Bandit Contextuel pour la Capture de Données Temps Réel sur les Médias Sociaux

Thibault Gisselbrecht[*+] — **Sylvain Lamprier**[*] — **Patrick Gallinari**[*]

[*] *Sorbonne Universités, UPMC Univ Paris 06*
CNRS, LIP6 UMR 7606, 4 place Jussieu, 75005 Paris, France
prénom.nom@lip6.fr
[+] *IRT SystemX, 8 Avenue de la Vauve, 91120 Palaiseau, France*
thibault.gisselbrecht@irt-systemx.fr

RÉSUMÉ. *La plupart des médias sociaux offrent un accès aux flux de données produites par leurs utilisateurs. L'utilisation des API fournies pour collecter ces données, relativement à un besoin spécifique, peut se révéler être une tâche complexe car elle nécessite une sélection soigneuse des sources. Cela représente un problème particulièrement difficile dans les réseaux sociaux de grandes tailles étant donné le nombre important d'utilisateurs potentiellement intéressants, la non-stationnarité intrinsèque de leur comportement, et les restrictions d'accès aux données. Dans cet article, nous proposons une approche permettant d'anticiper les profils les plus susceptibles de publier des contenus pertinents et de sélectionner un sous ensemble de comptes à chaque itération. Nous formalisons cette tâche comme un problème de bandit contextuel avec sélections multiples. Les expérimentations menées sur le réseau social Twitter montrent l'efficacité de notre approche dans un scenario réel.*

ABSTRACT. *Social media usually provide streaming data access that enable dynamic capture of the social activity of their users. Leveraging such APIs for collecting data that satisfy a given pre-defined need may constitute a complex task, that implies careful stream selections. On large social media, this represents a very challenging task due to the huge number of potential targets, the intrinsic non-stationarity of user's behavior, and restricted access to the data. We propose an approach that anticipates which profiles are likely to publish relevant contents and dynamically selects a subset of accounts to follow at each iteration using a contextual bandit algorithm. We conduct experiments on Twitter that demonstrate the empirical effectiveness of our approach in real-world settings.*

MOTS-CLÉS : *Bandit Manchot, Apprentissage Statistique, Médias Sociaux.*

KEYWORDS: *Multi-armed Bandit, Machine Learning, Social Media.*

1. Introduction

Au cours des dernières années, de nombreux réseaux sociaux permettant aux utilisateurs de publier et de partager du contenu en ligne ont fait leur apparition. Par exemple Twitter, avec 302 millions d'utilisateurs actifs et plus de 500 millions de messages chaque jour, est l'un des principaux acteurs du marché. Ces médias sociaux sont devenus une source de données très importante pour de nombreuses applications. Étant donné un besoin pré-défini, deux solutions sont habituellement disponibles pour collecter des données à partir de ces médias : 1) avoir accès à de grandes bases de données ou 2) utiliser les services de *streaming* que la plupart des médias proposent et qui permettent le suivi en temps réel de l'activité de leurs utilisateurs. Tandis que la première solution implique généralement des coûts très importants, la seconde constitue une alternative moins onéreuse. Toutefois, cela implique d'être en mesure de choisir efficacement quels flux de données sont les plus pertinents. Dans cet article, nous considérons des flux de données centrés sur les utilisateurs, chaque flux fournissant l'accès aux données publiées par un utilisateur particulier, avec des contraintes restrictives liées au nombre de flux pouvant être considérés simultanément (5000 sur Twitter). Étant donné le grand nombre d'utilisateurs (302 millions sur Twitter), le ciblage des sources pertinentes représente un effort non négligeable.

Imaginons une personne intéressée par l'informatique et souhaitant rester informée des dernières nouvelles relayées par les utilisateurs en relation avec ce sujet. Une solution consiste à sélectionner manuellement un ensemble de comptes et suivre leur activité au cours du temps. Cependant, Twitterest connu pour être extrêmement dynamique et pour une raison ou une autre, des utilisateurs pourraient se mettre à poster sur ce sujet tandis que d'autres pourraient cesser à tout moment. Donc, si l'utilisateur intéressé veut être à jour, il devrait changer le sous-ensemble des comptes suivis dynamiquement, son but étant d'anticiper les utilisateurs qui sont les plus susceptibles de produire des contenus pertinents dans un futur proche. Cette tâche semble toutefois difficile à exécuter manuellement pour deux raisons majeures. Tout d'abord, les critères choisis pour prédire si un compte sera potentiellement intéressant dans un futur proche peuvent être difficile à définir. Deuxièmement, même si nous étions en mesure de définir manuellement ces critères, la quantité de données à analyser serait trop grande pour pouvoir réaliser cette sélection dans un temps raisonnable.

Au vu de ces difficultés, nous proposons une solution qui, à chaque itération du processus, sélectionne automatiquement un sous-ensemble de comptes qui sont susceptibles d'être pertinents dans la fenêtre de temps suivante, en fonction de leur activité actuelle. Ces comptes sont ensuite suivis pendant un certain temps et les contenus publiés correspondants sont évalués - par l'utilisateur ou par un classifieur - afin de quantifier leur pertinence. L'algorithme derrière le système apprend alors une politique visant à améliorer la sélection à l'étape d'après. Nous abordons cette tâche comme un problème de bandit contextuel, dans lequel à chaque étape, l'algorithme choisit une action parmi un plus grand ensemble d'actions disponibles, sur la base des observations des caractéristiques de chaque action - le contexte de l'action - et reçoit une récompense qui quantifie la qualité de l'action choisie. Dans notre cas, étant

donné que suivre une personne correspond à une action, plusieurs actions peuvent être sélectionnées à chaque pas de temps, car nous voulons utiliser la capacité maximale de capture autorisée par l'API de *streaming*.
En définissant le contexte d'un utilisateur comme son activité présente, notre tâche s'inscrit bien le cadre du bandit contextuel. Cependant, dans son instance traditionnelle, le contexte de chaque action doit être observé à chaque itération afin d'effectuer les choix successifs et pour permettre l'apprentissage d'une politique. Dans notre cas, nous ne sommes pas en mesure d'observer l'activité présente de chaque utilisateur potentiel. Par conséquent, nous ne sommes pas en mesure d'obtenir tous les contextes et donc d'utiliser des algorithmes de bandits contextuels classiques tels que *LinUCB* (Chu *et al.*, 2011). D'autre part, nous ne pouvons pas traiter cela comme un problème de bandit traditionnel (non-contextuel) puisque nous perdrions les informations utiles fournies par les contextes observés. A notre connaissance, cette instance hybride du problème de bandit n'a pas encore été étudiée. Pour résoudre ce problème, nous proposons un algorithme de capture de données qui, en plus d'apprendre une politique de sélections des k meilleurs flux à chaque itération, apprend sur les distributions des contextes eux-mêmes, afin d'être en mesure de faire des estimations lorsque ces derniers ne sont pas observés.

2. État de l'art

Le problème du bandit manchot, proposé dans (Lai et Robbins, 1985) sous sa forme stationnaire a été largement étudié dans la littérature. Dans sa première instance, l'agent n'a accès à aucune information relative aux actions disponibles, mais fait l'hypothèse de distributions stationnaires sur les récompenses de chaque action. Un grand nombre de méthodes ont été proposées pour définir des politiques de sélection efficace, avec les garanties théoriques de convergence correspondantes. Le célèbre algorithme *UCB*, initialement proposé dans (Auer *et al.*, 2002), est basé sur la borne supérieure de l'intervalle de confiance des récompenses espérées pour chaque action disponible. Cet algorithme et ses très nombreuses variantes (Audibert *et al.*, 2007 ; Audibert et Bubeck, 2009) se sont révélés très efficaces pour la résolution du problème du bandit *stochastique*.
Dans (Chu *et al.*, 2011) entre autres, les auteurs étudient le problème du bandit contextuel, dans lequel l'agent observe des attributs sur chaque action avant d'effectuer la sélection, et proposent l'algorithme *LinUCB*. Cet algorithme fait l'hypothèse que la récompense espérée d'une action est une fonction linéaire de ses attributs et de paramètres inconnus à estimer. Que ce soit pour le bandit stochastique ou contextuel, d'autres types de stratégies appelées *Thompson Sampling* ont été proposés (Agrawal et Goyal, 2012a ; Chapelle et Li, 2011). Plus récemment, le cas où il est possible de sélectionner plusieurs actions simultanément a été formalisé dans (Chen *et al.*, 2013) (algorithme *CUCB*) et (Qin *et al.*, 2014) (algorithme C^2UCB) respectivement pour le cas non-contextuel et le cas contextuel.
Concernant les médias sociaux, plusieurs tâches existantes présentent des similarités

avec la nôtre. Dans (Li *et al.*, 2013), les auteurs construisent une plate-forme appelée ATM qui vise à cibler automatiquement des tweets en rapport avec un certain sujet. Ils développent un algorithme qui sélectionne efficacement des mots-clés pour couvrir les tweets cibles. Dans notre cas, nous ne nous concentrons pas sur des mots clés, mais sur les profils des utilisateurs. Dans (Colbaugh et Glass, 2011), les auteurs modélisent la blogosphère comme un réseau où chaque nœud est considéré comme un flux de données pouvant être suivi. Leur objectif est d'identifier des blogs qui contiennent des contenus pertinents pour suivre les sujets émergents. Cependant, leur approche est statique et leur modèle est basé sur des données déjà collectées. Dans (Hannon *et al.*, 2010), les auteurs construisent un système de recommandation appelé *Twittomender*, utilisant des méthodes de filtrage collaboratif pour trouver les comptes Twitter susceptibles d'intéresser un utilisateur cible. Les auteurs supposent la connaissance du graphe des *followers/followees*, ce qui est impossible pour les applications à grande échelle en raison des politiques de restriction de Twitter.
Les algorithmes de bandit ont déjà été appliqués à diverses tâches liées aux réseaux sociaux. Par exemple, dans (Kohli *et al.*, 2013), les auteurs traitent des tâches de minimisation d'abandon dans les systèmes de recommandation. Dans (Buccapatnam *et al.*, 2014), les bandits sont utilisés pour la publicité en ligne tandis que dans (Lage *et al.*, 2013), les auteurs utilisent un bandit contextuel pour une tâche de maximisation de l'audience. Dans (Gisselbrecht *et al.*, 2015), une tâche de capture de données similaire est abordée via des algorithmes de bandits non contextuels. Dans ce cas, chaque utilisateur est supposé avoir une distribution stationnaire et le processus doit trouver des utilisateurs avec les meilleures moyennes en jouant sur l'exploitation des bonnes sources déjà connues et l'exploration des inconnues. Notre approche diffère de ce travail principalement par la considération d'une hypothèse de non stationnarité de l'utilité des différents utilisateurs. Nous considérons toutefois ce dernier dans nos expérimentations pour nous comparer.

3. La Sélection de Flux de Données

3.1. *Contexte*

Notre objectif est de construire un système permettant de capturer des données pertinentes à partir des API proposées par les médias sociaux. Étant donné que les API limitent généralement le nombre d'utilisateurs pouvant être suivis de façon simultanée, l'objectif est de sélectionner dynamiquement ceux qui sont susceptibles de publier des contenus pertinents relativement à un besoin prédéfini. Il est important de noter que, quand bien même il n'y aurait pas de telles restrictions, notre approche reste valide vu l'énorme quantité de données publiées. La principale difficulté est donc de sélectionner efficacement les comptes à suivre étant donné le grand nombre d'utilisateurs et sachant qu'au début du processus, aucune information sur ces derniers n'est connue. En outre, même si certains comptes spécifiques pourraient être trouvés manuellement, le changement dynamique des sources semble complexe à réaliser ma-

nuellement. Pour ces raisons, la construction d'une solution automatique pour orienter le choix des sources apparaît pertinent.

3.2. *Un Processus de Décision sous Contraintes*

Notre tâche revient à sélectionner à chaque itération t du processus un sous-ensemble $\mathcal{K}_t$ de k profils à suivre parmi l'ensemble de tous les utilisateurs $\mathcal{K}$ ($\mathcal{K}_t \subseteq \mathcal{K}$), selon leur propension à poster des tweets pertinents. Étant donné un score $r_{a,t}$ associé au contenu produit par l'utilisateur a à l'étape t, le but est de sélectionner le sous-ensemble d'utilisateurs maximisant la somme des scores de pertinence :

$$\max_{(\mathcal{K}_t)t=1..n} \quad \sum_{t=1}^{n} \sum_{a \in \mathcal{K}_t} r_{a,t} \qquad [1]$$

Notons que, contrairement à la majorité des tâches classiques en recherche information, nous ne nous préoccupons pas ici de la précision des informations collectées. L'objectif est de maximiser le volume de messages pertinents collectés, un filtrage pouvant être éventuellement appliqué a posteriori. Cet aspect diffère des tâches habituelles en recherche d'information. Les scores de pertinence considérés dépendent d'un besoin d'information, défini par l'utilisateur du système, pouvant prendre des formes variées. Par exemple, l'utilisateur pourrait souhaiter suivre des profils actifs sur des thématiques précises, ou bien des profils influents (au sens où leurs messages sont souvent repostés par d'autres utilisateurs). Ces scores peuvent soit être assignés manuellement, relativement à une évaluation humaine des contenus, ou bien automatiquement, par exemple à l'aide d'un classifieur (voir partie 5).
Dans (Gisselbrecht *et al.*, 2015) les auteurs se basent sur une hypothèse de stationnarité des scores associés à chaque utilisateur, ce qui paraît une hypothèse peu réaliste sur des réseaux sociaux généralistes tels que Twitter, où les utilisateurs peuvent changer de centre d'intérêt à tout moment. Nous supposons ici qu'il est possible de mieux anticiper le comportement futur des utilisateurs (i.e. les contenus futurs) en considérant leur activité présente (i.e. les contenus déjà postés). En d'autres termes, si l'on représente l'activité de l'utilisateur a au temps $t-1$ par un vecteur $z_{a,t} \in \mathbb{R}^d$, il existe une fonction $h : \mathbb{R}^d \to \mathbb{R}$ permettant d'expliquer le score de a au temps t. Cette fonction de corrélation doit être apprise au fur et à mesure du processus de décision.
Cependant, la maximisation de la fonction définie dans la formule 1 est contrainte par différents facteurs :

– Les scores de pertinences $r_{a,t}$ sont seulement définis pour les utilisateurs suivis à l'itération t (i.e. pour les $a \in \mathcal{K}_t$) ;

– Les vecteurs de contextes $z_{a,t}$ sont seulement observés pour un sous-ensemble $\mathcal{O}_t$ des utilisateurs (i.e. il est impossible d'observer la totalité de l'activité du réseau) ;

Ces contraintes proviennent des restrictions liées aux API de *streaming* que nous utilisons pour la collecte des données :

– D'une part, une API *Sample streaming*, qui renvoie en temps réel 1% de tous les tweets publics. Nous utilisons cette API pour découvrir de nouveaux utilisateurs mais aussi pour récolter les contextes (activités) d'un grand nombre d'utilisateurs actifs à un moment donné.

– D'autre part, une API *Follow streaming*, qui permet d'obtenir en temps réel les contenus produits par 5000 utilisateurs définis. Nous utilisons cette API pour capturer les données produites par les utilisateurs sélectionnés par notre système.

Pour les profils appartenant à $\mathcal{O}_t$, les scores de pertinences peuvent être estimés grâce à la fonction de corrélation h, qui traduit les variations temporelles de la qualité d'un utilisateur. En revanche, pour les autres, nous proposons de considérer le cas stationnaire, en introduisant une tendance générale (ou qualité intrinsèque) pour chaque utilisateur. Nous sommes alors face à un problème hybride, où les contextes observés sont utilisés pour prédire des variations autour d'une utilité moyenne estimée.

3.3. *Description du système*

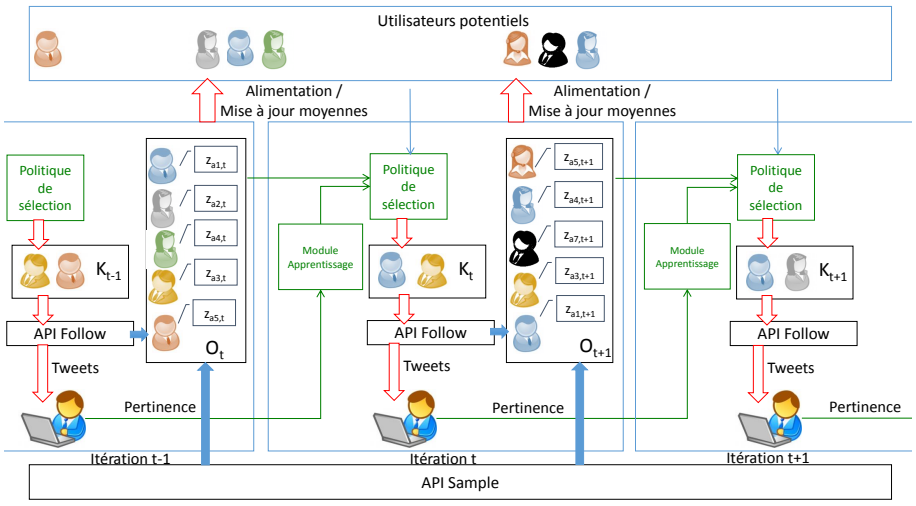

Figure 1. *Illustration du système.*

La figure 1 illustre le fonctionnement général du système. Ce dernier réitère les mêmes actions à chacune des itérations (trois itérations sont représentées). Au début de chaque étape, la politique de sélection des sources choisit un ensemble d'utilisateurs à suivre ($\mathcal{K}_t$) parmi tous les utilisateurs connus du système ($\mathcal{K}$), selon des observations et des connaissances fournies par un module d'apprentissage. Ensuite, les messages postés par les k profils sélectionnés sont collectés via l'API *Follow streaming*. Comme nous pouvons le voir dans la partie centrale du schéma, après avoir suivi

les utilisateurs de $\mathcal{K}_t$ pendant l'itération courante t, les messages collectés sont analysés (soit par un humain, soit automatiquement) et le résultat - traduisant la pertinence - est renvoyé au module d'apprentissage.
En parallèle, à chaque itération, l'activité courante de certains utilisateurs est capturée par l'API *Sample streaming*. Cela nous offre la possibilité d'enrichir la base des utilisateurs potentiels $\mathcal{K}$, mais aussi de construire l'ensemble des utilisateurs dont on connaît le contexte. Chaque profil ayant publié au moins un message parmi ceux qui ont été collecté via l'API *Sample streaming* à l'étape t sont inclus dans $\mathcal{O}_{t+1}$, les messages d'un même auteur a étant concaténés pour former $z_{a,t+1}$ (voir la partie 5 pour des exemples de construction de vecteurs de contexte à partir des messages). À cet ensemble sont ajoutés les utilisateurs de $\mathcal{K}_t$, puisque leur activité complète a été suivie pendant l'étape t.

4. Modèle et Algorithme

4.1. *Une approche de Bandit Contextuel*

Dans le problème du bandit contextuel, l'hypothèse classique considère l'existence d'un vecteur inconnu $\beta \in \mathbb{R}^d$ permettant d'estimer l'espérance de récompense que l'on peut observer pour chaque action et à chaque itération du processus : $\forall t \in \{1, .., T\}, \forall a \in \mathcal{K} : \mathbb{E}[r_{a,t}|z_{a,t}] = z_{a,t}^T \beta$ (Agrawal et Goyal, 2012b). Dans notre cas, afin de modéliser la qualité intrinsèque de chaque action (une action correspondant à la sélection d'un utilisateur), nous introduisons également un terme de biais $\theta_a \in \mathbb{R}$ pour chaque action a. Ainsi, sur les T itérations de notre processus de collecte :

$$\forall t \in \{1, .., T\}, \forall a \in \mathcal{K} : \mathbb{E}[r_{a,t}|z_{a,t}] = z_{a,t}^T \beta + \theta_a \quad [2]$$

Cette formulation correspond à un cas particulier de l'algorithme *LinUCB* hybride proposé dans (Li *et al.*, 2010) en considérant une valeur de 1 pour la valeur du contexte propre à chaque action. Il est important de noter que plusieurs paramètres individuels auraient pu être considérés dans notre étude, cependant cette approche ne nous semble pas adaptée étant donné le nombre très élevé d'actions disponibles (qui mènerait à un apprentissage d'autant plus complexe). D'autre part, l'utilisation d'un vecteur de paramètres commun permet une exploration facilitée, cela permettant la généralisation de corrélations observées pour des utilisateurs à l'ensemble des utilisateurs à l'activité courante proche. Ainsi, nous restreignons la modélisation individuelle à un terme unique de biais.
Comme précisé plus haut, notre tâche nécessite la sélection de multiples actions simultanément et donc à chaque itération t, l'algorithme doit sélectionner k ($k < K$) utilisateurs parmi un ensemble de K, selon les contextes observés et les qualités intrinsèques de ces derniers. Le gain obtenu après une période d'écoute correspond donc à la somme des gains individuels. Dans notre cas, une difficulté majeure provient du fait que tous les contextes ne sont pas observables, contrairement aux problèmes de

bandits contextuels traditionnels. Dans la suite de ce travail, nous considérons pour chaque utilisateur une probabilité p_a ($0 < p_a < 1$) de révéler son contexte. Notons que cette probabilité est contrainte par le problème considéré. Il s'agit alors de choisir à chaque itération t les k meilleurs utilisateurs à suivre parmi les K utilisateurs de $\mathcal{K} = \mathcal{O}_t \cup \bar{\mathcal{O}}_t$, avec $\bar{\mathcal{O}}_t$ l'ensemble des utilisateurs pour lesquels on ne connaît pas le contexte pour l'étape t. La prochaine sous-section concerne le cas où l'on observe le contexte de tous les utilisateurs, la sous-section suivante étendant ce travail pour la prise en compte des utilisateurs de $\bar{\mathcal{O}}_t$. Enfin, cette section se termine par une description de l'algorithme de collecte global.

4.2. *Une approche contextuelle pour la collecte de données dynamique*

Afin de dériver notre algorithme de collecte, nous considérons les hypothèses suivantes :

– **Vraisemblance :** Les scores de récompense sont identiquement et indépendamment distribués selon les contextes observés : $r_{a,t} \sim \mathcal{N}(z_{a,t}^T\beta + \theta_a, \lambda_a)$, avec λ_a correspondant à la variance de l'écart entre la récompense $r_{a,t}$ et l'application linéaire de prédiction $z_{a,t}^T\beta + \theta_a$;

– **Prior :** Les paramètres inconnus sont normalement distribués autour de l'origine : $\beta \sim \mathcal{N}(0, b\mathbb{I}_d)$ et $\theta_a \sim \mathcal{N}(0, \rho_a)$, où b et ρ_a permettent de contrôler la variance des paramètres et $\mathbb{I}_d$ est la matrice identité de taille d (taille des vecteurs de contexte).

L'idée est alors de construire un estimateur des paramètres β et θ par maximum a posteriori connaissant les récompenses collectées, et les contextes associés, jusqu'à l'itération courante. Pour simplifier, nous fixons b, ρ_a et λ_a à 1 pour tout a. Cependant, tous les résultats peuvent être étendus à des cas plus complexes.

Proposition 1 *En notant T_a l'ensemble des itérations où a a été choisi durant les n premières étapes ($T_a = \{t \leq n, a \in \mathcal{K}_t\}$, $|T_a| = \tau_a$), c_a le vecteur de récompenses obtenues par a lorsqu'il a été écouté ($c_a = (r_{a,t})_{t \in T_a}$) et D_a la matrice des contextes associée à a ($D_a = (z_{a,t}^T)_{t \in T_a}$), la distribution a posteriori de β et θ_a après n étapes, lorsque tous les contextes sont disponibles, est donnée par*[1] *:*

$$\beta \sim \mathcal{N}\left(\bar{\beta}, A_0^{-1}\right) \qquad [3] \qquad \theta_a + \bar{z}_a^T\beta \sim \mathcal{N}\left(\bar{\mu}_a, \frac{1}{\tau_a + 1}\right) \qquad [4]$$

$$\text{Avec : } A_0 = \mathbb{I}_d + \sum_{a=1}^{K}(\tau_a + 1)\bar{\Sigma}_a \ ; \ b_0^T = \sum_{a=1}^{K}(\tau_a + 1)\bar{\xi}_a \ ; \ \bar{\Sigma}_a = \frac{D_a^T D_a}{\tau_a + 1} - \bar{z}_a\bar{z}_a^T$$

$$\bar{\xi}_a = \frac{c_a^T D_a}{\tau_a + 1} - \bar{\mu}_a\bar{z}_a^T \ ; \ \bar{\beta} = A_0^{-1}b_0^T \ ; \ \bar{\mu}_a = \frac{\sum_{t \in T_a} r_{a,t}}{\tau_a + 1} \ ; \ \bar{z}_a = \frac{\sum_{t \in T_a} z_{a,t}}{\tau_a + 1}$$

[1] Les preuves complètes des propositions sont disponibles à l'URL `http://www-connex.lip6.fr/~lampriers/CORIA2016-supplementaryMaterial.pdf`

Notons que tous ces paramètres possèdent une dépendance en t, que nous avons volontairement omise afin d'alléger les notations. De plus tous ces paramètres peuvent être mis à jour de façon peu coûteuse à mesure que de nouveaux exemples d'apprentissage arrivent (la complexité de la mise à jour des paramètres est constante sur l'ensemble du processus). En combinant les équations 2, 3 et 4, la valeur espérée $\mathbb{E}[r_{a,t}|z_{a,t}]$, de la récompense de l'utilisateur a à l'itération t sachant son contexte $z_{a,t}$, suit la distribution postérieure :

$$\mathcal{N}\left(\bar{\mu}_a + (z_{a,t} - \bar{z}_a)^T\bar{\beta}, \frac{1}{\tau_a + 1} + (z_{a,t} - \bar{z}_a)^T A_0^{-1}(z_{a,t} - \bar{z}_a)\right) \quad [5]$$

Theorème 1 *Pour tout* $0 < \delta < 1$ *et* $z_{a,t} \in \mathbb{R}^d$*, en notant* $\alpha = \sqrt{2}erf^{-1}(1-\delta)$ [2]*, pour chaque action* a*, après* t *itérations*[1] *:*

$$P\Big(|\mathbb{E}[r_{a,t}|z_{a,t}] - \bar{\mu}_a - (z_{a,t} - \bar{z}_a)^T\bar{\beta}| \ \leq \ \alpha\,\sigma_a\Big) \geq (1-\delta)$$

$$\textit{avec } \sigma_a = \sqrt{\frac{1}{\tau_a + 1} + (z_{a,t} - \bar{z}_a)^T A_0^{-1}(z_{a,t} - \bar{z}_a)} \quad [6]$$

Cette formule peut directement être utilisée pour définir une borne supérieure de l'intervalle de confiance (*UCB - Upper Confidence Bound*) de la récompense espérée pour chaque utilisateur dont le contexte a été observé pour l'itération t :

$$s_{a,t} = \bar{\mu}_a + (z_{a,t} - \bar{z}_a)^T\bar{\beta} + \alpha\,\sigma_a \quad [7]$$

De manière classique, les approches type *UCB* étant des approches optimistes, l'idée est alors de sélectionner à chaque instant t les k utilisateurs ayant les k bornes $s_{a,t}$ les plus élevées. Nous rappelons qu'une formulation similaire est utilisée dans le bandit contextuel traditionnel. Dans la partie suivante, nous proposons une méthode permettant d'adapter cette dernière au cas où l'on ne peut pas observer tous les contextes.

4.3. *Prise en compte des utilisateurs avec contexte caché*

Les scores permettant de sélectionner les utilisateurs à écouter définis précédemment ne peuvent pas être calculés lorsque les contextes $z_{a,t}$ ne sont pas observés (i.e. lorsque les utilisateurs ne sont pas dans $\mathcal{O}_t$). Cela nécessite donc une méthode permettant de qualifier les personnes dont le contexte est inconnu. De plus, cela implique des problématiques de mise à jour des paramètres appris lorsqu'un utilisateur a été écouté mais dont le contexte n'a pas été révélé. Bien qu'il soit tentant de penser que le paramètre β commun à tous et les paramètres individuels θ_a peuvent être appris séparément, ces derniers sont en réalité corrélés, comme le montrent les formules 3

[2] erf^{-1} est la fonction erreur inverse, $erf(x) = 2/\pi \int_0^x e^{-t^2}\,\mathrm{d}x$.

et 4. Il est important de remarquer que pour conserver les garanties probabilistes, les paramètres du problème ne devraient être mis à jour que lorsqu'un profil appartient à $\mathcal{K}_t \cap \mathcal{O}_t$. Remplacer T_a par $T_a^{both} = \{t \leq n, a \in \mathcal{K}_t \cap \mathcal{O}_t\}$ (nous conservons la notation $|T_a^{both}| = \tau_a$), nous permet de réutiliser les formules de mise à jour des paramètres de la Proposition 1. Toutefois, le calcul du score de sélection d'un utilisateur dont le contexte est caché ne peut être effectué grâce à l'équation 7. Pour palier à ce problème, nous proposons d'utiliser un estimateur de la distribution moyenne du vecteur de contexte de chaque profil.

Nouvelles notations :

– L'ensemble des itérations où le contexte de a a été observé au bout de n étapes est noté $T_a^{obs} = \{t \leq n, a \in \mathcal{O}_t\}$ avec $|T_a^{obs}| = n_a$.

– La moyenne empirique des vecteurs de contexte de l'utilisateur a est noté $\hat{z}_a$, avec $\hat{z}_a = 1/n_a \sum_{t \in T_a^{obs}} z_{a,t}$. Remarquons que $\hat{z}_a$ est différent de $\bar{z}_a$ car le premier est mis à jour à chaque fois que le contexte est observé tandis que le second l'est uniquement lorsque l'utilisateur a été observé et sélectionné lors d'une même itération.

Hypothèses supplémentaires :

– Sans perte de généralité, nous supposons que β est borné par $M \in \mathbb{R}^{+*}$, i.e. $||\beta|| \leq M$, où $||\ ||$ désigne la norme L^2 sur $\mathbb{R}^d$.

– Pour tout a et à chaque t, les contextes $z_{a,t} \in \mathbb{R}^d$ sont iid depuis une distribution inconnue de moyenne $\mathbb{E}[z_a]$ et variance Σ_a finies.

Theorème 2 *Étant donné $0 < \delta < 1$ et $0 < \gamma < \frac{1}{2}$, en notant $\alpha = \sqrt{2} erf^{-1}(1-\delta)$, pour tout a après t itérations du processus*[1] *:*

$$P\left(|\mathbb{E}[r_a] - \bar{\mu}_a - (\hat{z_a} - \bar{z_a})^T \bar{\beta}| \leq \alpha\ \hat{\sigma}_a + \frac{1}{n_a^\gamma}\right) \geq (1-\delta)\left(1 - \frac{C_a}{n_a^{1-2\gamma}}\right)$$

$$\text{avec } \hat{\sigma}_a = \sqrt{\frac{1}{\tau_a + 1} + (\hat{z_a} - \bar{z_a})^T A_0^{-1} (\hat{z_a} - \bar{z_a})} \quad [8]$$

Où C_a est une constante positive spécifique à chaque utilisateur.

Étant donné que $\gamma < \frac{1}{2}$, la probabilité précédente tend vers $1 - \delta$ à mesure que le nombre d'observations du contexte de l'utilisateur a augmente (comme dans l'équation 6). Ainsi, l'inégalité ci-dessus nous donne une borne relativement serrée de l'intervalle de confiance associé à la récompense espérée de l'utilisateur a, d'où nous pouvons dériver un nouveau score de sélection. À chaque itération t, si le contexte de a n'est pas observé :

$$s_{a,t} = \bar{\mu}_a + (\hat{z_a} - \bar{z_a})^T \bar{\beta} + \alpha\ \hat{\sigma}_a + \frac{1}{n_a^\gamma} \quad [9]$$

Notons que puisque $\gamma > 0$, le terme d'exploration supplémentaire $1/n_a^\gamma$ tend vers 0 à mesure que le nombre d'observations de a augmente, et le score tend donc vers un score *LinUCB* classique (comme dans l'équation 7) dans lequel le contexte est remplacé par sa moyenne empirique $\hat{z_a}$. Finalement, comme chaque utilisateur a une probabilité $0 < p_a < 1$ de révéler son contexte à chaque temps t, nous pouvons affirmer que $1/n_a^\gamma$ tend vers 0 lorsque t augmente.

4.4. *Algorithme de Capture de Données Contextuel*

Les ensembles et paramètres manipulés sont tout d'abord initialisés :

– A_0 par une matrice identité de taille $d \times d$ et b_0 par un vecteur nul de taille d, avec d la taille des vecteurs de contexte ;

– $\mathcal{K}$ par l'ensemble vide ;

– $\mathcal{O}_1$ par les utilisateurs auteurs de messages collectés au cours d'une première utilisation de l'API *Sample* pendant une période de temps d'une durée $\mathcal{L}$. La représentation des messages de ces utilisateurs $a \in \mathcal{O}_1$ correspondent à leur contexte $z_{a,1}$.

L'algorithme procède ensuite, pour chaque itération t de la collecte[3], selon :

1) Insertion dans $\mathcal{K}$ de chaque utilisateur de $\mathcal{O}_t$ n'en faisant pas encore partie (en limitant le nombre de nouveau entrants à $newMax$ utilisateurs par itération afin d'éviter une sur-exploration pour le cas des très grands réseaux tels que Twitter) ;

2) Mise à jour des moyennes empiriques des contextes $\hat{z}_a$ pour les profils de $\mathcal{O}_t$;

3) Calcul de l'estimateur du paramètre $\bar{\beta}$ (voir proposition 1) ;

4) Calcul des scores de sélection $s_{a,t}$ de tous les utilisateurs dans $\mathcal{K}$, selon la formule 7 pour les utilisateurs de $\mathcal{O}_t$ et selon la formule 9 ceux de $\bar{\mathcal{O}}_t$. Le score des nouveaux utilisateurs est fixé à $+\infty$ pour forcer le système à les choisir une première fois ;

5) Sélection des k profils ayant les meilleurs scores $s_{a,t}$;

6) Capture simultanée via les APIs *Sample* et *Follow* pendant une période de durée $\mathcal{L}$. L'API *Follow* écoute les k utilisateurs sélectionnés $\mathcal{K}_t$;

7) Calcul des récompenses des messages collectés pour les utilisateurs de $\mathcal{K}_t$ à partir de l'API *Follow*, puis mise à jour des paramètres du modèle ;

8) Mise à jour de l'ensemble des utilisateurs observés $\mathcal{O}_{t+1}$, et des contextes associés, en considérant tous les utilisateurs pour lesquels au moins un message a été collecté pendant l'itération t (provenant soit de l'API *Sample* soit de l'API *Follow*) ;

[3] L'algorithme complet est disponible à l'URL `http://www-connex.lip6.fr/~lampriers/CORIA2016-supplementaryMaterial.pdf`

5. Expérimentations

5.1. *Description générale*

Outre une politique (appelée *Random*) sélectionnant uniformément les utilisateurs à suivre à chaque itération, nous comparons notre approche de collecte à deux autres algorithmes de bandits : *CUCB* et *CUCBV* respectivement proposés dans (Qin *et al.*, 2014) et (Gisselbrecht *et al.*, 2015). Ces algorithmes ne prennent pas en compte les contextes des actions. Dans toutes nos expérimentations, nous fixons les paramètres suivants : 1) Le terme d'exploration à $\alpha = 1.96$, ce qui correspond à un intervalle de confiance à 95% pour l'estimation de l'espérance de la récompense d'une action lorsque son contexte est observé ; 2) Le paramètre de réduction de l'intervalle de confiance des contextes inconnus à $\gamma = 0.25$; 3) Le nombre d'utilisateurs pouvant être ajoutés à $\mathcal{K}$ à chaque itération à $newMax = 200$.

5.2. *Expérimentations hors ligne*

Deux jeux de données sont utilisés pour nos expérimentations hors ligne :

1) *USElections* : Jeu de données de 2148651 messages collectés pendant les dix jours précédents les élections présidentielles américaines de 2012. Ces messages correspondent à la totalité de ceux postés pendant cette période par les 5000 premiers utilisateurs à avoir utilisé les mots "Obama", "Romney" ou "#USElections".

2) *Libya* : Jeu de données de 1211475 messages provenant de 17341 utilisateurs. Ces messages correspondent à l'ensemble de ceux postés sur une période de trois mois contenant le mot clé "Libya".

Modèle de contexte : Dans nos expérimentations, nous considérons que le contexte $z_{a,t}$ d'un utilisateur a pour une itération t correspond à une représentation de la concaténation des messages collectés pour cet utilisateur pendant l'étape $t-1$ selon l'une des deux APIs. Plutôt que d'utiliser directement une représentation fréquentielle en sac de mots, ce qui impliquerait de très longs vecteurs de contextes et un coût important lié à l'inversion des matrices requise lors de la mise à jour des paramètres, nous appliquons une réduction de dimensions à un espace de 30 concepts par l'emploi de l'algorithme LDA (Blei *et al.*, 2003). Outre une réduction de la complexité, l'utilisation de cet algorithme permet une meilleure généralisation des observations en regroupant sous un même concept des mots sémantiquement proches.

Modèle de récompense : Dans nos expérimentations, nous nous intéressons à des modèles de collecte visant à récompenser les messages ayant un fort impact sur une thématique donnée. Quatre thématiques sont considérées (correspondant à 4 différents modèles de récompense) : *politique*, *religion*, *sport* et *science*. L'appartenance à l'une de ces 4 thématiques se fait par l'application d'un classifieur linéaire entraîné sur le corpus *20 Newsgroups*. Ainsi, si un message correspond à la classe de la thématique recherchée, le score de récompense qui y est associé est alors égal au nombre de fois

où ce message est re-posté (*retweeté* sur Twitter) par d'autres utilisateurs pendant l'itération courante.

Afin d'obtenir des résultats généralisables, nous avons expérimenté plusieurs configurations de collecte en considérant plusieurs valeurs pour p, la probabilité d'observation des contextes, k, le nombre d'utilisateurs pouvant être suivis simultanément, et $\mathcal{L}$, la durée d'une itération. Plus précisément nous avons testé toutes les combinaisons possibles avec $p \in \{0.1, 0.5, 1.0\}$, $k \in \{50, 100, 150\}$ et $\mathcal{L} \in \{2min, 3min, 6min, 10min\}$. Néanmoins, pour des raisons de places, nous reportons seulement les résultats pour $k = 100$, $\mathcal{L} = 2min$. De la même façon, pour le jeu de données *USElections*, seule les expérimentations considérant le modèle de récompense centré sur la classe *politique* sont reportées, tandis que pour le jeu de données *Libya*, seul celui centré sur la classe *religion* l'est. Des tendances similaires ont été observées pour les autres configurations et modèles de récompense.

Résultats : La figure 2 représente le gain cumulé en fonction du temps, en haut à gauche pour *USElections* et en haut à droite pour *Libya*. Tout d'abord, nous remarquons que chaque politique fonctionne mieux que la politique *Random*, ce qui est un premier élément pour affirmer la pertinence des algorithmes de bandit pour la tâche en question. Deuxièmement, nous remarquons que *CUCBV* fonctionne mieux que *CUCB*, ce qui confirme les résultats obtenus dans (Gisselbrecht *et al.*, 2015) sur une tâche similaire de capture de données ciblée. Plus intéressant, lorsque chaque contexte est observable ($p = 1$), notre algorithme contextuel fonctionne mieux que les approches *CUCB* et *CUCBV*. Ce résultat montre que nous sommes en mesure de mieux anticiper les utilisateurs qui vont être les plus pertinents à l'étape suivante, compte tenu de ce qu'ils ont dit juste avant. Cela confirme aussi le comportement non-stationnaire des utilisateurs. Par exemple, les utilisateurs peuvent parler de science au cours de la journée tout en étant plus axé sur le sport quand ils reviennent à la maison. Considérer les contextes permet également de converger plus rapidement vers les utilisateurs intéressants puisque tous les comptes partagent le même paramètre β. En outre, les résultats montrent que, même pour de faibles probabilités d'observation des contextes p, notre politique contextuelle se comporte beaucoup mieux que les approches non-contextuelles, ce qui valide empiriquement notre approche : il est possible de tirer parti de l'information contextuelle, même si une grande partie de cette information est cachée. En particulier, pour le jeu de données *Libya* où aucune différence significative entre les deux algorithmes *CUCB* et *CUCBV* n'est remarquable, notre algorithme semble plus approprié. En donnant la possibilité de sélectionner des utilisateurs même si leur contexte est inconnu, nous permettons à l'algorithme de compléter sa sélection en choisissant les utilisateurs dont le contexte moyen correspond à un profil appris. Si aucun utilisateur dans $\mathcal{O}_t$ ne semble actuellement pertinent, l'algorithme peut alors compter sur les utilisateurs avec une bonne qualité intrinsèque moyenne. Numériquement pour $k = 100$, le nombre moyen d'utilisateurs sélectionnés pour lesquels le contexte a été observé à chaque pas de temps est de 43 pour $p = 0.1$ et 58 pour $p = 0.5$, ce qui confirme que l'utilisation de contextes intéressants lorsque ceux-ci sont disponibles, tout en conservant une probabilité de sélection non négligeable pour les utilisateurs de qualité dont le contexte n'est pas connu.

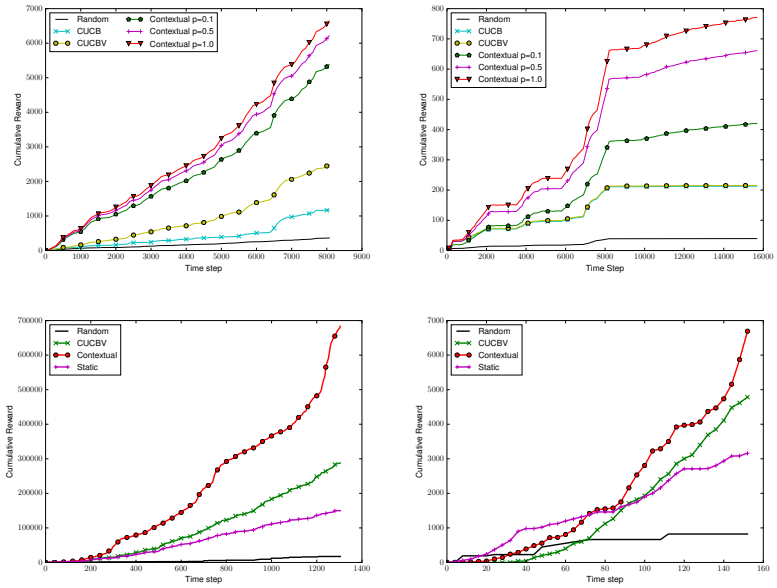

Figure 2. *Récompense cumulée en fonction du temps. Haut gauche : USElections/politique. Haut droite : Libya/religion. Bas gauche : Expérience en ligne. Bas droite : Expérience en ligne avec un zoom sur les premiers pas de temps.*

5.3. *Expérience en ligne*

Nous considérons notre tâche de collecte dans une expérimentation en ligne sur Twitter prenant en compte l'ensemble du réseau social ainsi que les contraintes de l'API. Pour ces expériences, nous avons utilisé tout le potentiel de l'API Twitter, à savoir 1% de tous les tweets publics pour l'API *Sample*, qui s'exécute en arrière-plan, et 5000 utilisateurs suivis simultanément ($k = 5000$) pour l'API *Follow*. À chaque itération, les utilisateurs sélectionnés sont suivis pendant $\mathcal{L} = 15minutes$. Compte tenu des restrictions de Twitter, chaque expérience nécessite un compte développeur, ce qui limite le nombre de politiques que nous pouvons tester en parallèle. Nous avons choisi de tester les quatre stratégies suivantes : notre approche contextuelle, *CUCBV*, *Random* et une autre appelée *Static*, qui suit les mêmes 5000 comptes, sélectionnés a priori, à chaque étape du processus. Les 5000 comptes suivis par l'approche *Static* ont été choisis en sélectionnant les 5000 utilisateurs ayant cumulé le plus fort volume de récompenses selon les messages collectés par l'API *Sample* sur une période de 24 heures. Pour information, certains comptes célèbres tels que *@dailytelegraph*, *@Independent* ou *@CNBC* en faisaient partie.

Résultats : La partie inférieure gauche de la figure 2 représente l'évolution du gain cumulé sur deux semaines pour les quatre algorithmes. On remarque le très bon com-

portement de notre algorithme dans un scénario réel, car la somme des récompenses qu'il accumule croît beaucoup plus rapidement que celle des autres politiques, surtout après les 500 premières itérations. Après ces premières itérations, notre algorithme semble avoir acquis une bonne connaissance sur les distributions de récompenses en fonction des contextes observés sur les différents utilisateurs du réseau. Afin d'analyser le comportement des politiques expérimentées pendant les premières itérations de la capture, nous représentons aussi en bas à droite de la figure 2 un zoom des mêmes courbes sur les 150 premiers pas de temps. Au début, la politique *Static* est plus performante que toutes les autres, ce qui peut s'expliquer par deux raisons : 1) les algorithmes de bandits utilisés ont besoin de sélectionner tous les utilisateurs au moins une fois pour initialiser les scores et 2) les utilisateurs faisant partie de l'ensemble *Static* sont supposés être des sources relativement pertinentes étant donné la façon dont ils ont été choisis. Vers l'itération 80, aussi bien l'algorithme *CUCBV* que l'algorithme contextuel deviennent meilleurs que *Static*, ce qui correspond au moment où ils commencent à pouvoir exploiter des "bonnes" actions identifiées. Ensuite, une période d'environ 60 itérations est observée (approximativement entre les itérations 80 et 140), au cours de laquelle l'algorithme *CUCBV* et notre approche contextuelle ont des performances comparables. Cette période s'explique par le fait que notre approche nécessite un certain nombre d'itérations pour apprendre des corrélations efficaces entre contextes et récompenses afin d'en tirer avantage. Enfin, après cette période d'initialisation, on peut remarquer un changement significatif de pente pour la courbe correspondant à notre algorithme, ce qui souligne sa faculté à être bien plus réactif dans un environnement dynamique.

6. Conclusion

Nous avons formalisé une tâche de collecte de données sur les réseaux sociaux comme une instance spécifique du problème de bandit contextuel, dans laquelle, dû aux restrictions imposées par les médias sociaux, seule une partie des contextes est observable à chaque itération. Pour résoudre cette tâche, nous avons proposé une adaptation de l'algorithme de bandit contextuel *LinUCB* pour le cas où certains contextes sont cachés. Bien qu'aucune borne supérieure sous-linéaire sur le regret ne puisse être atteinte, en raison de la grande incertitude induite par les contextes cachés, notre algorithme offre de bons résultats pour la collecte de données, même lorsque la proportion de contextes cachés est élevée. Cela permet en outre d'envisager son application à bien d'autres tâches, pour lesquelles des contraintes restreignent l'observation des contextes associés aux actions disponibles.

7. Remerciements

Ce travail a été effectué dans le cadre des recherches menées au sein de l'IRT SystemX et a ainsi bénéficié d'une aide de l'Etat au titre du programme d'Investissements

d'Avenir, ainsi que dans le cadre du projet LUXIDX du dispositif RAPID du fond de compétitivité des entreprises.

8. Bibliographie

Agrawal S., Goyal N., « Analysis of Thompson Sampling for the multi-armed bandit problem », *COLT*, 2012a.

Agrawal S., Goyal N., « Thompson Sampling for Contextual Bandits with Linear Payoffs », *CoRR*, 2012b.

Audibert J.-Y., Bubeck S., « Minimax policies for adversarial and stochastic bandits », *COLT*, 2009.

Audibert J.-Y., Munos R., Szepesvari C., « Tuning bandit algorithms in stochastic environments », *ALT*, 2007.

Auer P., Cesa-Bianchi N., Fischer P., « Finite-time Analysis of the Multiarmed Bandit Problem », *Mach. Learn.*, 2002.

Blei D. M., Ng A. Y., Jordan M. I., « Latent Dirichlet Allocation », *J. Mach. Learn. Res.*, 2003.

Buccapatnam S., Eryilmaz A., Shroff N. B., « Stochastic Bandits with Side Observations on Networks », *SIGMETRICS*, 2014.

Chapelle O., Li L., « An Empirical Evaluation of Thompson Sampling », *NIPS*, 2011.

Chen W., Wang Y., Yuan Y., « Combinatorial Multi-Armed Bandit : General Framework and Applications », *ICML*, JMLR Workshop and Conference Proceedings, 2013.

Chu W., Li L., Reyzin L., Schapire R. E., « Contextual Bandits with Linear Payoff Functions », *AISTATS*, 2011.

Colbaugh R., Glass K., « Emerging Topic Detection for Business Intelligence Via Predictive Analysis of 'Meme' Dynamics », *AAAI Spring Symposium : AI for Business Agility*, 2011.

Gisselbrecht T., Denoyer L., Gallinari P., Lamprier S., « WhichStreams : A Dynamic Approach for Focused Data Capture from Large Social Media », *ICWSM*, 2015.

Hannon J., Bennett M., Smyth B., « Recommending Twitter Users to Follow Using Content and Collaborative Filtering Approaches », *RecSys*, 2010.

Kohli P., Salek M., Stoddard G., « A Fast Bandit Algorithm for Recommendation to Users With Heterogenous Tastes. », *AAAI*, 2013.

Lage R., Denoyer L., Gallinari P., Dolog P., « Choosing Which Message to Publish on Social Networks : A Contextual Bandit Approach », *ASONAM*, 2013.

Lai T., Robbins H., « Asymptotically efficient adaptive allocation rules », *Advances in Applied Mathematics*, vol. 6, n° 1, p. 4 - 22, 1985.

Li L., Chu W., Langford J., Schapire R. E., « A Contextual-bandit Approach to Personalized News Article Recommendation », *WWW '10*, WWW '10, p. 661-670, 2010.

Li R., Wang S., Chang K. C.-C., « Towards Social Data Platform : Automatic Topic-focused Monitor for Twitter Stream », *Proc. VLDB Endow.*, 2013.

Qin L., Chen S., Zhu X., « Contextual Combinatorial Bandit and its Application on Diversified Online Recommendation », *SIAM*, 2014.