Stratégies de bandit pour les systèmes de recommandation

Jonathan Louëdec

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *04 Novembre 2016 (04/11/2016)* **par :**

JONATHAN LOUËDEC

Stratégies de Bandit pour les Systèmes de Recommandation

JURY

MAX CHEVALIER	Pr, Univ. de Toulouse 3, IRIT, Co-Directeur
AURÉLIEN GARIVIER	Pr, Univ. de Toulouse 3, IMT, Co-Directeur
JOSIANE MOTHE	Pr, Univ. de Toulouse 2, IRIT, Co-Directrice
OLIVIER CAPPÉ	DR CNRS, Télécom Paris, Rapporteur
VIOLAINE PRINCE	PR, Univ. de Montpellier 2, LIRMM, Rapporteur
FABIO CRESTANI	Pr, Univ. Lugano, Suisse, Examinateur
BRUNO GOUTORBE	C-Discount R&D, Examinateur

École doctorale et spécialité :
MITT : Image, Information, Hypermedia

Unité de Recherche :
Institut de Recherche en Informatique de Toulouse UMR 5505
Institut de Mathématiques de Toulouse UMR 5219

Directeur(s) de Thèse :
Max CHEVALIER, *Aurélien GARIVIER* et *Josiane MOTHE*

Rapporteurs :
Olivier CAPPÉ et *Violaine PRINCE*

Table des matières

Remerciements

Je tiens tout d'abord à remercier mes trois directeurs de thèse, Josiane Mothe, Max Chevalier et Aurélien Garivier. Merci à Josiane sans laquelle je ne serai certainement pas arrivé jusque là, qui m'a fait confiance durant ces six dernières années au travers d'un stage lors de ma troisième année de licence, de divers vacations et enfin au travers de cette thèse. Merci à Max pour tous ses conseils avisés durant la thèse qui m'ont permis de mieux m'organiser et progresser. Merci à Aurélien pour sa disponibilité, ses nombreuses explications et cours particuliers qui m'ont permis de très vite me familiariser avec les stratégies de bandit, même si parfois il a fallu m'expliquer plusieurs fois, preuve de sa patience. Enfin, merci à tous les trois de m'avoir donné ma chance, de toujours être bienveillants à mon égard, et de m'avoir toujours soutenu autant professionnellement que personnellement. Je m'estime vraiment chanceux de vous avoir eu comme directeurs de thèse.

J'adresse également mes remerciements à l'ensemble des membres de mon jury : Violaine Prince et Olivier Cappé, qui ont accepté d'être les rapporteurs de mon manuscrit.

Je souhaite également remercier l'ensemble des membres du bureau 424 de l'IRIT et du bureau 201 avec qui j'ai eu l'occasion de passer du temps. Merci Léa, Adrian et Anthony pour votre bonne humeur, votre amitié et pour tous ces moments de réflexion mais aussi de rigolade que nous avons pu partager tous les quatre. Merci également à Anne-Claire, Antoine, Brendan, Charlotte, Claire, Clément, Damien, Hélène, Magali, Nicolas et Sylvain pour tous ces agréables moments partagés. Enfin merci à Ngoc et Mahdi, qui représente l'avenir du bureau 424 ! J'espère que vous en garderez d'aussi bons souvenirs que moi.

Je remercie également Sébastien Gerchinovitz et Laurent Rossi pour toutes les discussions que l'on a pu avoir sur les approches d'apprentissage par renforcement et pour leur collaboration lors de travaux communs.

Je tiens à remercier plus généralement l'ensemble des membres des équipes SIG et IRIS de l'IRIT ainsi que ceux de l'équipe ESP de l'IMT, pour leur aide et leur gentillesse au quotidien.

Je tiens à remercier Agnès Requis et Martine Labruyère de l'École Doctorale pour leur gentillesse, leur disponibilité et leurs réponses claires à mes nombreuses interrogations.

Je remercie également le laboratoire d'excellence CIMI (Centre International de Mathématiques et d'Informatique de Toulouse) de m'avoir attribué une bourse doctorale afin de financer cette thèse.

Je remercie également toute ma famille et mes amis qui m'ont toujours soutenu et encouragé.

Enfin un merci tout particulier pour ma compagne, Virginie, qui me soutient et me conforte dans mes choix, qui a supporté les périodes les plus intenses de ma thèse qui me rendaient beaucoup moins disponible et pour tout ce qu'elle m'apporte au quotidien.

Résumé

Les systèmes de recommandation actuels ont besoin de recommander des objets pertinents aux utilisateurs (exploitation), mais pour cela ils doivent pouvoir également obtenir continuellement de nouvelles informations sur les objets et les utilisateurs encore peu connus (exploration). Il s'agit du dilemme exploration/exploitation. Un tel environnement s'inscrit dans le cadre de ce que l'on appelle « apprentissage par renforcement ».

Dans la littérature statistique, les stratégies de bandit sont connues pour offrir des solutions à ce dilemme. Les contributions de cette thèse multidisciplinaire adaptent ces stratégies pour appréhender certaines problématiques des systèmes de recommandation, telles que la recommandation de plusieurs objets simultanément, la prise en compte du vieillissement de la popularité d'un objet ou encore la recommandation en temps réel.

Introduction générale

Bien avant l'arrivée d'internet à la fin des années 1990, les individus se recommandaient des objets ou informations en fonction de leurs affinités par l'intermédiaire du «bouche à oreille». Ces dernières années, et principalement sur internet, le volume d'objets et d'informations disponibles augmente continuellement, à tel point que de nombreuses technologies sont imaginées et développées pour les gérer. Si le phénomène de «bouche à oreille» est toujours d'actualité, des techniques plus poussées sont nécessaires pour gérer cette masse d'informations afin de proposer des informations pertinentes aux utilisateurs. Les systèmes de recommandation contribuent à résoudre cette problématique.

Afin d'identifier les informations pertinentes pour un utilisateur, deux techniques sont généralement mises en œuvre : le filtrage basé sur le contenu et le filtrage collaboratif.

Le filtrage basé sur le contenu consiste à recommander à un utilisateur les documents similaires à ceux appréciés par cet utilisateur dans le passé. Si cette technique est efficace lorsque de nombreuses informations sont disponibles sur les documents, elle n'est pas utilisable lorsque le contenu de ces derniers n'est pas ou peu décrit.

Le filtrage collaboratif recommande les objets appréciés par les utilisateurs qui ont auparavant fait des choix similaires à ceux de l'utilisateur courant [Ricci 2011]. Cette technique, pour être efficace, requière que les utilisateurs aient suffisamment interagi avec le système.

Si ces techniques possèdent des limites communes connues, certaines d'entre elles sont spécifiques. Une idée naturelle est donc d'utiliser des approches hybrides basées sur le contenu des objets et sur la similarité des appréciations que les utilisateurs portent sur ces objets [Burke 2002].

Globalement, ces approches nécessitent que les objets soient suffisamment recommandés pour obtenir une estimation précise du gain associé. Ce gain peut être un taux de clics ou de ventes par exemple. Dans ce manuscrit nous considérons qu'un objet est *populaire* si son taux de clics est élevé. Cependant recommander tous les objets un grand nombre de fois est une stratégie dangereuse pour deux raisons : le gain moyen n'est pas optimal (car des objets non populaires sont trop souvent recommandés) et l'utilisateur peut choisir de ne plus utiliser le système (si les recommandations sont rarement intéressantes, voir [O'Brien 2008]). Il faut donc être capable de faire un apprentissage sur la popularité des objets disponibles, tout en maximisant le gain global : un tel problème est connu sous le nom de "dilemme exploration/exploitation". Les stra-

tégies dites «de bandit» offrent des solutions à ce dilemme [Bubeck 2012].
Le terme *bandit* tient son origine des machines à sous, dites «bandits manchots», dans un casino. A chaque tentative, le joueur doit décider quelle machine jouer parmi toutes celles disponibles. Afin de maximiser ses gains, il doit établir une stratégie lui permettant d'identifier rapidement la machine qui rapporte le plus, tout en testant suffisamment les autres machines pour ne pas risquer d'oublier des machines plus performantes. L'ensemble de la terminologie utilisée dans l'étude statistique des problèmes de bandit est inspirée de ce cadre, comme par exemple les termes "bras" (qui représente un appareil de machine à sous), "récompenses" (gain obtenu) ou encore "jouer". Ces approches sont décrites en détail dans le chapitre 2 de ce manuscrit.

Pour gérer le dilemme exploration/exploitation, les stratégies de bandit s'appuient sur une part d'aléatoire, des distributions de probabilité ou encore des intervalles de confiance. Dans la version stochastique des modèles de bandit, un agent choisit à chaque instant $t = 1, 2, \ldots$ un bras $A_t \in 1, \ldots, K$ et reçoit une récompense Z_t aléatoire dépendant de ce choix. Le cadre classique est stationnaire : les K bras sont disponibles du début à la fin et ont un gain moyen n'évoluant pas au cours du temps. Si ce modèle capture déjà l'essence du dilemme exploration/exploitation (il faut à la fois essayer les bras mal connus et tirer profit de ceux qui semblent le plus performants), il s'avère insuffisant dans de nombreux cas. En particulier, les données se renouvellent et vieillissent au fil du temps : c'est notamment le cas pour la recommandation de films, de produits ou de nouvelles. Il faudrait donc être capable de prendre en compte à la fois le **renouvellement** (i.e, l'apparition de nouveaux objets) et l'**obsolescence progressive** (i.e, la diminution de la popularité, voire la disparition de certains objets).

Dans la littérature statistique, des stratégies de bandit prenant en compte la non-stationnarité existent. Pour cela, certaines stratégies vont privilégier la fraîcheur des dernières interactions du système [Garivier 2008], tandis que d'autres considèrent que les objets ont une durée de vie prédéfinie [Chakrabarti 2009]. Cependant aucune approche ne considère d'informations a priori sur la manière dont la popularité des objets évolue.

Le premier axe de nos contributions consiste à considérer un modèle dans lequel la popularité des objets décroît de manière exponentielle en fonction du temps. Ce choix a été fait après l'observation préliminaire de l'évolution de la popularité des objets d'un jeu de données de recommandation d'information. Nous proposons une nouvelle stratégie de bandit pour tenir compte de ce modèle, et montrons à l'aide de simulations que la connaissance a priori de la manière dont la popularité des objets évolue au cours du temps permet d'obtenir de bien meilleures performances.

Les systèmes de recommandation doivent souvent être capables de gérer plusieurs

recommandations simultanément, il faudrait donc pouvoir générer une liste de recommandations à chaque instant : il s'agit d'un cas dit de «recommandation à tirages multiples». Pour cela, les approches de l'état-de-l'art considèrent autant d'instances d'une stratégie de bandit à tirages simples qu'il y a de recommandations à effectuer [Radlinski 2008, Kohli 2013].

Au contraire, dans **le deuxième axe de nos contributions**, nous proposons de nouvelles approches pour la recommandation à tirages multiples.

La première contribution de cet axe est la proposition d'une approche considérant une seule instance d'une stratégie de bandit. L'objectif est d'illustrer le fait que la gestion de l'ensemble des objets à l'aide d'une même instance permet d'apprendre beaucoup plus vite.

La littérature statistique récente propose de nouvelles stratégies de bandit, spécialement conçues pour recommander plusieurs objets à chaque instant [Uchiya 2010, Bubeck 2012, Komiyama 2015]. La seconde contribution de cet axe est la proposition d'une implémentation efficace de l'une d'entre elles, permettant ainsi son utilisation en recommandation. Nous montrons que cette stratégie, spécialement conçue pour le cadre de la recommandation à tirages multiples, permet d'obtenir de meilleures performances que les approches les plus efficaces de l'état-de-l'art [Radlinski 2008, Kohli 2013].

Lorsqu'une liste d'objets est recommandée, il est important que cette liste soit générée en fonction d'un objectif précis. L'objectif mis en avant dans ce manuscrit est la minimisation de *l'abandon*, c'est-à-dire la minimisation du nombre de fois où l'utilisateur ne clique sur aucune des recommandations. Il est important de noter que l'objectif n'est donc pas de maximiser le taux de clics : il n'y a pas de préférence entre un utilisateur cliquant sur une seule recommandation et un utilisateur cliquant sur l'ensemble des recommandations.

Dans le cadre de la minimisation de l'abandon, il est indispensable d'apporter de la diversité dans les listes de recommandations afin de maximiser le nombre de fois où un utilisateur clique sur au moins l'une des recommandations. Les approches de l'état de l'art apportant de la diversité en utilisant des stratégies de bandit sont peu performantes en terme de vitesse d'apprentissage [Radlinski 2008].

La troisième contribution de cet axe est la proposition d'une approche utilisant des stratégies de bandit à tirages simples basée sur un calcul de probabilité conditionnelle, afin d'apporter de la diversité. Nous montrons sur des données réelles que cette prise en compte de la diversité permet d'obtenir des proportions d'abandon plus faibles que celles obtenues par les approches de l'état-de-l'art, tout en proposant un apprentissage plus rapide. Ce manuscrit présente les travaux réalisés au cours des trois années de

thèse. Il est structuré en deux parties, chacune décomposée en plusieurs chapitres.

La première partie introduit les systèmes de recommandation ainsi que les stratégies de bandit, nous permettant d'identifier pourquoi ces stratégies représentent une alternative intéressante aux approches classiquement mises en œuvre.

- Le chapitre 1 présente les systèmes de recommandation et dresse une liste non-exhaustive de plusieurs exemples de systèmes de recommandation. Les techniques mises en œuvre dans la littérature sont également décrites ainsi que les différentes problématiques connues de la recommandation.
- Le chapitre 2 présente les problèmes de bandit. Il dresse un état-de-l'art des stratégies de la littérature statistique pour aborder ces problèmes.
- Le chapitre 3 met en évidence en quoi les stratégies de bandit représentent une alternative intéressante aux techniques classiques de recommandation. Il présente les différentes contributions. Les publications et communications réalisées au cours de la thèse sont détaillées.

La deuxième partie regroupe l'ensemble de nos contributions ainsi que plusieurs états-de-l'art représentant les problématiques spécifiques que nous abordons dans ce manuscrit, à savoir l'obsolescence progressive, la recommandation à tirages multiples et la diversification des listes de résultats.

- Le chapitre 4 aborde la problématique spécifique de la non stationnarité du gain moyen des bras. Un état-de-l'art des stratégies de bandit non stationnaires est dressé. Ce chapitre propose enfin un modèle et un formalisme où des hypothèses plus fortes sont faites sur la manière dont le gain moyen des bras évolue au cours du temps.
- Le chapitre 5 propose deux stratégies conçues pour aborder le modèle présenté dans le chapitre précédent. Il fournit une étude de la borne supérieure de regret de l'une de ces stratégies. A l'aide de simulations, les performances des stratégies proposées sont comparées à celles de l'état de l'art.
- Le chapitre 6 dresse un état de l'art complet de la problématique de diversification. Il met en évidence l'importance de la prise en compte de cette problématique en recommandation.
- Le chapitre 7 montre que les approches à tirages multiples de l'état de l'art utilisant des stratégies de bandit à tirages simples ont des vitesses d'apprentissage faibles. Nous proposons une nouvelle approche, permettant d'améliorer la vitesse d'apprentissage, tout en obtenant des performances sur le long terme équivalentes.
- Le chapitre 8 traite de la recommandation à tirages multiples en utilisant des stratégies de bandit spécialement conçues pour recommander plusieurs objets

à chaque instant. Il présente quelques stratégies récentes de la littérature. Une implémentation efficace de l'une d'entre elles permettant de la rendre utilisable en recommandation est proposée, ainsi qu'une évaluation sur deux jeux de données de l'état de l'art comparant les performances de cette stratégie avec celles des approches de l'état de l'art présentées dans le chapitre précédent.

— Le chapitre 9 propose d'améliorer les résultats obtenus jusqu'ici en prenant en compte la problématique de la diversification des listes de résultats. Pour cela une nouvelle approche utilisant des probabilités conditionnelles est proposée. Nous évaluons cette approche et montre que la prise en compte de la diversité permet d'améliorer les performances sur le long terme, mais également au début des expérimentations.

Première partie

Stratégies de Bandit et Systèmes de Recommandation : Etat de l'art et Objectifs

Résumé de la première partie

Dans cette première partie, nous présentons les deux principaux domaines dans lesquels s'inscrivent nos travaux : les systèmes de recommandation et les problèmes de bandit. Elle est divisée en trois chapitres.

Dans le chapitre 1, nous décrivons en détail ce qu'est un système de recommandation, ainsi que les principales problématiques liées. Nous présentons les techniques généralement utilisées dans ce contexte.

Dans le chapitre 2, nous présentons le principe de l'apprentissage par renforcement, dont découle celui des problèmes de bandit. Nous formalisons en détail ce problème afin de présenter plusieurs stratégies de la littérature. Les stratégies présentées sont conçues pour recommander un seul objet à chaque instant. Les stratégies capables de recommander plusieurs objets à chaque instant sont présentées dans la seconde partie.

Dans le chapitre 3, nous présentons les différentes problématiques qui sont abordées lorsque les stratégies de bandit sont appliquées à la recommandation. Nous détaillons ensuite l'ensemble de nos publications, soumises et acceptées ainsi que les communications effectuées dans le cadre de cette thèse.

Chapitre 1

Systèmes de recommandation : présentation et techniques

Sommaire

Résumé.

La recommandation représente un cadre de recherche complexe de la littérature informatique actuel, et plus particulièrement le cadre dans lequel nos travaux sont effectués. Le volume de données disponibles sur internet ne cesse de croître, dans ce cadre il est important de développer des technologies capables d'extraire les données les plus intéressantes en fonction d'un utilisateur.

Nos travaux s'inscrivent dans le contexte de la recommandation. Dans ce chapitre nous décrivons ce contexte, que nous illustrons par quelques exemples. Ensuite nous présentons les techniques mises en œuvre dans la littérature informatique, à savoir le filtrage collaboratif, le filtrage basé sur le contenu et les approches hybrides. Pour chacune de ces techniques, nous mettons en avant leurs avantages et inconvénients.

1.1 Introduction aux systèmes de recommandation (SR)

1.1.1 Description

Bien avant l'arrivée d'internet à la fin des années 1990, les individus se recommandaient des objets ou informations en fonction de leurs affinités par l'intermédiaire du «bouche à oreille». Ces dernières années, et principalement sur internet, le volume d'objets et d'informations disponibles augmente continuellement, à tel point que de nombreuses technologies sont imaginées et développées pour les gérer. Si le phénomène de «bouche à oreille» est toujours d'actualité, des techniques plus poussées sont nécessaires pour gérer cette masse d'informations afin de proposer des informations pertinentes aux utilisateurs. Les systèmes de recommandation (SR) contribuent à résoudre cette problématique [Ricci 2011, Kembellec 2014].

1.1.2 Diversité des objets et exemples de systèmes

Un SR suggère des objets à un utilisateur dans le but de le satisfaire. Le terme «objet» que nous utilisons dans l'ensemble de cette thèse est général : il désigne tout ce qui peut être recommandé. Un SR se focalise généralement sur un type d'objet, la figure 1.1 présentent plusieurs SR en fonction du type d'objet recommandé. Nous détaillons ici quelques uns de ces exemples :

- **Recommandation de produits**
  - *Amazon* est une entreprise de commerce en ligne américaine. Elle compte plus de 183 millions produits référencés. En France, *Amazon* compte 16,8 millions d'utilisateurs uniques par mois.
  - *Cdiscount* est l'entreprise leader de commerce en ligne française. Actuellement, *Cdiscount* est le deuxième site de commerce en ligne le plus visité en France, talonnant le géant américain *Amazon*.

- **Recommandation d'actualités**
  - *Google actualités* recense des milliers de sources d'actualités. Il est considéré comme un agrégateur d'informations externes, et doit donc être capable de faire le lien entre plusieurs actualités abordant le même sujet mais dont les sources sont distinctes.
  - *Yahoo ! News* est un service gratuit présentant aux utilisateurs des actualités issues d'autres sources d'actualités, telles que *ABC News* ou *Fox News*, il s'agit également d'un agrégateur d'informations externes. Début 2016, *Yahoo !* met à disposition de la communauté scientifique un jeu de données de 13,5 Terabytes, ce qui représente 110 billions d'interactions entre les utilisateurs et les actualités. Il s'agit du jeu de données d'apprentissage le plus volumineux rendu public à ce jour.
- **Recommandation de titres musicaux**
  - *Spotify* est un logiciel suédois de musique en ligne (streaming). En juin 2015, il compte plus de 30 millions de titres pour 75 millions d'utilisateurs actifs.
  - *Deezer* est un site web français d'écoute de musique à la demande, disposant de plus de 40 millions de titres pour 16 millions d'utilisateurs actifs.
- **Recommandation de films ou séries**
  - *Netflix* est un service américain de vidéos à la demande comptant plus de 75 millions d'utilisateurs. En 2006, *Netflix* lance un challenge avec un million de dollars de récompense pour l'équipe qui obtiendra une amélioration de 10% des prédictions de notes effectuées pour un film par l'algorithme utilisé par l'entreprise. Pour cela *Netflix* met à disposition 100 millions de notes, représentant 480 000 utilisateurs pour 17 000 films environ. Ce n'est qu'en 2009 qu'une équipe nommée "BellKor's Pragmatic Chaos" arrive à atteindre cet objectif. La solution est décrite dans l'article [Koren 2009].
  - *MovieLens* est un SR de films non-commercial créé en 1997 par l'université du Minnesota. Plusieurs jeux de données de référence en recommandation ont été fournis par *MovieLens*.

1.1.3 Retour utilisateur

Le retour utilisateur fait référence à l'ensemble des informations qu'il est possible d'extraire des interactions entre le SR et les utilisateurs. Le retour utilisateur est indispensable pour pouvoir optimiser les recommandations effectuées en permettant notamment de connaître les objets appréciés par les utilisateurs. Deux types de retours existent : le retour explicite, directement fourni par l'utilisateur, et le retour implicite,

Objets recommandés	Systèmes
Produits (CD, livres, ...)	Amazon, Rue du Commerce, Cdiscount
Actualités	Google Actualités, Yahoo ! News, BBC News, Le Monde
Films-Séries	Netflix, Allociné, IMDb, Vodkaster
Titres musicaux	Spotify, Deezer, Apple Music, Google Play Music
Emplois	LinkedIn, Viadeo, Indeed, Monster
Lieux-Vacances	TripAdvisor, Booking.com, Nomao, La Fourchette
Personnes	Facebook, Twitter, LinkedIn, Viadeo
Publicités	Google Search, Facebook, Amazon, Criteo

Table 1.1 – Exemples de SR classés par type d'objet recommandé

indirectement fourni par l'utilisateur [Montaner 2001, Ricci 2011].

1.1.3.1 Retour utilisateur explicite

Ces retours sont directement fournis par l'utilisateur. Par exemple, lorsqu'un nouvel utilisateur se connecte, beaucoup de SR commencent par demander à l'utilisateur de noter plusieurs objets afin de pouvoir par la suite recommander des objets lui correspondant ; c'est le cas de *Netflix* ou *Vodkaster* notamment. Il est ensuite possible de noter les objets au fur et à mesure des interactions avec le SR. Ces notes peuvent être :

- **numériques** : allant généralement de 1 à 5 ; 5 traduit le fait que l'objet plaît beaucoup à l'utilisateur. Ces notes sont souvent représentées par un nombre d'étoiles sur *Amazon* et *Allociné* notamment.
- **binaires** : la décision de l'utilisateur revient à choisir si l'objet est «bon» ou «mauvais». Ce type de notation peut être décliné de manière à ne considérer qu'un aspect : l'utilisateur a le choix de noter un objet «bon» ou de ne pas noter l'objet : c'est le cas de la fonction «like» du réseau social *Facebook*.
- **ordinales** : l'utilisateur doit choisir parmi une liste de termes celui qui lui parait être le plus adapté pour son sentiment vis à vis de l'objet en question. Un exemple de liste peut être «Très bon, bon, moyen, mauvais, très mauvais».
- **descriptives** : l'utilisateur choisit un ou plusieurs termes qui selon lui décrivent au mieux l'objet en question. Pour un film, ces termes peuvent être «passionnant», «drôle», «sans intérêt», «atypique», etc. De nombreux SR laissent également la possibilité aux utilisateurs de laisser des commentaires sur les objets. Dans ce cas un traitement sur le texte est nécessaire pour extraire les termes les plus intéressants.

1.1.3.2 Retour utilisateur implicite

Il regroupe l'ensemble des informations qui peuvent être extraites de la navigation de l'utilisateur sur le SR. Le premier retour implicite généralement considéré est le clic utilisateur, c'est à dire le fait que l'utilisateur sélectionne un objet. Les SR qui réalisent la vente ou la location d'objets ont un retour encore plus fort : l'achat ou la location de l'objet. Il est également possible de considérer le temps passé sur une page (*Dwell time*), il peut traduire le fait qu'un utilisateur a pris le temps de lire ou non le contenu de cette page.

1.2 Techniques classiquement mises en œuvre

Pour recommander des objets à des utilisateurs, un SR doit être capable d'identifier quels objets seront utiles pour un utilisateur donné. Il faut donc prédire l'utilité des objets en fonction des informations disponibles sur l'utilisateur. La plus simple des approches revient à recommander les objets les plus populaires par le passé. Cette approche s'avère être efficace si aucune information sur l'utilisateur n'est disponible, soit parce qu'il s'agit d'un nouvel utilisateur, soit parce que le SR ne conserve pas d'informations. Lorsque des informations sont disponibles sur les utilisateurs et sur les objets, les prendre en compte permet de personnaliser les recommandations effectuées. Pour cela plusieurs techniques de recommandation ont été proposées. Les deux principales sont le filtrage collaboratif et le filtrage basé sur le contenu des objets. Certains travaux proposent des approches hybrides basées sur les deux techniques précédemment citées.

1.2.1 Filtrage basé sur le contenu des objets

Cette approche se base sur la similarité entre les différents objets : les objets sont recommandés aux utilisateurs en fonction de leurs retours passés sur des objets similaires. L'enjeu ici est de trouver comment calculer la similarité entre les objets. Tout d'abord cela suppose que chaque objet possède suffisamment de métadonnées, dans le cas contraire il est difficile de calculer des similarités. Les métadonnées peuvent être la description d'un produit, le résumé d'un livre, le titre d'un film, le texte d'une actualité, etc. Pour calculer les similarités, les techniques d'analyse textuelle sont généralement utilisées, des approches de traitement du signal peuvent également être mises en œuvre lorsque des sons sont disponibles.

Lorsque le filtrage basé sur le contenu des objets est utilisé, cela revient à ordonner

l'ensemble des objets en fonction de l'utilisateur u_i. Les modèles de recherche pour l'ordonnancement, plus connus en recherche d'information, peuvent être divisés en deux catégories : les modèles basés sur la similarité entre les objets et l'utilisateur courant et les modèles basées sur l'importance des documents [Liu 2011]. Ces derniers sont applicables uniquement sur des pages web et ne sont pas développées dans cette thèse.

1.2.1.1 Modèles de recherche basés sur la similarité utilisateur-objet

Le but de ces modèles conçus pour la recherche d'information (Recherche d'Information (RI)), est d'estimer la similarité entre les termes d'une requête et l'ensemble des termes des documents disponibles. Appliqués aux SR, ces modèles estiment la similarité entre un utilisateur et l'ensemble des termes disponibles dans les métadonnées des objets (Description, titre, ...). L'utilisateur peut être représenté par les termes des métadonnées des objets qu'il a cliqués par le passé.

Le modèle booléen [Lan 1974] est le premier modèle proposé, dans les années 1970. Il retourne l'ensemble des objets dont les métadonnées contiennent exactement tous les termes représentant l'utilisateur. Ce modèle est difficilement applicable à la recommandation car un utilisateur est représenté par beaucoup plus de termes qu'une requête. Les modèles suivants sont préférés car ils permettent d'obtenir de meilleures performances.

Le modèle vectoriel [Salton 1975] représente l'utilisateur et les objets à l'aide de vecteurs dans l'espace des termes et utilise des mesures de similarité (cosinus par exemple) afin d'évaluer la proximité entre les utilisateurs et/ou les objets.

Le modèle Latent Semantic Indexing [Deerwester 1990] utilise la décomposition en valeurs singulières pour réduire l'espace des termes à un espace de concepts dans lequel la similarité entre utilisateur et objet est calculée.

Les modèles probabilistes, comme BM25 [Robertson 1994] ou les modèles de langues [Ponte 1998] se basent sur la théorie des probabilités pour mesurer la similarité entre utilisateurs et objets.

Afin d'illustrer comment ces modèles fonctionnent, nous prenons l'exemple du modèle vectoriel.

Le modèle Vectoriel. Il permet de calculer un score de similarité entre les utilisateurs et les objets disponibles. Un utilisateur est représenté par un vecteur des poids des termes contenus dans les métadonnées des objets qu'il a appréciés par le passé. Un objet est représenté par un vecteur des poids des termes contenus dans ses mé-

tadonnées. Le score de similarité est exprimé comme la similarité entre le vecteur de l'utilisateur et le vecteur de l'objet.

Définition 1 (Modèle vectoriel)**.**

Soient $u = [w_{1,u}, \ldots, w_{m,u}]$ et $o_i = [w_{1,i}, \ldots, w_{m,i}]$ les représentations de l'utilisateur et des objets dans l'espace des termes, avec m le nombre de termes, $w_{t,u}$ le poids du terme t pour l'utilisateur u et $w_{t,i}$ le poids du terme t pour l'objet i. Alors le score de similarité de l'objet o_i par rapport à l'utilisateur u est donné par la mesure de similarité exprimée comme le cosinus entre les deux vecteurs telle que :

$$sim(o_i, u) = \frac{\sum_{t=1}^{m}(w_{t,i}w_{t,u})}{\sqrt{\sum_{t=1}^{m}(w_{t,u})^2}\sqrt{\sum_{t=1}^{m}(w_{t,i})^2}} \tag{1.1}$$

Le poids d'un terme pour un utilisateur ou un objet est généralement calculé à l'aide de la pondération $TF.IDF$ [Robertson 1976].

La quantité TF représente le nombre d'occurrences d'un terme dans les métadonnées d'un objet, ou dans les métadonnées de l'ensemble des objets appréciés par l'utilisateur. $TF(t, o_i)$ est le nombre d'occurrence du terme t dans les métadonnées de $l'objeto_i$

La quantité IDF exprime l'importance d'un terme au sein du corpus considéré, telle que :

$$IDF(t) = \log(\frac{N}{n_t}) \tag{1.2}$$

où N est le nombre total d'objets et n_t est le nombre d'objets dont les métadonnées contiennent le terme t. La pondération $TF.IDF$ d'un terme t pour un objet o_i est alors exprimée comme le produit des deux quantités précédentes :

$$TF.IDF(t, o_i) = TF(t, o_i).IDF(t) \tag{1.3}$$

Cette pondération prend en compte le fait que tous les objets n'ont pas le même pouvoir discriminant. En effet l'importance d'un terme est définie à partir de sa fréquence dans les métadonnées d'un objet, mais les termes trop fréquents, donc peu discriminants, sont pénalisés par la quantité IDF.

1.2.1.2 Avantages et inconvénients du filtrage basé sur le contenu des objets

Ce type de filtrage possède plusieurs avantages : lorsqu'un SR possède peu d'utilisateurs, les recommandations effectuées seront tout de même de bonne qualité, car chaque utilisateur est indépendant des autres. Ensuite la connaissance du domaine n'est pas nécessaire, car les recommandations sont construites en se basant sur le corpus. Enfin plus les utilisateurs utilisent le SR, plus les recommandations effectuées seront précises [Poirier 2010].

Cependant plusieurs inconvénients peuvent être cités : lorsqu'un nouvel utilisateur interagit avec le SR, aucune information sur ses préférences n'est disponible. La problématique devient donc quels objets lui proposer lors de ses premières interactions. Cette problématique est connue sous le nom de *départ à froid côté utilisateur*. Un inconvénient majeur intervient si les objets possèdent peu de contenu : il devient difficile de calculer des similarités entre chacun de ces objets. Si les techniques pour traiter le contenu textuel sont nombreuses, le contenu n'est pas forcement textuel, ce qui rend plus complexe l'utilisation des données disponibles. C'est le cas par exemple pour les SR qui recommandent des titres musicaux, dans ce cas des approches de traitement du signal, plus difficiles à utiliser, sont généralement mises en place. Enfin il est important de ne pas toujours recommander l'objet le plus proche des préférences de l'utilisateur. En effet l'utilisateur risque de se voir recommander des objets très similaires en permanence. Ce problème est connu sous le nom de *sur-spécialisation (overspecialization)* [Montaner 2001].

1.2.2 Filtrage collaboratif

Contrairement au filtrage basé sur le contenu, le filtrage collaboratif est basé sur la similarité entre les utilisateurs. Ce filtrage recommande les objets appréciés par les utilisateurs qui ont auparavant fait des choix similaires à ceux de l'utilisateur courant.

1.2.2.1 Mesure de similarité : exemple

Afin d'observer la proximité entre les utilisateurs, une mesure de similarité est généralement utilisée. Pour notre exemple nous utilisons le coefficient de corrélation de Pearson r (Définition 2) souvent utilisé par ailleurs [Resnick 1994]. Ce coefficient est compris entre -1 et 1 ; des valeurs proches de 1 signifient que les deux utilisateurs ont des préférences très proches, tandis que des valeurs proches de -1 caractérisent deux utilisateurs dont les préférences sont opposées. Les valeurs proches de 0 traduisent une corrélation faible.

Notes	**Utilisateur A**	**Utilisateur B**	**Utilisateur C**	**Utilisateur D**	**Utilisateur E**
Objet 1	3	4	5	2	5
Objet 2	5	4	2	4	1
Objet 3	2	2	3	3	2
Objet 4	3	5	5	5	4
Objet 5	3	1	2	2	3

TABLE 1.2 – Exemples de notes

Définition 2 (Coefficient de corrélation de Pearson)**.**

Soient
- α_i, la note donnée par l'utilisateur α à l'objet i,
- β_i, la note donnée par l'utilisateur β à l'objet i,
- $\bar{\alpha}$, la moyenne des notes données par l'utilisateur α
- $\bar{\beta}$, la moyenne des notes données par l'utilisateur β

Le coefficient de corrélation de Pearson r est défini comme :

$$r = \frac{\sum_{i=1}^{N}(\alpha_i - \bar{\alpha})(\beta_i - \bar{\beta})}{\sqrt{\sum_{i=1}^{N}(\alpha_i - \bar{\alpha})^2}\sqrt{\sum_{i=1}^{N}(\beta_i - \bar{\beta})^2}} \tag{1.4}$$

La table 1.2 présente un jeu de données illustratif contenant des appréciations ou notes, de 1 à 5, données par cinq utilisateurs sur cinq objets.

La table 1.3 présente les coefficients de corrélation calculés sur ce jeu de données. Les utilisateurs C et E sont fortement corrélés positivement, ainsi un objet apprécié par l'utilisateur C pourra être recommandé à l'utilisateur E. L'utilisateur A est fortement corrélé négativement avec les utilisateurs B, C et E. Dans ce cas un objet qui n'est pas apprécié par l'un de ces utilisateurs pourra être recommandé à l'utilisateur A.

1.2.2.2 Avantages et inconvénients

Le premier avantage du filtrage collaboratif est qu'il ne nécessite pas d'étudier le contenu des objets, mais se base uniquement sur les profils des utilisateurs. Ainsi lorsque le contenu est faible, ou non-textuel, le filtrage collaboratif est tout de même capable de recommander des objets. De plus, tout comme le filtrage basé sur le contenu,

Corrélation	**Utilisateur A**	**Utilisateur B**	**Utilisateur C**	**Utilisateur D**	**Utilisateur E**
Utilisateur A	1	-0.839	-0.958	-0.343	-0.707
Utilisateur B	-0.839	1	0.662	0.677	0.289
Utilisateur C	-0.958	0.662	1	0.202	0.834
Utilisateur D	-0.343	0.677	0.202	1	-0.243
Utilisateur E	-0.707	0.289	0.834	-0.243	1

Table 1.3 – Corrélation entre les utilisateurs

la connaissance du domaine n'est pas nécessaire. Ce type de filtrage est capable de s'adapter à des contextes différents ; pour un utilisateur inconnu dans un contexte, ce type de filtrage utilise les préférences sur ce contexte des utilisateurs avec un comportement similaire au sein d'un autre contexte où l'utilisateur est connu. Enfin, comme le filtrage basé sur le contenu, la qualité des recommandations croît avec l'utilisation du système.

Cependant à la création du SR, il est impossible de mettre en place une approche par filtrage collaboratif, il est nécessaire qu'un certain nombre d'interactions ait eu lieu par le passé pour calculer des similarités entre utilisateurs. Ce problème est valable également lorsqu'un nouvel objet ou un nouvel utilisateur est ajouté.

1.2.3 Approches hybrides

Des études ont démontré l'efficacité relative des systèmes de filtrage collaboratif par rapport aux systèmes de filtrage basé sur le contenu [Candillier 2009]. Cependant, de nombreux systèmes utilisent des approches hybrides pour compenser les limitations de chacun des filtrages [Burke 2007]. Certains inconvénients du filtrage basé sur le contenu sont corrigés par le filtrage collaboratif, et inversement. Par exemple lorsqu'un nouvel objet est disponible, le filtrage collaboratif ne peut pas être utilisé. Par contre le filtrage basé sur le contenu est capable de calculer la similarité de cet objet avec ceux déjà existants. A l'inverse, si le contenu de certains objets est pauvre, le filtrage basé sur le contenu ne pourra pas être utilisé. Le filtrage collaboratif pourra tout de même

faire le lien avec d'autres objets car ils ont été appréciés par des utilisateurs similaires. Il reste tout de même des inconvénients qui sont communs aux deux types de filtrage, notamment le départ à froid côté utilisateur, le problème est de décider quels objets recommander à un nouvel utilisateur [Burke 2002].

1.2.4 Autres techniques

Si les approches présentées précédemment représentent les techniques fondamentales de la recommandation, d'autres techniques peuvent être citées :

- **Le filtrage démographique** consiste à utiliser les informations démographiques des utilisateurs pour effectuer des recommandations. Par exemple le système peut personnaliser les recommandations en fonction du pays d'origine de l'utilisateur, de son âge, de son sexe, etc. Si ce type de filtrage est utilisé dans la littérature marketing, peu de travaux de la littérature informatique s'intéressent à ce type de filtrage.
- **Le filtrage basé sur la communauté** tient son origine des réseaux sociaux. Ce type de filtrage se base sur les préférences des amis de l'utilisateur [Arazy 2009, Ben-Shimon 2007]. Si le filtrage collaboratif calcule la proximité entre les utilisateurs, ici le fait d'être ami avec un autre utilisateur garantit cette proximité. Ce type de filtrage suit l'idée de l'épigramme "Dis moi qui sont tes amis, je te dirai qui tu es".
- **Le filtrage basé sur la connaissance** est basé sur la connaissance d'un domaine en particulier : le système possède des connaissances a priori sur la capacité des objets à répondre à un besoin d'un utilisateur. Le besoin utilisateur est considéré comme la description d'un problème et les recommandations possibles comme des solutions envisageables [Bridge 2005].

Conclusion

Dans ce chapitre nous avons introduit plusieurs concepts de recommandation. Pour cela nous avons commencé par citer plusieurs exemples de systèmes de recommandation connus et décrit les techniques classiquement mises en œuvre. Nous avons mis en évidence que si l'utilisation d'une technique seule impose plusieurs inconvénients, ces derniers peuvent être limités en hybridant plusieurs techniques. Des travaux récents proposent d'utiliser des approches de la littérature statistique afin d'optimiser les recommandations effectuées : les stratégies de bandit. Les travaux effectués durant cette

thèse traitent de l'adaptation de ces stratégies pour leur utilisation en recommandation et le chapitre suivant présente les bases de ces approches.

Chapitre 2

Problèmes de bandit : principe, formalisme et stratégies

Sommaire

Résumé.

Pour maximiser le gain obtenu, il est important d'exploiter les objets les plus populaires par le passé tout en étant capable d'explorer les objets peu connus afin de ne pas passer à côté d'un objet encore plus populaire. Les stratégies de bandit sont connues pour proposer des solutions efficaces à ce problème.

Dans ce chapitre nous présentons les approches statistiques que nous utilisons dans nos travaux : les stratégies de bandit. Tout d'abord nous introduisons les «problèmes de bandit». Ensuite nous recensons quelques unes des stratégies utilisées dans la littérature statistique pour aborder cette problématique. Enfin nous présentons plusieurs problématiques spécifiques aux problèmes de bandit, dont certaines sont abordées dans nos travaux.

2.1 Introduction aux problèmes de bandit

2.1.1 Contexte

Même si l'avènement d'internet et des systèmes de recommandation a fortement relancé l'intérêt qui leur est porté, l'étude des problèmes de bandit a débuté il y a longtemps et leur intérêt applicatif est bien plus vaste. Ils modélisent des situations où un agent, plongé à chaque instant dans un certain contexte, doit choisir séquentiellement une suite d'actions qui lui assure un certain gain aléatoire, et qui influe sur ses observations futures. Il s'agit de concevoir et d'analyser des règles de décision dynamiques, appelées politiques ou stratégies, en utilisant les observations passées pour optimiser les choix futurs. Une bonne politique doit réaliser un équilibre entre l'exploitation des actions qui se sont révélées payantes par le passé et l'exploration de nouvelles possibilités qui pourraient s'avérer encore meilleures. Ce problème est connu sous le nom de "dilemme exploration/exploitation".

Initialement motivés essentiellement par la thématique des essais cliniques, ces modèles interviennent désormais dans de nombreux autres domaines industriels, les technologies de l'information en ayant multiplié les opportunités.

Domaines d'application Historiquement les premières stratégies ont été développées il y a plus de 80 ans [Thompson 1933] pour le domaine médical, et plus particulièrement les essais cliniques, afin de réduire le nombre de patients ne bénéficiant pas du meilleur traitement.

Dans les logiciels de jeux, chaque mouvement doit être choisi en simulant et évaluant comment va évoluer le jeu suite à ce mouvement. Les stratégies de bandit peuvent être utilisées pour se focaliser sur l'exploration des mouvements les plus prometteurs. Par exemple cette idée a été implémentée pour le jeu de Go, où le programme-joueur MoGo développé par [Gelly 2006] est basé sur une stratégie de type UCB appliquée aux arbres de décision. MoGo a déjà battu plusieurs des meilleurs joueurs mondiaux. Le programme AlphaGo, développé par Google DeepMind, est considéré comme le

meilleur programme-joueur mondial actuellement, il est également basé sur une stratégie de bandit.

Ces stratégies sont également utilisées dans le routage de paquets sur un réseau. Une séquence de paquets doit être acheminée d'un hôte de départ à un hôte d'arrivée, plusieurs chemins sont disponibles et chaque paquet peut être envoyé sur un chemin différent. L'objectif est de délivrer le plus rapidement possible la séquence entière de paquets.

Les méthodes d'A/B testing consistent à proposer deux versions d'un logiciel ou d'une interface web un certain nombre de fois donné afin de pouvoir identifier la version la plus efficace selon un critère donné. Les stratégies de bandit sont utilisées dans ce domaine afin de minimiser le nombre de fois où la version la moins efficace est proposée. Le livre *Bandit algorithms for website optimization* [White 2012] décrit de manière très accessible pourquoi ces stratégies peuvent être utiles dans ce domaine et comment elles peuvent être implémentées.

Durant cette thèse nous nous sommes intéressés en particulier au domaine de la recommandation décrit dans le précédent chapitre. Des travaux proposent d'utiliser les stratégies de bandit dans ce domaine, afin d'améliorer les recommandations effectuées, qu'il s'agisse de publicités, d'objets, de musiques, de vidéos, etc. Plus de détails sont donnés dans la suite de ce manuscrit.

2.1.2 Définition et bornes de regret

2.1.2.1 Problème de bandit stochastique

Dans la version stochastique [1] la plus simple du problème,

- à chaque étape $t = 1, 2, \ldots$, l'agent choisit un bras A_t parmi la collection de K bras $a \in 1, \ldots, K$
- et il reçoit une récompense X_t telle que, conditionnellement au choix du bras joué, les récompenses soient indépendantes et telles que l'espérance de X_t est p_{A_t} avec $a \in 1, \ldots, K$. Cette espérance est généralement estimée par

$$\hat{p}_a(t) = \frac{1}{N_a(t)} \sum_{s=1}^{t} X_s \mathbb{I}_{A_s=a} \tag{2.1}$$

On appelle *politique* ou stratégie la règle de décision qui, aux observations passées, associe le prochain choix A_t. Le meilleur choix, inconnu de l'agent, est le bras a^* qui correspond à la récompense moyenne maximale p_{a^*}.

La performance d'une politique est mesurée par le regret cumulé R_n, il est présenté à la définition 3.

Définition 3 (Regret cumulé Rn)**.**

Soient

- X_t la récompense obtenue par l'agent à l'instant t,
- a_* est le bras qui correspond à la récompense maximale p_{a^*}

Le regret cumulé R_n est la différence moyenne entre les récompenses qu'elle permet d'accumuler jusqu'au temps $t = n$ et celles qui auraient pu être obtenues pendant la même période si le meilleur bras était connu à l'avance :

$$R_n = np_{a^*} - \mathbb{E}\left[\sum_{t=1}^{n} X_t\right] \tag{2.2}$$

1. Il existe une variante dite *adversariale*, relevant de la théorie des jeux, où les récompenses sont choisies non pas au hasard mais par un adversaire ; il n'en sera pas question dans ce manuscrit. Outre [Cesa-Bianchi 2006], le lecteur intéressé pourra se référer aux travaux de Gilles Stoltz pour en saisir les subtilités.

2.1.2.2 Borne inférieure de regret

Une stratégie de bandit ne peut pas être arbitrairement bonne : il existe une borne inférieure de regret qu'elle doit encourir dès lors qu'il offre des garanties uniformes de performances. La plus célèbre de ces bornes est celle de [Lai 1985] (voir l'article [Kaufmann 2014] pour une preuve moderne et plus générale). Dans le cas de récompenses binaires (suivant une loi de Bernoulli), elle stipule que si une politique assure dans tout environnement un regret sous-polynômial, alors celui-ci est toujours logarithmique. Cette borne inférieure de R_n (Théorème 1) correspond à la performance maximale que la politique peut obtenir.

Théorème 1 (Borne inférieure de Lai et Robbins [Lai 1985])**.**

Si une politique assure dans tout environnement un regret sous-polynomial, si les récompenses sont binaires et quels que soient $p_1, p_2, \ldots, p_K \in]0,1[$, alors on obtient la borne inférieure pour le regret suivante :

$$R_n \geq \left(\sum_{a:p_a<p_{a*}} \frac{p_{a*} - p_a}{kl(p_a, p_{a*})} \right) \log(n)(1 - o(1)), \tag{2.3}$$

où kl désigne l'entropie binaire :

$$kl(x, y) = x \log\left(\frac{x}{y}\right) + (1 - x) \log\left(\frac{1 - x}{1 - y}\right) \tag{2.4}$$

2.1.2.3 Borne supérieure de regret

Généralement, une borne supérieure de regret est également étudiée. Cette borne permet de certifier le niveau de performance minimale de la politique en question. Ces bornes diffèrent selon les politiques : par exemple, cette borne peut être d'ordre linéaire, ce qui traduit plutôt un comportement où la part d'exploration est constante, ou encore d'ordre logarithmique, ce qui démontre que la part d'exploration décroit avec le temps, au profit de l'exploitation. Lorsque nous présentons une politique, nous précisons quel type de borne elle réussi à atteindre asymptotiquement, c'est à dire lorsque T tend vers l'infini. Il est possible d'obtenir une borne de regret avec un horizon T fini, pour illustrer cela nous présentons la preuve permettant d'obtenir ce type de regret pour la stratégie *UCB1* présentée dans la section suivante.

2.2 Stratégies à tirages simples de la littérature statistique

A un instant t donné, la première stratégie qui nous vient à l'esprit consiste à recommander le bras a tel que $a = \text{argmax}_{a \in (1,\dots,K)} \hat{p}_a$, c'est à dire le bras ayant obtenu la moyenne des récompenses la plus forte par le passé. Cette stratégie n'est pas la bonne : elle se fie, à l'étape t, aux récompenses obtenues dans le passé pour construire des estimateurs $\hat{p}_1, \dots, \hat{p}_K$ en faisant comme si on avait pour chaque bras un échantillon indépendant et identiquement distribué. Cette stratégie peut être trompée par des observations malchanceuses lors des premiers essais du meilleur bras, et elle ne réalisera jamais son erreur si celui-ci n'est plus tiré. S'il est nécessaire de se concentrer sur les bras qui ont montré de bonnes performances dans le passé, il convient de ne jamais exclure complètement les autres. Dans la suite de cette section, nous décrivons plusieurs stratégies de bandit parmi les plus populaires.

2.2.1 Stratégie ϵ-*greedy* et variantes

2.2.1.1 ϵ-greedy

Algorithme 2.1 : ϵ-greedy

K bras sont disponibles
Initialisation : jouer chaque bras $a \in (1, \dots, K)$ une fois
pour $t = K+1, \dots, T$ **faire**
  - Avec une probabilité $1 - \epsilon$, jouer $A_t = \text{argmax}_{a \in (1,\dots,K)} \hat{p}_a$
  - Sinon, jouer $A_t = a$ avec a tiré selon une loi uniforme dans $(1, \dots, K)$

fin

La stratégie ϵ-*greedy* [Watkins 1989] est sûrement la plus simple à comprendre car l'exploration et l'exploitation sont clairement distinguées. L'idée est de jouer majoritairement le bras avec le gain moyen estimé le plus fort, mais également d'explorer en tirant un bras aléatoirement, selon une loi uniforme, avec une faible probabilité ϵ.

2.2.1.2 ϵ-first

La variante la plus simple est la stratégie ϵ-*first* ; elle consiste à réaliser toute l'exploration au début de l'expérience. Pour un horizon donné $T \in \mathbb{N}$ d'interactions, le bras joué est tiré suivant une loi uniforme parmi les bras disponibles durant les $T\epsilon$ premières

Algorithme 2.2 : ϵ-first

K bras sont disponibles
Initialisation : jouer chaque bras $a \in (1, \ldots, K)$ une fois
pour $t = K + 1, \ldots, T$ **faire**
  si *Si* $t \leq T\epsilon$ **alors**
    jouer $A_t = a$ avec a tiré selon une loi uniforme dans $(1, \ldots, K)$
  sinon
    jouer $A_t = \text{argmax}_{a \in (1,\ldots,K)} \hat{p}_a$
  fin
fin

interactions. Durant les $(1 - \epsilon)T$ interactions restantes, le bras avec le gain moyen estimé le plus fort est joué. Even-Dar *et al.* [Even-Dar 2002] prouvent que $O\left(\frac{K}{\alpha^2} \log \frac{K}{\eta}\right)$ tirages aléatoires suffisent pour trouver un bras optimal avec une probabilité $1 - \eta$.

2.2.1.3 ϵ-decreasing

Algorithme 2.3 : ϵ-decreasing

K bras sont disponibles
ϵ_0 est la valeur d'origine de ϵ
Initialisation : jouer chaque bras $a \in (1, \ldots, K)$ une fois
pour $t = K + 1, \ldots, T$ **faire**
  $\epsilon_t = min\{1, \frac{e_0}{t}\}$
    – Avec une probabilité $1 - \epsilon_t$, jouer $A_t = \text{argmax}_{a \in (1,\ldots,K)} \hat{p}_a$
    – Sinon, jouer $A_t = a$ avec a tiré selon une loi uniforme dans $(1, \ldots, K)$
fin

Lorsque la valeur de ϵ est constante sur toute la durée de l'expérimentation, la part d'exploration reste la même, malgré le fait qu'il est possible que chaque bras soit joué suffisamment pour estimer précisément son gain moyen. Le regret croît donc de manière linéaire ($O(T)$). Dans ce cas avoir une valeur constante pour η est clairement sous-optimale, une valeur de η qui diminue au fur et à mesure que les interactions passent semble être plus appropriée. C'est de cette idée que découle la stratégie ϵ-*decreasing*. Le principe est le même que pour la stratégie ϵ-*greedy*, la seule distinction est le fait que la part d'exploration ϵ_t diminue avec le nombre d'interactions. Cette valeur est donnée par $\epsilon_t = min\{1, \frac{e_0}{t}\}$ avec $e_0 > 0$. Auer, Cesa-Bianchi et Fisher [Auer 2002a] proposent une analyse permettant d'obtenir un regret de l'ordre

de $O(\log(T))$ où T est le nombre d'interactions. Plusieurs variantes de la stratégie ϵ-*decreasing* on été proposées, elles diffèrent en un point : la façon dont ϵ_t décroit [Cesa-Bianchi 1998, Auer 2002a].

2.2.2 Stratégies de type UCB

Afin de canaliser l'exploration vers les bras qui en valent vraiment la peine, les stratégies de type UCB (Upper Confidence Bound) ont été développées. Elles utilisent non pas des estimateurs, mais des bornes de confiance supérieures pour l'espérance de chaque bras comme indices de qualité. Ces stratégies sont dites "optimistes dans l'incertain", c'est à dire que le bras ayant potentiellement l'espérance la plus forte est recommandé. Ce choix peut se faire au détriment d'un bras ayant une meilleure espérance estimée, mais dont la borne de confiance supérieure est plus faible, car il a été beaucoup tiré par le passé. Globalement, ces stratégies tirent le bras qui maximise la quantité

$$A_t = \underset{a\in(1,\dots,K)}{\operatorname{argmax}}\, \hat{p}_a(t) + B_a(t) \tag{2.5}$$

où $B_a(t)$ est un bonus de confiance. Les stratégies de type UCB se distinguent principalement par la manière dont la borne de confiance supérieure est calculée.

2.2.2.1 UCB1

Algorithme 2.4 : UCB1

K bras sont disponibles
Initialisation : jouer chaque bras $a \in (1, \dots, K)$ une fois
pour $t = K + 1, \dots, T$ **faire**
    jouer $A_t = \operatorname{argmax}_{a\in(1,\dots,K)} \hat{p}_a + \sqrt{\frac{2\log(t)}{N_a(t)}}$
fin

Cette stratégie, dont le bonus est calculé en utilisant l'inégalité de Hoeffding, permet d'obtenir une borne de regret logarithmique non-asymptotique [Auer 2002a] (Voir théorème 2). Nous fournissons également la preuve complète de cette borne, car dans nos travaux elle est adaptée pour un cadre où les récompenses décroissent avec le temps. Il est donc important de bien comprendre la preuve initiale pour appréhender la preuve décrite dans le chapitre 5.

Toutefois, la stratégie *UCB1* a un comportement sous-optimal qui peut s'avérer assez décevant dans le cas (fréquent en recommandation) où les récompenses moyennes

sont toutes très faibles. Cela se conçoit bien : la borne de Hoeffding est alors très pessimiste car les valeurs des différents p_a sont très petites.

Théorème 2 (UCB1 : borne supérieure de regret non asymptotique [Auer 2002a])**.**

Pour tout $K > 1$, si la stratégie UCB1 est utilisée sur K bras ayant des récompenses suivant des distributions de Bernoulli de paramètres $p_1, \ldots, p_K$, alors le regret cumulé après t interactions est au plus de

$$\left[8 \sum_{a:p_a<p_{a^*}} \left(\frac{\log(t)}{\Delta_a}\right)\right] + \left(1 + \frac{\pi^2}{3}\right) \left(\sum_{j=1}^{K} \Delta_j\right) \tag{2.6}$$

où $\Delta_a = p_{a^*} - p_a$

Preuve théorème 2 [Auer 2002b] : On considère un entier n, un bras optimal a^* et un bras sous-optimal a tel que $p_a < p_{a^*}$. Pour un bras b, la moyenne des récompenses obtenues par le passé en jouant le bras b est notée $\hat{p}_b(t) = \frac{1}{N_a(t)} \sum_{r=1}^{t} X_r \mathbb{I}_{A_r=b}$. Par commodité, pour un entier positif s nous notons également $\hat{p}_{b,s} = (X_{b,1} + \ldots + X_{b,s})/s$, tel que $\hat{p}_b(t) = \hat{p}_{b,N_b(t)}$. La stratégie UCB joue le bras qui maximise la quantité

$$u_b(t) = \hat{p}_b(t) + B_{t,N_b(t)} \text{ avec } B_{t,N_b(t)} = \sqrt{\frac{2\ln(t)}{N_b(t)}} \tag{2.7}$$

C est un entier positif.

$$\begin{aligned}
N_a(n) &= 1 + \sum_{t=K+1}^{n} \{A_t = a\} && (2.8)\\
&\leq C + \sum_{t=K+1}^{n} \{A_t = a, N_a(t-1) \geq C\} && (2.9)\\
&\leq C + \sum_{t=K+1}^{n} \{u_{a^*}(t-1) \leq u_a(t-1), N_a(t-1) \geq C\} && (2.10)\\
&\leq C + \sum_{t=K+1}^{n} \{\min_{0<s<t} u_{a^*}(s) \leq \max_{C \leq s_a < t} u_a(s_a)\} && (2.11)\\
&\leq C + \sum_{t=1}^{\infty} \sum_{s=1}^{t-1} \sum_{s_a=C}^{t-1} \{u_{a^*}(s) \leq u_a(s_a)\} && (2.12)
\end{aligned}$$

Observer $u_{a^*}(s) \leq u_a(s_a)$ implique l'un des cas suivants :

$$\hat{p}_{a^*}(s) \leq pa^* - B_{t,s} \tag{2.13}$$
$$\hat{p}_a(s_a) \geq p + B_{t,s_a} \tag{2.14}$$
$$p_{a^*} < p_a + 2B_{t,s_a} \tag{2.15}$$

La probabilité des événements 2.13 et 2.14 est bornée en utilisant l'inégalité de Chernoff-Hoeffding :

$$\Pr(\hat{p}_{a^*}(s) \leq p_{a^*} - B_{t,s}) \leq \exp\left(-2\frac{B_{t,s}^2}{1}\right) \tag{2.16}$$
$$\leq \exp\left(-4\frac{\ln(t)}{s}\right) \tag{2.17}$$
$$\leq \exp(-4\ln(t)) \tag{2.18}$$
$$\leq t^{-4} \tag{2.19}$$

De même, $\Pr(\hat{p}_a(s_a) \geq p_a + c_{t,s_a}) \leq t^{-4}$

Si 2.15 est vraie, alors $p_{a^*} - p_a - 2B_{t,s} < 0$. Pour $C = \frac{8\ln(n)}{\Delta^2}$ avec $\Delta = (p_{a^*} - p_a)$, 2.15 n'est pas vraie. En effet

$$p_{a^*} - p_a - 2B_{t,s_a} = p_1 - p_a - 2\sqrt{\frac{2\ln(t)\Delta^2}{8\ln(n)}} \tag{2.20}$$
$$= p_1 - p_a - \sqrt{\frac{\ln(t)\Delta^2}{\ln(n)}} \tag{2.21}$$
$$\geq p_1 - p_a - \Delta = 0 \tag{2.22}$$
$$\tag{2.23}$$

Pour $s_a > \frac{8\ln(n)}{\Delta^2}$, nous obtenons

$$N_a(n) = \frac{8\ln(n)}{\Delta^2} + \sum_{t=1}^{\infty}\sum_{s=1}^{t-1}\sum_{s_a=\frac{8\ln(n)}{\Delta^2}}^{t-1} (\Pr(\hat{p}_a(s) \geq p_a + B_{t,s_a}) + \Pr(\hat{p}_1(s) \leq p_1 - B_{t,s})) \tag{2.24}$$

$$\leq \frac{8\ln(n)}{\Delta^2} + \sum_{t=1}^{\infty}\sum_{s=1}^{t-1}\sum_{s_a=1}^{t-1} (2t^{-4}) \tag{2.25}$$

$$\leq \frac{8\ln(n)}{\Delta^2} + \sum_{t=1}^{\infty} (2t^{-2}) \tag{2.26}$$

$$\leq \frac{8\ln(n)}{\Delta^2} + \frac{\pi^2}{3} + 1 \tag{2.27}$$

□

2.2.2.2 KL-UCB

Algorithme 2.5 : KL-UCB

K bras sont disponibles
Initialisation : jouer chaque bras $a \in (1, \ldots, K)$ une fois
pour $t = K+1, \ldots, T$ **faire**
    jouer $A_t =$
    $\text{argmax}_{a\in(1,\ldots,K)} \max \{q \in [0,1] : N_a(t) kl(\hat{p}_a, q) \leq \log(t) + c\log(\log(t))\}$
fin

Cette variante (algorithme 2.5) est proposée dans [Garivier 2011]. Dans cet article une borne non-asymptotique de regret est montrée d'où peut être déduite l'optimalité au sens de la borne de Lai et Robbins 1. Le calcul de la borne supérieure de confiance est un peu plus complexe, mais le gain de performance n'est pas seulement théorique. Cette stratégie permet d'obtenir de bonnes performances en pratique, même si les récompenses moyennes sont très faibles. Dans la stratégie 2.5, le paramètre c est présent pour obtenir des garanties sur les bornes de regret, mais en pratique les auteurs conseillent d'utiliser $c = 0$.

2.2.3 Thompson Sampling

Une stratégie ancienne d'inspiration bayésienne est aujourd'hui considérée comme l'un des plus prometteuses à la fois d'un point de vue théorique et d'un point de vue

Algorithme 2.6 : Thompson Sampling

Pour chaque bras $a \in (1, \dots, K)$, initialiser $S_a = 0$ et $F_a = 0$
pour $t = 1, 2, \dots$ **faire**
  Pour chaque bras a avec $a = 1, \dots, K$, tirer $\theta_a(t)$ selon une distribution Beta($S_a + 1, F_a + 1$).
  Jouer $A_t = \text{argmax}_{a \in (1,\dots,K)} \, \theta_a(t)$ et observer la récompense X_t
  si $X_t = 1$ **alors**
    $S_{A_t} = S_{A_t} + 1$
  sinon
    $F_{A_t} = F_{A_t} + 1$.
  fin
fin

pratique : *Thompson sampling* [Thompson 1933]. Cette stratégie (décrite en détail dans l'algorithme 2.6) consiste à recommander le bras avec l'espérance, tiré aléatoirement selon la loi de probabilité associée à ce bras, la plus forte. La loi de probabilité est ensuite mise à jour en fonction du retour utilisateur obtenu. Si la stratégie a plus de 80 ans, son étude récente permet d'obtenir des garanties théoriques fortes, et en pratique, de bonnes performances en terme de regret. En effet l'étude détaillée de cette stratégie permet d'obtenir une borne supérieure de regret asymptotique logarithmique [Agrawal 2012, Kaufmann 2013].

2.2.4 Successive Elimination

Cette stratégie, proposée par [Even-Dar 2006], consiste à jouer à tour de rôle les bras, et à les éliminer les uns après les autres. A un instant donné, il reste S bras disponibles, et chaque bras restant a été joué τ fois. Un bras a est éliminé si

$$\hat{p}_a < \max_{i \in 1,\dots,S} \hat{p}_i - \gamma U(\tau, t) \tag{2.28}$$

$$\text{où } U(\tau, t) = 2\sqrt{\frac{2\overline{\log}(T/\tau)}{\tau}} \text{ avec } \overline{\log(t)} = \log(t) \vee 1 \tag{2.29}$$

2.3 Stratégies de bandit et problématiques spécifiques

Le cadre stochastique tel qu'il est décrit dans la section 2.1 de ce chapitre peut paraître simpliste, et nécessite souvent d'être adapté lorsque ces stratégies sont appliquées sur des problèmes réels. Plusieurs problématiques peuvent être citées, les deux dernières sont développées plus en détail dans la suite de ce document, car elles représentent la base de certains de nos travaux.

- **Nombre de bras infini**. Il est possible de faire face à un problème où tous les bras ne peuvent pas être testés, mais où des informations sur les relations entre ces bras sont disponibles [Berry 1997, Carpentier 2015]. Dans ce cas, il convient de se focaliser sur certains bras, répartis dans la population, pour identifier dans la population un bras qui obtient des performances satisfaisantes. C'est le cas notamment pour le jeu de Go, où le nombre de possibilités après un coup est très grand. Pour aborder ce grand nombre de bras disponibles, des stratégies permettant d'explorer des arbres de décision sont notamment développées [Gelly 2006, Coquelin 2007].
- **Aspect contextuel**. En recommandation, lorsque des informations contextuelles sont disponibles sur les objets et/ou les utilisateurs, il est important de les prendre en compte (Filtrage collaboratif, filtrage basé sur le contenu, etc.). Des stratégies ont été développées afin de gérer ces informations. Cette problématique est connue sous plusieurs noms dans la littérature statistique : bandits avec contexte [Li 2010], bandits associatifs [Strehl 2006], bandits avec conseil d'expert [Auer 2002a], ou encore bandits avec covariables [Sarkar 1991, Perchet 2013]. L'approche la plus connue est de type UCB : *LinUCB* [Li 2010]. Elle consiste à maintenir un vecteur de contexte, contenant plusieurs paramètres, afin d'ajuster la borne supérieure de confiance calculée.
- **Non-stationnarité du gain moyen associé à un bras**. Il est peu réaliste, surtout en recommandation, qu'un bras permette d'obtenir toujours la même récompense en moyenne. Par exemple, les actualités anciennes recommandées par des sites comme *Le Monde*[2] ou encore *Le Figaro*[3] doivent être mises de côté pour privilégier des actualités plus récentes. Cette problématique est plus détaillée dans le chapitre 4, dans lequel nous réalisons un état de l'art des approches existantes. Les approches issues de nos travaux associées à cette problématique

2. http ://www.lemonde.fr
3. http ://www.lefigaro.fr

sont présentées dans le chapitre 5.

— **Tirages multiples**. De nombreux systèmes de recommandation doivent recommander plusieurs documents à chaque instant. Dans ce cas des stratégies capables de jouer plusieurs bras à chaque instant sont nécessaires. Dans ce cas le problème est de construire une liste de bras à jouer permettant à l'agent d'obtenir un gain maximal. Nous décrivons dans le chapitre 6 cette problématique, nous décrivons les approches existantes en recommandation qui l'aborde et faisons notamment le lien avec la recherche d'information. Dans le chapitre suivant (chapitre 7), nous montrons comment les stratégies de bandit à tirages simples, comme ceux présentés dans la section précédente, sont utilisés en recommandation pour recommander plusieurs documents à chaque instant. Dans le chapitre 8, des stratégies de bandit récentes de la littérature statistique, capables de recommander plusieurs items, sont présentées. Nous décrivons également comment une de ces approches a été adaptée pour pouvoir être appliquée en recommandation. Enfin dans le chapitre 9, nous abordons la problématique de la diversité des éléments au sein de la liste recommandée. Nous proposons une approche qui expérimentalement obtient de très bons résultats.

Conclusion

Dans ce chapitre nous avons présenté les «problèmes de bandit». Nous avons montré que plusieurs familles de stratégies peuvent être utilisées. Dans la littérature statistique, de nombreux travaux s'intéressent à la quantification des performances minimales et maximales de ces stratégies en étudiant les bornes inférieures et supérieures de regret. Afin d'illustrer comment ces bornes peuvent être obtenues, nous avons repris une preuve permettant d'obtenir une borne supérieure de regret logarithmique pour la stratégie UCB1. Nous avons également listé un certain nombre de problématiques spécifiques complexifiant le modèle initial.

Un lien entre ces problématiques et la recommandation peut facilement être fait. Par exemple des problématiques communes sont la prise en compte de la non stationnarité du gain moyen associé à un bras ou encore le tirage de plusieurs bras à chaque itération. Dans le chapitre suivant nous mettons en évidence pourquoi l'utilisation de ces stratégies en recommandation se révèle être une alternative intéressante aux approches mises en œuvre habituellement.

Chapitre 3

Stratégies de bandit et recommandation : problématiques et contributions

Sommaire

Résumé.

En recommandation, des utilisateurs interagissent de manière séquentielle avec le système. Le choix des objets recommandés à chaque instant permet d'obtenir de nouvelles informations concernant leurs pertinences. Mais il est également important de recommander des objets connus pour être pertinents. Nous sommes en présence du dilemme exploration/exploitation et les stratégies présentées dans le chapitre précédent sont connues pour proposer des solutions à ce dilemme.

Dans le premier chapitre, nous avons décrit les SR ainsi que les techniques habituellement utilisés dans ce cadre. Dans le second chapitre, nous avons introduit les problèmes de bandit, et montré plusieurs approches conçues pour aborder ces problèmes. Nous présentons maintenant notre problématique de recherche : comment adapter les stratégies de bandit pour les appliquer à la recommandation ?

3.1 Dilemme exploration/exploitation : problématique commune

Afin d'améliorer la pertinence des recommandations pour un utilisateur, un SR peut considérer l'historique des interactions passées avec les utilisateurs. Lorsque toutes les données sont connues, il est possible d'estimer la pertinence des objets, il s'agit alors du cadre de l'apprentissage supervisé [Hastie 2009]. Ce n'est pas un cadre réaliste pour un SR : de nouveaux utilisateurs et de nouveaux objets apparaissent continuellement. Par ailleurs il est souhaitable de pouvoir apprendre continuellement tout en recommandant des objets. Un tel environnement s'inscrit dans le cadre de ce que l'on appelle l'*apprentissage par renforcement* [Sutton 1999]. Il s'agit d'implémenter une stratégie pour explorer de nouvelles données d'apprentissage (les interactions des utilisateurs), tout en assurant que les résultats puissent être directement exploités. Ce problème est connu sous le nom de « dilemme exploration/exploitation ». Les stratégies de bandit sont connues pour offrir des solutions à ce dilemme [Bubeck 2012]. Elles sont généralement conçues pour recommander un seul objet à chaque instant : ce sont des stratégies à tirages simples. Plusieurs familles de stratégies de bandit se distinguent de par la stratégie choisie pour appréhender le « dilemme exploration/exploitation ».

Nos travaux se placent dans le cadre de l'application des stratégies de bandit à la recommandation. Plus précisément, ils se focalisent sur deux problématiques de la

recommandation. La première est l'*obsolescence progressive* de la popularité d'un objet, en effet il parait irréaliste de considérer que la popularité d'un objet n'évolue pas dans le temps. La seconde est *la recommandation à tirages multiples*, qui peut être divisée en plusieurs sous-problématiques : comment recommander plusieurs objets et comment ajouter de la diversité au sein de la liste des objets recommandés.

3.2 Contributions

3.2.1 Contribution 1 : proposition de deux approches permettant de prendre en compte l'obsolescence progressive de la popularité des objets

Les stratégies de bandit sont des approches séquentielles et sont conçues pour que les mises à jour nécessaires pour maintenir la stratégie soient peu nombreuses. Elles sont particulièrement adaptées à un cadre en ligne, où une réponse très rapide est nécessaire. C'est le cas de beaucoup de SR. Une problématique importante dans un cadre en ligne est la non-stationnarité des récompenses obtenues. Dans notre première contribution, introduite dans un article publié à la conférence CAP 2016, nous proposons deux approches permettant de prendre en compte un cadre où la popularité des objets diminue avec le temps [Louëdec 2016].

Dans le chapitre 4, nous présentons les stratégies de bandit existantes capables d'aborder la problématique de la non-stationnarité des récompenses. Ensuite nous définissons le modèle utilisé, et le formalisons afin de pouvoir présenter nos approches.

Dans le chapitre 5, nous introduisons les stratégies que nous proposons *Fading-UCB* et *Trust and Abandon*. Afin d'identifier les performances minimales de la stratégie *Fading-UCB*, Nous fournissons une borne supérieure de regret pour celle-ci. Enfin nous évaluons à l'aide de simulations les performances en terme de regret cumulé des deux stratégies proposées par rapport aux approches de l'état de l'art.

3.2.2 Contribution 2 : proposition d'une approche utilisant des stratégies de bandit à tirages simples pour recommander plusieurs objets simultanément

La seconde contribution est une approche pour la recommandation à tirages multiples. Elle consiste à adapter les stratégies de bandit à tirages simples pour pouvoir recommander plusieurs objets à chaque instant. Elle a été introduite dans un article

publié en 2015 dans la conférence nationale *CORIA* [Louëdec 2015d]. Cet article a ensuite été sélectionné par le comité de programme de la conférence pour être étendu dans le numéro spécial de *Document Numérique* dédié aux conférences CORIA 2014 et 2015 [Louëdec 2015a].

Dans le chapitre 6, nous commençons par présenter la problématique de recommandation à tirages multiples. Nous citons ensuite plusieurs travaux de la littérature informatique proposant des solutions pour recommander plusieurs objets à chaque instant. Nous finissons en décrivant comment les approches de recommandation sont évaluées, en faisant la différence entre l'évaluation en ligne et l'évaluation hors-ligne.

Dans le chapitre 7, nous présentons les différentes approches de l'état de l'art qui utilisent des stratégies de bandit à tirages simples pour recommander plusieurs objets simultanément. Nous montrons ensuite les limites de ces approches et nous introduisons ensuite l'approche «Multiple-Play Bandit» (L'approche «Multiple-Play Bandit» (MPB)) dont nous comparons les performances avec les approches de l'état de l'art en utilisant deux jeux de données de référence : *Jester* et *Movielens*.

3.2.3 Contribution 3 : adaptation et application d'une stratégie de bandit à tirages multiples pour la recommandation à tirages multiples

Des travaux récents de la littérature statistique proposent des stratégies de bandit conçues spécialement pour recommander plusieurs objets simultanément [Bubeck 2012]. Dans un article publié dans la conférence FLAIRS [Louëdec 2015c], nous proposons une adaptation de l'une de ces stratégies afin de l'utiliser en recommandation.

Dans le chapitre 8, nous commençons par présenter les stratégies de bandit issues de la littérature statistique capables de recommander plusieurs objets à chaque instant. Nous présentons ensuite une adaptation de la stratégie *Exp3.M* pour la recommandation, que nous évaluons sur deux jeux de données de référence : *Jester* et *Movielens*.

3.2.4 Contribution 4 : proposition de deux approches permettant d'apporter de la diversité dans les listes de résultats

Si des travaux utilisant des stratégies de bandit pour effectuer de la recommandation à tirages multiples sont disponibles, notamment de part les contributions précédentes, une problématique connexe n'est jusqu'ici pas abordée : comment apporter de la diversité dans les listes d'objets recommandés. Dans notre quatrième contribution, introduite dans un article en cours de soumission, nous proposons une nouvelle

approche, utilisant des probabilités conditionnelles, capable d'apporter de la diversité dans les listes d'objets recommandés. Nous décrivons également pourquoi apporter de la diversité peut être important.

Dans le chapitre 9, nous commençons par introduire la notion de diversité et les cas dans lesquels la diversité peut être déterminante. Nous introduisons ensuite les approches «2-Diversified Bandit» et «K-Diversified Bandit» que nous avons proposées et qui permettent d'apporter de la diversité à deux degrés différents dans les listes d'objets recommandés. Nous comparons ensuite ces approches avec celles de l'état de l'art sur deux jeux de données de référence : *Jester* et *Movielens*.

3.3 Publications

Les contributions et travaux ont donné lieu à des publications, ainsi qu'à des présentations dans des conférences ou séminaires. Une liste détaillée des publications acceptées, des communications ainsi que des articles soumis ou en préparation est présentée ci-après. Pour les articles précédés d'un astérisque j'ai moi même présenté oralement le travail à la conférence liée.

3.3.1 Publications acceptées

Article dans revue nationale

- [Louëdec 2015a] Jonathan Louëdec, Max Chevalier, Aurélien Garivier et Josiane Mothe. *Algorithmes de bandits pour la recommandation à tirages multiples*. Document numérique, Hermès, Vol. 18/2-3/2015, p. 59-79, 2015.
- [Cabanac 2015] Guillaume Cabanac, Amira Derradji, Ali Jaffal, Jonathan Louëdec, Gloria Elena Jaramillo Rojas. *Forum Jeunes Chercheurs à Inforsid 2014*. Ingénierie des Systèmes d'Information, Hermès Science, Vol. 20, N. 2, p. 119-143, 2015.

Article dans conférence internationale avec actes et comités de lecture

- *[Louëdec 2015c] Jonathan Louëdec, Max Chevalier, Josiane Mothe, Aurélien Garivier. *A Multiple-play Bandit Algorithm Applied to Recommender Systems (regular paper)*. Florida Artificial Intelligence Research Society (FLAIRS 2015), Hollywood, Floride, Etats-Unis, 18/05/15-20/05/15, AAAI Press, p. 67-72, mai 2015.

Article dans conférence nationale avec actes et comité de lecture

- *[Louëdec 2016] Jonathan Louëdec, Laurent Rossi, Max Chevalier, Aurélien Garivier, Josiane Mothe. *Algorithme de bandit et obsolescence : un modèle pour la recommandation*. Conférence francophone sur l'apprentissage automatique (CAP), Marseille, France, 04/07/16-08/07/16, Accepté, à paraître, juillet 2016.
- [Louëdec 2015b] Jonathan Louëdec, Max Chevalier, Aurélien Garivier, Josiane Mothe. *Systèmes de recommandations : algorithmes de bandits et évaluation experimentale*. Journées de Statistique de la SFdS (JDS 2015), Lille, France, 01/05/15-05/05/15, Société Française de Statistiques (SFdS), mai 2015.
- *[Louëdec 2015d] Jonathan Louëdec, Max Chevalier, Josiane Mothe, Aurélien Garivier, Sébastien Gerchinovitz. *Algorithmes de bandit pour les systèmes de recommandation : le cas de multiples recommandations simultanées (regular paper)*. Conférence francophone en Recherche d'Information et Applications (CORIA 2015), Paris, 18/03/15-20/03/15, LIMSI, p. 73-88, mars 2015.
- *[Louëdec 2014] Jonathan Louëdec. *Algorithmes de bandits pour les systèmes de recommandation (student paper)*. INFORSID Forum Jeunes Chercheurs, Lyon, France, 20/05/14-23/05/14, Hermès, p. 17-20, mai 2014.

3.3.2 Communications dans des conférences et séminaires sans actes

- *Adaptation des stratégies de bandit pour la recommandation* Séminaire étudiant IMT, Université Paul Sabatier, Toulouse, 26 mai 2016.
- *Bandit et recommandation* Séminaire étudiant IMT, Université Paul Sabatier, Toulouse, 10 octobre 2013.

3.3.3 Publications en préparation

- Jonathan Louëdec, Max Chevalier, Aurélien Garivier, Josiane Mothe. *Diversified Bandit : Taking into Account the Combinatorial Aspect of the Multiple-Play Recommendation*.

Conclusion

Nous avons présenté dans ce chapitre la problématique principale des travaux de cette thèse : comment adapter les stratégies de bandit pour qu'elles puissent être utilisées en recommandation. En effet le modèle présenté dans le chapitre 2 parait peu réa-

liste en recommandation ; cela implique une adaptation des différentes stratégies pour aborder des problématiques plus spécifiques. Nous avons en particulier mis en évidence pourquoi ces stratégies représentent une alternative intéressante aux approches classiquement utilisées. Nous avons brièvement résumé chacune des contributions présentées dans ce manuscrit, et plus particulièrement celles qui sont abordées dans nos contributions :

- Tirages multiples : comment effectuer plusieurs recommandations à chaque instant ?
- Diversité : comment apporter de la diversité dans une liste de recommandation ?
- Non stationnarité : Comment prendre en compte la non stationnarité de la popularité de chaque objet ?

Ces contributions sont détaillées dans la suite de ce manuscrit.

Deuxième partie

Stratégies de Bandit pour la Recommandation : Obsolescence Progressive, Tirages Multiples et Diversité

Résumé

Dans cette seconde partie, nous nous focalisons sur l'adaptation des approches de bandit pour la recommandation. nous abordons trois problématiques importantes des systèmes de recommandation :

— Obsolescence progressive : Comment gérer l'arrivée de nouveaux objets et la décroissance de leur popularité avec le temps ?
— Tirages multiples : Comment recommander plusieurs objets à chaque instant ?
— Diversité : Comment apporter de la diversité dans les listes d'objets retournés ?

Dans le chapitre 4, nous présentons quelques travaux permettant de prendre en compte la non-stationnarité de la popularité des objets. Un formalisme sera fourni afin de pouvoir ensuite décrire un modèle où la popularité des objets décroit de manière exponentielle.

Dans le chapitre 5, nous proposons deux stratégies de bandit conçues pour gérer l'apparition régulière de nouveaux objets tout en étant capable de prendre en compte l'obsolescence progressive de la popularité de chaque objet : les stratégies *Fading-UCB* et *Trust and Abandon*. Nous fournissons une analyse du regret de la stratégie *Fading-UCB* afin de proposer une borne supérieure de regret. Nous montrons que les performances de ces stratégies permettent d'obtenir de meilleures performances que celles obtenues en utilisant les approches de l'état de l'art en simulant un système respectant notre modèle.

Dans le chapitre 6, nous présentons plus en détail les nouvelles problématiques et les enjeux liées à la recommandation à tirages multiples. Nous décrivons quelques approches qui sont utilisées en recommandation pour aborder ces problématiques. Pour cela nous faisons notamment le lien avec des approches de RI. Nous mettons en évidence par la suite que l'évaluation dans un cadre séquentiel est difficile, et listons les différents cadres proposés dans l'état de l'art pour évaluer de nouvelles approches en recommandation à tirages multiples.

Dans le chapitre 7, nous montrons comment les stratégies de bandit conçues pour recommander un seul objet à chaque instant sont adaptées dans les travaux de l'état de l'art pour être capables de recommander plusieurs objets à chaque instant. Nous mettons en évidence les limites de ces méthodes et proposons une nouvelle approche permettant de traiter ces limites. Cette approche est évaluée sur deux jeux de données populaires en recommandation : Jester et MovieLens.

Dans le chapitre 8, nous présentons des stratégies de bandit spécialement conçues pour recommander plusieurs objets à chaque instant issues de travaux récents de la littérature statistique. Nous proposons une adaptation de l'une de ces stratégies et l'éva-

luons dans un cadre de recommandation à tirages multiples.

Dans le chapitre 9, nous nous focalisons sur la problématique de la diversité. Nous montrons tout d'abord comment les approches de l'état de l'art tentent d'apporter de la diversité dans les listes de résultats. Nous décrivons ensuite deux approches que nous avons proposées pour apporter de la diversité en utilisant des probabilités conditionnelles. Nous comparons ces approches à celles de l'état de l'art et montrons qu'elles permettent d'apporter plus de diversité et obtiennent de meilleures vitesses d'apprentissage ainsi que des proportions d'abandon plus faibles sur le long terme.

Chapitre 4

Stratégies de bandit dans un cadre non stationnaire

Sommaire

Résumé.

Dans ce chapitre nous abordons la problématique spécifique de la non stationnarité du gain moyen des bras. Dans un premier temps nous présentons les approches de la littérature statistique traitant cette problématique, ces approches sont nommées *bandits non stationnaires*. Nous proposons ensuite un nouveau modèle dans lequel des hypothèses plus fortes sont faites sur la manière dont le gain moyen des bras évolue au cours du temps. Nous décrivons enfin un formalisme pour ce modèle afin de clairement présenter dans le chapitre suivant deux contributions.

En recommandation, de nombreux systèmes considèrent des objets dont la pertinence n'est pas constante avec le temps. C'est le cas notamment de la recommandation de films, de nouvelles ou encore de musiques. Il est important d'être capable de prendre en compte la non stationnarité de la popularité des objets. Dans la littérature statistique, plusieurs approches existent, mais ne font pas d'hypothèses fortes sur la manière dont la popularité des objets évolue.

4.1 Bandits non stationnaires

4.1.1 Justification

Dans la version stochastique des modèles de bandit, un agent choisit à chaque instant $t = 1, 2, \ldots$ un *bras* $A_t \in \{1, \ldots, K\}$ et reçoit une récompense X_t aléatoire dépendant de ce choix. Le cadre classique est stationnaire : les K bras sont disponibles du début à la fin et ont un gain moyen n'évoluant pas au cours du temps. Si ce modèle capture déjà l'essence du dilemme exploration/exploitation (il faut à la fois essayer les bras mal connus et tirer profit de ceux qui semblent le plus performants), il s'avère insuffisant dans de nombreuses applications. En particulier, de nombreux domaines voient leurs données se renouveler et vieillir au fil du temps. Il faudrait donc être capable de prendre en compte à la fois ce renouvellement et cette obsolescence progressive.

4.1.2 Stratégies prenant en compte la non-stationnarité

Les stratégies présentées dans le chapitre 2 ne sont pas conçues pour appréhender un modèle non stationnaire. Les stratégies de bandit généralement connues telles que le *Thompson Sampling* ([Thompson 1933]), les approches de type *UCB* ([Auer 2002a, Audibert 2009, Garivier 2011]) ou encore les approches *Softmax* ([Auer 2002b]) ne sont pas conçues pour appréhender un modèle non stationnaire.

Cependant plusieurs travaux de l'état de l'art considèrent des problèmes de bandit dans un cadre non stationnaire. Garivier et Moulines (2008) [Garivier 2008] présentent une analyse d'une version pondérée en fonction du temps de l'approche UCB. Dans ce modèle, les observations les plus récentes sont considérées comme plus importantes. Dans ce même article les auteurs analysent une stratégie UCB utilisant une fenêtre glissante de taille fixe dans laquelle uniquement les informations les plus récentes sont prises en compte dans le calcul de la borne supérieure de confiance : *sliding-window UCB (SW-UCB)* (détaillé dans l'algorithme 4.1). Ces approches sont conçues pour un modèle où aucune information sur la manière dont le gain moyen évolue dans

le temps n'est disponible. Chakrabarti et al. (2009) [Chakrabarti 2009] proposent un modèle de "bandit mortel" dans lequel un bras a une durée de vie prédéfinie au delà de laquelle le bras disparaît. Ce modèle s'est inspiré de ce que l'on peut observer lors de la recommandation de nouvelles (Yahoo ! actualités, Le Monde, ...). Slivkins et Upfal (2007) [Slivkins 2007] considèrent un cas où le gain moyen de chaque bras suit un mouvement brownien. Besbes et al. (2014) [Besbes 2014] proposent un modèle où le gain moyen des bras peut changer. Pour ces deux cas, l'exploration doit être suffisamment importante pour être capable d'identifier le possible changement de bras optimal. Tracà et Rubin [Tracà 2015] proposent une approche de type UCB prenant en compte les variations périodiques du nombre de visiteurs sur un système de recommandation. Cette approche suggère de réguler la part d'exploration selon l'affluence : plus il y a de visiteurs, plus il faut favoriser l'exploitation à l'exploration. L'exploration se fait donc quand peu de visiteurs sont en ligne. Dans l'article [Komiyama 2014], les auteurs proposent d'apprendre une fonction décroissante considérée comme une combinaison linéaire de fonctions connues. Cet article considère que la moyenne des récompenses d'un bras décline en fonction de son âge, comme nous le suggérons dans notre article.

Algorithme 4.1 : Sliding-Window UCB (SW-UCB)

A : vecteur contenant les bras disponibles
S : taille de la fenêtre
$N_t(S,a) = \sum_{s=t-S}^{t} \mathbb{1}_{A_s=a}$
$\hat{X}_t(S,a) = \frac{1}{N_t(S,a)_s} \mathbb{1}_{A_s=a}$
pour $t = 0, \ldots, KL$ **faire**
    si $t\%(L+1) = 0$ **alors**
        $A = A \cup a_r$
    fin
    $A_t = \text{argmax}_{a \in A}\, \hat{X}_t(S,a) + \sqrt{\frac{2\log(t \wedge S)}{N_t(S,a)}}$
    Jouer A_t
fin

Un autre moyen de prendre en compte l'évolution temporelle du gain moyen consiste à utiliser des approches contextuelles. Par exemple, la stratégie *LinUCB* proposée par Li et al. [Li 2010] est capable de prendre en compte des variables telles que l'âge du bras et l'heure. Dans notre modèle, nous disposons a priori d'informations sur la façon dont la popularité des bras décroit. Ainsi il est possible d'anticiper le décroissance de la popularité et d'adapter nos estimateurs à chaque instant. Cela nous permet de proposer des algorithmes plus performants. C'est ce que nous proposons dans la section

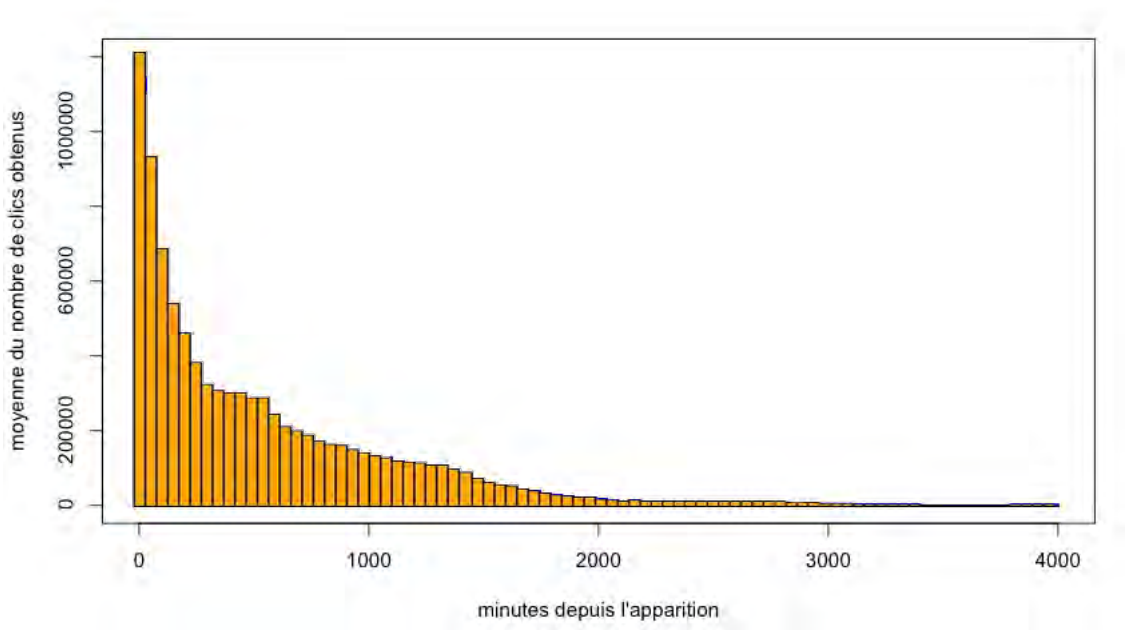

FIGURE 4.1 – Obsolescence de la popularité des bras. Données issues des 100 documents les plus cliqués du challenge CLEF-NEWSREEL

suivante.

4.2 Définition du modèle et formalisme

4.2.1 Modèle

Dans le modèle que nous proposons, nous considérons que de nouveaux bras apparaissent de manière régulière (flux de bras) et que la popularité de l'ensemble des bras décroit de manière exponentielle (obsolescence progressive). Cette décroissance peut être paramétrée selon la nature des bras disponibles. Ce type de décroissance est fondée sur des observations empiriques sur les données du challenge CLEF-NEWSREEL [1]. Ce challenge fournit un jeu de données de deux mois d'interactions entre un système de recommandation d'information et ses utilisateurs. Nous avons pu observer que la popularité d'une information à tendance à décroître de façon exponentielle. Il apparaît que l'information atteint généralement sa popularité optimale dès son apparition, pour décroître très rapidement avec le temps qui passe. Ce phénomène est visible sur

1. News Recommendation Evaluation Lab de la conférence CLEF, 2015 [Hopfgartner 2014]

la Figure 4.1, qui représente la moyenne du nombre de clics obtenus sur les 100 informations les plus cliquées de la collection en fonction de leur âge.

Considérer que la popularité des bras diminue selon un facteur exponentiel constant présente un fort avantage : l'ordre des bras ne change pas. Ainsi nous garantissons qu'un bras non-optimal à un instant donné ne le sera plus jamais, ou encore que si un bras obtient un gain moyen meilleur qu'un autre bras, il le sera toujours.

4.2.2 Formalisme

Nous considérons une succession de périodes de taille fixe L. Au début de la période r, commençant à l'instant $1 + rL$, le bras a_r entre en jeu et son instant d'apparition est $t_a = 1 + rL$. La période en cours à l'instant t est $r(t) = \lfloor \frac{t}{L} \rfloor$. L'expérience se termine après K périodes, la période finale est $r = K - 1$. À chaque instant t, l'agent choisit un bras A_t parmi ceux disponibles et reçoit une récompense $Z_t \in \{0, 1\}$. Lors de l'apparition des bras, les récompenses sont d'espérances p_a, où a est le bras associé.

Les espérances associées aux bras décroissent à la même vitesse, selon le facteur d'obsolescence progressive τ. Comme nous l'avons souligné plus tôt dans ce chapitre, il est important de préciser qu'un bras qui n'est pas optimal à l'instant t ne le sera jamais par la suite. Le bras optimal durant la période r est noté a_r^*. En outre, un bras a reste optimal un nombre r_a de périodes, pouvant être nul, et ne le sera plus jamais. Lors de son apparition, nous supposons qu'un bras a une espérance supérieure à une certaine valeur η fixée a priori. Cela induit qu'un bras apparu il y a plus de $\tau \log \frac{1}{\eta}$ instants ne peut pas être le bras optimal.

On note $\widetilde{N_a}(T)$ le nombre de fois où le bras a a été joué en étant sous-optimal jusqu'à l'instant T :

$$\widetilde{N_a}(T) = \sum_{t=t_a+r_aL}^{\left(t_a+\tau\log(\frac{1}{\eta})\right)\wedge T} \mathbb{1}_{\{I_t=a\}} \tag{4.1}$$

où r_a est le nombre de périodes pendant lesquelles a est optimal.
Une stratégie de bandit vise à minimiser la différence entre la somme des récompenses obtenues en utilisant le bras optimal a_r^* à chaque période et la somme des récompenses obtenues via la stratégie utilisée. Cette différence est appelée le regret cumulé moyen :

$$R(T) = \sum_{r=0}^{K-1} \sum_{t=1+rL}^{(r+1)L} \mathbb{E}[Z^{a_r^*}] e^{-\frac{rL-t_{a_r^*}}{\tau}} - \sum_{t=1}^{T} \mathbb{E}[Z_t] \tag{4.2}$$

Conclusion

Nous avons présenté dans ce chapitre comment l'évolution temporelle du gain moyen des bras est traité dans les travaux de l'état de l'art. Nous avons constaté que les approches découlant de ces travaux ne font pas d'hypothèses sur la façon dont le gain moyen des bras évolue. Nous avons ensuite défini un modèle, où le gain moyen des bras décroit de façon exponentielle. Cette hypothèse forte s'appuie sur l'étude d'un jeu de données issues d'un système de recommandation de nouvelles.

Dans le chapitre suivant nous proposons deux stratégies adaptées au modèle présenté dans ce chapitre.

Chapitre 5

Stratégies de bandit et obsolescence progressive

Sommaire

Résumé.

En prenant en compte le fait que le gain moyen des bras décroit de façon exponentielle et que de nouveaux bras entrent régulièrement, les stratégies proposées dans ce chapitre permettent d'obtenir des valeurs de regret cumulé jusqu'à quatre fois plus faibles que celles obtenues avec les stratégies de l'état de l'art. Nous proposons également une étude théorique du regret cumulé pour l'une de ces stratégies, nous permettant de fournir une borne supérieure de regret.

Le modèle proposé dans le chapitre précédent fait l'hypothèse que le gain moyen des bras décroit de façon exponentielle. Cette hypothèse forte permet de proposer des approches spécifiques à ce type de décroissance pouvant obtenir de bien meilleures performances que les approches de l'état de l'art ; ces dernières ne prenant pas en compte d'informations a priori sur la manière dont le gain moyen évolue avec le temps.

Dans ce chapitre nous proposons deux stratégies de bandit pour ce modèle : *Fading-UCB* et *Trust and Abandon*. Une borne supérieure de regret est fournie pour l'algorithme *Fading-UCB*. Nous évaluons ensuite les performances de ces stratégies en terme de regret cumulé à l'aide de simulations.

5.1 Fading-UCB (F-UCB)

5.1.1 Présentation de la stratégie

L'objectif de cette stratégie que nous proposons dans l'article [Louëdec 2016] est d'adapter la stratégie *Upper Confidence Bound* [Auer 2002b] pour la prise en compte de l'obsolescence des bras et d'un flux constant de bras. Cette stratégie utilise non pas l'espérance estimée de chaque bras, mais des bornes de confiance supérieures de cette espérance. C'est une stratégie dite "optimiste dans l'incertain" : c'est à dire qu'elle consiste à jouer le bras ayant potentiellement la plus forte récompense.

L'espérance d'un bras lors de son apparition p_a est estimée par $\hat{p}_a(t)$:

$$\hat{p}_a(t) = \frac{1}{N_a(t)} \sum_{s=1}^{t} \left(Z_s \exp\left(\frac{s - t_a}{\tau}\right) \mathbb{1}_{A_s=a} \right) \tag{5.1}$$

avec $N_a(t)$ le nombre de fois où le bras a a été joué aux t premiers instants et τ le facteur d'obsolescence, défini a priori.

Pour la suite de l'analyse, nous projetons l'ensemble des estimateurs à l'instant t en cours. Pour cela, nous proposons d'utiliser non pas l'espérance du bras lors de son apparition p_a mais l'espérance à l'instant t $\mu_a(t)$, estimée par :

$$\hat{\mu}_a(t) = \hat{p}_a(t) \exp\left(-\frac{t - t_a}{\tau}\right) \tag{5.2}$$

5.1.2 Calibration de l'intervalle de confiance

Notre objectif est de trouver un intervalle de risque α pour l'espérance en début de période $\mu_a(t)$, variable comprise entre 0 et $M = e^{-\frac{t-t_a}{\tau}}$. En appliquant l'inégalité d'Hoeffding, décrite en annexe, on obtient la valeur :

$$M\sqrt{\frac{2}{n}\log\frac{1}{\alpha}} \tag{5.3}$$

Classiquement la valeur choisie pour α est $1/t$. Ainsi au fur et à mesure des interactions, la borne croît lentement si le bras associé n'est pas joué. Le bras sera rejoué à un moment ou un autre sur le long terme. Mais ici, un bras qui est apparu il y a plus de $\tau \log \frac{1}{\eta}$ instants ne peut pas être le bras optimal, il est donc inutile de faire grandir α au delà de cette valeur, cela revient à fixer $\frac{1}{\alpha} = \tau \log \frac{1}{\eta}$.

Au final, en majorant M par 1, on obtient la borne de confiance supérieure $U_a(t)$ suivante :

$$U_a(t) = \hat{\mu}_a(t) + \sqrt{\frac{2\log\left(\tau \log \frac{1}{\eta}\right)}{N_a(t)}} \tag{5.4}$$

5.1.3 Algorithme

La stratégie $F-UCB$ consiste à jouer le bras avec la borne de confiance supérieure $U_a(t)$ la plus grande. Elle est décrite dans l'algorithme 5.1.

Algorithme 5.1 : Fading-UCB (F-UCB)

A : vecteur contenant les bras disponibles
pour $r = 0, \ldots, K-1$ **faire**
  $A = A \cup a_r$
  pour $t = 1 + rL, \ldots, (r+1)L$ **faire**
    $$U_a(t) = \hat{\mu}_a(t) + \sqrt{\frac{2\log\left(\tau \log \frac{1}{\eta}\right)}{N_a(t)}}$$
    Jouer $A_t = \text{argmax}_{a \in A}\, U_a(t)$
  fin
fin

5.1.4 Étude du regret

5.1.4.1 Performance minimale : borne supérieure de regret

L'espérance d'un bras a à un instant $t \geq t_a$ est notée :

$$\mu_a(t) = p_a \exp^{-\frac{t-t_a}{\tau}} \tag{5.5}$$

et l'espérance d'un bras optimal a_t^* au même instant est :

$$\mu^*(t) = \mu_{a_t^*}(t) = \max_a \mu_a(t) \tag{5.6}$$

Pour un bras a sous-optimal lors de la dernière période, l'écart minimal Δ_a entre l'espérance du bras optimal lors de la période en cours et l'espérance du second bras optimal est défini par :

$$\Delta_a = \min_{t_a + r_a L \leq t \leq t_a + \tau \log(1/\eta)} \mu^*(t) - \mu_a(t) \tag{5.7}$$

Cet écart minimal est utilisé dans le théorème 3 qui donne une borne pour le regret cumulé moyen :

Théorème 3.

Le regret cumulé moyen $R(T)$ de la stratégie F-UCB vérifie :

$$R(T) \leq \left(8 \log \left(\tau \log \frac{1}{\eta} \right) \sum_a \frac{1}{\Delta_a^2} + (K-1) \right.$$

$$\left. \left(1 + \frac{2}{\tau \log \frac{1}{\eta}} \right) \right) \wedge \left((K-1)\, \tau \log \frac{1}{\eta} \right)$$

en sommant sur les bras a sous-optimaux.

5.1.4.2 Preuve

Les éléments principaux de la preuve du théorème 3 sont donnés ci-dessous. Un bras optimal lors de la dernière période est optimal sur toute sa durée de vie ; on s'intéresse à un bras a sous-optimal à un instant donné donc forcément sous-optimal lors de la dernière période. Soit $N_a(T)$ le nombre de fois que le bras a a été joué jusqu'à l'instant T et $\widetilde{N}_a(T)$ le nombre de fois que le bras a a été joué en étant sous-

optimal jusqu'à l'instant T, l'inégalité $\widetilde{N}_a(T) \leq N_a(T)$ est immédiate. L'obsolescence des bras assure également $N_a(T) \leq \tau \log(1/\eta)$. Le bras a sera sous-optimal entre les instants $t_1 = t_a + r_a L$ et $t_2 = (t_a + \tau \log(1/\eta)) \wedge T$. Par conséquent, en posant $t_1 = t_a + (r_a L) \vee 1$,

$$\begin{aligned}
\widetilde{N}_a(T) &= \sum_{t=t_a+r_aL}^{t_2} \mathbb{1}_{\{I_t=a\}} && (5.8)\\
&\leq 1 + \sum_{t=t_1}^{t_2} \mathbb{1}_{\{I_t=a\}} && (5.9)\\
&\leq l + \sum_{t=t_1}^{t_2} \mathbb{1}_{\left\{I_t=a, \widetilde{N}_a(t-1)\geq l\right\}} && (5.10)\\
&\leq l + \sum_{t=t_1}^{t_2} \mathbb{1}_{\left\{U_{a^*}(t-1)\leq U_a(t-1), \widetilde{N}_a(t-1)\geq l\right\}} && (5.11)\\
&\leq l + \sum_{t=t_1}^{t_2} \mathbb{1}_{\{U_{a^*}(t-1)\leq U_a(t-1), N_a(t-1)\geq l\}} && (5.12)
\end{aligned}$$

en notant $a^*(t)$ le bras optimal à l'instant t, $N^*(t) = N_{a^*(t)}(t)$ et $U^*(t) = U_{a^*(t)}(t)$. La moyenne empirique $\hat{\mu}_a(t)$ fait intervenir les t valeurs de la suite :

$$\left(Z_s \mathbb{1}_{A_s=a} \exp^{\frac{s}{\tau}}\right)_{1\leq s\leq t} \tag{5.13}$$

Cette suite est constituée des $N_a(t)$ valeurs $Z_s \exp^{\frac{s}{\tau}}$ lorsque le bras a est tiré et de $t - N_a(t)$ valeurs nulles lorsque le bras a n'est pas tiré. Soit $(X_{a,i})_{1\leq i\leq N_a(t)}$ la suite constituée des valeurs (comprises entre 0 et 1) obtenues lorsque le bras a est tiré. En utilisant cette suite, $\hat{\mu}_a(t)$ s'écrit

$$\hat{\mu}_a(t) = \frac{1}{N_a(t)} \sum_{i=1}^{N_a(t)} X_{a,i} \exp^{-\frac{t}{\tau}} \tag{5.14}$$

et on introduit l'expression :

$$\hat{\mu}_{a,s}(t) = \frac{1}{s} \sum_{i=1}^{s} X_{a,i} \exp^{-\frac{t}{\tau}} \tag{5.15}$$

Cette notation et la définition de la borne supérieure de confiance permet d'écrire les inclusions suivantes :

$$\left\{U_{a^*(t-1)}(t-1) \leq U_a(t-1), N_a(t-1) \geq l\right\} \subset \tag{5.16}$$

$$\left\{\min_{1\leq s^*\leq N^*(t-1)} \hat{\mu}_{a^*_{t-1},s^*}(t-1) + \sqrt{\frac{2m}{s^*}} \leq \right. \tag{5.17}$$

$$\left. \max_{l\leq s\leq N_a(t-1)} \hat{\mu}_{a,s}(t-1) + \sqrt{\frac{2m}{s}} \right\} \subset \tag{5.18}$$

$$\bigcup_{s^*=1}^{N^*(t-1)} \bigcup_{s=l}^{N_a(t-1)} A_{s,s^*}(t-1) \tag{5.19}$$

en notant $m = \log\left(\eta \log \frac{1}{\tau}\right)$ et $A_{s,s^*}(t)$ l'événement :

$$\left\{\hat{\mu}_{a,s^*}(t) + \sqrt{\frac{2m}{s^*}} \leq \hat{\mu}_{a,s}(t) + \sqrt{\frac{2m}{s}}\right\}. \tag{5.20}$$

Observer l'événement $A_{s,s^*}(t-1)$ implique au moins l'un des 3 cas suivants :

$$A^1_s(t-1) = \left\{\hat{\mu}_{a,s}(t-1) - \sqrt{\frac{2m}{s}} \geq \mu_a(t-1)\right\} \tag{5.21}$$

$$A^2_{s^*}(t-1) = \left\{\hat{\mu}_{a^*_{t-1},s^*}(t-1) + \sqrt{\frac{2m}{s^*}} \leq \mu^*(t-1)\right\} \tag{5.22}$$

$$A^3_s(t-1) = \left\{\mu^*(t-1) \leq \mu_a(t-1) + 2\sqrt{\frac{2m}{s}}\right\} \tag{5.23}$$

La probabilité de $A^1_s(t-1)$ peut s'écrire :

$$\mathbb{P}\left(\sum_{i=1}^{s^*} X_{a^*_{t-1},i} \exp^{-\frac{t-1}{\tau}} \leq s^* p_{a^*_{t-1}} e^{-\frac{t-1-T_a}{\tau}} - \sqrt{2s^*m}\right) \tag{5.24}$$

et l'inégalité de Hoeffding, dont un énoncé est rappelé en fin de preuve, permet de majorer cette probabilité par :

$$\exp\left(-2\frac{\left(\sqrt{2s^*m}\right)^2}{s^*}\right) = \left(\tau\log\frac{1}{\eta}\right)^{-4} \tag{5.25}$$

De manière analogue, $\mathbb{P}\left(A^2_{s^*}(t-1)\right)$ se majore par la même expression. Lorsque s devient supérieur à la valeur

$$\left\lceil \frac{8\log\left(\eta\log\frac{1}{\tau}\right)}{\Delta_a^2} \right\rceil \tag{5.26}$$

la probabilité $\mathbb{P}\left(A^3_s(t-1)\right)$ devient nulle. Finalement, en choisissant comme valeur de l l'expression ci-dessus, $\mathbb{E}\left(\widetilde{N}_a(T)\right)$ est majoré par :

$$\left\lceil \frac{8\log\left(\tau\log\frac{1}{\eta}\right)}{\Delta_a^2} \right\rceil + \sum_{t=t_1}^{t_2}\sum_{s^*=1}^{N^*(t-1)}\sum_{s=1}^{N_a(t-1)} 2\left(\tau\log\frac{1}{\eta}\right)^{-4} \tag{5.27}$$

$$\leq \frac{8\log\left(\tau\log\frac{1}{\eta}\right)}{\Delta_a^2} + 1 + 2\left(\tau\log\frac{1}{\eta}\right)^{3}\left(\tau\log\frac{1}{\eta}\right)^{-4} \tag{5.28}$$

en utilisant que $t_2 - t_1 + 1$, $N_a(t)$ et $N^*(t)$ sont majorés par $\tau\log\frac{1}{\eta}$.

Finalement,

$$\mathbb{E}\left(\widetilde{N}_a(T)\right) \leq \frac{8\log\left(\tau\log\frac{1}{\eta}\right)}{\Delta_a^2} + 1 + \frac{2}{\tau\log\frac{1}{\eta}} \tag{5.29}$$

ce qui permet d'obtenir une borne pour le regret $R(T)$ en sommant sur les bras a sous-optimaux :

$$\begin{aligned} R(T) &\leq \sum_a \mathbb{E}\left(\widetilde{N}_a(T)\right) && (5.30)\\ &= \sum_a \left(\frac{8\log\left(\tau\log\frac{1}{\eta}\right)}{\Delta_a^2} + 1 + \frac{2}{\tau\log\frac{1}{\eta}}\right) && (5.31)\\ &\leq 8\log\left(\tau\log\frac{1}{\eta}\right)\sum_a \frac{1}{\Delta_a^2} && (5.32)\\ &\quad + (K-1)\left(1 + \frac{2}{\tau\log\frac{1}{\eta}}\right) && (5.33) \end{aligned}$$

où K est le nombre total de périodes.

□

Inégalité d'Hoeffding : Soit (X_i) avec $1 \leq i \leq n$ une suite de variables aléatoires indépendantes de même espérance p avec $X_i \in [a_i, b_i]$. Pour tout $t > 0$,

$$\mathbb{P}(|\bar{X} - p| \geq t) \leq 2 \exp \frac{-2n^2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2} \tag{5.34}$$

5.2 La stratégie Trust and Abandon

5.2.1 Motivation

Nous allons maintenant définir une politique où l'on fait l'hypothèse qu'au début d'une période r, les bras a_i avec $i \in 1, \ldots, r-1$ ont été joués un nombre de fois suffisant pour avoir des estimations assez précises de leur popularité. Les espérances μ_a de ces mêmes bras au début de la période r sont comprises entre $\eta \exp\left(\frac{rL-t_a}{\tau}\right)$ et $\exp\left(\frac{rL-t_a}{\tau}\right)$. Le bras entrant a_r a besoin d'être testé un certain nombre de fois pour pouvoir estimer son espérance, située entre η et 1 durant la période r. Ce bras est le seul à pouvoir atteindre une espérance très proche de 1. L'approche *TA* que nous proposons n'est pas sans rappeler la stratégie *Successive Elimination* [Even-Dar 2006].

Tout comme la stratégie *F-UCB*, nous projetons l'ensemble des estimateurs à l'instant de départ de cette période. Pour cela nous utilisons les estimateurs $\hat{\mu}_a(t)$ et les bornes supérieures de confiance $U_a(t)$ définies pour la stratégie *F-UCB*.

5.2.2 Algorithme

La stratégie *Trust and Abandon* consiste à jouer durant la période r le bras a_r tant qu'il est impossible de certifier que son espérance est plus faible que celle d'un autre bras. Afin de certifier cela, nous utilisons la borne de confiance supérieure $U_a(t)$ de $\mu_a(t)$. Il faut ensuite choisir le moment où $U_{a_r}(t)$ devient trop faible pour certifier qu'un autre bras obtient en moyenne un meilleur gain moyen. Plusieurs choix sont possibles.

Dans une première version, *TA-μ* (algorithme 5.2), nous cherchons l'instant où cette borne devient inférieure à $\max_{a \in A} \hat{\mu}_a(t)$, l'espérance estimée la plus grande au début de la période en cours. Le bras a_r n'est alors plus joué, au profit du bras $\operatorname{argmax}_{a \in A} \hat{\mu}_a(t)$.

Dans une seconde version, *TA-UCB* (algorithme 5.3), nous cherchons l'instant où cette borne devient inférieure à $\max_a U_a(t)$, la plus grande borne de confiance au début

de la période en cours. Le bras a_r n'est alors plus joué, au profit du bras $\mathrm{argmax}_a \hat{\mu}_a(t)$. Si cette deuxième version ressemble à l'approche *F-UCB*, la principale différence réside dans le choix du bras joué une fois le bras a_r éliminé : *TA-UCB* joue le bras avec l'espérance la plus forte, tandis que *F-UCB* joue le bras avec la borne de confiance supérieure du gain associé la plus forte.

Algorithme 5.2 : Trust and Abandon μ (TA-μ)

```
A : vecteur contenant les bras disponibles
pour r = 0, ..., K − 1 faire
    A = A ∪ a_r
    pour t = 1 + rL, ..., (r + 1)L faire
        si max_{a∈A} μ̂_a(t) > U_{a_r}(t) alors
            Jouer A_t = argmax_{a∈A} μ̂_a(t)
        sinon
            Jouer A_t = a_r
        fin
    fin
fin
```

Algorithme 5.3 : Trust and Abandon UCB (TA-UCB)

```
A : vecteur contenant les bras disponibles
pour r = 0, ..., K − 1 faire
    A = A ∪ a_r
    pour t = 1 + rL, ..., (r + 1)L faire
        si max_{a∈A} U_a(t) > U_{a_r}(t) alors
            Jouer A_t = argmax_{a∈A} μ̂_a(t)
        sinon
            Jouer A_t = a_r
        fin
    fin
fin
```

5.3 Comparaison avec les approches de l'état de l'art : simulations

5.3.1 Cadre expérimental

Pour évaluer les performances de nos approches et les comparer à celles de l'état de l'art, nous avons mis en place des simulations. Lors de ces simulations, nous varions le nombre d'itérations contenu dans une période L et le paramètre d'obsolescence τ afin d'évaluer leurs effets sur le regret cumulé. Chaque expérimentation est effectuée sur un intervalle de temps $T = 10\,000$. Le nombre total de bras k ainsi que le nombre de périodes R sont égaux à $\lfloor \frac{T}{L} \rfloor$. Les résultats présentés sont la moyenne de 200 répétitions de chaque expérimentation. Les performances de notre second algorithme *TA-UCB* sont toujours meilleurs que celles de notre second algorithme *TA-*μ. Pour alléger les graphiques nous ne présenterons que les résultats de l'algorithme *TA-UCB*.

Les approches *F-UCB* et *TA-UCB* de ce chapitre sont comparées à deux approches de l'état de l'art : *UCB1* (algorithme 2.4) et *SW-UCB* (algorithme 4.1). La première approche ne prend pas en compte l'obsolescence du gain moyen des bras. Ce gain va diminuer avec le temps car les récompenses diminuent, cela implique que les bras avec un gain moyen faible seront rejoués régulièrement. L'approche *SW-UCB* utilise une fenêtre glissante de taille fixe S, où uniquement les S dernières interactions sont prises en compte dans le calcul de la borne supérieure de confiance. Selon la taille de la fenêtre les résultats varient. Une fenêtre de petite taille implique qu'il faut régulièrement rejouer les bras, car les interactions passées avec ces bras ne sont plus prises en compte. Une fenêtre de très grande taille revient à ne pas prendre en compte l'obsolescence progressive des bras. Nous avons donc testé plusieurs valeurs possibles pour la taille de la fenêtre $S = xL$ avec $x \in [0.5, 1, 1.5, 2, 2.5, 3]$.

Les bornes supérieures de confiance de l'ensemble des algorithmes utilisés dans cette expérimentation sont calculées en utilisant la constante 0.5 au lieu de 2 à l'intérieur de la racine. Si cette valeur habituelle 2 permet d'analyser l'algorithme afin d'obtenir des bornes de regret supérieures, elle donne une part trop forte à l'exploration.

Lors de nos expérimentations, plusieurs valeurs de $\tau = xL$ ont été testées avec $x \in [3, 4, 5, 6, 7, 8, 9, 10]$ et $L \in [500, 1000, 2000, 5000]$. La figure 5.1 est obtenue en utilisant $L = 2000$ et $\tau = 5L$ et la figure 5.2 en utilisant $L = 2000$ et $\tau = 10L$. Ce choix permet de mettre en évidence l'impact de la vitesse d'obsolescence sur les résultats. Il est important de rappeler que le choix des valeurs de τ et η induisent la durée maximum de vie d'un bras. Par exemple si $\eta = 0.1$ et $\tau = 5L$, un bras ne peut plus

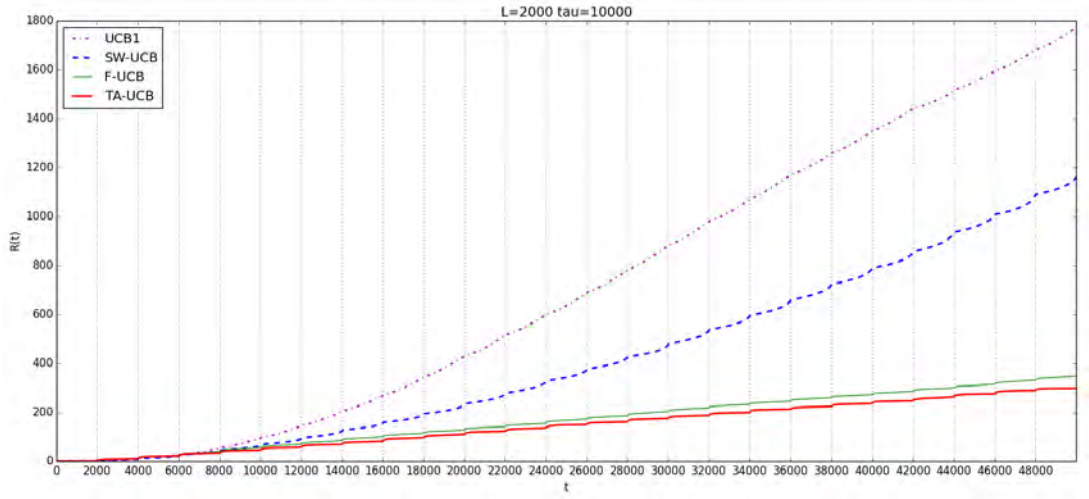

FIGURE 5.1 – Simulation avec $L = 2000$ et $\tau = 5L$. Valeurs du regret cumulé en fonction du nombre d'interactions avec le système

être optimal à partir de $\tau \log \frac{1}{\eta} = 5L$, ce qui signifie qu'un bras peut être optimal au maximum 5 périodes après son apparition, et ensuite mis de côté. Les expérimentations sont réalisées avec $\eta = 0.1$, ainsi un bras peut être optimal au maximum τ périodes. Concernant l'optimisation de la taille S de la fenêtre pour l'algorithme *SW-UCB*. Expérimentalement la valeur qui permet d'obtenir les meilleurs résultats sont obtenus avec $L = 2000$ est $S = 2L$.

5.3.2 Résultats

Sur l'ensemble des cas expérimentés selon le cadre précédemment défini, les différents regrets cumulés $R^{\alpha}(t)$ peuvent être classés comme suit :

$$R^{UCB}(t) > R^{SW-UCB}(t) > R^{F-UCB}(t) \geq R^{TA-UCB}(t) \tag{5.35}$$

L'approche *UCB1* est celle qui obtient les performances les plus faibles en terme de regret cumulé. En ne prenant pas en compte l'obsolescence progressive des bras, cette approche rejoue régulièrement les bras avec des gains cumulés faibles, car le gain moyen du bras optimal diminue et cette décroissance n'est pas répercutée sur l'ensemble des bornes. En utilisant une fenêtre glissante, l'approche *SW-UCB* permet d'obtenir de meilleurs résultats que l'approche *UCB1*, mais ces résultats restent bien plus faibles que ceux obtenus par les approches présentées dans ce chapitre.

L'approche *TA-UCB* est celle qui obtient le regret cumulé le plus faible sur les deux

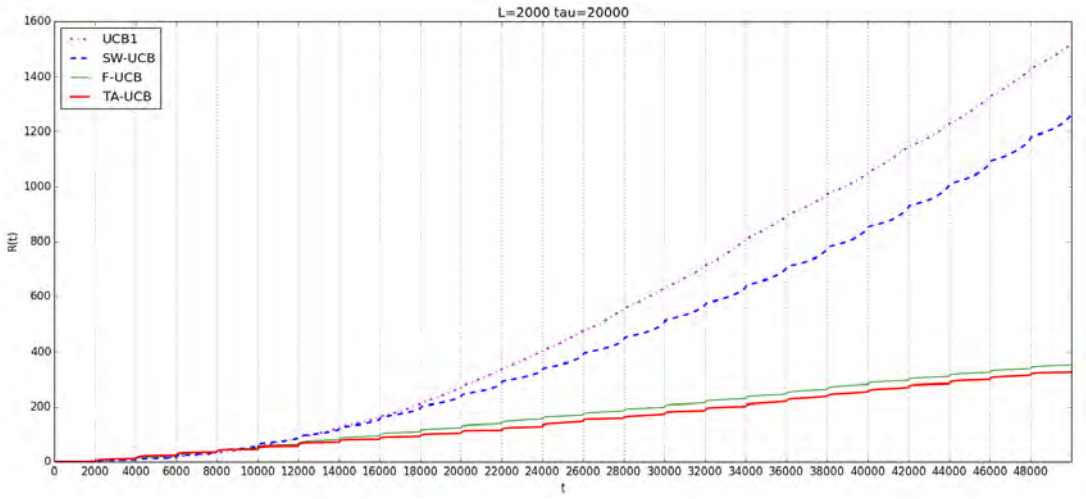

FIGURE 5.2 – Simulation avec $L = 2000$ et $\tau = 10L$. Valeurs du regret cumulé en fonction du nombre d'interactions avec le système

figures 5.1 et 5.2. L'approche *F-UCB* obtient des performances très proches de l'approche *TA-UCB*. Lorsque le nouveau bras entrant n'est pas le meilleur, l'utilisation de la borne de confiance calculée par *F-UCB* fait qu'il est assez long d'éliminer les autres bras s'ils n'ont pas été suffisamment joués par le passé, alors que *TA-UCB* va privilégier le bras qui a la plus forte espérance estimée directement. Pour de grandes valeurs de $L \geq 10000$, il se trouve que *TA-UCB* fait aussi bien que *F-UCB* car leur comportement est très similaire lorsque les bras sont joués un grand nombre de fois avant l'insertion d'un nouveau bras.

Conclusion

Dans ce chapitre nous avons proposé un modèle dans lequel de nouveaux bras apparaissent régulièrement et où le gain moyen associé à chaque bras décroît de manière exponentielle. Ce modèle est inspiré de plusieurs observations empiriques sur un jeu de données du challenge NEWS-REEL de la conférence CLEF, où l'obsolescence des bras est visible. Une première approche *F-UCB* est proposée. Elle représente une adaptation pour ce modèle de la stratégie UCB, une analyse nous a permis de quantifier une borne supérieure de regret pour cette stratégie. La seconde approche proposée, *Trust and Abandon*, repose sur une hypothèse stricte : lors de l'apparition d'un bras, nous estimons que l'ensemble des autres bras ont été suffisamment joués pour avoir une estimation précise des gains moyens associés. Nous en avons proposé deux variantes.

Plusieurs simulations nous permettent d'observer que ces deux approches, *F-UCB* et *Trust and Abandon*, obtiennent de bien meilleures performances en terme de regret cumulé que l'approche de l'état de l'art *Sliding-Window UCB*, conçue pour prendre en compte la non-stationnarité du gain moyen de chaque bras. L'ensemble de ces approches surpassent largement la stratégie *UCB1* qui ne prend pas du tout en compte l'obsolescence progressive des bras.

Le chapitre suivant présente la recommandation à tirages multiples. Plus précisément il décrit en détail l'une des problématiques qui en découle : la diversité, et présente la manière dont l'évaluation est généralement mise en œuvre en recommandation à tirages multiples.

Chapitre 6

Recommandation à tirages multiples : présentation et problématiques

Sommaire

Résumé.

Lorsque plusieurs objets doivent être recommandés à chaque instant, il est important de réfléchir à quel objectif doit être atteint. Si la maximisation du taux de clics est souhaitée, recommander les objets les plus populaires constitue une solution efficace. Nos travaux visent plutôt à minimiser la proportion d'abandon. Dans ce cadre une réflexion plus approfondie lors de la constitution des listes de résultats est nécessaire. Une problématique doit principalement être prise en compte : l'apport de diversité.

Dans ce chapitre, nous commençons par montrer que le passage de la recommandation à tirages simples à la recommandation à tirages multiples apportent deux nouvelles problématiques importantes : comment ordonner les recommandations au sein de la liste recommandée ? et comment apporter de la diversité au sein de cette liste ? Nous décrivons ensuite l'impact de ces deux nouvelles problématiques en recommandation. Enfin nous présentons comment il est possible d'évaluer des approches séquentielles à tirages multiples.

6.1 Ordonnancement et diversité

6.1.1 Problématiques

Les SR actuels sont utilisés dans de nombreuses applications, particulièrement en ligne : il peut s'agir de recommander des objets marchants comme des films, articles, musiques ou des contenus comme des pages web, des traitements médicaux, ... De nombreux SR recommandent à chaque instant plusieurs objets à un utilisateur simultanément ; ils sont qualifiés de SR à tirages multiples.

Deux nouvelles problématiques doivent alors être traitées : l'ordonnancement et la diversité.

La problématique d'*ordonnancement* est très populaire en recherche d'information, mais au contraire très peu utilisée en recommandation pour plusieurs raisons : tout d'abord parce que les listes de recommandations sont de tailles suffisamment faibles pour que l'utilisateur puisse les parcourir, mais aussi parce que les recommandations ne sont pas forcement présentées sous la forme de listes, mais distribuées sur une page.

La problématique de *diversité* est au contraire populaire en recommandation, tout comme en recherche d'information [Agrawal 2009, Santos 2010]. Lorsque plusieurs objets sont recommandés à chaque instant, il est important d'avoir une réflexion sur l'objectif du système. Si le système souhaite simplement maximiser le taux de clics global, une solution consiste à recommander les objets les plus populaires. Cela reste risqué, car il est possible que ces objets soient tous similaires et provoquent chez les utilisateurs n'appréciant pas ce type d'objets un arrêt de l'utilisation du système. L'apport de diversité dans la liste des recommandations peut permettre de satisfaire plus d'utilisateurs. Dans la sous-section suivante, nous décrivons plus en détails cette problématique.

6.1.2 Recommandation à tirages multiples et diversité

Les approches de recommandation visent habituellement à recommander des objets dont les pertinences estimées pour un utilisateur sont les plus fortes. La notion de diversité est généralement mise de côté. Il peut être intéressant d'ajouter de la diversité aux recommandations : cela rend possible la découverte de l'intérêt d'un utilisateur pour un domaine inconnu jusqu'ici, augmentant ainsi les perspectives de clics, ventes, ... C'est également une solution idéale pour aborder la problématique de départ à froid côté utilisateur : un nouvel utilisateur apparaît, lui recommander une liste d'objets d'horizons différents augmente la probabilité qu'il trouve au moins un objet pertinent dans la liste et permet de guider les recommandations futures [Ziegler 2005].

En recherche d'information, elle est notamment abordée pour gérer l'ambiguïté de certains termes d'une requête. Par exemple, dans l'article [Clarke 2008], les auteurs proposent de fournir à l'utilisateur un ensemble d'objets qui correspondent aux différents sens du terme. Pour cela les objets dont le contenu est trop similaire à un objet déjà sélectionné est pénalisé. Une approche non supervisée de classification regroupant les objets selon le sens d'un terme ambigu a également été proposée [Chifu 2012].

La diversité est généralement abordée selon deux angles : la diversité individuelle et la diversité agrégée [Foulonneau 2013].

6.1.2.1 La diversité individuelle

Elle représente la diversité des objets recommandés à un même utilisateur. Son objectif est de proposer à l'utilisateur des objets distincts afin d'éviter la redondance. Ce type de diversité est celui auquel nous faisons référence dans les chapitres suivants de ce manuscrit.

En recommandation, l'objectif n'est pas de désambiguïser un terme, car généralement chaque système est associé à un domaine connu. L'objectif est plutôt de minimiser la proportion d'abandon, c'est à dire le nombre de fois où un utilisateur ne sélectionne aucun des objets qui lui ont été recommandés. L'abandon a généralement un impact négatif sur l'utilisateur et peut entraîner le retrait de celui-ci [O'Brien 2008]. Pour un système de recommandation commercial, ne recommander que des objets pertinents mais trop similaires entre eux à un utilisateur n'est pas forcement utile, en effet un trop grande similarité entre les objets recommandés a un impact négatif sur la satisfaction utilisateur [Brynjolfsson 2003]. A l'inverse, la diversité des objets recommandés conforte l'utilisateur dans son choix et a donc un impact positif sur la décision d'achat [Castagnos 2010]. Si un utilisateur aime les films d'un réalisateur en particulier, si l'ensemble des recommandations ne contient que des films de ce réalisateur, l'utilisateur

ne sera sûrement pas pleinement satisfait.

La diversité individuelle est liée à d'autres problématiques connues en recommandation [Adamopoulos 2011] :

- **La nouveauté** : un nouveau type d'objets apparaît, recommander un objet de ce type peut être susceptible de plaire à l'utilisateur, car il s'agit d'un type d'objets qu'il ne connaît pas.
- **La sérendipité** : un objet pertinent mais inattendu est recommandé, créant ainsi la surprise chez l'utilisateur.

Afin d'accroître la diversité individuelle, plusieurs approches ont été proposées dans la littérature. Dans l'article [Zhang 2009], les auteurs proposent d'ajouter des objets choisis de manière aléatoire dans la liste des recommandations. *Ziegler et al.* tentent plutôt d'identifier les différents intérêts de l'utilisateur pour composer une liste d'objets correspondante [Ziegler 2005]. Les approches permettant d'aborder les problématiques de nouveauté et de sérendipité apportent généralement de la diversité dans la liste de résultats. *Onuma et al.* proposent d'utiliser les zones inexplorées du graphe des relations entre les objets [Onuma 2009], tout en conservant un niveau élevé de pertinence [Zhou 2010].

6.1.2.2 La diversité agrégée

Elle représente la capacité d'un SR à proposer l'ensemble des objets disponibles tout en évitant de recommander les mêmes objets à tous les utilisateurs. Cela implique de prendre en compte d'une part la diversité de l'ensemble des objets, mais également d'évaluer la diversité des objets recommandés, car le système ne recommande pas forcement l'ensemble des objets à sa disposition. Ce type de diversité est très importante pour les systèmes de recommandation devant gérer des stocks, car il peut être plus intéressant de recommander des objets moins pertinents pour l'utilisateur mais dont la disponibilité est avérée, plutôt que des objets pertinents et populaires, mais en rupture de stock [Konstan 2012].

Si un système de recommandation peut posséder énormément d'objets, un nombre limité d'entre eux sont réellement utilisés (la tête) alors que la plupart des objets ne le sont que très peu (la longue traîne). Ce phénomène est généralement appelé *économie de la longue traîne* [Anderson 2006]. Une minorité des objets représente une majorité des récompenses positives. Ce phénomène est généralement modélisé selon le principe de Pareto, qui dit que 20% des objets représentent 80% des récompenses positives. Avec l'avènement du e-commerce, plusieurs travaux suggèrent que l'exploitation, même exceptionnel, des objets de la longue traîne permet d'obtenir plus de

récompenses positives qu'en exploitant uniquement les objets de la tête [Tucker 2007, Adomavicius 2011].

Pour évaluer les objets de la longue traîne, *Park et al.* [Park 2008] proposent de regrouper ces objets dans des classes et de les évaluer ensemble plutôt qu'individuellement. Dans l'article [Adomavicius 2011], les auteurs utilisent un graphe des objets, et suggèrent de recommander les objets appartenant à des zones inexplorées, mais proches du graphe des ressources explorées. Il s'agit ici d'une approche inspirée du filtrage basé sur le contenu. D'autres approches plus classiques sont également proposées. Il s'agit de réorganiser les listes recommandées [Adomavicius 2012], ajouter aléatoirement des éléments de la longue traîne [Lemire 2008] ou encore combiner plusieurs stratégies de recommandation [Burke 2007].

6.2 Comment évaluer des approches séquentielles à tirages multiples ?

La première question que l'on peut se poser est "comment évaluer les approches séquentielles ?", dans lesquelles les interactions entre le SR et les utilisateurs ont lieu les unes après les autres. Le contexte idéal d'évaluation est d'avoir accès à un système de recommandation où des utilisateurs se connectent régulièrement. Ce n'est généralement pas le cas en recherche académique. Cependant il existe le challenge CLEF-NEWSREEL [1]. Il met à disposition un SR en ligne et deux mois d'interactions passées entre ce même SR d'information et ses utilisateurs. Malheureusement le nombre d'interactions passées est grand, tandis que le nombre de recommandations qu'il est possible de faire en ligne est très faible. Cette opportunité est idéale lorsque des approches nécessitent de construire un contexte, comme une matrice pour certaines approches collaboratives, afin de tester notamment si les temps de réponses sont satisfaisants. Cependant dans notre cas ce challenge ne convient pas : nous voulons évaluer une approche séquentielle, et les interactions séquentielles en ligne mises à notre disposition sont trop peu nombreuses pour tester correctement nos stratégies.

6.2.1 Simulation de l'aspect séquentiel de la recommandation

Lorsque nous sommes dans l'impossibilité de tester les algorithmes sur des SR en ligne, il convient de simuler cet aspect séquentiel. Pour cela il faut disposer d'un jeu de données possédant une évaluation utilisateur d'un certain nombre d'objets. La majorité

1. News Recommendation Evaluation Lab de la conférence CLEF, 2013-2016 [Hopfgartner 2014]

des jeux de données nécessite de faire l'hypothèse que la popularité des objets n'évolue pas dans le temps. A chaque instant, la simulation consiste à :

- Tirer aléatoirement un utilisateur U, disposant d'un contexte ou non.
- Effectuer la recommandation d'un objet O à cet utilisateur.
- Récupérer dans les données l'évaluation donnée par U à O et comparer.

Une limite forte de cette approche de simulation est qu'un utilisateur peut ne pas avoir noté l'ensemble des objets. Dans ce cas, une note fictive est allouée au couple utilisateur/objet. Cette note peut être arbitraire, comme la moyenne de toutes les autres notes, la note minimale, ..., ou alors elle peut être prédite avec des approches de complétion de matrices.

Dans l'article [Li 2010], les auteurs proposent un nouveau cadre d'évaluation, basé sur le jeu de données *Yahoo ! Front Page Today Module User Click Log*[2]. Ce jeu de données contient plus de 45 millions d'interactions entre des utilisateurs et le "Today module" où quatre informations sont affichées et l'une d'entre elles est mise en avant. Ce jeu de données a été construit en choisissant aléatoirement l'information qui est mise en avant et les données sont ordonnées en fonction du temps. Le cadre d'évaluation consiste donc à choisir à chaque instant parmi les informations disponibles celle qui doit être mise en avant. Si cette information est bien celle mise en avant dans la ligne actuelle du jeu de données, la récompense associée (clic/non clic) est prise en compte, sinon la ligne actuelle n'est pas prise en compte et on continue en utilisant la ligne suivante. Ce cadre applicatif est particulièrement adapté pour les approches de bandit car aucune information qui ne découle pas de nos choix passés n'est prise en compte.

6.2.2 Jeux de données

De nombreux jeux de données sont disponibles pour évaluer les approches de recommandation. Cependant chacun de ces jeux possèdent des avantages et des limites : certains prennent en compte le fait que la popularité des objets peut évoluer dans le temps, d'autres permettent de recommander plusieurs objets à chaque instant. Le choix des jeux de données à utiliser dépend de la problématique considérée. Dans le tableau 6.1, nous listons un certain nombre d'entre eux, avec le nombre d'interactions contenues dans celui-ci. Nous précisons également les problématiques qui peuvent être abordées à l'aide de ce jeu. Certains challenges, comme ceux de la conférence CLEF et RECSYS, proposent d'utiliser des jeux non cités ici.

2. Jeu de données disponible sur le site http ://webscope.sandbox.yahoo.com/

Jeu de données	Taille	Problématiques		
		Contextuelle	Tirages multiples	Évolution temporelle
Movielens 100 [Kohli 2013]	16,476	non	oui	non
Movielens 100k [Harper 2015]	100,000	non	oui	non
Movielens 1M [Harper 2015]	1,000,000	non	oui	non
Movielens 10M [Harper 2015]	10,000,000	non	oui	non
Jester [Goldberg 2001]	4,100,000	non	oui	non
Douban [Ma 2011]	16,830,839	non	oui	non
Yahoo ! Front Page Today Module [Li 2010]	45,811,883	oui	non	oui

TABLE 6.1 – Jeux de données utilisables en recommandation et problématiques liées

6.3 Cadre expérimental utilisé dans ce manuscrit

Pour évaluer nos approches et la comparer aux méthodes de l'état de l'art, nous utilisons dans les chapitres 7, 8 et 9 le cadre expérimental défini par Kohli et al. [Kohli 2013]. Les expérimentations font appel aux jeux de données MovieLens-100 et Jester.

Le premier jeu de données, MovieLens-100, contient 943 utilisateurs qui ont noté 100 films. Les notes sont comprises entre 1 (mauvais) et 5 (bon) (si un film n'est pas noté par un utilisateur, la note minimale de 1 lui est affectée). Pour traduire les notes en actions utilisateurs, Kohli et al. ont choisi de fixer un seuil de pertinence. Dans leurs expérimentations, deux valeurs ont été utilisées : 2 et 4. Lorsque le seuil est 2 (resp. 4), tous les films qui ont une note strictement supérieure à 2 (resp. 4) sont considérés comme pertinents, c'est à dire que l'utilisateur les a cliqués.

Le deuxième jeu de données, Jester, contient 25 000 utilisateurs qui ont noté 100 blagues. Les notes sont comprises entre -10 (pas drôle) et 10 (très drôle). Kohli et al. ont choisi de fixer le seuil de pertinence à 7 : toutes les blagues qui ont une note strictement supérieure à 7 sont considérées comme cliquées par l'utilisateur et donc pertinentes.

Pour simuler l'aspect temps réel du SR, à chaque instant t, un utilisateur est choisi aléatoirement et m = 5 objets lui sont recommandés. L'utilisateur choisi est toujours considéré comme un nouvel utilisateur. Ce dernier clique sur un objet seulement si la note associée est strictement supérieure au seuil choisi. Si l'utilisateur clique au moins sur un objet, la récompense obtenue est de $Z_t = 1$, sinon $Z_t = 0$. Notre objectif est de maximiser la somme des Z_t, autrement dit de minimiser l'abandon.

Chaque expérimentation est réalisée sur un intervalle de temps de longueur T = 100 000. Les figures présentées au sein de chaque chapitre le taux de clics sur les 1 000 derniers instants, moyenné sur 200 expérimentations de *Monte Carlo*.

Conclusion

La recommandation à tirages multiples apportent deux nouvelles problématiques importantes : l'ordonnancement et la diversité. Ces problématiques sont très populaires en recherche d'information. La problématique d'ordonnancement est peu abordée en recommandation, principalement car les listes de recommandations sont généralement d'une taille suffisamment faible pour que l'utilisateur puisse visualiser facilement l'ensemble des recommandations. Au contraire, nous avons montré que les récompenses obtenues sont fortement impactées par la problématique de la diversité. En effet plusieurs travaux montrent que diversifier les listes de résultats peut améliorer le taux de ventes, clics, vues, ... Nous avons également indiqué que l'évaluation des approches séquentielles à tirages multiples en recherche académique est majoritairement basée sur des simulations utilisant des jeux de données statiques.

Des approches utilisant des stratégies de bandit pour recommander plusieurs objets à chaque instant ont été proposées dans des travaux de l'état de l'art. Ces approches et leurs limites sont présentées dans le chapitre suivant, ainsi que l'une de nos contributions, permettant de corriger ces limites.

Chapitre 7

Bandit à tirages simples et recommandation à tirages multiples

Sommaire

Résumé.

les approches de l'état de l'art utilisent autant d'instances d'une stratégie de bandit à tirages simples qu'il y a d'objets à recommander. Nous proposons au contraire de gérer l'ensemble des recommandations par une seule instance d'une stratégie de bandit pour rendre l'apprentissage plus efficace. Nous montrons sur deux jeux de données de référence (Movielens et Jester) que l'approche que nous proposons permet d'obtenir des vitesses d'apprentissage jusqu'à treize fois plus rapides tout en obtenant des taux de clics équivalents.

Les stratégies de bandit de l'état de l'art utilisent autant d'instances d'une stratégie de bandit qu'il y a de recommandations à effectuer à chaque instant. Si sur le long

terme ces approches obtiennent de bons résultats, l'utilisation d'une seule instance d'une stratégie de bandit pour gérer l'ensemble des recommandations semble pouvoir permettre d'atteindre plus rapidement des proportions d'abandon satisfaisantes. Dans ce chapitre nous commençons par fournir un formalisme, afin de pouvoir décrire par la suite les approches de l'état de l'art ainsi que notre proposition, l'approche *Multiple-Play Bandit*. Une évaluation est ensuite effectuée en utilisant deux jeux de données populaires

7.1 Formalisme et solutions sous-optimales

7.1.1 Formalisme

Nous nous intéressons à l'application des stratégies de bandit dans le domaine de la recommandation. Ainsi par rapport au vocabulaire utilisé dans le chapitre 2 qui présentait les principes de ces stratégies, les bras deviennent des objets, l'agent devient un système de recommandation, les récompenses deviennent des clics utilisateurs.

Lorsque plusieurs objets sont recommandés à chaque instant, l'utilisateur peut choisir de cliquer un ou plusieurs objets parmi ceux qui lui ont été recommandés. Les recommandations cliquées sont généralement considérées comme pertinentes. Si l'utilisateur ne clique sur aucune des recommandations, il s'agit d'un *abandon*. Dans le but d'optimiser les performances d'un SR, nous considérons dans ce chapitre le problème de la minimisation de l'abandon. Ce problème peut également être vu comme la maximisation du nombre de fois où les utilisateurs sélectionnent au moins un objet parmi ceux recommandés.

Une recommandation cliquée par l'utilisateur est considérée comme pertinente comme cela est généralement le cas en recommandation. La principale source d'information pour évaluer l'efficacité d'un SR est le retour utilisateur [Ricci 2011].

Considérons une collection de k objets notés I_i avec $i \in \{1, \ldots, K\}$. À chaque instant t, m objets sont recommandés à un utilisateur. Si cet utilisateur clique sur au moins un objet, nous obtenons une récompense de 1, sinon la récompense est de 0. L'objectif est de minimiser la fraction de 0 obtenus, autrement dit de minimiser la proportion d'abandon. P_m^K est l'ensemble des combinaisons de m objets qu'il est possible d'obtenir avec K objets. Un utilisateur est représenté par un vecteur de pertinence $X = \{0, 1\}^K$ où $X_i = 1$ si l'objet i est cliqué par l'utilisateur. A chaque instant t, l'utilisateur est représenté par X_t et une combinaison $A_t \in P_m^K$ lui est recommandée.

Z_t est la récompense obtenue pour la combinaison A_t associée au vecteur de per-

tinence X_t. Il est défini par :

$$Z_t = \max_{i \in A_t} X_{i,t} \tag{7.1}$$

Chaque composant $i \in \{1, \ldots, K\}$ du vecteur X suit une distribution de Bernoulli de paramètre p_i inconnu. La probabilité p_i peut être estimée à chaque instant t par :

$$\hat{p}_i(t) = \frac{1}{N_i(t)} \sum_{t\ :\ i \in A_t} X_{i,t} \quad \text{avec} \quad N_i(t) = \sum_{t\ :\ i \in A_t} 1 \tag{7.2}$$

La proportion d'utilisateurs qui considèrent comme pertinent au moins un objet de A_t est $E[Z_t]$, l'espérance de la variable Z_t. Maximiser $E[Z_t]$ est équivalent à minimiser la proportion d'abandon. La ou les combinaisons optimales A^* sont donc la ou les combinaisons qui provoquent au moins un clic chez un maximum d'utilisateurs. A^* est définie par :

$$A^* = \operatorname*{argmax}_{A \in P_m^K} E[Z] \tag{7.3}$$

Pour rappel, le but d'un SR dans le contexte temps réel peut être défini comme la minimisation de la différence entre $\sum_{t=1}^{T} Z^*$ (la somme des récompenses obtenues en utilisant une combinaison optimale A^*) et $\sum_{t=1}^{T} Z_t$ (la somme des récompenses obtenues avec la combinaison recommandée par l'approche utilisée). Cette différence est appelée le regret cumulé :

$$R(T) = T \times E[Z^*] - \sum_{t=1}^{T} E[Z_t] \tag{7.4}$$

7.1.2 Solutions sous-optimales

Trouver une combinaison optimale A^* est un problème NP-complet [Radlinski 2008]. C'est pourquoi une combinaison sous-optimale mais plus facile à atteindre semble plus appropriée au contexte temps réel. Kohli et al. [Kohli 2013] utilisent deux autres combinaisons : la combinaison indépendante et la combinaison diversifiée.

La combinaison indépendante est imaginée selon le principe de classement par probabilité [Robertson 1977]. Elle est définie comme la combinaison des m objets les plus cliqués.

$$A^{\text{indépendante}} = \operatorname*{argmax}_{A \in P_m^K} \sum_{i \in A} E[X_i] \tag{7.5}$$

Cette combinaison est visée par l'approche *IBA* qui est conçue pour maximiser :

$$Z_t^{\text{indépendante}} = \sum_{i \in A_t} X_{i,t} \tag{7.6}$$

La notion de diversité individuelle est essentielle en recommandation, elle n'est pourtant pas prise en compte avec la combinaison indépendante [Chen 2006]. Prenons un exemple de recommandation de films : admettons que les films les plus populaires sont ceux l'hexalogie « Star Wars ». Dans ce cadre, il y a des chances que ces 6 films soient aimés par des personnes aux goûts similaires, qui représentent la majorité des utilisateurs. Toutefois pour tous les utilisateurs qui ont des goûts différents, aucune de ces 6 propositions ne les satisfera. Une combinaison composée de films de genre différents est plus appropriée, particulièrement dans le cas de la minimisation de l'abandon et/ou dans le cas où peu d'informations sur les objets/utilisateurs sont disponibles.

Pour prendre en compte la notion de diversité, une notion de combinaison diversifiée a été définie [Radlinski 2008]. Il s'agit de la combinaison qui propose l'objet le plus populaire en première position, et ensuite les objets les plus populaires lorsque les objets aux positions précédentes ne sont pas cliqués :

$$A_1^{\text{diversifiée}} = \underset{i \in K}{\operatorname{argmax}} E[X_i] \quad \text{et} \tag{7.7}$$

$$A_k^{\text{diversifiée}} = \underset{i \in K / \left\{A_{1,\dots,k-1}^{\text{diversifiée}}\right\}}{\operatorname{argmax}} E\left[X_i | X_j = 0 \ \forall j \in \left\{A_{1,\dots,k-1}^{\text{diversifiée}}\right\}\right] \tag{7.8}$$

Cette combinaison prend en compte la notion de diversité car elle est constituée des objets les plus populaires lorsque les objets recommandés aux positions précédentes ne sont pas cliqués. Il s'agit tout de même d'une combinaison sous-optimale qui peut être différente de la combinaison ayant la plus grande probabilité d'obtenir au moins un clic. En effet la combinaison diversifiée recommande en première position l'objet le plus populaire, cet objet ne fait pas obligatoirement partie de la combinaison optimale A^*.

La combinaison indépendante $A^{\text{indépendante}}$ est considérée comme moins bonne que la combinaison diversifiée $A^{\text{diversifiée}}$ lorsque le but est la minimisation du nombre d'abandon [Radlinski 2008].

7.2 Stratégies à tirages simples pour le cas à tirages multiples

Dans cette section, nous détaillons deux approches de la littérature utilisant autant d'instances d'une stratégie de bandit qu'il y a de recommandations à effectuer, *RBA* et *IBA*. Nous les utilisons comme approches de références. L'approche *RBA* vise la combinaison diversifiée, tandis que l'approche *IBA* vise la combinaison indépendante.

Nous présentons dans un deuxième temps notre proposition, l'approche *MPB*, qui vise la combinaison diversifiée.

Les premières approches de recommandation utilisant des stratégies de bandit sont antérieures aux versions à tirages multiples de la littérature statistique. Au delà de l'aspect contextuel, se posait la question de comment utiliser ces stratégies conçues pour recommander un seul objet à chaque instant, lorsque plusieurs recommandations sont nécessaires. Dans ce cadre, Radlinski et al. [Radlinski 2008] ont développé l'approche *Ranked Bandit Algorithm* (*RBA*) qui utilise autant d'instances d'une stratégie de bandit à tirages simples que de recommandations à soumettre. Ces instances peuvent être des stratégies telles que *UCB1* [Auer 2002a], *ϵ-greedy* [Sutton 1999] ou encore *Exp3* [Auer 2002b]. Plus récemment Kohli et al. [Kohli 2013] ont créé l'approche *Independent Bandits Algorithm* (*L'approche «Independent Bandit Algorithm» (IBA)*). La principale différence avec l'approche *RBA* est la manière dont l'action de l'utilisateur est prise en compte (voir section 7.2.1.2).

L'approche *Combinatorial Multi-Armed Bandit (CMAB)* proposée par Chen et al. [Chen 2013] utilise une seule instance d'une stratégie de bandit pour effectuer plusieurs recommandations. L'approche *Multiple Plays Bandit (MPB)* que nous proposons dans cet article repose sur la même idée, mais se démarque sur la manière dont le retour utilisateur est considéré. Alors que l'approche *CMAB* optimise le nombre total de clics à chaque instant, l'approche *MPB* tend à optimiser le nombre de fois où au moins un clic est observé. Cette différence permet d'assurer une certaine diversité dans les résultats.

7.2.1 Ranked Bandit Algorithm (RBA)

L'approche *RBA (Algorithme 7.1)* a été développée par Radlinski et al. en 2008 [Radlinski 2008] et nécessite l'utilisation en parallèle de m instances d'une stratégie de bandit à tirages simples. À chaque instant t, les m objets choisis par les m instances de la stratégie de bandit sont recommandés à l'utilisateur. L'information de chaque instance est mise à jour de la manière suivante : l'instance correspondant au premier objet cliqué obtient une récompense de 1, tandis que tous les autres obtiennent une récompense de 0. De cette manière, la première instance tend à recommander l'objet avec le taux de clics le plus haut. La deuxième instance peut recevoir une récompense de 1 uniquement si le premier objet n'est pas cliqué. Elle tend ainsi à recommander l'objet qui obtient le meilleur taux de clics conditionnellement au fait que l'objet avec le taux de clics le plus haut n'ai pas été cliqué et ainsi de suite pour les instances suivantes.

Algorithme 7.1 : Ranked Bandit Algorithm (RBA)

BTS_i : instance d'une stratégie de bandit à tirages simples pour la recommandation en position i
pour $t = 1, \ldots, T$ **faire**
  pour $i = 1, \ldots, m$ **faire**
    $a_i \leftarrow$ SélectionnerObjet(BTS_i, K)
    si $a_i \in \{a_1, \ldots, a_{i-1}\}$ **alors**
      $a_i \leftarrow$ choix arbitraire parmi les documents non sélectionnés
    fin
  fin
  $A_t \leftarrow \cup_i a_i$
  Recommander A_t à l'utilisateur, récupérer le retour utilisateur X_t
  pour $i = 1, \ldots, m$ **faire**
    Retour utilisateur :

$$z_i = \begin{cases} 1 & \text{si l'objet } a_i \text{ est le premier cliqué} \\ 0 & \text{sinon} \end{cases}$$

    MiseAJour(BTS_i, z_i)
  fin
fin

7.2.2 Independent Bandits algorithm (IBA)

Plus récemment, l'approche *IBA (Algorithme 7.2)* a été développée par Kohli et al. [Kohli 2013]. Tout comme l'approche *RBA*, elle nécessite l'utilisation en parallèle de m instances d'une stratégie de bandit à tirages simples. La principale différence entre les deux approches est la manière dont le retour utilisateur est pris en compte. Dans l'approche *RBA* seulement le premier objet est considéré comme pertinent, tandis que dans l'approche *IBA* tous les objets cliqués sont considérés comme pertinents. Ceci induit que la $k^{\text{ième}}$ instance tend à recommander l'objet avec le $k^{\text{ième}}$ taux de clics le plus haut. L'approche *IBA* vise la combinaison indépendante.

La proportion d'abandon est généralement plus grande pour la combinaison indépendante que pour la combinaison diversifiée, ce qui induit que l'approche *RBA* est, dans la plupart des cas, meilleure que l'approche *IBA* sur le très long terme (lorsque t est grand). Cependant, Kohli et al. ont montré dans leurs simulations que le temps d'apprentissage nécessaire pour l'approche *IBA* est significativement plus court (la proportion d'abandon diminue donc plus rapidement qu'avec l'approche *RBA*), ce qui est

Algorithme 7.2 : Independent Bandits Algorithm (IBA)

BTS_i : instance d'une stratégie de bandit à tirages simples pour la recommandation en position i
pour $t = 1, \ldots, T$ **faire**
  pour $i = 1, \ldots, m$ **faire**
    $a_i \leftarrow$ SélectionnerObjet(BTS_i, $K \setminus A_{t,i-1}$)
  fin
  $A_t \leftarrow \cup_i a_i$
  Recommander A_t à l'utilisateur, récupérer le retour utilisateur X_t
  pour $i = 1, \ldots, m$ **faire**
    Retour utilisateur :

$$z_i = \begin{cases} 1 & \text{si l'objet } a_i \text{ est cliqué} \\ 0 & \text{sinon} \end{cases}$$

    MiseAJour(BTS_i, z_i)
  fin
fin

un indicateur important pour la qualité d'un SR, en particulier dans un environnement qui change souvent ou en début de mise en service.

Dans les deux approches précédentes, les m instances d'une stratégie de bandit à tirages simples fonctionnent de façon (quasiment) indépendantes les unes des autres. À chaque instant t, chaque instance apprend uniquement sur un seul objet. Dans ces conditions, les deux approches convergent uniquement lorsque toutes les instances ont convergé, ce qui augmente le regret. Au contraire, une approche permettant d'utiliser une seule instance d'une stratégie de bandit pour gérer l'ensemble des recommandations permettrait d'apprendre sur m objets à chaque instant. On peut penser que l'apprentissage sera alors plus efficace.

Algorithme 7.3 : Multiple-Play Bandit

BTS : instance d'une stratégie de bandit à tirages simples pour la recommandation en position i
pour $t = 1, \ldots, T$ **faire**
  pour $i = 1, \ldots, m$ **faire**
    $a_i \leftarrow$ SélectionnerObjet(BTS, $K \backslash A_{t,i-1}$)
  fin
  $A_t \leftarrow \cup_i a_i$
  Recommander A_t à l'utilisateur, récupérer le retour utilisateur X_t
  pour $i = 1, \ldots, m$ **faire**
    Retour utilisateur :

$$z_i = \begin{cases} 1 & \text{si l'objet } a_i \text{ est le premier cliqué} \\ -1 & \text{si } \forall a_j \text{ tel que } j <= i, \text{ aucun } a_j \text{ n'a été cliqué} \\ 0 & \text{sinon} \end{cases}$$

    MiseAJour(BTS, z_i)
  fin
fin

7.3 Contribution : Utilisation d'une seule instance d'une stratégie de bandit pour générer la liste de recommandations

7.3.1 Présentation de l'approche Multiple-Play Bandit (MPB)

L'approche Multiple-Play Bandit (*MPB*) [Louëdec 2015d] que nous proposons est décrite dans l'*algorithme 7.3*. La principale différence avec les approches *RBA* et *IBA* est que nous utilisons une seule instance d'une stratégie de bandit. Pour recommander simultanément une liste de m objets, le premier objet est sélectionné selon la règle utilisée par le bandit. Le second objet est sélectionné parmi l'ensemble des objets disponibles excepté celui précédemment recommandé, et ainsi de suite pour les recommandations suivantes. Les estimateurs de la pertinence des m objets sont mis à jour, estimateurs gérés par la même instance d'une stratégie de bandit. De cette manière, m estimateurs seront mis à jour à chaque instant. Ainsi l'estimateur de la pertinence d'un objet approche plus rapidement les véritables valeurs associées, et la vitesse d'apprentissage sera améliorée.

Une autre différence importante concerne la façon dont le retour de l'utilisateur est pris en compte. À chaque instant une liste de m recommandations classée par ordre de pertinence estimée décroissant, est proposée à l'utilisateur. Lors de la recommandation, lorsque l'on suppose qu'une liste ordonnée est recommandée, tous les objets non cliqués et qui sont recommandés avant le premier objet cliqué sont pénalisés, avec une récompense négative (-1). De cette manière, les objets qui sont peu cliqués sont vite mis de côté. Le choix de la valeur -1 est arbitraire, l'ajustement de cette valeur sera une piste pour de futurs travaux. Cette méthode permet de prendre en compte plus facilement le vieillissement d'un objet : si un objet était populaire mais ne l'est plus, beaucoup de récompenses négatives seront retournées et l'estimateur de la pertinence lié à cet objet va rapidement décroître. Pour les autres objets, seulement le premier cliqué obtient une récompense positive (1), tandis que les autres obtiennent tous une récompense nulle (0). De cette manière cette approche vise la combinaison diversifiée.

7.3.2 Évaluation en utilisant deux jeux de données de référence

7.3.2.1 Cadre expérimental

Les jeux de données utilisés et la manière dont l'expérimentation est conduite sont décrits dans la section 6.3.

Afin d'avoir une comparaison efficace entre notre approche et les approches existantes (*RBA* et *IBA*), nous avons choisi de les implémenter tout en utilisant les stratégies de bandit ϵ-*greedy* et *Upper Confidence Bound 1 (UCB1)*. Ce sont les stratégies utilisées dans l'article de Kohli et al. [Kohli 2013] qui nous sert de référence pour le cadre expérimental.

Les résultats expérimentaux sont représentés dans la figure 7.1, présentant l'expérimentation sur la collection Movielens avec comme seuil de pertinence 2, la figure 7.2, présentant l'expérimentation sur la collection Movielens avec comme seuil de pertinence 4, et la figure 7.3 présentant l'expérimentation sur la collection Jester avec comme seuil de pertinence 7. Dans ces figures, la combinaison optimale et la combinaison indépendante sont représentées par les courbes horizontales. Dans les trois expérimentations mises en place, la combinaison diversifiée est la même que la combinaison optimale. Pour rappel, la combinaison indépendante est constituée des m objets les plus cliqués, et la combinaison diversifiée est constituée de l'objet le plus populaire en première position et des objets les plus populaires lorsque les objets aux positions précédentes ne sont pas cliqués. L'approche *IBA* vise la combinaison indépendante tandis que l'approche *RBA* et l'approche *MPB* visent la combinaison diversifiée. La différence entre ces deux combinaisons est approximativement de 1 % pour l'expéri-

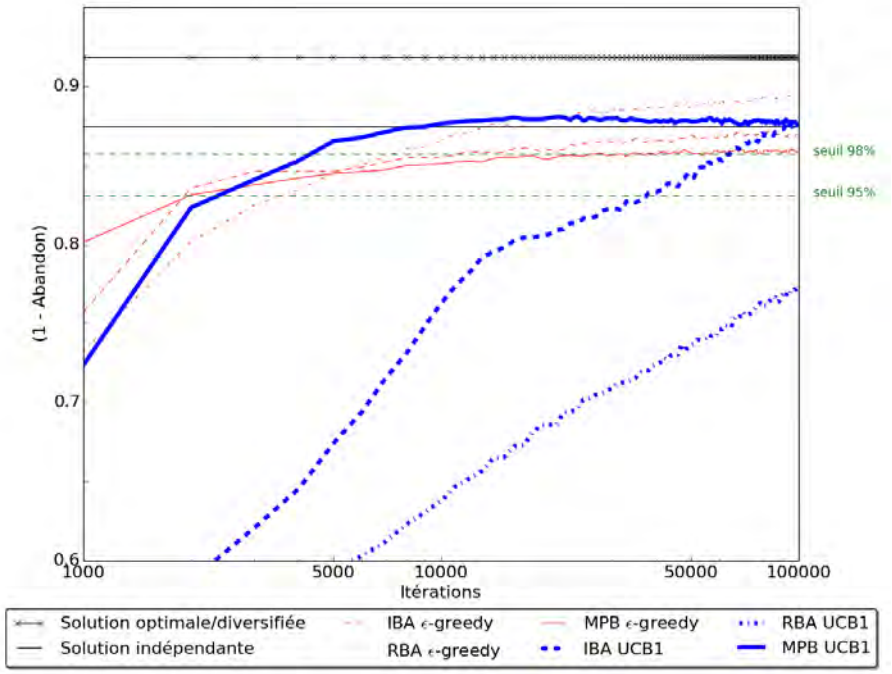

FIGURE 7.1 – Comparaison des différentes approches avec le jeu de données MovieLens avec un seuil de pertinence = 2. (1 - la proportion d'abandon) en fonction du nombre d'interaction avec le système.

mentation sur la collection Movielens avec le seuil de pertinence 4 (voir figure 7.2) et l'expérimentation sur la collection Jester (voir figure 7.3). Cette différence permet d'obtenir environ 2 % d'augmentation potentielle du revenu du service de recommandation. Cette différence est plus importante dans le cadre de l'expérimentation sur la collection MovieLens avec le seuil de pertinence 2 avec un écart de 4 % environ (voir figure 7.1). Cette différence permet d'obtenir environ 5 % d'augmentation potentielle du revenu du service de recommandation.

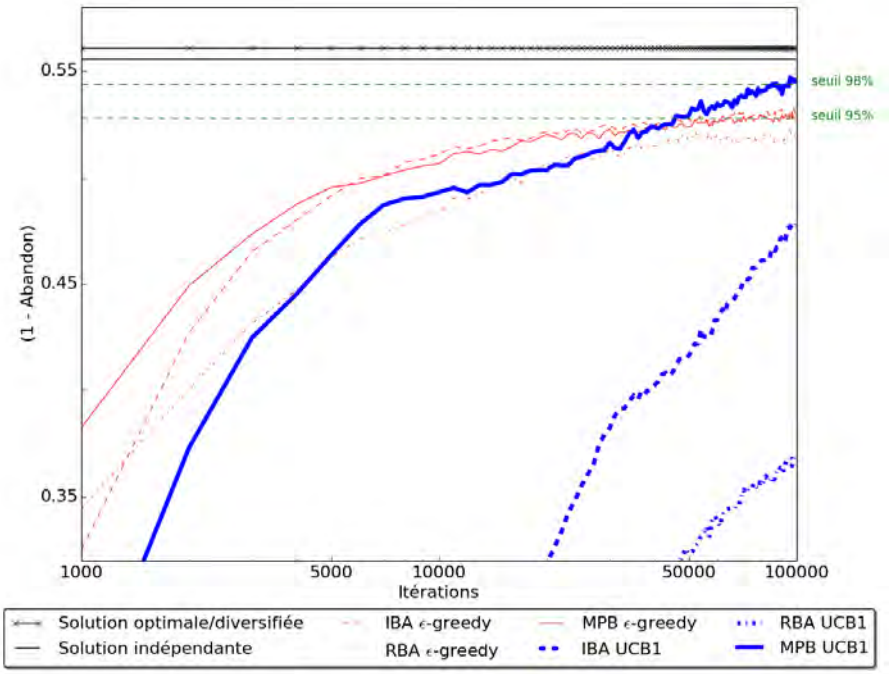

FIGURE 7.2 – Comparaison des différentes approches avec le jeu de données MovieLens avec un seuil de pertinence = 4. (1 - la proportion d'abandon) en fonction du nombre d'interaction avec le système.

7.3.2.2 Commentaires sur les résultats

UCB1

Nous avons souhaité observer le comportement des trois approches étudiées lorsque celles-ci sont implémentées avec la stratégie de bandit *UCB1*. L'approche *MPB* obtient des proportions d'abandon équivalentes ou plus faibles ce qui signifie que l'approche est plus efficace que les approches *RBA* et *IBA* au terme des 3 expérimentations. En outre, pour l'approche *MPB*, le temps d'apprentissage est bien plus court. Pour vérifier la pertinence statistique de l'amélioration du temps d'apprentissage, un test unilatéral de Wilcoxon a été mis en place. Ce test permet de donner un niveau de confiance sur l'hypothèse que la vitesse d'apprentissage de l'approche *MPB* est moins bonne ou identique à l'approche comparée, l'hypothèse alternative étant que l'approche *MPB* apprend plus vite. Très clairement, l'approche *MPB* obtient de meilleurs résultats, avec des p-values toutes inférieures à $2,2 \times 10^{-16}$.

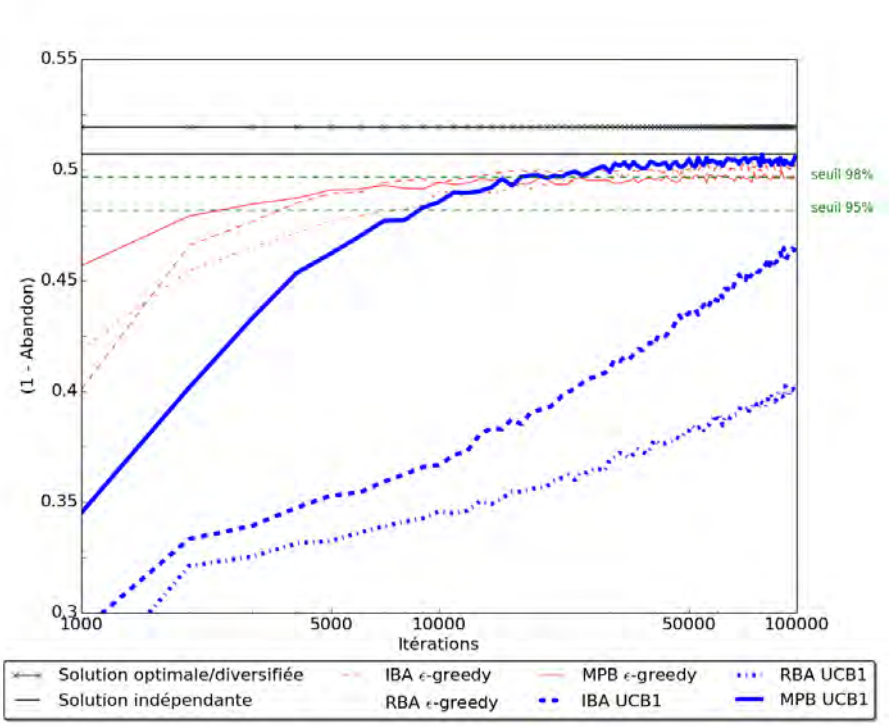

FIGURE 7.3 – Comparaison des différentes approches avec le jeu de données Jester avec un seuil de pertinence = 7. (1 - la proportion d'abandon) en fonction du nombre d'interaction avec le système.

Pour quantifier l'amélioration de la vitesse d'apprentissage, nous comparons le nombre d'étapes de temps nécessaire aux différentes approches pour atteindre 95 % et 98 % de la combinaison indépendante en moyenne. Les méthodes de l'état de l'art atteignent ces valeurs uniquement sur la collection MovieLens avec le seuil de pertinence 2 (figure 7.1). Pour cette expérimentation, l'approche *MPB* atteint treize fois plus vite les deux seuils. Concernant les expérimentations sur la collection MovieLens avec le seuil de pertinence 4 et sur la collection Jester, les seuils 95 % et 98 % ne sont pas atteints en moyenne par les approches *RBA* et *IBA*. La forme des courbes laisse supposer que ces deux approches n'ont pas eu assez des 100 000 étapes de temps pour atteindre les deux valeurs seuils. Cependant dans la figure 7.1, l'approche *MPB* atteint la valeur seuil de 95 % en 3 000 itérations et 98 % en 5 000 itérations en moyenne sur la collection MovieLens avec le seuil de pertinence de 2. Dans la figure 7.3, pour la collection Jester, 9 000 itérations sont nécessaires pour atteindre 95 % de la combinaison indépendante et 18 000 itérations pour la valeur seuil de 98 %. Dans la figure 7.2, avec le seuil de

% (1 - abandon) de la combinaison indépendante		**95 %**	**98 %**
collection	approche	p-value	p-value
MovieLens seuil = 2	**IBA-Egreedy**	0,0010	0,9385
	RBA-Egreedy	1,0e-11	0,0067
	RBA-UCB1	<2,2e-16	<2,2e-16
	IBA-UCB1	<2,2e-16	<2,2e-16
MovieLens seuil = 4	**IBA-Egreedy**	0,82	0,84
	RBA-Egreedy	8,0e-6	0,00032
	RBA-UCB1	<2,2e-16	<2,2e-16
	IBA-UCB1	<2,2e-16	<2,2e-16
Jester seuil = 7	**IBA-Egreedy**	9,3e-11	0,0012
	RBA-Egreedy	<2,2e-16	1,3e-15
	RBA-UCB1	<2,2e-16	<2,2e-16
	IBA-UCB1	<2,2e-16	<2,2e-16

TABLE 7.1 – Résultats du Test de Wilcoxon unilatéral pour comparer le nombre d'étapes nécessaire pour atteindre 95 % et 98 % de la combinaison indépendante en utilisant les méthodes de l'état de l'art avec les stratégie de bandit ϵ-*greedy* et *UCB1* et cette même quantité en utilisant l'approche *MPB*. Hypothèse alternative : l'approche *MPB* apprend plus vite. L'hypothèse alternative est retenue pour l'ensemble des comparaisons, hormis la comparaison avec l'approche *IBA* utilisant la stratégie ϵ-greedy

pertinence 4 sur la collection MovieLens, les valeurs seuils sont atteintes moins rapidement, mais toujours avant les 100 000 itérations (95 % en 46 000 itérations, 98 % en 96 000 itérations en moyenne).

ϵ-greedy

Regardons maintenant comment se comporte l'approche *MPB* lorsque la stratégie de bandit ϵ-*greedy* est utilisée. Pour les 3 expérimentations mises en œuvre, globalement l'approche *MPB* obtient des proportions d'abandon équivalentes à celles obtenues par les méthodes de l'état de l'art. D'après les tableaux 7.1 et 7.2, une amélioration significative de la vitesse d'apprentissage peut être constatée en comparant l'approche *MPB* avec l'approche *RBA*. Par contre, cette implémentation avec ϵ-*greedy* ne permet pas d'observer une nette amélioration de la vitesse d'apprentissage en comparant l'approche *MPB* avec l'approche *IBA*. Il apparaît tout de même que pour le seuil de 95 %, notre approche est au moins aussi rapide que les approches existantes. Ce n'est pas

collection	approche	ratio (95 %)	ratio (98 %)
MovieLens seuil = 2	**IBA-Egreedy**	$\approx$ 1,0	$\approx$ 0,5
	RBA-Egreedy	$\approx$ 2,0	$\approx$ 0,3
	IBA-UCB1	(+)	(+)
	RBA-UCB1	(+)	(+)
MovieLens seuil = 4	**IBA-Egreedy**	$\approx$ 1,0	(-)
	RBA-Egreedy	(+)	(-)
	IBA-UCB1	(+)	(+)
	RBA-UCB1	(+)	(+)
Jester seuil = 7	**IBA-Egreedy**	$\approx$ 1,5	$\approx$ 0,3
	RBA-Egreedy	$\approx$ 2,5	$\approx$ 0,6
	IBA-UCB1	(+)	(+)
	RBA-UCB1	(+)	(+)

TABLE 7.2 – Ratios approximatifs entre le nombre d'étapes nécessaire pour atteindre 95 % et 98 % de la combinaison indépendante en utilisant les méthodes de l'état de l'art avec la stratégie de bandit ϵ-*greedy* et cette même quantité en utilisant l'approche *MPB*. (+) : l'approche *MPB* atteint le seuil, mais pas l'approche comparée, (-) : aucune des deux approches n'atteint le seuil. L'approche *MPB* atteint le seuil de 95 % au moins aussi rapidement que les approches comparées. Cependant, l'approche *MPB* met plus de temps à atteindre le seuil 98 % lorsque la stratégie ϵ-greedy est utilisé que les approches *RBA* et *IBA*

le cas pour le seuil 98 % dans la figure 7.2, sur la collection MovieLens avec le seuil de pertinence 4, *MPB* n'atteint pas ce seuil en moyenne. Pour la collection MovieLens avec le seuil de pertinence 2 de la figure 7.1, le seuil est atteint mais deux fois moins rapidement qu'avec l'approche *IBA*. Enfin dans la figure 7.3 avec la collection Jester, trois fois plus d'itérations sont nécessaires pour atteindre le seuil de 98 %.

La stratégie de bandit ϵ-*greedy* réalise son exploration en choisissant quelques objets au hasard avec une probabilité de ϵ. En utilisant une seule stratégie de bandit comme dans l'approche *MPB*, le risque est de recommander les objets avec les meilleures proportions de clics très rarement au profit d'autres. Ceci est particulièrement vrai si ces objets ne sont pas cliqués les premières fois qu'ils sont recommandés.

Expérimentalement nous avons pu observer que l'utilisation d'autant d'instances d'une stratégie de bandit que de objets à recommander, comme dans les approches *RBA* et *IBA*, permet de réduire ce problème. En effet si un objet n'est pas cliqué à un instant donné, une seule instance le prendra en compte, laissant ainsi toutes ses chances d'être

recommandé par l'une des autres instances.

L'amélioration significative des vitesses d'apprentissage apportée par l'approche *MPB* lorsque la stratégie de bandit *UCB1* est utilisée est intéressante. En visant la combinaison diversifiée, cette approche atteint des proportions d'abandon inférieures aux autres approches sur le long terme ($t > 50\ 000$). Cependant, lorsque la stratégie de bandit ϵ-*greedy* est utilisée, l'approche *MPB* n'améliore pas significativement la vitesse d'apprentissage et l'approche *IBA* reste la plus rapide.

Conclusion

Dans ce chapitre, nous nous sommes intéressés au problème de la recommandation à tirages multiples en utilisant des stratégies de bandit à tirages simples. Habituellement, les approches de l'état de l'art utilisent autant d'instances d'une stratégie de bandit à tirages simples qu'il y a d'objets à recommander. Pour améliorer la vitesse d'apprentissage, nous avons proposé de gérer l'ensemble des recommandations simultanément, en utilisant une seule instance d'une stratégie de bandit à tirages simples. Dans le cas de l'utilisation de la stratégie de bandit *UCB1*, l'approche proposée *MPB* apprend beaucoup plus rapidement (jusqu'à treize fois plus rapidement). De plus elle obtient des proportions d'abandon équivalentes à celles de l'approche la plus rapide de l'état de l'art, *IBA*. Cependant, en utilisant la stratégie de bandit ϵ-*greedy*, l'approche *MPB* n'atteint pas toujours des proportions d'abandon aussi faibles que *IBA*, pour une vitesse d'apprentissage équivalente. Nous avons pu observer expérimentalement que la nature aléatoire de la stratégie ϵ-*greedy* peut expliquer qu'aucune amélioration n'est observable.

En recommandation, les seules stratégies de bandit utilisées sont celles conçues pour recommander un seul objet à chaque instant. L'état de l'art récent propose pourtant de nouvelles stratégies capables de recommander plusieurs objets à chaque instant. Dans le chapitre suivant nous proposons une implémentation de l'un d'eux afin qu'il soit utilisable en recommandation. Nous comparons également les performances obtenues en utilisant cette stratégie avec les performances obtenues en utilisant les approches de l'état de l'art.

Chapitre 8

Bandit à tirages multiples : état de l'art et applications

Sommaire

Résumé.

. La littérature en statistique récente propose de nouvelles stratégies capables de recommander plusieurs objets à chaque instant, mais ne les évaluent pas sur des jeux de données réelles. L'implémentation de l'une de ces stratégies que nous proposons nous permet de l'utiliser dans le cadre de la recommandation. Lors de l'évaluation sur deux jeux de données de référence (Jester et Movielens), cette stratégie nous permet d'obtenir des proportions d'abandon équivalentes à l'approche *MPB* tout en obtenant des vitesses d'apprentissage jusqu'à trois fois plus rapides.

Les approches présentées dans le chapitre précédent proposent d'adapter des stratégies de bandit conçues pour recommander un seul objet à chaque instant afin d'effectuer de multiples recommandations. Plus récemment, quelques travaux de la littérature statistique et recherche d'information proposent des stratégies conçues spécialement pour recommander plusieurs objets à chaque instant. Ces stratégies possèdent des garanties théoriques de performances minimales et maximales (bornes de regret). Si il semble très intéressant d'utiliser ces stratégies en recommandation, cela n'a pas encore été proposé, principalement car la complexité algorithmique de ces stratégies est plus grande. Une implémentation efficace, répondant aux contraintes de la recommandation, peut permettre d'obtenir des vitesses d'apprentissages plus rapides que celles des approches de l'état de l'art.

8.1 Stratégies de bandit à tirages multiples : approches récentes de la littérature

Les stratégies de bandit présentées dans le chapitre 2 sont conçues pour recommander un seul objet à chaque instant. Des travaux récents en apprentissage par renforcement et en recherche d'information proposent de nouvelles stratégies capables de recommander plusieurs objets à chaque instant. Lorsque plusieurs objets sont recommandés à chaque instant, l'objectif n'est plus de trouver l'objet le plus populaire, mais de trouver la combinaison d'objets permettant d'obtenir les meilleures récompenses globalement.

8.1.1 Approches inspirées des bandits à tirages simples

La stratégie *Exp3_M* [Uchiya 2010, Bubeck 2012] est une variante de la stratégie *Exp3* capable de recommander plusieurs objets à chaque instant. Dans la section 8.2, nous décrivons en détail cette stratégie et proposons une implémentation efficace de celle-ci.

La stratégie *Thompson Sampling* présentée dans le chapitre 2 consiste à recommander l'objet avec l'espérance la plus forte. Cette espérance est tirée aléatoirement selon la loi de probabilité Bêta associée à cet objet. La loi de probabilité est ensuite mise à jour en fonction du retour utilisateur obtenu. La stratégie *Thompson Sampling Multiple-Play* [Komiyama 2015] est une variante de cette stratégie capable de recommander plusieurs objets à chaque instant. Elle fonctionne de la même manière, mais au lieu de jouer l'objet avant l'espérance la plus forte, elle joue les k objets avec les k

Algorithme 8.1 : Multiple Play Thompson Sampling

k objets sont recommandés à chaque instant
Pour chaque objet $a \in (1, \ldots, K)$, initialiser $S_a = 0$ et $F_a = 0$
pour $t = 1, 2, \ldots$ **faire**
    Pour chaque objet a avec $a = 1, \ldots, K$, tirer $\theta_a(t)$ selon une distribution Beta($S_a + 1, F_a + 1$).
    Jouer $A_t =$ top-k objets ordonnés selon $\theta_a(t)$ et récupérer les récompenses $X_t(a)$ avec $a \in A_t$
    pour $a \in A_t$ **faire**
        si $X_t(a) = 1$ **alors**
            $S_a = S_a + 1$
        sinon
            $F_a = F_a + 1.$
        fin
    fin
fin

espérances les plus fortes, k étant le nombre de recommandations à effectuer. Plus de détails sont apportées dans l'algorithme 8.1.

8.1.2 Approches inspirées des modèles de clics de la recherche d'information

Récemment, les stratégies de bandit ont été adaptées en utilisant les modèles de clics de la littérature de la recherche d'information. La récente stratégie *cascading bandit* [Kveton 2015] est une variante pour l'apprentissage du modèle en cascade plus connu en recherche d'information [Craswell 2008]. Ce modèle suppose que l'utilisateur regarde les objets du haut vers le bas de la liste. Pour chaque objet, il choisit de cliquer sur l'objet qui est alors le seul objet pertinent ou de passer à l'objet suivant. L'utilisateur clique ainsi sur un seul objet qui lui paraît pertinent compte tenu des objets précédents. La stratégie *cascading bandit* considère donc que le retour utilisateur est observé jusqu'à ce qu'un des objets soit cliqué, aucun retour utilisateur n'est observé pour les objets classés plus bas dans la liste. Expérimentalement, placer les objets les plus mauvais en début de liste permet d'obtenir plus d'informations sans accroître le regret. Cela ne convient pas dans un cadre de recommandation, car les utilisateurs peuvent être agacés si les objets les mieux classés ne sont pas pertinents.

La stratégie *Parsimonious Item Exploration* [Combes 2015] permet d'aborder ce pro-

Algorithme 8.2 : Exp3.M

Initialisation : $p_1 = (\frac{m}{K}, \ldots, \frac{m}{K}) \in \mathbb{R}^K$

pour *Chaque instant* $t \geq 1$ **faire**

- Construire $A_t \subset \{1, \ldots, K\}$ aléatoirement selon $\mathbb{P}[i \in A_t] = p_{i,t}, \forall i = 1, \ldots, K$
- Recommander A_t à l'utilisateur et recevoir la récompense $X_{i,t}, \forall i \in A_t$
- construire $q_{t+1} \in \mathbb{R}^K$ comme

$$q_{i,t+1} = m \frac{p_{i,t} \ \exp(\eta \tilde{X}_{i,t})}{\sum_{1 \leq j < K} p_{j,t} \ \exp(\eta \tilde{X}_{j,t})} \quad (1)$$

  où $\tilde{X}_{i,t} = \frac{X_{i,t}}{p_{i,t}} \mathbb{1}_{i \in A_t}$
- mise à jour : construire $p_{t+1} \in \mathbb{R}^K$ comme

$$p_{i,t+1} = \min\{C q_{i,t+1}, 1\} \quad (2)$$

  avec C tel que $\sum_{i=1}^{K} \min\{C q_{i,t+1}, 1\} = m$

fin

blème en définissant une position l dans la liste comme position d'exploration. A chaque instant une liste de k objets est construite. Elle est ordonnée par ordre décroissant en fonction de la moyenne des récompenses de chaque objet. Ensuite à la position l, avec une probabilité $1/2$, un nouvel objet remplace l'objet actuellement en position l. Il est choisi aléatoirement parmi l'ensemble des objets qui possède une borne supérieure de confiance KL-UCB plus grande que la moyenne des récompenses de l'objet actuellement en position l.

8.2 Contribution : Adaptation et application de la stratégie *EXP3.M* à la recommandation

8.2.1 Stratégie EXP3.M et son implémentation

La stratégie *Exp3.M* (algorithme 8.2) est une stratégie récente de bandit conçue spécialement pour aborder le problème du tirage multiple. Il a été initialement proposé et étudier théoriquement en 2010 par Uchiya, Nakamura et Kudo [Uchiya 2010]. Plus récemment, en 2012, Bubeck et Cesa-Bianchi [Bubeck 2012] ont analysé une généralisation de cette stratégie pour une classe complexe de problèmes de bandit : les bandits combinatoires. Dans la suite de cette section, nous décrivons la stratégie *Exp3.M* et l'adaptation que nous proposons de certains composants dans le but de permettre une

implémentation efficace de cette stratégie pour la recommandation.

8.2.1.1 Intuitions mathématiques

La stratégie *Exp3.M* (décrite dans l'algorithme 8.2) repose sur le principe de *descente miroir*, une méthodologie générale pour minimiser le regret en optimisation convexe en ligne. Plus précisément, la stratégie *Exp3.M* peut être vue comme un cas particulier de l'algorithme *descente miroir convexe en ligne* [1] utilisant la fonction d'*entropie négative* $F(p) = \sum_{i=1}^{K}(p_i \log(p_i) - p_i)$ pour l'ensemble convexe $\mathcal{C}_m = \{p \in [0,1]^K : \sum_i p_i = m\}$. Effectivement, les étapes (1) et (2) de l'algorithme *Exp3.M* sont équivalentes aux deux mises à jour de descente miroir convexe en ligne :

$$\nabla F(q_{t+1}) = \nabla F(p_t) + \eta \tilde{X}_t \tag{8.1}$$

$$p_{t+1} = \operatorname*{argmin}_{p \in \mathcal{C}_m} D_F(p, q_{t+1}) \ , \tag{8.2}$$

où $D_F(p,q) = \sum_{i=1}^{K}(p_i \log(p_i/q_i) + q_i - p_i)$ est la divergence de Bregman associée à la fonction F. En d'autres termes, pour construire le nouveau vecteur p_{t+1}, la stratégie *Exp3.M* commence à p_t et se déplace selon la direction du vecteur de récompenses estimée $\tilde{X}_t = (\tilde{X}_{i,t})_{1 \leq i \leq K}$, et se projette finalement sur l'ensemble convexe $\mathcal{C}_m$.

8.2.1.2 Implémentation efficace

Dans la stratégie *Exp3.M*, deux points nécessitent une attention particulière dans la manière dont ils sont traités afin de minimiser le nombre d'opérations nécessaires à chaque instant :

- Il existe plusieurs solutions pour construire l'ensemble $A_t \subset \{1, \ldots, K\}$ aléatoirement selon $\mathbb{P}[i \in A_t] = p_{i,t}$, $\forall i = 1, \ldots, K$. Une approche naïve consiste à remarquer que $p_t \in \mathcal{C}_m$ peut être décomposé comme une combinaison convexe de la forme $\sum_A \alpha_A \mathbb{1}_A$ (où les sous-ensembles $A \subset \{1, \ldots, K\}$ sont de cardinalité m), c'est donc suffisant pour construire A aléatoirement selon α_A. Malheureusement, la complexité de calcul de cette étape en utilisant cette approche est de l'ordre de $\binom{K}{m}$, soit l'ensemble des combinaison de m éléments parmi K. Nous proposons d'utiliser la fonction *Dependent Rounding (algorithme 8.3)* conçue par Gandhi et al. [Gandhi 2006] et Uchiya, Nakamura, et Kudo [Uchiya 2010] dont la complexité de calcul est de l'ordre de $\mathcal{O}(K)$.

1. décrit dans l'article de Bubeck et Cesa-Bianchi [Bubeck 2012] dans les sections 5.3 et 5.4.

Algorithme 8.3 : Dependent Rounding

Entrée : Vecteur $p \in \mathbb{R}^K$ de probabilités avec $\sum_{i=1}^{K} p_i = m$
Sortie : sous-ensemble de m objets
tant que *il existe un* i *tel que* $0 < p_i < 1$ **faire**
- Choisir i, j distincts, avec $0 < p_i < 1, 0 < p_j < 1$
- Fixer $\alpha = \min\{1 - p_i, p_j\}$ et $\beta = \min\{p_i, 1 - p_j\}$
- Mettre à jour p_i et p_j comme

$$(p_i, p_j) = \begin{cases} (p_i + \alpha, p_j - \alpha) \text{ avec probabilité } \frac{\beta}{\alpha+\beta} \\ (p_i - \beta, p_j + \beta) \text{ avec probabilité } \frac{\alpha}{\alpha+\beta} \end{cases}$$

fin
Retourner $\{i : p_i = 1, 1 \leq i \leq K\}$

— Le second problème est comment implémenter la constante numérique C définie implicitement par l'équation $\sum_{i=1}^{K} \min\{Cq_{i,t+1}, 1\} = m$. Il se trouve que, après le tri décroissant des composants $q_{i,t+1}$, $i = 1, \ldots, K$, ce problème se résume à résoudre successivement des équations linéaires à une variable de la forme :

$$k + \sum_{i=k+1}^{K} Cv_i = m \ , \quad \text{avec } C \in [1/v_k, 1/v_{k+1}] \tag{8.3}$$

où $v_1 \geq v_1 \geq \ldots \geq v_K$ sont les valeurs $q_{i,t+1}$ triées dans l'ordre croissant. La complexité de calcul globale associée est de l'ordre de $\mathcal{O}(K \log K)$.

8.2.1.3 Garanties théoriques

Uchiya, Nakamura, et Kudo [Uchiya 2010] ont prouvé que la stratégie *Exp3.M* obtient un regret de l'ordre de $\mathcal{O}(\sqrt{mTKlog(K/m)}))$ si les récompenses sont définies comme $Z_t = \sum_{i \in A^t} X_{i,t}$ (nombre total de clics). Cette borne de regret améliore de $\sqrt{m}$ la borne obtenue avec l'approche *IBA* utilisant m instances de la stratégie à tirages simples *Exp3*.

8.2.2 Évaluation en utilisant deux jeux de données de références

Pour évaluer les performances en terme d'abandon de la stratégie *Exp3.M* et les comparer aux approches de l'état de l'art, nous avons mené plusieurs expérimentations avec les jeux de données Movielens-100 et Jester qui sont utilisés dans l'article de Kohli et al. [Kohli 2013].

8.2.2.1 Cadre expérimental

Les jeux de données utilisés et la manière dont l'expérimentation est conduite sont décrits dans la section 6.3.

Afin de comparer efficacement entre les performances de la stratégie Exp3.M et celles des approches existantes *RBA* et *IBA*, nous avons choisi d'implémenter les approches existantes en utilisant les stratégies de bandit *Exp3* [Auer 2002b] et ϵ-*greedy* [Auer 2002b].

Avec la stratégie *Exp3*, une probabilité de $1/K$ est allouée à chaque objet. Selon ces probabilités, un objet est recommandé aléatoirement. Les probabilités sont ensuite mises à jour selon les retours utilisateurs et un paramètre η. Nous avons choisi d'utiliser la stratégie *Exp3* car la stratégie *Exp3.M* en est une variante. La valeur du paramètre η a besoin d'être réglé, de manière indépendante pour *RBA* et *IBA*. Pour cela nous avons utilisé une grille de valeurs 2^i avec $i \in [-7, -6, \ldots, 0]$ et nous avons choisi la valeur qui réalise le meilleur compromis entre la vitesse d'apprentissage et la valeur moyenne d'abandon à la fin de l'expérimentation. La valeur retenue pour η est 2^{-6} pour les deux approches.

La seconde stratégie ϵ-*greedy* ajoute une part aléatoire dans le choix de l'objet recommandé. A chaque instant t, un objet tiré selon une loi uniforme est recommandé avec une probabilité ϵ, sinon, avec une probabilité $(1 - \epsilon)$, l'objet avec le meilleur taux de clics est recommandé. Cette stratégie est la stratégie utilisée avec *RBA* et *IBA* qui donne les meilleurs résultats dans l'article de Kohli et al. [Kohli 2013]. Dans cet article, les auteurs indiquent que la valeur $\epsilon = 0.05$ donne les meilleurs résultats durant leurs tests initiaux.

La stratégie *Exp3.M* nécessite également d'être paramétrée. En utilisant la même méthode que pour la stratégie *Exp3*, la valeur 2^{-5} a été retenue pour η.

Nos résultats expérimentaux sont montrés dans les figures 8.1, 8.2 et 8.3.

Pour rappel, dans ces figures, la combinaison optimale et la combinaison indépendante sont représentées par les courbes horizontales. Dans les trois expérimentations mises en place, la combinaison diversifiée est la même que la combinaison optimale. la combinaison indépendante est constituée des m objets les plus cliqués. La combinaison diversifiée quant à elle est constituée de l'objet le plus populaire en première position et des objets les plus populaires lorsque les objets aux positions précédentes ne sont pas cliqués.

L'approche *RBA* vise la solution optimale, tandis que l'approche *IBA* et la stratégie *Exp3.M* visent la solution indépendante. Les différences de performance en terme d'abandon entre la solution indépendante et la solution optimale est d'environ 1% pour

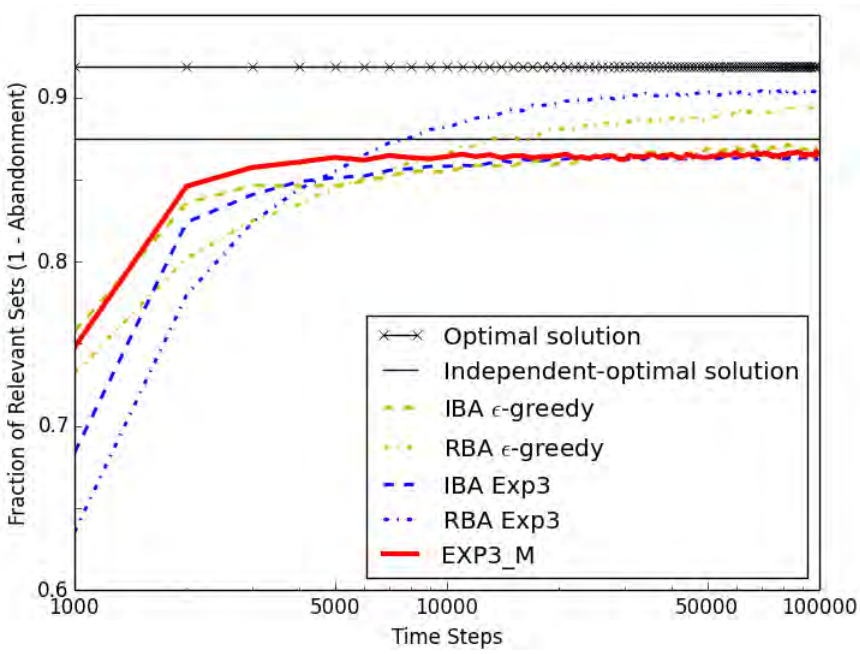

FIGURE 8.1 – Jeu de données MovieLens avec un seuil de pertinence = 2. (1 - la proportion d'abandon) en fonction du nombre d'interaction avec le système.

les figures 8.2 et 8.3. Cependant, dans les cas présentés dans la figure 8.1, cette différence est plus importante, avec 4% d'abandon en plus.

8.2.2.2 Résultats

Comparons tout d'abord la stratégie *Exp3.M* avec les méthodes de l'état de l'art. La stratégie *Exp3.M* converge vers des proportions d'abandon équivalentes à celles obtenues en utilisant l'approche *IBA-ϵ-greedy*, la méthode de l'état de l'art avec la vitesse d'apprentissage la plus rapide. De plus, la stratégie *Exp3.M* apprend plus vite. Pour mettre en évidence cette amélioration, nous utilisons un test unilatéral de Wilcoxon (Voir Table 8.1). La stratégie *Exp3.M* permet d'apprendre de 1.5 à 12 fois plus rapidement que les méthodes de l'état de l'art.

Pour confirmer cette amélioration de la vitesse d'apprentissage, nous utilisons la même approche que dans le chapitre précédent : nous comparons le nombre d'étapes nécessaire pour atteindre 95% et 98% de la combinaison indépendante en utilisant les méthodes de l'état de l'art avec les stratégies de bandit *ϵ-greedy* et *Exp3* et cette même quantité en utilisant la stratégie *Exp3.M* (Voir Table 8.2).

Nous comparons maintenant les performances des approches *RBA* et *IBA* en utili-

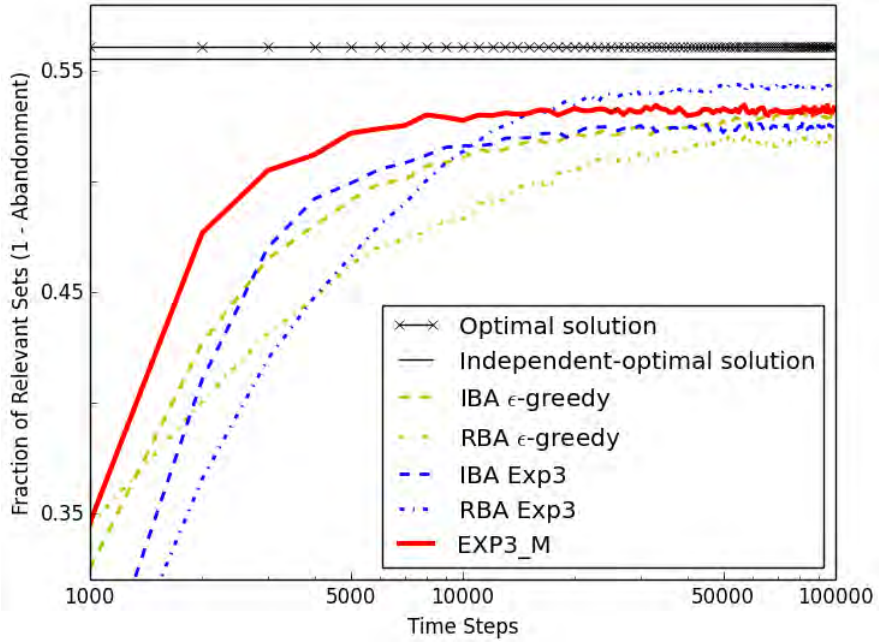

FIGURE 8.2 – Jeu de données MovieLens avec un seuil de pertinence = 4. (1 - la proportion d'abandon) en fonction du nombre d'interaction avec le système.

sant la stratégie ϵ-*greedy*, comme implémenté dans l'article de Kohli et al. [Kohli 2013], avec notre implémentation utilisant la stratégie *Exp3*. L'approche *RBA-Exp3* a une proportion (1 - abandon) meilleure que *RBA-Egreedy* mais seulement après 4 000 itérations pour Movielens et 10 000 itérations pour Jester. Concernant les approches *IBA-Exp3* et les approches *IBA-*ϵ*-greedy*, les performances sont très proches (voir Figures 8.1, 8.2 et 8.3).

Sur le long terme, l'approche *RBA-Exp3* est toujours la meilleure, car elle vise la solution optimale, au contraire des autres approches qui visent la solution indépendante. Sur la Figure 8.1, l'approche *RBA-Exp3* dépasse la solution indépendante après 8 000 itérations et après 50000 itérations pour les résultats de la Figure 8.3. Sur la Figure 8.2, l'approche *RBA-Exp3* ne dépasse pas la solution indépendante, mais c'est la seule approche qui atteint 98% de la solution indépendante. Néanmoins sa vitesse d'apprentissage est plus faible que les implémentations de l'approche *IBA* et de la stratégie *Exp3.M*.

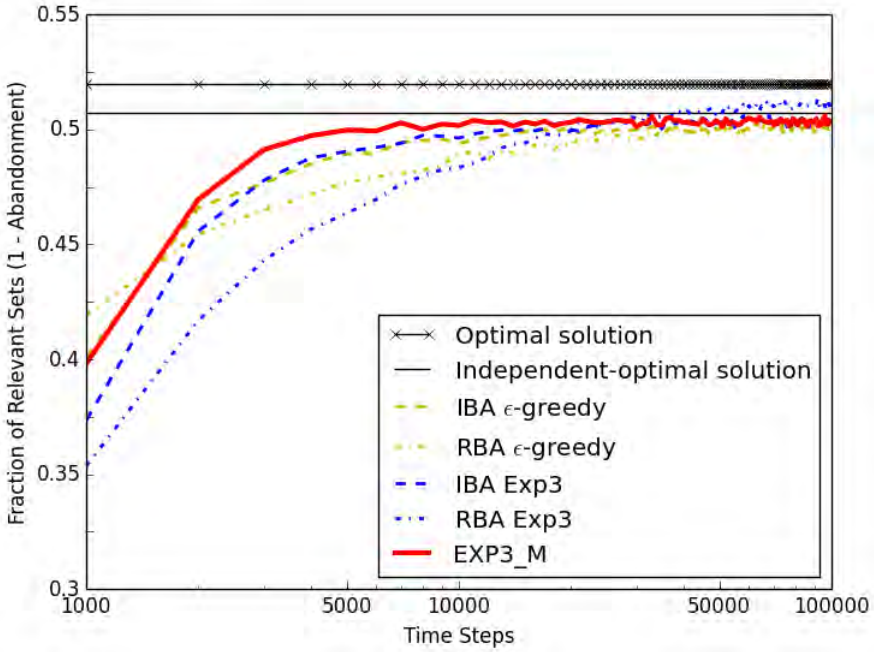

FIGURE 8.3 – Jeu de données Jester avec un seuil de pertinence = 7. (1 - la proportion d'abandon) en fonction du nombre d'interaction avec le système.

Conclusion

Dans ce chapitre, nous avons présenté une stratégie de bandit conçue spécifiquement pour recommander plusieurs objets à chaque instant : *Exp3.M*. Les approches de l'état de l'art utilisent autant d'instances d'une stratégie de bandit qu'il y a de recommandations à effectuer. En gérant l'ensemble des recommandations simultanément, la stratégie *Exp3.M* converge vers des valeurs d'abandon équivalentes et apprend plus vite que l'approche *IBA* de l'état de l'art qui a les meilleures performances en terme de vitesse d'apprentissage. Nous avons également montré que l'approche *RBA* implémentée en utilisant la stratégie de bandit *Exp3* peut être plus intéressante sur le long terme.

la stratégie *Exp3.M* est conçue pour maximiser la somme des clics. Lorsque la minimisation de l'abandon est attendue, la stratégie *Exp3.M* vise une solution sous-optimale.

Dans le chapitre suivant, nous proposons une nouvelle approche utilisant des stratégies de bandit à tirages simples et des probabilités conditionnelles pour apporter de la diversité dans la liste de résultats. Nous montrons que l'apport de diversité permet d'obtenir des proportions d'abandon plus faibles sur le long terme, pour des vitesses d'apprentissage plus rapides que les approches de l'état de l'art.

TABLE 8.1 – Résultats du Test de Wilcoxon unilatéral pour comparer le nombre d'étapes nécessaire pour atteindre 95% et 98% de la solution indépendante en utilisant les méthodes de l'état de l'art avec les stratégies de bandit ϵ-greedy et *Exp3* et cette même quantité en utilisant la stratégie *Exp3.M*. Hypothèse alternative : la stratégie *Exp3.M* apprend plus vite. L'hypothèse alternative est retenue pour l'ensemble des comparaisons, hormis la comparaison avec l'approche IBA utilisant la stratégie ϵ-greedy

		95 %	**98 %**
collection	Approche	p-value	p-value
MovieLens threshold 2	**IBA-Egreedy**	0.054	0.0012
	IBA-Exp3	4.2e-09	<2.2e-16
	RBA-Egreedy	2.9e-13	2.9e-15
	RBA-Exp3	<2.2e-16	<2.2e-16
MovieLens threshold 4	**IBA-Egreedy**	1.0e-08	0.0016
	IBA-Exp3	9.4e-15	7.6e-07
	RBA-Egreedy	<2.2e-16	1.0e-10
	RBA-Exp3	<2.2e-16	1.1e-07
Jester threshold 7	**IBA-Egreedy**	8.8e-06	8.7e-09
	IBA-Exp3	2.4e-14	1.9e-14
	RBA-Egreedy	<2.2e-16	<2.2e-16
	RBA-Exp3	<2.2e-16	<2.2e-16

TABLE 8.2 – Ratios approximatifs entre le nombre d'étapes nécessaire pour atteindre 95% et 98% de la combinaison indépendante en utilisant les méthodes de l'état de l'art avec les stratégies de bandit ϵ-*greedy* et *Exp3* et cette même quantité en utilisant la stratégie *Exp3.M*. (— : aucune des approches n'atteint le seuil). L'approche *Exp3.M* atteint les différents seuils au moins aussi rapidement que les approches comparées. Cependant, dans certains cas aucune des approches n'atteint le seuil.

collection	approach	ratio (95%)	ratio (98%)
MovieLens	**IBA-Egreedy**	1	3
	IBA-Exp3	1.5	2
threshold 2	**RBA-Egreedy**	2	2
	RBA-Exp3	2	1
MovieLens	**IBA-Egreedy**	7	—
	IBA-Exp3	—	—
threshold 4	**RBA-Egreedy**	—	—
	RBA-Exp3	2	—
Jester	**IBA-Egreedy**	1.5	3
	IBA-Exp3	1.5	2
threshold 7	**RBA-Egreedy**	2.5	6
	RBA-Exp3	3	4

Chapitre 9

Diversité et recommandation à tirages multiples

Sommaire

Résumé.

En apportant de la diversité dans les listes d'objets recommandées aux utilisateurs, les deux stratégies que nous proposons permettent d'obtenir des vitesses d'apprentissage jusqu'à dix fois plus rapides que les approches de l'état de l'art, tout en obtenant des proportions d'abandon de 1,5% à 6% plus faibles. Des proportions d'abandon plus faibles peuvent impliquer une augmentation du revenu du système de recommandation.

L'apport de diversité dans les listes d'objets recommandées aux utilisateurs représente une problématique importante de la recommandation. Lors de la construction d'une liste, la prise en compte de la proximité entre les différents objets peut permettre d'apporter de la diversité. Des proportions d'abandon plus faibles peuvent être obtenues et une augmentation du revenu du système de recommandation peut en résulter.

Dans ce chapitre nous présentons tout d'abord comment les approches existantes tentent d'apporter de la diversité dans la liste de résultats. Nous montrons que ces approches possèdent quelques inconvénients dont le principal est la faible vitesse d'apprentissage. Nous proposons ensuite deux stratégies basées sur le calcul de probabilités conditionnelles afin d'apporter de la diversité. Nous évaluons ensuite ces approches à l'aide de simulations.

9.1 Diversité et recommandation

9.1.1 Enjeux

Le chapitre 6 décrit en détail la problématique de la diversité.

Pour minimiser la proportion d'abandon, nous proposons de diversifier la liste des recommandations effectuées. Pour cela, nous voulons maximiser la probabilité qu'au moins un objet satisfera l'utilisateur, comme dans l'article [Candillier 2012]. Reprenons l'exemple d'un SR proposant trois films à chaque instant. Nous supposons que les films les plus populaires sont ceux de la trilogie "Retour vers le Futur". Lorsque le système recommande trois films, l'idéal est que le plus d'utilisateurs clique sur au moins un film. Si le système recommande les trois films les plus populaires, un utilisateur qui n'apprécie pas la trilogie "Retour vers le Futur" ne cliquera sur aucun des films. Pour cet utilisateur, il serait plus approprié de recommander une liste de trois autres films. Globalement, une solution intermédiaire capable de satisfaire une majorité des utilisateurs est préférable : le système devrait recommander un film de la trilogie et diversifier les autres films de la liste.

9.1.2 Solutions sous-optimales

Comme présenté plus tôt dans ce manuscrit, trouver une combinaison optimale A^* est un problème NP-complet [Radlinski 2008]. C'est pourquoi une combinaison sous-optimale mais plus facile à atteindre semble plus appropriée au contexte temps réel. Kohli et al. [Kohli 2013] utilisent deux autres combinaisons : la combinaison indépendante et la combinaison diversifiée (définies dans la section 7.1.2)

Pour rappel, La combinaison indépendante est définie comme la combinaison des m objets les plus cliqués :

$$A^{\text{indépendante}} = \underset{A \in P_m^K}{\operatorname{argmax}} \sum_{i \in A} E[X_i] \tag{9.1}$$

et la combinaison k-diversifiée est composée de l'objet le plus populaire en première position, et ensuite les objets les plus populaires lorsque les objets aux positions précédentes ne sont pas cliqués :

$$A_1^{\text{k-div}} = \underset{i \in K}{\operatorname{argmax}} E[X_i] \quad \text{et} \tag{9.2}$$

$$A_l^{\text{k-div}} = \underset{i \in K / \left\{A_{1,\dots,l-1}^{\text{k-div}}\right\}}{\operatorname{argmax}} E\left[X_i | X_j = 0 \ \ \forall j \in \left\{A_{1,\dots,l-1}^{\text{k-div}}\right\}\right]. \tag{9.3}$$

Cette solution est celle visée par les approches *RBA*, *MPB* et *k-diversity bandit*. Cette dernière est présentée dans la suite de ce chapitre.

La combinaison indépendante $A^{\text{indépendante}}$ est considérée comme moins bonne que la combinaison k-diversifiée $A^{\text{k-div}}$ lorsque le but est la minimisation du nombre d'abandon [Radlinski 2008].

Nous proposons également une solution intermédiaire : la combinaison 2-diversifiée $A^{\text{2-div}}$, qui tente d'apporter de la diversité dans la liste, tout en continuant de recommander les objets les plus cliqués. Cette solution est visée par l'approche proposée dans la suite de ce chapitre *2-Diversified Bandit Algorithm*. A chaque position l de la liste, uniquement l'objet recommandé à la position $l-1$ est pris en compte. Cette solution est la combinaison composée de l'objet le plus cliqué en première position et, aux positions suivantes, les objets les plus cliqués lorsque l'objet recommandé à la position précédente n'est pas cliqué. Plus formellement, la combinaison $A^{\text{2-div}}$ est définie comme :

$$A_1^{\text{2-div}} = \underset{i \in K}{\operatorname{argmax}} E[X_i] \quad \text{et} \tag{9.4}$$

$$A_l^{\text{2-div}} = \underset{i \in K / \left\{A_{1,\dots,l-1}^{\text{2-div}}\right\}}{\operatorname{argmax}} E\left[X_i | X_{A_{l-1}^{\text{2-div}}} = 0\right] \tag{9.5}$$

Cette solution prend en compte la notion de diversité, mais seulement entre l'objet recommandé en position l et celui recommandé à la position $l-1$. En reprenant l'exemple de la trilogie "Retour vers le futur", la liste recommandée pourrait contenir un épisode de la trilogie en première position et en troisième position.

9.2 Adaptation des stratégies de bandit à tirages simples pour apporter de la diversité

9.2.1 Comment les approches existantes tentent de diversifier les listes de résultats ?

La stratégie *Exp3.M* présentée dans le chapitre précédent, et l'approche *IBA* présentée dans le chapitre 7 sont toutes deux conçues pour recommander plusieurs objets à chaque instant, mais ont pour objectif de maximiser le taux de clics, et non la proportion d'abandon.

Les approches *MPB* et *RBA* tentent d'apporter de la diversité en prenant en compte uniquement le premier clic de l'utilisateur (voir sections 7.2.1 et 7.3).

9.2.2 Contribution : utilisation des probabilités conditionnelles au sein de la liste pour la diversifier

9.2.2.1 2-Diversified Bandit Algorithm (2-DBA)

Considérer la corrélation entre des clics sur plusieurs objets au sein d'une même liste de recommandation à un instant t peut permettre de diversifier la liste des résultats. Cela est possible puisque que nous pouvons recommander des objets qui sont rarement cliqués simultanément.

La solution optimale A^* peut être obtenue en utilisant l'ensemble des combinaisons de k objets avec leurs récompenses respectives. Cela implique d'utiliser une matrice de dimension k et de trouver la combinaison qui minimise la proportion d'abandon. Si le nombre d'objets disponibles est relativement grand, une telle matrice est difficile à stocker. Pour 100 objets par exemple, cela revient à stocker une matrices comportant 10 milliards d'éléments. Pour rappel, *Amazon* possède un catalogue de 183 millions d'objets. Afin de réduire cette contrainte forte, nous proposons de considérer uniquement les clics simultanés de chaque couple d'objets en utilisant deux matrices de dimension 2. La première contient le nombre de clics simultanés, et la seconde le nombre de fois où chaque couple d'objets a été recommandé au sein d'une même liste. Ainsi, il est possible de quantifier la popularité de chaque objet conditionnellement au fait qu'un objet donné ne soit pas cliqué. Autrement dit, nous pouvons connaître l'objet qui à la plus forte probabilité d'être cliqué lorsqu'un objet donné n'est pas cliqué. Cependant, des informations pouvant contribuer à la diversité individuelle au sein de la liste ne sont pas considérées, par exemple nous ne savons pas exactement si trois objets sont

Algorithme 9.1 : 2-Diversified Bandit Algorithm (2-DBA)

K : Nombre d'objets disponibles
k : Nombre d'objets à recommander à chaque instant
T : Horizon
BTS : instance d'une stratégie de bandit à tirages simples
SC : Matrice de clics simultanés
N : Matrice de recommandations simultanées
pour $t = 1, \ldots, T$ **faire**
    a(1)$\leftarrow$ SelectionnerObjet(BTS, K)
    $A_{t,1} \leftarrow a(1)$
    pour $l = 2, \ldots, m$ **faire**
        $a(l) \leftarrow$ SelectionnerObjet(BTS, $K \backslash A_{t,l-1}$) en utilisant $\hat{p}_i^l(t)$ $A_{t,l} \leftarrow a(l)$
    fin
    $A_t \leftarrow \cup_l a_l$
    Recommander A_t à l'utilisateur, récupérer le retour utilisateur X_t
    pour $l = 1, \ldots, m$ **faire**
        Retour utilisateur :

$$z_l = \begin{cases} 1 & \text{si } a_i \text{ est cliqué} \\ 0 & \text{sinon} \end{cases}$$

        MiseAJour(BTS, z_i)
    fin
    Mettre à jour $SC_{i,j}$ et $N_{i,j}$
fin

cliqués au sein d'une même liste.

Quelques notations supplémentaires sont nécessaires pour décrire cette approche. a^l avec $i \in (1, ..., k)$ est l'objet recommandé à la position l. $SC_{i,j}$ est le nombre de clics simultanés pour les objets i et j, et $N_{i,j}$ est le nombre de fois où les objets i et j sont recommandés au sein d'une même liste.

A chaque position l d'une liste donnée, l'approche 2-DBA calcule les estimateurs de chaque objet en utilisant une stratégie de bandit. Ces estimateurs sont calculés en utilisant les probabilités conditionnelles de clic sur un objet i sachant que i^{l-1} n'est pas cliqué. La probabilité $\hat{p}_i(t)$ que l'objet i soit cliqué est définie comme :

$$\hat{p}_i(t) = \frac{1}{N_i(t)} \sum_{t\,:\,i \in A_t} X_{i,t} \quad \text{avec} \quad N_i(t) = \sum_{t\,:\,i \in A_t} 1 \tag{9.6}$$

$\mathbb{P}(i|\overline{i^{l-1}})$ est la probabilité que l'objet i soit cliqué conditionnellement au fait que l'objet i^{l-1} n'est pas cliqué. Elle est calculée comme suit :

$$\mathbb{P}(i|\ \overline{i^{l-1}}) = \frac{\mathbb{P}(i \cap \overline{i^{l-1}})}{\mathbb{P}(\overline{i^{l-1}})} = \frac{\frac{N_{i,i^{l-1}} - SC_{i,i^{l-1}}}{N_{i,i^{l-1}}}}{1 - \hat{p}_{i^{l-1}}(t)} \tag{9.7}$$

$\hat{p}_i^l(t)$ est le produit de la probabilité $\mathbb{P}(i|\overline{i^{l-1}})$ et la probabilité $\hat{p}_i(t)$. Il prend en compte à la fois la popularité de l'objet et la corrélation de cet objet avec l'objet recommandé à la position précédente.

$$\hat{p}_i^l(t) = \hat{p}_i(t) \times \mathbb{P}(i|\ \overline{i^{l-1}}) \tag{9.8}$$

Pour minimiser la proportion d'abandon un clic est suffisant pour recevoir la récompense globale maximale de 1. A la position l, nous devons choisir l'objet à recommander parmi les $K - l + 1$ restants. Un objet i parmi les objets restants et l'objet i^{l-1} sont souvent cliqués simultanément lorsqu'ils sont recommandés au sein d'une même liste. Si l'utilisateur ne clique pas sur i^{l-1}, il y a peu de chance que l'utilisateur clique sur i. Donc recommander un objet qui est rarement cliqué en même temps que i^{l-1} semble plus favorable à la minimisation de la proportion de l'abandon. C'est ce qui est mesuré par la probabilité $\mathbb{P}(i|\overline{i^{l-1}})$. De cette manière les objets qui ont une forte probabilité $\mathbb{P}(i|\overline{i^{l-1}})$ doivent être mis en avant, tandis que les objets avec une faible probabilité doivent être pénalisés.

L'approche 2-DBA (voir Algorithme 9.1) que nous proposons recommande en première position l'objet le plus cliqué par le passé. Aux positions suivantes, les objets les plus cliqués conditionnellement au fait que l'objet recommandé à la position précédente ne soit pas cliqué sont recommandés. Ainsi, l'objet recommandé en seconde position est celui qui est le plus cliqué lorsque l'objet le plus populaire ne l'est pas. Mais il est important de souligner qu'aucune garantie n'est donnée sur le fait que les objets recommandés en première position et en troisième position ne soient pas cliqués en même temps. En reprenant l'exemple de la trilogie "Retour vers le futur", la liste recommandée pourrait contenir un épisode de la trilogie en première position et en troisième position. Si deux groupes de films sont plus cliqués que les autres, mais en même temps rarement cliqués simultanément, les recommandations tendront à recommander une liste composée des films de ces deux groupes. Dans le cadre de la minimisation de l'abandon, une solution composée d'un film de k groupes de films semble plus adaptée. Pour approcher cette solution, nous généralisons l'approche *2-DBA* dans la sous-section suivante.

9.2.2.2 k-Diversified Bandit Algorithm (k-DBA)

L'approche k-DBA est une généralisation de l'approche *2-DBA*, présentée dans la sous-section précédente. Son but est d'apporter plus de diversité dans la liste d'objets recommandée. L'approche *2-DBA* choisit un objet à recommander à la position l conditionnellement au fait que l'objet recommandé à la position $l-1$ ne soit pas cliqué. L'approche *k-DBA* prend en compte l'ensemble des objets recommandés aux positions précédentes. L'objectif de cette approche est de recommander un objet à la position l conditionnellement au fait que l'ensemble des objets recommandés aux $l-1$ premières positions ne soient pas cliqués.

$\hat{p}_i^l(t)$ est la probabilité d'obtenir un clic pour l'objet i à la position l, conditionnellement au fait que tous les objets recommandés aux positions précédentes ne soient pas cliqués. Lorsque l = 2, $p_i^2(t)$ peut être définie de la même manière que pour l'approche *2-DBA* :

$$\hat{p}_i^2(t) = \hat{p}_i(t) \times \mathbb{P}(i|\ \overline{i^1}) \tag{9.9}$$

Mais dans le cas où $l > 2$, il est plus difficile de définir $p_i^l(t)$ principalement à cause de l'aspect combinatoire de la recommandation à tirages multiples. Par exemple pour calculer $\hat{p}_i^3(t)$, l'idéal est de pouvoir prendre en compte $\mathbb{P}(i|\overline{i^1}, \overline{i^2})$, la probabilité que l'objet i soit cliqué sachant que les objets i^1 et i^2 ne sont pas cliqués. La formule de la probabilité $\mathbb{P}(i|\overline{i^1}, \overline{i^2})$ est :

$$\mathbb{P}(i|\ \overline{i^1}, \overline{i^2}) = \frac{\mathbb{P}(i \cap \overline{i^1} \cap \overline{i^2})}{\mathbb{P}(\overline{i^1}|\ \overline{i^2}) \times \mathbb{P}(\overline{i^2})} \tag{9.10}$$

Malheureusement, les informations à disposition dans les matrices SC et N ne nous permettent pas de calculer $\mathbb{P}(i \cap \overline{i^1} \cap \overline{i^2})$. $\mathbb{P}(i \cap \overline{i^1})$, $\mathbb{P}(i \cap \overline{i^2})$ et $\mathbb{P}(\overline{i^1} \cap \overline{i^2})$ sont connues, mais aucune information sur le fait que les trois objets sont cliqués au sein d'une même liste n'est disponible.

Afin de réduire cette limitation, nous proposons de multiplier $\hat{p}_i(t)$ par le produit de l'ensemble des probabilités conditionnelles de i sachant qu'un objet recommandé à l'une des positions précédentes n'est pas cliqué. Dans ce cas, si l'objet i est souvent cliqué en même temps que l'un des objets recommandés aux positions précédentes, il est pénalisé. Cette pénalisation est d'autant plus importante si cet objet est souvent cliqué avec plusieurs objets déjà présents dans la liste. Formellement, $\hat{p}_i^l(t)$ est définie comme :

$$\hat{p}_i^l(t) = \hat{p}_i(t) \times \prod_{s=1}^{l-1} \mathbb{P}(i|\ \overline{i^s}) \tag{9.11}$$

TABLE 9.1 – Ratios approximatifs entre le nombre d'étapes nécessaire pour atteindre 95 % et 98 % et 100 % de la combinaison indépendante en utilisant les approches *RBA*, *MPB* et *2-DBA* et cette même quantité en utilisant l'approche *k-DBA*. (+) : l'approche *MPB* atteint le seuil, mais pas l'approche comparée, (-) : aucune des deux approches n'atteint le seuil.

Jeu de données	Approche	$threshold$		
		95%	98%	100%
MovieLens	**2-DBA**	1	1	3.5
	RBA	2.5	3	3.5
seuil 2	**MPB**	1	11	(+)
MovieLens	**2-DBA**	1	3	(-)
	RBA	(+)	(+)	(-)
seuil 4	**MPB**	10.1	(+)	(-)
Jester	**2-DBA**	0.5	1.5	(+)
	RBA	(+)	(+)	(+)
seuil 7	**MPB**	10.1	(+)	(+)

9.2.3 Évaluation en utilisant deux jeux de données de références : Jester et Movielens

9.2.3.1 Cadre expérimental

Les jeux de données utilisés et la manière dont l'expérimentation est conduite sont décrits dans la section 6.3.

Afin de pouvoir comparer efficacement les approches présentées dans ce chapitre, *2-DBA*, *k-DBA* et les approches de l'état de l'art (*RBA* et *MPB*), il est important de les implémenter en utilisant la même stratégie de bandit. Nous avons choisi la stratégie ϵ-*greedy* car elle est utilisée dans l'article définissant le cadre expérimental utilisé [Kohli 2013]. Les autres stratégies de bandit à tirages simples peuvent également être utilisées (UCB, Exp3, Thompson Sampling, ...).

La stratégie ϵ-*greedy* [Watkins 1989] est décrite en détail dans la section 2.2.1. Dans l'article [Kohli 2013], les auteurs indiquent que la valeur $\epsilon = 0.05$ donnent les meilleurs résultats durant les tests initiaux. Cette valeur est donc utilisée dans nos expérimentations.

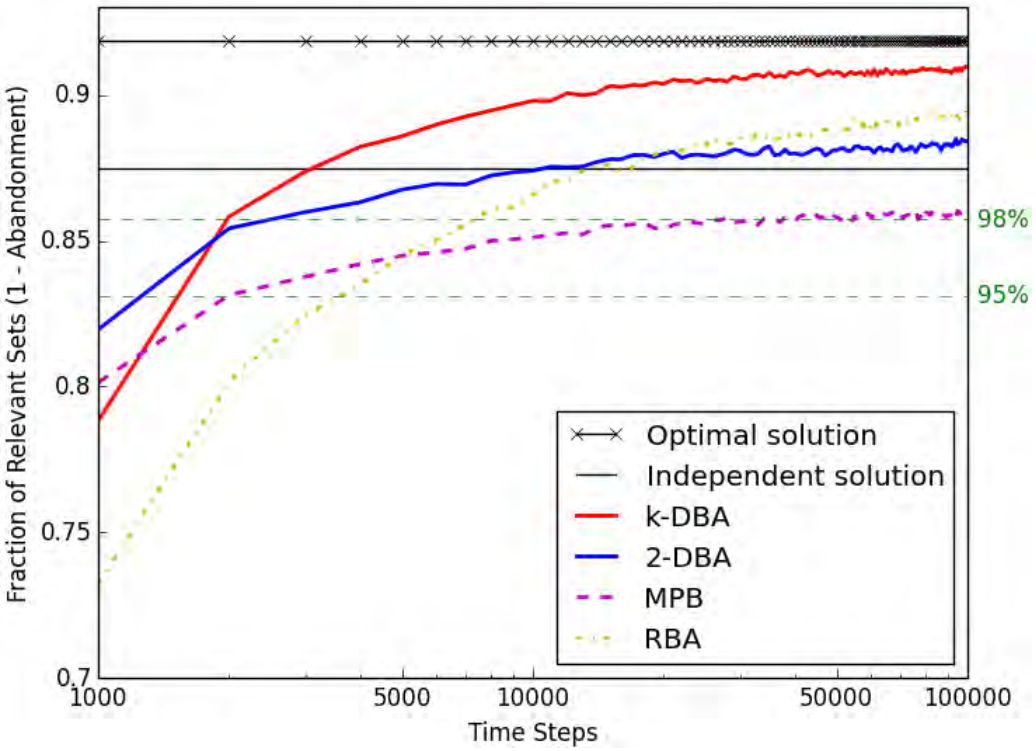

FIGURE 9.1 – Comparaison des différentes approches avec le jeu de données Movielens avec un seuil de pertinence = 2. (1 - la proportion d'abandon) en fonction du nombre d'interaction avec le système.

9.2.3.2 Résultats et Discussion

Les résultats expérimentaux sont présentés dans les Figures 9.1, 9.2 et 9.3. Dans ces figures, nous avons tracé la solution optimale, celle qui minimise la proportion d'abandon, avec une ligne horizontale contenant des croix, et la solution indépendante, celle qui maximise le taux de clics, avec une ligne solide horizontale. Dans nos expérimentations, la solution optimale et la solution k-diversifiée sont composées de la même combinaison de cinq objets. Sur la Figure 9.1, la solution optimale permet d'obtenir une proportion d'abandon $\approx 4\%$ plus faible que la solution indépendante ($\approx 1\%$ dans les Figures 9.2 et 9.3).

En comparant l'approche *2-DBA* avec l'approche *k-DBA*, nous évaluons l'impact de considérer tous les objets recommandés aux positions précédentes dans le calcul de la probabilité conditionnelle. En recommandant aléatoirement un objet avec une probabilité $\epsilon = 5\%$, la proportion d'abandon obtenue par la solution optimale ne peut pas être atteinte, mais seulement approchée. Pour l'atteindre, une approche de type

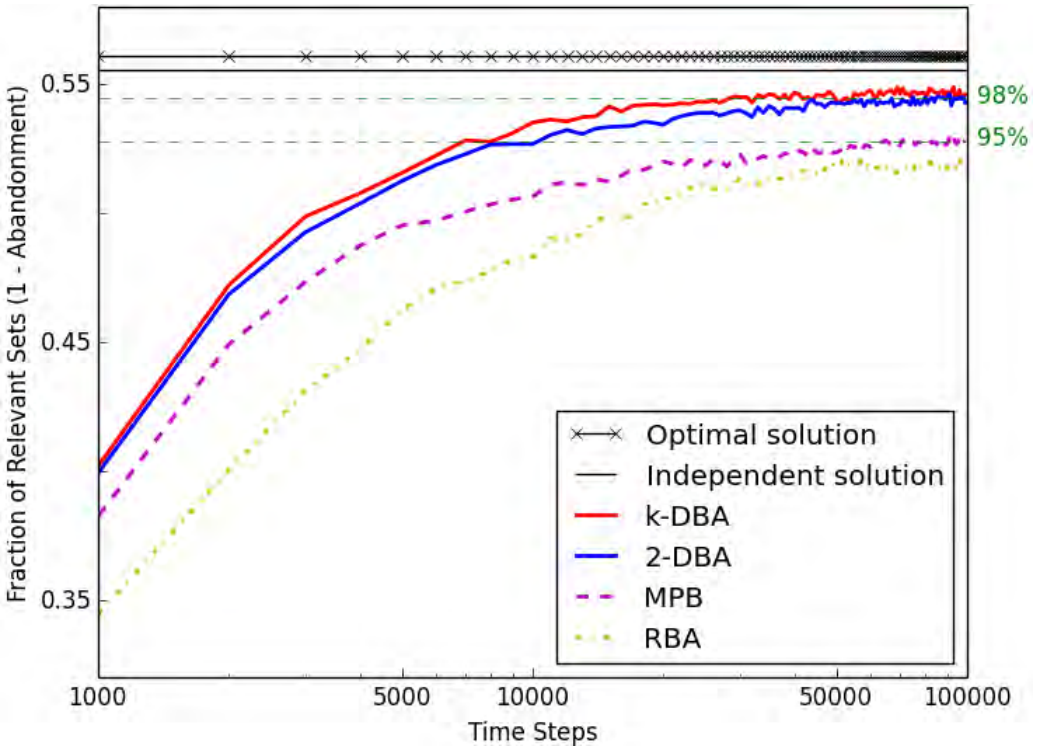

FIGURE 9.2 – Comparaison des différentes approches avec le jeu de données Movielens avec un seuil de pertinence = 4. (1 - la proportion d'abandon) en fonction du nombre d'interaction avec le système.

ϵ-decreasing semble plus adaptée. Nous pouvons observer sur les figures 9.1, 9.2 et 9.3 que *k-DBA* converge vers de meilleures proportions d'abandon que l'approche *2-DBA*, pour des vitesses d'apprentissage équivalentes. Le fait de considérer l'ensemble des objets recommandés aux positions précédentes plutôt qu'uniquement celui à la position précédente dans le calcul de la probabilité conditionnelle permet donc d'améliorer les résultats.

Comparons maintenant les résultats des deux approches proposées dans ce chapitre avec ceux des approches de l'état de l'art. Tout d'abord, l'approche *k-DBA* permet de converger vers de meilleures proportions d'abandon que les approches *RBA* et *MPB*, quelque soit le jeu de données et le seuil de pertinence utilisés. Avec le jeu de données MovieLens et le seuil de pertinence 2 (Figure 9.1), L'approche *k-DBA* atteint en $\approx 2,000$ interactions la proportion d'abandon obtenue par l'approche *MPB* à la fin de l'expérimentation ($\approx 6,000$ interactions pour les Figures 9.2 et 9.3). De plus, à la fin de l'expérimentation, *k-DBA* obtient $\approx 6\%$ moins d'abandon en moyenne que l'approche *MPB* ($\approx 2\%$ moins d'abandon sur la Figure 9.2 et $\approx 1.5\%$ moins d'abandon sur la Figure9.3).

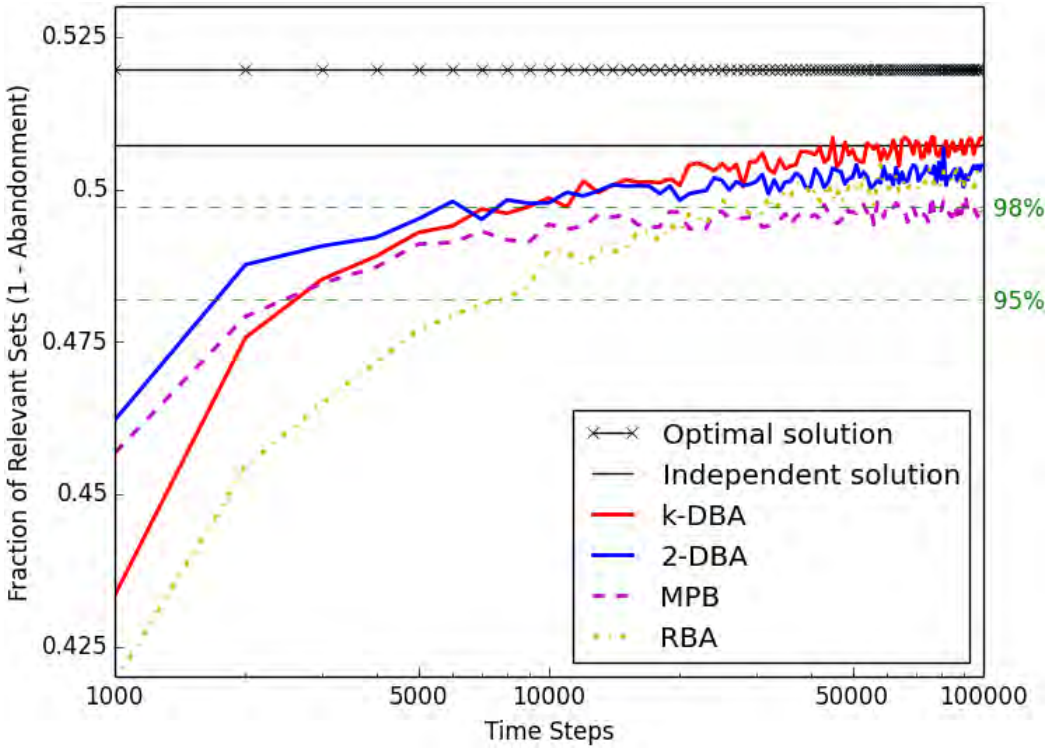

FIGURE 9.3 – Comparaison des différentes approches avec le jeu de données Jester avec un seuil de pertinence = 7. (1 - la proportion d'abandon) en fonction du nombre d'interaction avec le système.

L'approche *k-DBA* apprend 1.5 à 11 fois plus vite que les approches de l'état de l'art. Afin de confirmer cette amélioration de la vitesse d'apprentissage, nous comparons le nombre d'interactions nécessaires par les différentes approches pour atteindre 95%, 98% et 100% de la proportion d'abandon obtenue en utilisant la solution indépendante (Voir Tableau 9.1). Ce tableau confirme que les approches proposées dans ce chapitre apprennent plus rapidement que les approches de l'état de l'art, et permettent d'obtenir de plus faibles proportions d'abandon. L'approche *RBA* est celle qui apprend le moins vite, mais sur la Figure 9.1, cette approche atteint de plus faibles proportions d'abandon sur le long terme que l'approche *2-DBA*. Ce n'est pas le cas pour les Figures 9.2 et 9.3 pour lesquelles l'approche *RBA* obtient de plus faibles proportions d'abandon que l'approche *2-DBA*. Donc, sur le long terme, il est possible que l'approche *RBA* soit plus efficace que l'approche *2-DBA*, cela s'explique principalement par le fait que *RBA* vise la solution k-diversifiée.

Conclusion

Dans ce chapitre, nous avons abordé la problématique de la diversité en recommandation à tirages multiples. Nous avons proposé *2-Diversified Bandit Algorithm (2-DBA)* et sa généralisation *k-Diversified Bandit Algorithm (k-DBA)* qui visent toutes deux à apporter de la diversité dans la liste des objets recommandés. Dans le contexte de la minimisation de l'abandon, autrement dit la maximisation du nombre de fois où au moins un objet de la liste est cliqué, nous avons montré que nos approches apprennent plus rapidement, et convergent vers des proportions d'abandon plus faibles que les approches de l'état de l'art, *RBA* et *MPB*. Les expérimentations ont été conduites sur deux jeux de données de référence. En comparant les deux approches proposées, nous avons montré que considérer l'ensemble des objets recommandés aux positions précédentes plutôt qu'uniquement celui à la position précédente dans le calcul de la probabilité conditionnelle permet d'obtenir de plus faibles proportions d'abandon.

Conclusion et perspectives

Conclusion

Nos travaux se situent dans le cadre de la recommandation et plus précisément sur l'utilisation des stratégies de bandit dans ce cadre.

De nombreux SR voient la popularité de leurs objets décroître avec le temps. Nous avons proposé une nouvelle stratégie capable de prendre en compte l'obsolescence progressive de la popularité des objets.

A chaque instant, un SR peut devoir recommander plusieurs objets à chaque instant, il s'agit de SR à tirages multiples. Dans ce cadre, nous avons proposé une nouvelle approche utilisant des stratégies de bandit à tirages simples pour recommander plusieurs objets à chaque instant. Dans la littérature récente en apprentissage, quelques stratégies spécialement conçues pour recommander plusieurs objets à chaque instant ont été proposées d'un point de vue théorique. Dans nos travaux nous avons proposé une implémentation efficace de l'un d'eux pour le rendre utilisable en recommandation. Enfin, lorsque l'objectif est de minimiser l'abandon, c'est à dire maximiser le nombre de fois où l'utilisateur clique sur au moins un des objets recommandés, il est important de prendre en compte la notion de diversité. Nous proposons deux approches prenant en compte cette notion. Dans les sous-sections suivantes, nous résumons les différentes contributions ainsi que les résultats obtenus.

Bandit et obsolescence progressive

Dans le chapitre 4, nous avons présenté les stratégies de bandit capables de prendre en compte la non-stationnarité de la moyenne des récompenses d'un bras. Nous avons proposé un modèle plus restreint dans lequel de nouveaux bras apparaissent régulièrement et où le gain moyen associé à chaque bras décroît de manière exponentielle. Ce modèle est inspiré de plusieurs observations empiriques sur un jeu de données du challenge NEWS-REEL de la conférence CLEF où l'obsolescence des bras est visible.

Deux approches ont ensuite été proposées dans le chapitre 5. La première, l'approche *Fading-UCB*, représente une adaptation pour ce modèle de la stratégie UCB, une analyse nous permet de quantifier une borne supérieure de regret pour cette stratégie. La seconde approche proposée, *Trust and Abandon* repose sur une hypothèse stricte : lors de l'apparition d'un bras, nous estimons que l'ensemble des autres bras ont été suffisamment joués pour avoir une estimation précise des gains moyens associés.

Plusieurs simulations nous permettent d'observer que ces deux approches, *Fading-UCB* et *Trust and Abandon*, obtiennent de bien meilleures performances en terme de regret cumulé que l'approche de l'état de l'art *Sliding-Window UCB*, conçue pour prendre en compte la non-stationnarité du gain moyen de chaque bras. L'ensemble de ces approches surpassent largement la stratégie *UCB1*, qui ne prend pas du tout en compte l'obsolescence progressive des bras avec le temps.

Stratégies de bandit et recommandation à tirages multiples

Dans le chapitre 6, nous avons décrit en détail l'impact de recommander plusieurs objets à chaque instant au lieu d'un seul objet. Nous avons en particulier apporté un état de l'art sur la prise en compte de la diversité en recommandation. Nous décrivons également comment les approches utilisant des stratégies de bandit peuvent être évaluées. Nous avons ensuite dans les 3 chapitres suivants proposé plusieurs solutions.

Approches utilisant des stratégies de bandit à tirages simples

Le chapitre 7 s'intéresse au problème de la recommandation à tirages multiples en utilisant des stratégies de bandit à tirages simples. Les approches de l'état de l'art utilisent autant d'instances d'une stratégie de bandit à tirages simples qu'il y a d'objets à recommander. Pour améliorer la vitesse d'apprentissage, nous avons proposé au chapitre 7 de gérer l'ensemble des recommandations simultanément en utilisant une seule instance d'une stratégie de bandit à tirages simples.

Dans le cas de l'utilisation de la stratégie de bandit *UCB1*, l'approche proposée *MPB* apprend beaucoup plus rapidement (jusqu'à treize fois plus rapidement). De plus elle obtient des proportions d'abandon équivalentes à celles de l'approche la plus rapide de l'état de l'art *IBA*.

Cependant en utilisant la stratégie ϵ-*greedy*, l'approche *MPB* n'atteint pas toujours des proportions d'abandon aussi faibles que *IBA*, pour une vitesse d'apprentissage équivalente. Nous avons pu observer expérimentalement que la nature aléatoire de la stratégie ϵ-*greedy* peut expliquer qu'aucune amélioration n'est observable.

Stratégies de bandit à tirages multiples

L'état de l'art récent en apprentissage statistique propose des stratégies de bandit conçues spécifiquement pour recommander plusieurs objets à chaque instant. Les approches présentées dans le chapitre 7 utilisent autant d'instances d'une stratégie de bandit qu'il y a de recommandations à effectuer. Dans le chapitre 8, nous proposons une

implémentation efficace, utilisable en recommandation, de l'une des stratégies conçues pour recommander plusieurs objets à chaque instant : *Exp3.M*. En gérant l'ensemble des recommandations simultanément, la stratégie *Exp3.M* converge vers des proportions d'abandon équivalentes et apprend plus vite que l'approche de l'état de l'art *IBA* qui obtient les meilleures performances en terme de vitesse d'apprentissage.

La stratégie *Exp3.M* est conçue pour maximiser le taux des clics. Lorsque la minimisation de l'abandon est attendue, la stratégie *Exp3.M* vise une solution sous-optimale. Nous avons également montré que l'approche *RBA* implémentée en utilisant la stratégie de bandit *Exp3* peut être plus intéressante sur le long terme.

Apport de diversité dans les listes de résultats

Le chapitre 9 s'intéresse à la prise en compte de la diversité. Deux approches ont été proposées : *2-Diversified Bandit Algorithm (2-DBA)* et sa généralisation *k-Diversified Bandit Algorithm (k-DBA)*. Ces deux approches visent à apporter de la diversité dans la liste des objets recommandés.

Dans le contexte de la minimisation de l'abandon, autrement dit la maximisation du nombre de fois où au moins un objet de la liste recommandée est cliqué, nous avons montré que nos approches apprennent plus rapidement, et convergent vers de plus faibles et donc meilleures proportions d'abandon que les approches de l'état de l'art, *RBA* et *MPB*.

Les expérimentations ont été conduites sur deux jeux de données de référence. En comparant les deux approches proposées, nous avons montré que considérer l'ensemble des objets recommandés aux positions précédentes plutôt qu'uniquement l'objet à la position précédente dans le calcul de la probabilité conditionnelle permet d'obtenir de meilleures proportions d'abandon.

Perspectives

Si les travaux réalisés au cours de cette thèse donnent de bons résultats, il est important tout de même d'évaluer leurs limites et d'identifier les prolongements possibles qui constitueront nos pistes de recherche à venir. Nous détaillons dans cette section nos perspectives de recherche.

Évaluation des stratégies *Fading-UCB* et *Trust and Abandon* à l'aide de jeux de données réelles

Les stratégies proposées dans le chapitre 5 sont conçues pour prendre en compte l'obsolescence progressive. Afin de les évaluer, nous avons simuler cette obsolescence. Seulement nous avons pu observer que cette obsolescence est réelle sur un jeu de données du challenge NEWS-REEL de la conférence CLEF. Une perspective intéressante pour la suite de nos travaux est d'évaluer nos approches sur ce jeu de données. Afin d'avoir une évaluation efficace, nous nous placerons dans le cadre d'évaluation proposée dans l'article [Li 2010] (voir détails dans la section 2 du chapitre 6).

Nous souhaitons également proposer un nouveau cadre expérimental utilisant des données issues de systèmes de recommandation d'information, tels que le jeu de données *Yahoo News Feed dataset* qui contient plus d'un tera octet d'interactions ou encore les données proposées par les différents challenges NEWSREEL de la conférence CLEF.

Borne de regret pour la stratégie *Trust and Abandon*

Si nous fournissons une analyse de la borne de regret supérieure pour la stratégie *Fading-UCB*, ce n'est pas le cas pour la stratégie *Trust and Abandon*. Nous souhaitons donc dans la suite de nos travaux analyser la borne supérieure de regret de l'approche *Trust and Abandon* qui obtient de bons résultats expérimentalement.

Stratégies de bandit à tirages multiples et diversité

La stratégie *Exp3.M* est conçue pour maximiser le taux de clics. Donc, lorsque l'objectif est de minimiser l'abandon, la stratégie *Exp3.M* vise une solution sous-optimale. Nous avons montré que viser une solution prenant en compte la notion de diversité, comme le font les approches RBA et k-DBA, peut être plus intéressant sur le long terme. Dans nos travaux futurs, nous pourrions adapter la stratégie *Exp3.M* pour qu'elle soit capable d'apporter de la diversité dans les listes de recommandations produites.

Réflexion sur un compromis entre la minimisation de l'abandon et la maximisation du taux de clics

Les approches 2-DBA et k-DBA apportent plus ou moins de diversité au sein des listes de résultats. Nous souhaitons dans nos travaux futurs intégrer plus d'éléments sur l'apport plus ou moins important de diversité. Nous voudrions étudier si le nombre d'objets pris en compte dans le calcul des probabilités conditionnelles peut impacter

le taux de clics et la proportion d'abandon. En recommandation, une solution qui peut limiter le désengagement utilisateur en diversifiant les listes de recommandations tout en conservant des taux de clics acceptables peut être plus efficace qu'une solution considérant un seul de ces deux aspects.

Bibliographie Personnelle

[Cabanac 2015] Guillaume Cabanac, Amira Derradji, Ali Jaffal, Jonathan Louëdec et Gloria Elena Jaramillo Rojas. *Forum Jeunes Chercheurs à Inforsid 2014.* Ingénierie des Systèmes d'Information, vol. 20, no. 2, pages 119–143, 2015. Sélection des meilleurs articles du FJC 2014.

[Louëdec 2014] Jonathan Louëdec. *Algorithmes de bandits pour les systèmes de recommandation (student paper).* In INFORSID Forum Jeunes Chercheurs, Lyon, 20/05/2014-23/05/2014, pages 17–20, http ://www.editions-hermes.fr/, mai 2014. Hermès.

[Louëdec 2015a] Jonathan Louëdec, Max Chevalier, Aurélien Garivier et Josiane Mothe. *Algorithmes de bandits pour la recommandation à tirages multiples.* Document numérique, vol. 18/2-3/2015, pages 59–79, 2015.

[Louëdec 2015b] Jonathan Louëdec, Max Chevalier, Aurélien Garivier et Josiane Mothe. *Systèmes de recommandations : algorithmes de bandits et évaluation experimentale (education paper).* In Journées de Statistique de la SFdS (JDS), Lille, 01/05/2015-05/05/2015, page (en ligne), http ://www.sfds.asso.fr/, mai 2015. Société Française de Statistiques (SFdS).

[Louëdec 2015c] Jonathan Louëdec, Max Chevalier, Josiane Mothe et Aurélien Garivier. *A Multiple-play Bandit Algorithm Applied to Recommender Systems (regular paper).* In Florida Artificial Intelligence Research Society (FLAIRS), Hollywood, Floride, Etats Unis, 18/05/2015-20/05/2015, pages 67–72. AAAI Press, mai 2015.

[Louëdec 2015d] Jonathan Louëdec, Max Chevalier, Josiane Mothe, Aurélien Garivier et Sébastien Gerchinovitz. *Algorithmes de bandit pour les systèmes de recommandation : le cas de multiples recommandations simultanées (regular paper).* In Conférence francophone en Recherche d'Information et Applications (CORIA), Paris, 18/03/2015-20/03/2015, pages 73–88, http ://www.limsi.fr, mars 2015. LIMSI.

[Louëdec 2016] Jonathan Louëdec, Laurent Rossi, Max Chevalier, Josiane Mothe, Aurélien Garivier et Sébastien Gerchinovitz. *Algorithme de bandit et obsolescence : un modèle pour la recommandation.* In Conférence francophone sur l'apprentissage automatique (CAP), Marseille, 04/07/2016-08/07/2016, pages Accepté, à paraître, Juillet 2016.

Bibliographie

[Adamopoulos 2011] Panagiotis Adamopoulos et Alexander Tuzhilin. *On unexpectedness in recommender systems : Or how to expect the unexpected.* In Workshop on Novelty and Diversity in Recommender Systems (DiveRS 2011), at the 5th ACM International Conference on Recommender Systems (RecSys' 11), pages 11–18. Chicago, Illinois, USA : ACM, 2011.

[Adomavicius 2011] Gediminas Adomavicius et YoungOk Kwon. *Maximizing aggregate recommendation diversity : A graph-theoretic approach.* In Proc. of the 1st International Workshop on Novelty and Diversity in Recommender Systems (DiveRS 2011), pages 3–10. Citeseer, 2011.

[Adomavicius 2012] Gediminas Adomavicius et YoungOk Kwon. *Improving aggregate recommendation diversity using ranking-based techniques.* Knowledge and Data Engineering, IEEE Transactions on, vol. 24, no. 5, pages 896–911, 2012.

[Agrawal 2009] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson et Samuel Ieong. *Diversifying search results.* In Proceedings of the Second ACM International Conference on Web Search and Data Mining, pages 5–14. ACM, 2009.

[Agrawal 2012] Shipra Agrawal et Navin Goyal. *Analysis of Thompson sampling for the multi-armed bandit problem.* In Proceedings of the 25th Conference on Learning Theory (COLT), 2012.

[Anderson 2006] Chris Anderson. The long tail : Why the future of business is selling more for less. Hyperion, 2006.

[Arazy 2009] Ofer Arazy, Nanda Kumar et Bracha Shapira. *Improving social recommender systems.* IT Professional Magazine, vol. 11, no. 4, page 38, 2009.

[Audibert 2009] Jean-Yves Audibert, Rémi Munos et Csaba Szepesvári. *Exploration–exploitation tradeoff using variance estimates in multi-armed bandits.* Theoretical Computer Science, vol. 410, no. 19, pages 1876–1902, 2009.

[Auer 2002a] Peter Auer, Nicolo Cesa-Bianchi et Paul Fischer. *Finite-time analysis of the multiarmed bandit problem.* Machine learning, vol. 47, no. 2-3, pages 235–256, 2002.

[Auer 2002b] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund et Robert E Schapire. *The nonstochastic multiarmed bandit problem.* SIAM Journal on Computing, vol. 32, no. 1, pages 48–77, 2002.

[Ben-Shimon 2007] David Ben-Shimon, Alexander Tsikinovsky, Lior Rokach, Amnon Meisles, Guy Shani et Lihi Naamani. *Recommender system from personal social networks*. In Advances in Intelligent Web Mastering, pages 47–55. Springer, 2007.

[Berry 1997] Donald A Berry, Robert W Chen, Alan Zame, David C Heath et Larry A Shepp. *Bandit problems with infinitely many arms*. The Annals of Statistics, pages 2103–2116, 1997.

[Besbes 2014] Omar Besbes, Yonatan Gur et Assaf Zeevi. *Optimal exploration-exploitation in a multi-armed-bandit problem with non-stationary rewards*. arXiv preprint arXiv :1405.3316, 2014.

[Bridge 2005] Derek Bridge, Mehmet H Göker, Lorraine McGinty et Barry Smyth. *Case-based recommender systems*. The Knowledge Engineering Review, vol. 20, no. 03, pages 315–320, 2005.

[Brynjolfsson 2003] Erik Brynjolfsson, Yu Hu et Michael D Smith. *Consumer surplus in the digital economy : Estimating the value of increased product variety at online booksellers*. Management Science, vol. 49, no. 11, pages 1580–1596, 2003.

[Bubeck 2012] Sebastien Bubeck et Nicolo Cesa-Bianchi. *Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems*. Foundations and Trends in Machine Learning, vol. 5, no. 1, pages 1–122, 2012.

[Burke 2002] Robin Burke. *Hybrid recommender systems : Survey and experiments*. User modeling and user-adapted interaction, vol. 12, no. 4, pages 331–370, 2002.

[Burke 2007] Robin Burke. *Hybrid web recommender systems*. In The adaptive web, pages 377–408. Springer, 2007.

[Cabanac 2015] Guillaume Cabanac, Amira Derradji, Ali Jaffal, Jonathan Louëdec et Gloria Elena Jaramillo Rojas. *Forum Jeunes Chercheurs à Inforsid 2014*. Ingénierie des Systèmes d'Information, vol. 20, no. 2, pages 119–143, 2015. Sélection des meilleurs articles du FJC 2014.

[Candillier 2009] Laurent Candillier, Kris Jack, Françoise Fessant et Frank Meyer. *State-of-the-art recommender systems*. Collaborative and Social Information Retrieval and AccessTechniques for Improved User Modeling, 2009.

[Candillier 2012] Laurent Candillier, Max Chevalier, Damien Dudognon, Josiane Mothe et Ebuzzing OverBlog. *Multiple similarities for diversity in recommender systems*. International Journal on Advances in Intelligent Systems Volume 5, Number 3 & 4, 2012, 2012.

[Carpentier 2015] Alexandra Carpentier et Michal Valko. *Simple regret for infinitely many armed bandits.* arXiv preprint arXiv :1505.04627, 2015.

[Castagnos 2010] Sylvain Castagnos, Nicolas Jones et Pearl Pu. *Eye-tracking product recommenders' usage.* In Proceedings of the fourth ACM conference on Recommender systems, pages 29–36. ACM, 2010.

[Cesa-Bianchi 1998] Nicolò Cesa-Bianchi et Paul Fischer. *Finite-Time Regret Bounds for the Multiarmed Bandit Problem.* In Proceedings of the 15th International Conference on Machine Learning (ICML), pages 100–108. Citeseer, 1998.

[Cesa-Bianchi 2006] Nicolo Cesa-Bianchi et Gábor Lugosi. Prediction, learning, and games. Cambridge university press, 2006.

[Chakrabarti 2009] Deepayan Chakrabarti, Ravi Kumar, Filip Radlinski et Eli Upfal. *Mortal multi-armed bandits.* In Advances in Neural Information Processing Systems, pages 273–280, 2009.

[Chen 2006] Harr Chen et David R Karger. *Less is more : probabilistic models for retrieving fewer relevant documents.* In Proceedings of the 29th international ACM Conference on Research and Development in Information Retrieval (SIGIR), pages 429–436, 2006.

[Chen 2013] Wei Chen, Yajun Wang et Yang Yuan. *Combinatorial multi-armed bandit : General framework and applications.* In Proceedings of the 30th International Conference on Machine Learning (ICML), pages 151–159, 2013.

[Chifu 2012] Adrian-Gabriel Chifu et Radu-Tudor Ionescu. *Word sense disambiguation to improve precision for ambiguous queries.* Open Computer Science, vol. 2, no. 4, pages 398–411, 2012.

[Clarke 2008] Charles LA Clarke, Maheedhar Kolla, Gordon V Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher et Ian MacKinnon. *Novelty and diversity in information retrieval evaluation.* In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pages 659–666. ACM, 2008.

[Combes 2015] Richard Combes, Stefan Magureanu, Alexandre Proutiere et Cyrille Laroche. *Learning to Rank : Regret Lower Bounds and Efficient Algorithms.* SIGMETRICS Perform. Eval. Rev., vol. 43, no. 1, pages 231–244, Juin 2015.

[Coquelin 2007] Pierre-Arnaud Coquelin et Rémi Munos. *Bandit algorithms for tree search.* arXiv preprint cs/0703062, 2007.

[Craswell 2008] Nick Craswell, Onno Zoeter, Michael Taylor et Bill Ramsey. *An experimental comparison of click position-bias models.* In Proceedings of the 2008

International Conference on Web Search and Data Mining, pages 87–94. ACM, 2008.

[Deerwester 1990] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer et Richard Harshman. *Indexing by latent semantic analysis*. Journal of the American society for information science, vol. 41, no. 6, page 391, 1990.

[Even-Dar 2002] Eyal Even-Dar, Shie Mannor et Yishay Mansour. *PAC bounds for multi-armed bandit and Markov decision processes*. In Computational Learning Theory, pages 255–270. Springer, 2002.

[Even-Dar 2006] Eyal Even-Dar, Shie Mannor et Yishay Mansour. *Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems*. The Journal of Machine Learning Research, vol. 7, pages 1079–1105, 2006.

[Foulonneau 2013] Muriel Foulonneau, Valentin Grouès, Yannick Naudet et Max Chevalier. *Recommandeurs et diversité : Exploitation de la longue traîne et diversité des listes de recommandations*, 2013.

[Gandhi 2006] Rajiv Gandhi, Samir Khuller, Srinivasan Parthasarathy et Aravind Srinivasan. *Dependent Rounding and Its Applications to Approximation Algorithms*. J. ACM, vol. 53, no. 3, pages 324–360, 2006.

[Garivier 2008] Aurélien Garivier et Eric Moulines. *On upper-confidence bound policies for non-stationary bandit problems*. arXiv preprint arXiv :0805.3415, 2008.

[Garivier 2011] Aurélien Garivier et Olivier Cappé. *The KL-UCB algorithm for bounded stochastic bandits and beyond*. In Proceedings of the 24th Conference on Learning Theory (COLT), 2011.

[Gelly 2006] Sylvain Gelly, Yizao Wang, Olivier Teytaud, Modification Uct Patterns et Projet Tao. *Modification of UCT with patterns in Monte-Carlo Go*. 2006.

[Goldberg 2001] Ken Goldberg, Theresa Roeder, Dhruv Gupta et Chris Perkins. *Eigentaste : A constant time collaborative filtering algorithm*. Information Retrieval, vol. 4, no. 2, pages 133–151, 2001.

[Harper 2015] F Maxwell Harper et Joseph A Konstan. *The MovieLens Datasets : History and Context*. ACM Transactions on Interactive Intelligent Systems (TiiS), vol. 5, no. 4, page 19, 2015.

[Hastie 2009] Trevor Hastie, Robert Tibshirani et Jerome Friedman. The elements of statistical learning. Springer, 2nd édition, 2009.

[Hopfgartner 2014] Frank Hopfgartner, Benjamin Kille, Andreas Lommatzsch, Till Plumbaum, Torben Brodt et Tobias Heintz. *Benchmarking News Recommendations in a Living Lab.* In CLEF'14 : Proceedings of the 5th International Conference of the CLEF Initiative, LNCS, pages 250–267. Springer Verlag, 09 2014.

[Kaufmann 2013] Emilie Kaufmann, Nathaniel Korda et Rémi Munos. *Thompson sampling : An asymptotically optimal finite-time analysis.* In Proceedings of the 23th International Conference in Algorithmic Learning Theory (ALT), 2013.

[Kaufmann 2014] Emilie Kaufmann, Olivier Cappé et Aurélien Garivier. *On the complexity of best arm identification in multi-armed bandit models.* arXiv preprint arXiv :1407.4443, 2014.

[Kembellec 2014] Gérald Kembellec, Ghislaine Chartron et Imad Saleh. Les moteurs et systèmes de recommandation. ISTE editions, 2014.

[Kohli 2013] Pushmeet Kohli, Mahyar Salek et Greg Stoddard. *A Fast Bandit Algorithm for Recommendation to Users With Heterogenous Tastes.* In Proceedings of the 27th AAAI Conference on Artificial Intelligence, pages 1135–1141, 2013.

[Komiyama 2014] Junpei Komiyama et Tao Qin. *Time-Decaying Bandits for Non-stationary Systems.* In Web and Internet Economics, pages 460–466. Springer, 2014.

[Komiyama 2015] Junpei Komiyama, Junya Honda et Hiroshi Nakagawa. *Optimal Regret Analysis of Thompson Sampling in Stochastic Multi-armed Bandit Problem with Multiple Plays.* In Proceedings of the 32th International Conference on Machine Learning (ICML), 2015.

[Konstan 2012] JA Konstan et J Riedl. *Deconstructing Recommender Systems : How Amazon and Netflix predict your preferences and prod you to purchase.* IEEE Spectrum, vol. 49, 2012.

[Koren 2009] Yehuda Koren. *The bellkor solution to the netflix grand prize.* Netflix prize documentation, vol. 81, pages 1–10, 2009.

[Kveton 2015] Branislav Kveton, Csaba Szepesvari, Zheng Wen et Azin Ashkan. *Cascading bandits : Learning to rank in the cascade model.* In Proceedings of the 32nd International Conference on Machine Learning (ICML-15), pages 767–776, 2015.

[Lai 1985] Tze Leung Lai et Herbert Robbins. *Asymptotically efficient adaptive allocation rules.* Advances in applied mathematics, vol. 6, no. 1, pages 4–22, 1985.

[Lan 1974] *Information Retrieval On-Line. F. W. Lancaster and E. G. Fayen. Los Angeles : Wiley-Becker & Hayes. 417 pp. (1974).* Journal of the American Society for Information Science, vol. 25, no. 5, pages 336–337, 1974.

[Lemire 2008] Daniel Lemire, Stephen Downes et Sébastien Paquet. *Diversity in open social networks.* published online, 2008.

[Li 2010] Lihong Li, Wei Chu, John Langford et Robert E.Schapire. *A Contextual-Bandit Approach to Personalized News Article Recommandation.* In Proceedings of 19th International World Wide Web Conference, pages 661–670, 2010.

[Liu 2011] Tie-Yan Liu. Learning to rank for information retrieval. Springer Science & Business Media, 2011.

[Louëdec 2014] Jonathan Louëdec. *Algorithmes de bandits pour les systèmes de recommandation (student paper).* In INFORSID Forum Jeunes Chercheurs, Lyon, 20/05/2014-23/05/2014, pages 17–20, http ://www.editions-hermes.fr/, mai 2014. Hermès.

[Louëdec 2015a] Jonathan Louëdec, Max Chevalier, Aurélien Garivier et Josiane Mothe. *Algorithmes de bandits pour la recommandation à tirages multiples.* Document numérique, vol. 18/2-3/2015, pages 59–79, 2015.

[Louëdec 2015b] Jonathan Louëdec, Max Chevalier, Aurélien Garivier et Josiane Mothe. *Systèmes de recommandations : algorithmes de bandits et évaluation experimentale (education paper).* In Journées de Statistique de la SFdS (JDS), Lille, 01/05/2015-05/05/2015, page (en ligne), http ://www.sfds.asso.fr/, mai 2015. Société Française de Statistiques (SFdS).

[Louëdec 2015c] Jonathan Louëdec, Max Chevalier, Josiane Mothe et Aurélien Garivier. *A Multiple-play Bandit Algorithm Applied to Recommender Systems (regular paper).* In Florida Artificial Intelligence Research Society (FLAIRS), Hollywood, Floride, Etats Unis, 18/05/2015-20/05/2015, pages 67–72. AAAI Press, mai 2015.

[Louëdec 2015d] Jonathan Louëdec, Max Chevalier, Josiane Mothe, Aurélien Garivier et Sébastien Gerchinovitz. *Algorithmes de bandit pour les systèmes de recommandation : le cas de multiples recommandations simultanées (regular paper).* In Conférence francophone en Recherche d'Information et Applications (CORIA), Paris, 18/03/2015-20/03/2015, pages 73–88, http ://www.limsi.fr, mars 2015. LIMSI.

[Louëdec 2016] Jonathan Louëdec, Laurent Rossi, Max Chevalier, Josiane Mothe, Aurélien Garivier et Sébastien Gerchinovitz. *Algorithme de bandit et obsolescence :*

un modèle pour la recommandation. In Conférence francophone sur l'apprentissage automatique (CAP), Marseille, 04/07/2016-08/07/2016, pages Accepté, à paraître, Juillet 2016.

[Ma 2011] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu et Irwin King. *Recommender systems with social regularization.* In Proceedings of the fourth ACM international conference on Web search and data mining, WSDM '11, pages 287–296, Hong Kong, China, 2011.

[Montaner 2001] Miquel Montaner. *A Taxonomy of Personalized Agents on the Internet.* Technical Report, 2001.

[O'Brien 2008] Heather L O'Brien et Elaine G Toms. *What is user engagement ? A conceptual framework for defining user engagement with technology.* Journal of the American Society for Information Science and Technology, vol. 59, no. 6, pages 938–955, 2008.

[Onuma 2009] Kensuke Onuma, Hanghang Tong et Christos Faloutsos. *TANGENT : a novel,'Surprise me', recommendation algorithm.* In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 657–666. ACM, 2009.

[Park 2008] Yoon-Joo Park et Alexander Tuzhilin. *The long tail of recommender systems and how to leverage it.* In Proceedings of the 2008 ACM conference on Recommender systems, pages 11–18. ACM, 2008.

[Perchet 2013] Vianney Perchet, Philippe Rigollet*et al. The multi-armed bandit problem with covariates.* The Annals of Statistics, vol. 41, no. 2, pages 693–721, 2013.

[Poirier 2010] Damien Poirier, Françoise Fessant et Isabelle Tellier. *De la Classification d'Opinion à la Recommandation : l'Apport des Textes Communautaires.* TAL : traitement automatique des langues : revue semestrielle de l'ATALA, vol. 51, no. 3, pages 19–46, 2010.

[Ponte 1998] Jay M Ponte et W Bruce Croft. *A language modeling approach to information retrieval.* In Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pages 275–281. ACM, 1998.

[Radlinski 2008] Filip Radlinski, Robert Kleinberg et Thorsten Joachims. *Learning Diverse Rankings with Multi-Armed Bandits.* In Proceedings of the 25th International Conference on Machine Learning (ICML), pages 784–791, 2008.

[Resnick 1994] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom et John Riedl. *GroupLens : an open architecture for collaborative filtering of netnews.* In Proceedings of the 1994 ACM conference on Computer supported cooperative work, pages 175–186. ACM, 1994.

[Ricci 2011] Francesco Ricci, Lior Rokach et Bracha Shapira. *Introduction to Recommender Systems Handbook.* In Recommender Systems Handbook, pages 1–35. Springer, 2011.

[Robertson 1976] Stephen E Robertson et K Sparck Jones. *Relevance weighting of search terms.* Journal of the American Society for Information science, vol. 27, no. 3, pages 129–146, 1976.

[Robertson 1977] Stephen E Robertson. *The probability ranking principle in IR.* Journal of documentation, vol. 33, no. 4, pages 294–304, 1977.

[Robertson 1994] Stephen E Robertson et Steve Walker. *Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval.* In Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, pages 232–241. Springer-Verlag New York, Inc., 1994.

[Salton 1975] Gerard Salton, Anita Wong et Chung-Shu Yang. *A vector space model for automatic indexing.* Communications of the ACM, vol. 18, no. 11, pages 613–620, 1975.

[Santos 2010] Rodrygo LT Santos, Craig Macdonald et Iadh Ounis. *Selectively diversifying web search results.* In Proceedings of the 19th ACM international conference on Information and knowledge management, pages 1179–1188. ACM, 2010.

[Sarkar 1991] Jyotirmoy Sarkar. *One-armed bandit problems with covariates.* The Annals of Statistics, pages 1978–2002, 1991.

[Slivkins 2007] Aleksandrs Slivkins et Eli Upfal. *Adapting to a Stochastically Changing Environment : The Dynamic Multi-Armed Bandits Problem.* Rapport technique, Technical Report CS-07-05, Brown University, 2007.

[Strehl 2006] Alexander L Strehl, Chris Mesterharm, Michael L Littman et Haym Hirsh. *Experience-efficient learning in associative bandit problems.* In Proceedings of the 23rd international conference on Machine learning, pages 889–896. ACM, 2006.

[Sutton 1999] Richard S Sutton et Andrew G Barto. *Reinforcement learning.* Journal of Cognitive Neuroscience, vol. 11, no. 1, pages 126–134, 1999.

[Thompson 1933] William R Thompson. *On the likelihood that one unknown probability exceeds another in view of the evidence of two samples.* Biometrika, pages 285–294, 1933.

[Tracà 2015] Stefano Tracà et Cynthia Rudin. *Regulating Greed Over Time.* arXiv preprint arXiv :1505.05629, 2015.

[Tucker 2007] Catherine Tucker et Juanjuan Zhang. *Long tail or steep tail ? A field investigation into how online popularity information affects the distribution of customer choices.* 2007.

[Uchiya 2010] Taishi Uchiya, Atsuyoshi Nakamura et Mineichi Kudo. *Algorithms for Adversarial Bandit Problems with Multiple Plays.* In Algorithmic Learning Theory, LNCS Springer, pages 375–389. 2010.

[Watkins 1989] Christopher John Cornish Hellaby Watkins. *Learning from delayed rewards.* 1989.

[White 2012] John Myles White. Bandit algorithms for website optimization. O'Reilly Media, December 2012.

[Zhang 2009] Mi Zhang et Neil Hurley. *Statistical modeling of diversity in top-n recommender systems.* In Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 01, pages 490–497. IEEE Computer Society, 2009.

[Zhou 2010] Tao Zhou, Zoltán Kuscsik, Jian-Guo Liu, Matúš Medo, Joseph Rushton Wakeling et Yi-Cheng Zhang. *Solving the apparent diversity-accuracy dilemma of recommender systems.* Proceedings of the National Academy of Sciences, vol. 107, no. 10, pages 4511–4515, 2010.

[Ziegler 2005] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan et Georg Lausen. *Improving recommendation lists through topic diversification.* In Proceedings of the 14th international conference on World Wide Web, pages 22–32. ACM, 2005.

Liste des abréviations

RI Recherche d'Information

SR Système de Recommandation

MPB L'approche «Multiple-Play Bandit»

RBA L'approche «Ranked Bandit Algorithm»

IBA L'approche «Independent Bandit Algorithm»

2-DBA L'approche «2-Diversified Bandit Algorithm»

k-DBA L'approche «k-Diversified Bandit Algorithm»

UCB Upper Confidence Bound

F-UCB Fading Upper Confidence Bound

Table des figures

Liste des tableaux

Liste des Algorithmes