (resp. d'observation), nous sélectionnons les visiteurs arrivés avant le 15/07/16 inclus (resp. arrivés après le 16/07/16 inclus).

Étape de pré-analyse Nous construisons un arbre de classification exclusivement sur les visiteurs de *train_set* qui ont vu la version originale A . Nous cherchons à identifier les caractéristiques des populations ayant un taux de conversions différent (l'objectif ici est "Achat", "Pas d'achat"). Trois caractéristiques sont associées pour chaque visiteur.

1. Le cluster de visites basé sur la série temporelle de visites (présence ou non présence du visiteur sur le site, pour chaque jour du test).
2. Le cluster de pages basé sur la série temporelle de pages consultées (nombre de pages consultées sur le site, pour chaque jour du test).
3. Le topic associé aux pages qu'il a consulté lors de ces visites passées².

Pour simplifier le test, nous considérerons que dans nos expériences tous les visiteurs sont parfaitement associés à leurs clusters.

La figure 2 présente l'arbre de classification issu du jeu de données de pré-analyse *train_set* pour la variation A . Cet arbre identifie 10 profils (clusters) distincts. Il a été

². pour des raisons techniques, nous utiliserons ici uniquement la dernière page consultée avant qu'il n'arrive sur la page test.

TABLE 1 – Comparaison des méthodes

Bras	Notre méthode	Bandit Bayésien	LinUCB
Gagnant	X	B	B
Gain	1457	1252	1300
Visiteurs	11168	11168	11168
Proba	X	0.69	X

construit par l’algorithme CTREE de façon à ce que la différence entre les taux de conversions des différents profils soit significative avec un indice de confiance de 95% et en évaluant l’association entre le taux de conversions et toutes les variables explicatives via une statistique de corrélation quadratique. Selon les variables impliquées (numériques ou catégorielles), différents types de scores sont utilisés dans ces tests de corrélation. Par défaut, les p -valeurs résultantes sont corrigées par une correction de Bonferroni [13] pour des tests multiples à travers le nombre de variables utilisées pour la régression.

3.2 Analyse des résultats d’observation

Afin de valider notre méthode, nous la comparons avec la même méthode de bandit, mais sans contexte (un seul bandit, pas de segment particulier), ainsi qu’avec l’algorithme LinUCB très utilisé actuellement pour ce genre de test. Nous appliquons ces trois méthodes sur le même jeu de données *exp_set*. Ainsi pour simuler la phase d’observation, nous prenons les visiteurs un à un dans le jeu de test et nous le soumettons à chacune des trois méthodes.

Pour simuler notre méthode, nous avons utilisé 10 bandits chargés d’allouer la proportion du trafic vers A ou B pour des visiteurs aux caractéristiques spécifiques (ces caractéristiques étant identifiés par les feuilles de l’arbre de pré-analyse). Ainsi, par exemple, le bandit *node3* (Figure 2) travaille sur les visiteurs associés à la fois au cluster de pages n° 3 ($ClustPage \in \{3\}$) et aux clusters de visites n° 2 ou 3 ($ClustVisite \in \{2, 3\}$). Il est à noter qu’un visiteur ne peut pas être associé à plusieurs bandits car l’arbre détermine des populations disjointes.

La méthode du bandit LinUCB identifie le meilleur bras selon différentes variables explicatives (ici les informations liées aux contextes sont les mêmes informations que nous utilisons pour réaliser notre arbre de pré-analyse). Lorsque LinUCB affecte ce «nouveau» visiteur NV à une variation différente de celle qui avait été appliquée dans la réalité, nous cherchons un visiteur NA aux mêmes caractéristiques, affecté à l’autre variation et qui n’a pas encore été proposé à l’algorithme. Nous inversons alors ce deux visiteurs NV et NA . Dans le cas (rare) où il n’existe plus de visiteurs NA dans le jeu de données, nous réalisons un tirage aléatoire parmi tous les visiteurs de même profil dans le jeu de données. Cette méthode comporte donc une part d’aléatoire. Néanmoins, les expériences ont montré que lorsque cette problématique se produit (pour les derniers visiteurs), LinUCB a en général déjà convergé vers la variation optimale.

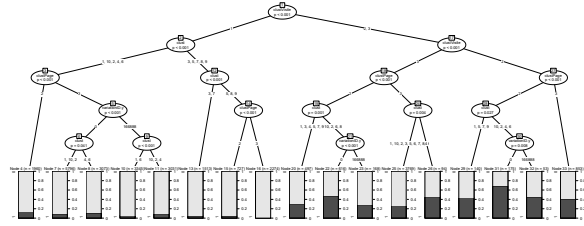


FIGURE 3 – Arbre de régression issu des données *exp_set*

Le tableau 1 compare les trois approches suivant deux critères :

1. Le nombre de conversions (gain) obtenues.
2. La probabilité d’être le meilleur bras.

Le tableau 2 reprend les résultats de notre méthode suivant les différents segments.

Notre méthode aboutit à 6 variations optimales avec une probabilité de plus de 70%. Le gain total réalisé est de 1457 conversions. Aucun nouveau visiteur n’a été alloué au bandit 5. Certains bandits convergent rapidement.

Pour la méthode du bandit binomial sans contexte prédéfini, la probabilité d’avoir identifié le meilleur bras ne dépasse les 70% qu’à partir de 4000 visiteurs. Cette méthode réalise un gain total inférieur à 14% à notre méthode. La méthode du bandit LinUCB réalise un gain moyen de l’ordre de 1300 conversions : à aucun moment, LinUCB n’a dépassé le gain obtenu par notre méthode.

Nous avons également cherché à savoir si notre méthode retrouvait les résultats réels du jeu de données initial.

La figure 4 reprend la figure 3, soit les variables discriminantes sur l’ensemble de la population testée via l’allocation aléatoire, et présente également quels ont été les profils affectés à chaque bandit. Comme l’arbre de construction des bandits a été réalisé sur une population différente (visiteur arrivé entre le 11/07/16 et 14/07/16 contre les visiteurs du 15/07/16 au 25/07/16 pour la phase de test), cet arbre est plus précis que celui de la phase d’apprentissage. Il comprend également les variations A et B qui permettent d’observer si une population a été impactée par le test (Dans notre jeu de donnée, la variation A est appelée 0 et la variation B : 166888). La variable *Clust* est le "topic" associé à un visiteur. La variable *clustPage* est le cluster construit à partir des pages parcourues, enfin *clustVisite* est le cluster construit à partir des visites réalisées.

Dans tous les cas, la variation A a été meilleure que la variation B . D’après l’arbre de la figure 4, on distingue 3 profils impactés par le test :

- Les visiteurs à la variable $clust \in \{1, 10, 2, 4, 6\}$, $clustPage \in \{3\}$ et $clustVisit \in \{1\}$. Ces visiteurs ont été traités par le bandit 16. La meilleure variation était la variation A .
- Les visiteurs à la variable $clust \in \{10, 2, 6, 8\}$, $clustPage \in \{2\}$ et $clustVisit \in \{3\}$. Ces vi-

TABLE 2 – Notre méthode

Bras	3	5	7	8	11	12	15	16	18	19	TOTAL
Gagnant	A	B	A	B	A	A	X	A	A	A	X
Gain	573	50	39	51	10	9	0	628	6	91	1457
Visiteurs	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

siteurs ont été traités par les bandits 7 et 8. La meilleure variation était la variation A.

- Les visiteurs à la variable $clust = \in \{10, 2, 4, 6\}$, $clusPage \in \{2\}$ et $clusVisit \in \{2\}$. Ces visiteurs ont été traités par les bandits 7 et 8. La meilleure variation était la variation A.

Dans notre méthode

- Le bandit 7 converge vers A avec une probabilité à 88% d’être le meilleur.
- Le bandit 8 converge vers B avec une probabilité à 63% d’être le meilleur.
- Le bandit 16 converge vers A avec une probabilité à 96% d’être le meilleur.

Notre méthode est donc aussi fiable qu’une méthode fréquentielle, lorsque la variation impacte faiblement une population, mais elle apporte les avantages supplémentaires suivants :

- Identification de la variation optimale pour une segmentation particulière.
- Convergence plus rapide vers la phase d’exploitation.
- Limitation des convergences vers la mauvaise variation.
- Gain comparable à une allocation aléatoire lorsque la variation impacte une faible proportion de la population.

3.3 Conclusion et travaux futurs

Une des problèmes majeurs des bandits dans un trafic particulièrement volatil qu’est celui du e-commerce est que la convergence vers une mauvaise solution est difficile à contrer. En utilisant des bandits de façon indépendantes, nous limitons les erreurs d’identification de la meilleure variation. De par notre utilisation du clustering, nous tirons profit des profils issus des parcours et des centres d’intérêt des visiteurs. Ces profils permettent de limiter l’information émise pour le contexte et anticipent la création trop importante de bandits, qui seraient alors soumis à un trafic trop limité pour converger. Actuellement nous avons automatisé le nombre de sujets à définir et testé cette approche avec d’autres e-commerçants. Nous travaillons également sur la qualité des clusters, et abordons les sujets d’intérêt de visiteurs de façon évolutive. Enfin, nous souhaitons utiliser des paramètres relatifs au trafic global du site afin d’anticiper les "pics" et "creux" de trafic qui peuvent nuire à la phase d’exploration (en période de faible trafic, les différences entre les variations sont plus réduites).

Références

- [1] S. Roukine, *Améliorer ses taux de conversions web : vers la performance des sites web au-delà du webmarketing*, Eyrolles, 2011.
- [2] E. Kaufmann, O. Cappé et A. Garivier, On the Complexity of A/B Testing, *Conference on Learning Theory, Barcelona, Spain, JMLR : Workshop and Conference Proceedings*, Vol. 35, pp 461-481, 2014.
- [3] A. Wald, Sequential tests of statistical hypotheses. *Annals of Mathematical Statistics*, Vol. 16, pp 117-186, 1945.
- [4] A. Petitjean, A. Ketterlin et P. Gançarski, A global averaging method for dynamic time warping, with applications to clustering *Pattern Recognition archive*, Vol. 44, pp 678-693, 2011.
- [5] A. Sarda-Espinosa, 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>
- [6] B. Grun et K. Hornik, An R Package for Fitting Topic Models. *Journal of Statistical Software*, Vol. 40, pp 1-30, 2011.
- [7] H. Strasser et C. Weber On the Asymptotic Theory of Permutation Statistics *Mathematical Methods of Statistics*, Vol. 8, pp 220–250, 1991
- [8] T. Hothorn, K. Hornik et A. Zeileis, Unbiased Recursive Partitioning : A Conditional Inference Framework, *Journal of Computational and Graphical Statistics*, Vol. 15, pp 651–674, 2006.
- [9] AM. Molinaro, S. Dudoit et MJ. van der Laan, Tree-Based Multivariate Regression and Density Estimation with Right-Censored Data, *Journal of Multivariate Analysis*, Vol. 90, pp 154–177, 2004.
- [10] , L. Li, W. Chu, J. Langford, John et E. Schapire, A contextual-bandit approach to personalized news article recommendation, *Proceedings of the Nineteenth International Conference on the World Wide Web*, pp 661-670, 2010.
- [11] S. Scott, A Modern Bayesian Look at the Multi-armed Bandit, *Applied Stochastic Models Business and Industry*, Vol. 26, pp 639–658, 2010.
- [12] P. Auer, N. Cesa-Bianchi, Y. Freund, et R. E. Schapire, The nonstochastic multiarmed bandit problem, *SIAM Journal on Computing*, Vol. 32 pp 48–77, 2002.

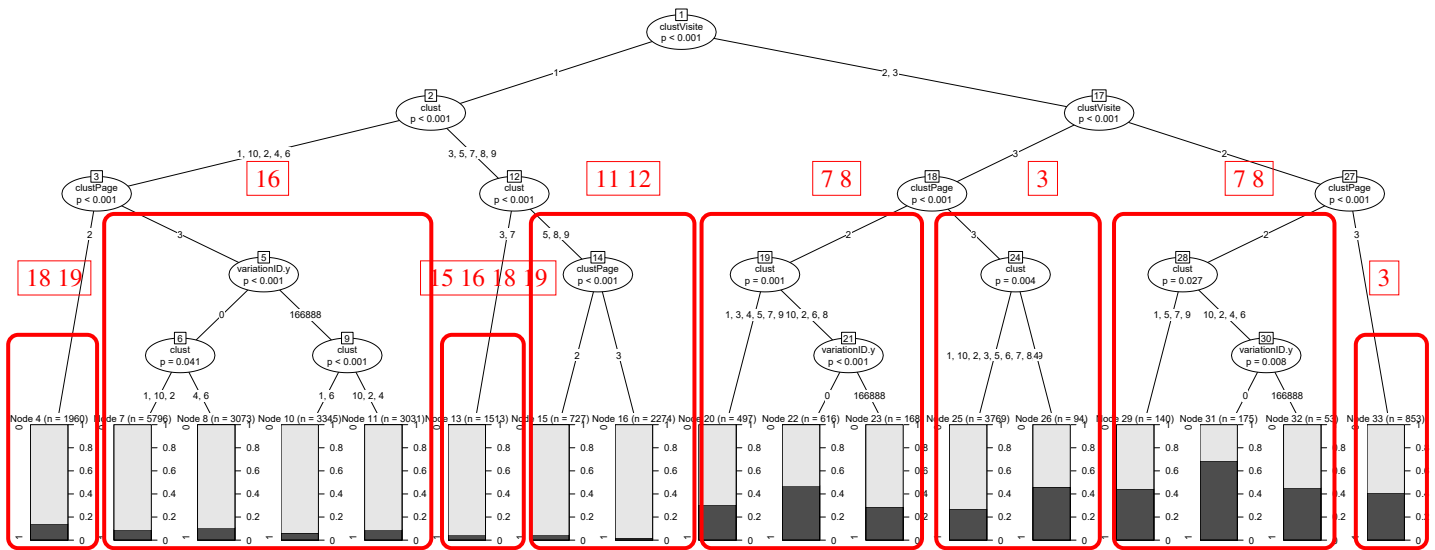


FIGURE 4 – Arbre de régression de *exp_set* (avec bandits)

[13] W-Y. Loh Fifty Years of Classification and Regression Trees *International Statistical Review*, Vol. 82 pp 329–348, 2014.