Recommandation contextuelle de services : application à la recommandation d'évènements culturels dans la ville intelligente

Nicolas Gutowski

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ D'ANGERS
COMUE **U**NIVERSITÉ **B**RETAGNE **L**OIRE

École Doctorale N° 601
Mathématique et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par

Nicolas GUTOWSKI

Recommandation contextuelle de services :
Application à la recommandation d'événements culturels dans la ville intelligente

Thèse présentée et soutenue à Angers, le 04/11/2019

Unité de recherche : LERIA (EA 2645)

Thèse N° : 181209

Rapporteurs avant soutenance :
Allel HADJALI, Professeur des Universités, ISAE-ENSMA, Poitiers
Armelle BRUN, Maître de Conférences (HDR), LORIA, Université de Lorraine

Composition du jury :

Président :	Éric MONFROY, Professeur des Universités, LERIA, Université d'Angers
Rapporteurs :	Allel HADJALI, Professeur des Universités, ISAE-ENSMA, Poitiers
	Armelle BRUN, Maître de Conférences (HDR), LORIA, Université de Lorraine
Examinateurs :	Florence SÈDES, Professeur des Universités, IRIT, Université Toulouse 3 - Paul Sabatier
	Bruno ZANUTTINI, Professeur des Universités, GREYC, Université de Caen Normandie
	Éric MONFROY, Professeur des Universités, LERIA, Université d'Angers
	Raphaël FÉRAUD, Docteur en Informatique, Chercheur, Orange Labs Lannion
Directeur de thèse :	Tassadit AMGHAR, Maître de Conférences (HDR), LERIA, Université d'Angers
Co-encadrant de thèse :	Olivier CAMP, Docteur en Informatique, Enseignant-Chercheur, ESEO, Angers

REMERCIEMENTS

« *La science a la chance et la modestie de savoir qu'elle est dans le provisoire, de déplacer les frontières de l'inconnu et d'avancer.* » ***Marc Augé***

Je souhaiterais tout d'abord remercier toutes les personnes impliquées, de près comme de loin, de m'avoir accordé la chance de réaliser cette thèse de doctorat dans de bonnes conditions et de m'avoir permis de participer à mon niveau à déplacer « les frontières de l'inconnu ».

Merci à mes encadrants, Tassadit Amghar et Olivier Camp, pour leur précieuse aide dans les moments de doutes et les maintes relectures d'articles. Merci particulièrement à Tassadit Amghar pour ses orientations scientifiques avisées et Olivier Camp pour m'avoir transmis le goût de la rigueur.

Merci aux membres de mon comité de suivi extérieur : Sylvain Lamprier et Matthieu Basseur pour leurs conseils justes qui m'ont permis d'atteindre le niveau de précision nécessaire dans mes communications scientifiques.

Merci aux rapporteurs de ma thèse Allel HadjAli et Armelle Brun, pour avoir accepté d'évaluer mon travail ainsi qu'à tous les examinateurs : Florence Sèdes, Bruno Zanuttini, Raphaël Féraud et Éric Monfroy.

Merci à Olivier Beaudoux, Chef du Département Informatique & Systèmes et à Guy Plantier Directeur de la Recherche de l'ESEO qui ont su m'aménager les conditions de travail adéquates pour la réalisation du doctorat. Merci à tous mes collègues du département pour leur accompagnement et plus particulièrement à Slimane Hammoudi pour ses conseils et ses collaborations dans le domaine du contexte. Merci à mon collègue Fabien Chhel, véritable encyclopédie humaine du domaine de l'apprentissage, avec qui j'ai pu apprendre énormément et qui m'a inspiré bon nombre de méthodes. Merci à Daniel Schang d'avoir pris le temps de relire ce mémoire bénévolement. Je remercie également Olivier Paillet, Directeur Général de l'ESEO pour m'avoir octroyé sa confiance avant même le début du projet et d'avoir soutenu le financement nécessaire à sa réalisation.

Merci à ma famille et ma belle-famille pour son soutien, ma femme pour sa patience et sa disponibilité, ainsi qu'à mon fils qui de nature m'a permis de sortir le nez de mon écran quand il fallait revenir à la vie.

Enfin je souhaiterais consacrer ma dernière dédicace à la mémoire de mon père défunt qui est la première personne à m'avoir mis un clavier sous la main et m'avoir donné goût à l'informatique.

SOMMAIRE

INTRODUCTION

Systèmes de recommandation : De la donnée à la décision...

En raison du volume croissant de données disponibles, les systèmes de recommandation sont de plus en plus utilisés et font l'objet de nombreuses recherches. Ils permettent de suggérer du contenu web [CPF17 ; Li+10], des films [Lev+12], de la musique [HBC10], des informations de voyage personnalisées à des touristes [Gav+14] ou encore divers éléments (comme des suggestions d'amis ou des événements) dans les réseaux sociaux [Bao+15].

Certains de ces systèmes de recommandation s'appuient désormais sur la croissance importante du nombre d'objets connectés dans le monde et leur utilisation à travers l'internet des objets *(Internet of Things : IoT)*. Ces systèmes font partie des applications de *Mobile Crowd Sensing and Computing (MCSC)* [Had14] utilisables entre autres dans le cadre de la ville intelligente. En effet, les applications de *MCSC* permettent de collecter une grande variété de données à partir des téléphones mobiles des utilisateurs tout en préservant le respect de leur vie privée [Had14]. Les systèmes de recommandation appliqués aux *MCSC* prennent tout leur sens au vu de la multitude d'informations disponibles dans les villes intelligentes puisqu'il devient impossible pour un utilisateur mobile de sélectionner les plus appropriées. C'est pourquoi, par exemple, un système de recommandation proposant des services pertinents à des utilisateurs mobiles évoluant dans une ville intelligente semble aujourd'hui incontournable. Il peut aussi devenir nécessaire de répondre à d'autres besoins plus implicites engendrés par l'*IoT*, comme par exemple optimiser la découverte dynamique de services *IoT* adéquats pour des clients mobiles situés dans un bâtiment intelligent de la ville [Wan+16].

Le contexte technologique de notre travail s'inscrit dans cette dynamique orientée *MCSC* et se concentre principalement sur la recommandation de services à des utilisateurs mobiles.

Plus particulièrement, la principale perspective de cette thèse est d'intégrer et d'évaluer à terme, des algorithmes de bandits-manchots pour la recommandation au sein de notre application mobile de visualisation et de recommandation d'événements culturels : *scéno*[1]. À ce titre, dans la Figure 1 nous illustrons le processus de recommandation à des utilisateurs mobiles dans la ville intelligente en se basant sur une approche de type apprentissage par renforcement.

1. Disponible sur Google Play : `https://play.google.com/store/apps/details?id=sceno.android.pfe.com.sceno&hl=fr`

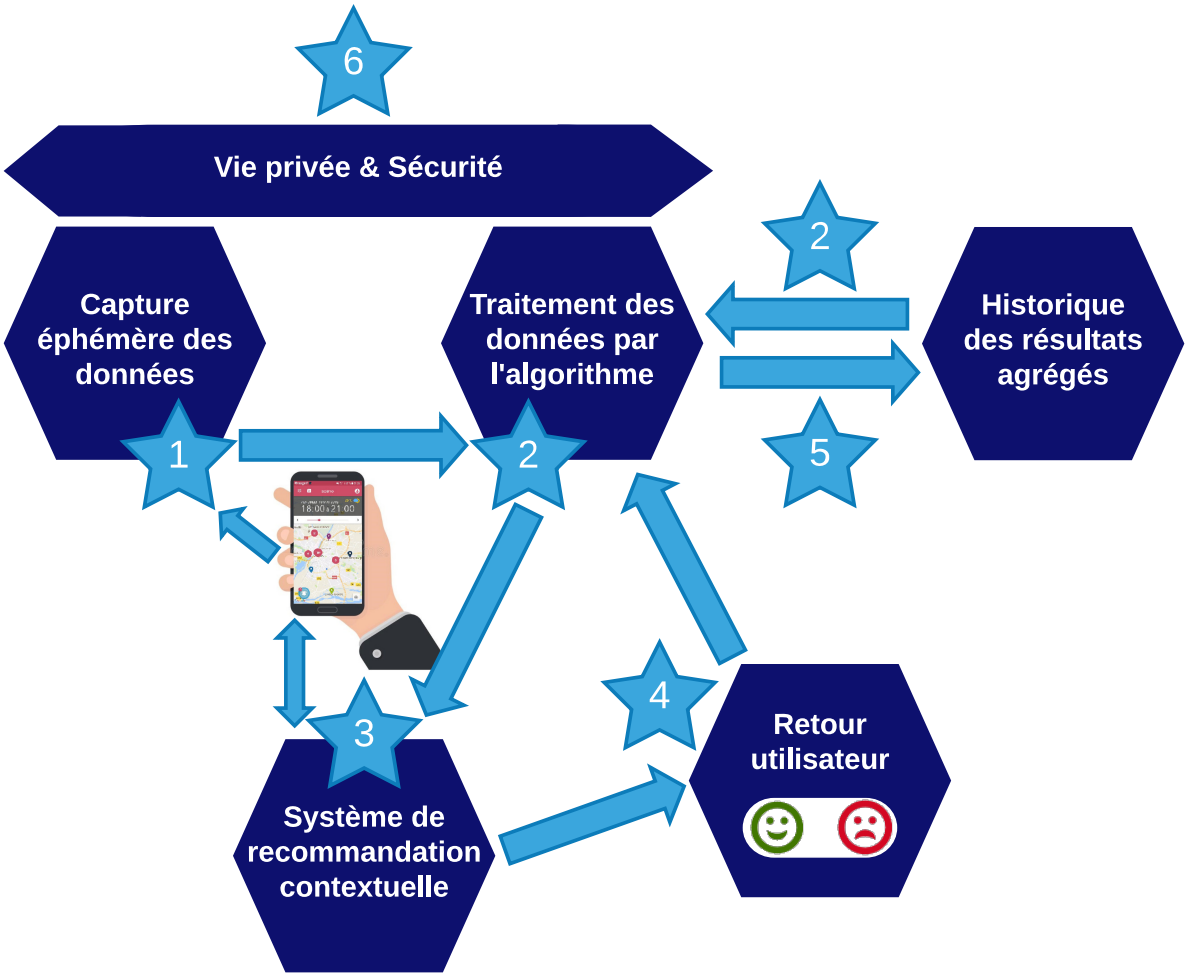


FIGURE 1 – De la donnée à la décision : processus de recommandation à des utilisateurs mobiles dans la ville intelligente basé sur un apprentissage par renforcement.

Chaque étape de ce processus est le point d'entrée d'un certain nombre de problématiques pour lesquelles il est nécessaire d'apporter une solution. Nous détaillons ces étapes ci-dessous :

1. **Capture éphémère des données.** Dans le cadre d'une approche contextuelle de la recommandation, cette première étape est une phase « cruciale » de capture de l'information contextuelle. L'objectif est de pouvoir acquérir et transmettre du contexte pertinent sous forme de représentation structurée et exploitable par tout algorithme de recommandation contextuel. C'est à ce titre que l'adjectif « crucial » prend tout son sens puisque dans tout problème de décision, si les données fournies en entrée ne sont pas pertinentes, sont erronées, ou incomplètes, alors les décisions qui en découlent peuvent s'avérer inadéquates. Cette problématique est plus communément connue sous l'allé-

gation : « *Garbage in, Garbage out* » accréditée à Wilf Hey (IBM, 1965) par le *FOLDOC*[2] (Free On-line Dictionary of Computing) et notamment reprise dans le domaine de l'informatique décisionnelle [BBM81]. C'est pourquoi dans cette thèse nous avons porté une attention particulière à la bonne réalisation de cette phase de capture ;

2. **Traitement des données par l'algorithme et historique des résultats agrégés (1/2).** Cette seconde étape correspond au traitement des informations de contextes (précédemment capturées) par un algorithme d'apprentissage par renforcement dédié à la recommandation et permettant de sélectionner l'élément le plus pertinent à recommander. Plus particulièrement, en apprentissage par renforcement, nous faisons face à un problème de décisions séquentielles. Ainsi, à chaque itération dans un environnement interactif, un agent réalise une action (dans notre cas une recommandation) pour laquelle il reçoit un retour d'informations sous forme de récompense. Celle-ci lui permet d'acquérir de l'expérience et ainsi d'apprendre. Dans notre cas, durant l'étape de traitement, l'algorithme d'apprentissage par renforcement dédié à la recommandation tient compte des données contextuelles courantes qui lui ont été transmises en les confrontant à un historique de résultats des recommandations passées. Plus précisément, l'algorithme recommande l'élément qui maximise l'espérance d'obtenir une récompense sachant le contexte courant observé et l'historique des recommandations. Dans le cadre de cette thèse, en termes d'approche d'apprentissage par renforcement, nous avons choisi les algorithmes de bandits-manchots contextuels pour la recommandation basés sur un modèle linéaire (p. ex., *LinUCB* [Li+10], *Contextual Thompson Sampling* [AG13]). Ceux-ci utilisent à la fois le contexte courant observé et un historique sous forme de coefficients (pondérations) de confiance accordés à chaque dimension du contexte ;
3. **Système de recommandation contextuel.** Cette étape tire partie des calculs effectués par l'algorithme d'apprentissage traitant des données, et notifie physiquement l'élément à l'utilisateur mobile. Ensuite, nous attendons une interaction de l'utilisateur par rapport à la recommandation qui lui a été faite (retour utilisateur). Si l'utilisateur fait un retour (positif ou négatif), celui-ci est transmis à l'algorithme d'apprentissage ;
4. **Retour utilisateur.** Une fois que la recommandation a été physiquement effectuée à l'utilisateur, celui-ci a la possibilité : soit de l'ignorer ; soit de répondre de manière positive à celle-ci ; soit de répondre négativement. Si l'utilisateur ignore la recommandation, rien n'est fait en aval, et nous considérons qu'aucune recommandation n'a été effectuée. Si l'utilisateur répond (positivement ou négativement), la recommandation est alors prise en compte et est transmise pour traitement à l'algorithme d'apprentissage ;
5. **Traitement des données par l'algorithme et historique des résultats agrégés (2/2).** Cette étape de traitement dépend de l'algorithme d'apprentissage qui est employé pour l'effectuer et varie donc en fonction de sa politique de prise en compte des retours utilisateurs. Durant cette étape, le retour utilisateur est traité de manière à ce qu'un historique de réussites (retours positifs) et échecs (retours négatifs) des recommandations pas-

2. `http://foldoc.org/`

sées soit pris en compte. Dans le cadre des bandits-manchots pour la recommandation par exemple, à partir de cet historique, il s'agira de maximiser le cumul des « récompenses » (réussites) et ainsi de minimiser les « regrets » (échecs) ;

6. **Vie privée et sécurité.** Tout au long du processus, il est nécessaire de préserver la vie privée de l'utilisateur en restant conforme à la politique de confidentialité de l'application (p. ex., pour l'application mobile *scéno*, voir Annexe D.3). Ainsi tant pour la capture de l'information contextuelle que pour le stockage de l'historique, les traitements sont effectués de manière sécurisée et veillent à ne stocker que de l'information agrégée. C'est à dire plus concrètement que les données capturées circulent de manières chiffrées et sont détruites après traitement. Les seules données d'historiques stockées sont celles des poids de confiance accordés à chaque dimension du contexte.

Ainsi, dans le cadre de cette thèse et en perspective de proposer la solution scientifique la plus adéquate pour chaque étape du processus de recommandation de l'application *scéno* :

— nous avons évalué nos algorithmes de recommandation hors ligne afin de pouvoir choisir ceux qui correspondraient le mieux à nos besoins et/ou de proposer de nouvelles méthodes plus performantes (voir nos publications : [Gut+18a ; Gut+18b ; Gut+19a ; Gut+19d]) ;
— nous avons capturé des informations contextuelles issues du monde réel et avons raisonné sur ce contexte afin de pouvoir fournir à nos algorithmes de recommandation une représentation structurée du contexte suffisamment pertinente et exploitable (voir nos publications : [Gut+17 ; Gut+18b ; Gut+18c ; Gut+18d ; Gut+18e ; Gut+19d]).

Plus précisément, pour répondre à ces enjeux inhérents aux **systèmes de recommandation**, nos travaux se sont découpés comme suit :

1. Capturer **de la donnée** contextuelle et la rendre exploitable par des systèmes de recommandations c.-à-d., transformer le contexte brut capturé en représentation structurée de l'information (Voir Partie III, Chapitres 6 et 7) ;

2. Implémenter et comparer des algorithmes de bandits-manchots pour la recommandation, capables de mener **à la décision** de recommandation la plus pertinente possible pour chaque utilisateur rencontré dans un contexte donné (Voir Partie II, Chapitre 4) ;

3. Mesurer et analyser la performance de nos algorithmes de bandits-manchots pour la recommandation afin de les améliorer, selon trois critères : la précision globale, la diversité des recommandations, la précision individuelle (Voir Partie II, Chapitres 4 et 5 ; Partie III, Chapitre 7).

Dans la section suivante, nous repositionnons nos publications.

Publications

Journal à comité de lecture

N Gutowski, T Amghar, O Camp, S Hammoudi. A Framework for Context-Aware Service Recommendation for Mobile Users : A Focus on Mobility in Smart Cities. *From Data To Decision Journal (ISTE Press Ltd.)*, 2017.

Conférences internationales à comité de lecture

— N Gutowski, T Amghar, O Camp, F Chhel. Context Enhancement for Linear Contextual Multi-Armed Bandits. *30th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, Volos (GR), 2018
— N Gutowski, T Amghar, O Camp, F Chhel. Using Individual Accuracy to Create Context for Non-Contextual Multi-Armed Bandit Problems. *16th IEEE Research, Innovation and Vision for the Future (RIVF)*, Danang (VN), 2019
— N Gutowski, T Amghar, O Camp, F Chhel. Global Versus Individual Accuracy in Contextual Multi-Armed Bandit. *34th ACM/SIGAPP Symposium on Applied Computing (SAC)*, Limassol (CY), 2019
— N Gutowski, T Amghar, O Camp, F Chhel. Gorthaur : A Portfolio Approach for Dynamic Selection of Multi-Armed Bandit Algorithms for Recommendation. *31st IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, Portland (US), 2019
— N Gutowski, O Camp, T Amghar, F Chhel, P Albers. Improving Bandit-Based Recommendations with Spatial Context Reasoning : An Online Evaluation. *31st IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, Portland (US), 2019

Conférences nationales à comité de lecture

N Gutowski, T Amghar, O Camp, F Chhel. Bandits-Manchots Contextuels : Précision Globale Versus Individuelle. *4ème conférence sur les Applications Pratiques de l'Intelligence Artificielle (APIA)*, Nancy (FR), 2018.

Ateliers de conférences nationales à comité de lecture

— N Gutowski, T Amghar, O Camp, S Hammoudi. Recommandation et Visualisation dans les Villes Intelligentes : Projet EVENT-AI. Atelier Data Intelligence dans le cadre de la *36ème conférence INFormatique des ORganisations et Systèmes d'Information et de Décision (Inforsid)*, Nantes (FR), 2018
— N Gutowski, O Camp, P Albers, T Amghar, F Chhel, S Hammoudi, G Paller. Déduction de quartiers à partir des connexions au Wifi Urbain. *Atelier EXtraction de Connaissances à partir de donnéEs Spatialisées (EXCES) dans le cadre de la 14ème conférence in-*

ternationale francophone Spatial Analysis and GEOmatics (SAGEO), Montpellier (FR), 2018

Posters et démonstrations à comité de lecture

— N Gutowski, T Amghar, O Camp, F Chhel. Bandits-Manchots Contextuels : Précision Globale Versus Individuelle. *4ème conférence sur les Applications Pratiques de l'Intelligence Artificielle*, Nancy (FR), 2018.
— N Gutowski, T Amghar, O Camp. From mobility to adaptive services : Targeted display application. *BIG IA - Journée AfIA-MaDICS*, Lyon (FR), 2016

Chapitre de livre

Notre contribution à la revue *From Data To Decision* a été sélectionnée pour être intégrée dans un chapitre de livre. Afin de cadrer aux exigences de l'ouvrage, le titre de la contribution d'origine a été revu, l'abstract légèrement modifié, des paragraphes ont été ajoutés afin de lier nos travaux à la thématique métier que représente la gestion de crise et le nombre de pages a été réduit.

N Gutowski, T Amghar, O Camp, S Hammoudi. Mobility and Prediction : an Asset for Crisis Management. *In book : How Information Systems Can Help in Alarm/Alert Detection, (Elsevier, ISTE Press Ltd.)*, 2018

Publication à comité de lecture en marge de la thèse

N Gutowski, O Camp, E Chauveau. Measuring the Energy Consumption of Massive Data Insertions : An Energy Consumption Assessment of the PL/SQL FOR LOOP and FORALL Methods. *13th IEEE Green Computing and Communications (GreenCom)*, Exeter (UK), 2017

Positionnement des contributions

Principe

L'ensemble de nos publications cité ci-dessus se positionne dans la littérature de trois grands domaines :

— le traitement du contexte (de sa modélisation et son acquisition jusqu'au raisonnement que nous lui appliquons) ;
— les systèmes de recommandation et notamment ceux qui s'appuient sur les données contextuelles pour gagner en précision ;
— les bandits-manchots pour la recommandation.

Le positionnement de ces publications vis à vis de l'état de l'art est précisément illustré à la Figure 2. Nous y introduisons les trois grands chapitres de ce mémoire au titre d'état de

l'art : les systèmes de recommandation, les bandits-manchots pour la recommandation et le contexte. De plus, nous positionnons nos publications selon leur proximité et leur pertinence vis à vis de la contribution apportée à chacun des 3 chapitres.

Nous proposons de décrire la Figure 2 dans la sous-section suivante.

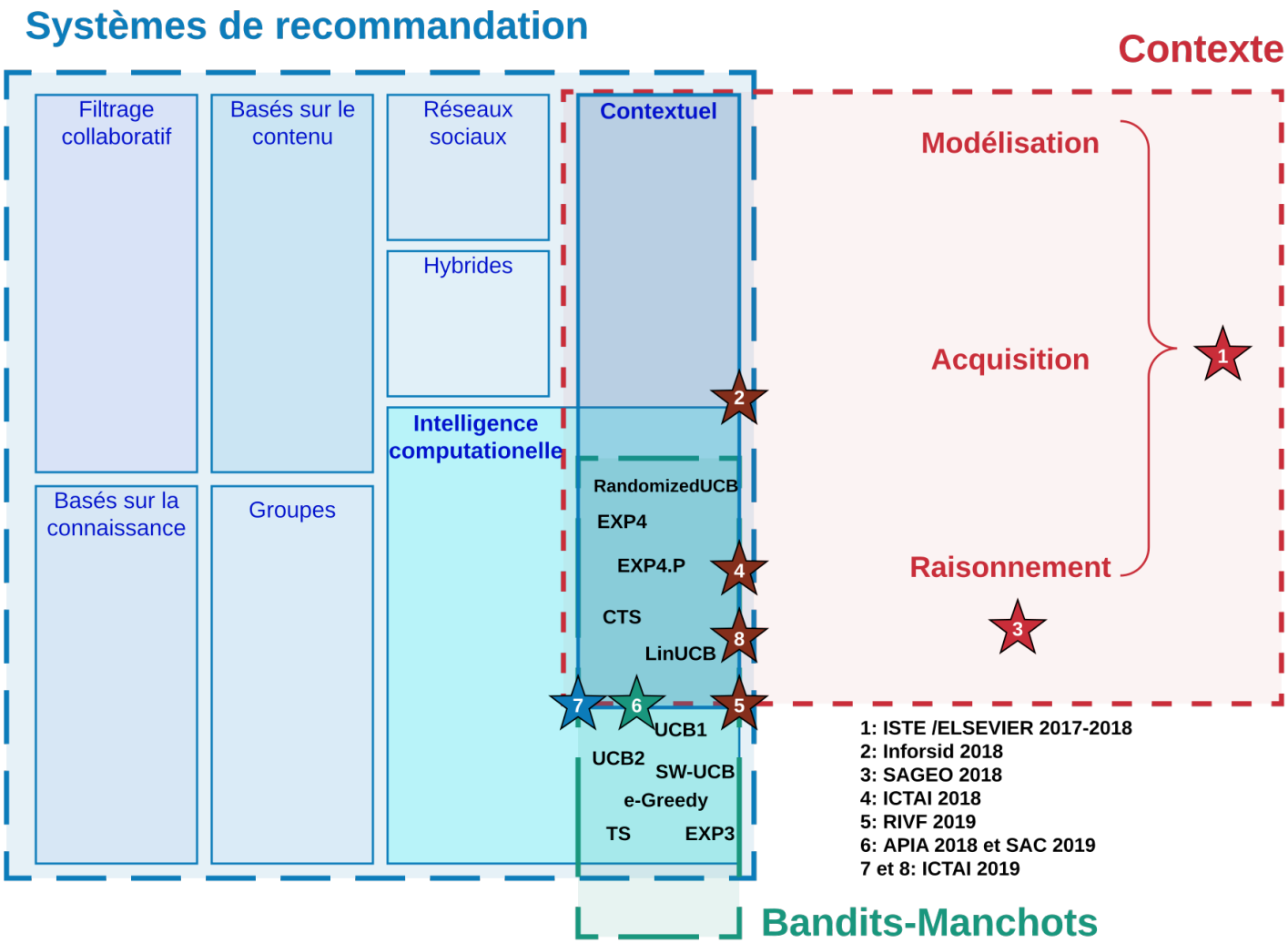


FIGURE 2 – Plan de la partie état de l'art et positionnement de nos contributions par rapport aux domaines.

Description de la Figure 2

Dans cette sous-section, nous détaillerons la Figure 2 et ses différentes aires représentant les trois chapitres de notre État de l'art.

Important : la surface des aires dans cette figure n'illustre pas l'importance des domaines qu'elles représentent vis à vis de la littérature, elle met en lumière les axes sur lesquels portent les contributions de cette thèse.

Systèmes de recommandation (Aire bleue)

Nous représentons les systèmes de recommandation selon les cinq principales approches suivantes :

1. Les systèmes de recommandation s'appuyant sur le filtrage collaboratif ;
2. Les systèmes de recommandation basés sur le contenu ;
3. Les systèmes de recommandation basés sur la connaissance ;
4. Les systèmes de recommandation contextuels ;
5. Les systèmes de recommandation à des groupes d'utilisateurs.

Chacune de ces trois approches peut être mise en œuvre dans trois types de systèmes que nous positionnons de manières « transversales » :

1. Les systèmes utilisant des techniques basées sur des modèles (*Computational Intelligence (CI)* : Intelligence Computationnelle (IC)) ;
2. Les systèmes utilisant des techniques basées sur des heuristiques ;
3. Les systèmes hybrides combinant des techniques variées.

Contexte (Aire rouge)

Le contexte est abordé selon 3 grandes thématiques de recherche :

1. La modélisation du contexte ;
2. L'acquisition du contexte ;
3. Le raisonnement contextuel.

Les bandits-manchots (Aire verte)

L'une des mises en œuvre (basée sur les modèles) des systèmes de recommandation consiste à poser le problème sous la forme de bandits-manchots et à utiliser les algorithmes adaptés pour le résoudre. Il existe de nombreux algorithmes permettant de résoudre le problème de bandits-manchots, soit contextuels, soit non contextuels. Aussi, nous avons positionné les principaux algorithmes de la littérature selon leur prise en compte ou non du contexte.

Pour illustrer cela, à la Figure 2 nous avons mis en exemple des algorithmes de bandits-manchots contextuels et non contextuels que nous aborderons au Chapitre 2 de l'état de l'art. Ainsi *UCB* [Aue02] et *EXP3* [Aue+02] sont hors de la zone de contexte car non contextuel tandis que *LinUCB* [Li+10] et *EXP4* [Aue+02] sont dans la zone de contexte car ce sont des algorithmes de bandits-manchots prenant en compte les informations contextuelles.

Intersections entre les différentes aires

Les principales contributions de cette thèse se situent à **la convergence** entre les **systèmes de recommandation**, le domaine du **contexte** et les **bandits-manchots** puisque nous abordons le problème de recommandation contextuelle comme un problème de bandit-manchot.

Positionnement de nos publications (Étoiles)

L'étoile verte numérotée 6 fait référence à nos articles des conférences *PFIA APIA 2018* [Gut+18a] et *ACM SAC 2019* [Gut+19a] c.-à-d., contributions sur les algorithmes de bandits-manchots contextuels basés sur un modèle linéaire et sur la mesure de diversité et de précision individuelle des recommandations. Ces publications décrivant des travaux orientés uniquement sur les modèles de bandits-manchots (contextuels et non contextuels), nous avons donc positionné l'étoile numéro 6 à la frontière entre l'aire bandits-manchots contextuels et non contextuels. Les étoiles marrons [3] numérotées 2, 4 et 5 font référence à des publications contribuant à l'amélioration du contexte pour les bandits-manchots contextuels. L'étoile numéro 2 fait référence au travail que nous avons présenté lors de l'atelier « Data Intelligence » de la conférence *Inforsid 2018* [Gut+18e] c.-à-d., contribution à l'acquisition du contexte via l'application mobile *scéno* du projet *EVENT-AI* et contribution à l'utilisation d'algorithmes de bandits-manchots pour la recommandation en ligne. L'étoile numéro 4 fait référence à notre publication à *IEEE ICTAI 2018* [Gut+18b] et se situe donc à la frontière entre les bandits-manchots contextuels et le raisonnement contextuel c.-à-d., contribution au raisonnement contextuel via une technique de déduction de contexte (*ICE*) pouvant se rapprocher du *LAD* et de techniques de déduction des préférences utilisateurs. L'étoile numéro 5 fait référence à notre publication à *IEEE RIVF 2019* [Gut+19d] et se situe donc à la frontière entre les bandits-manchots contextuels et non contextuels ainsi que le raisonnement contextuel c.-à-d., contribution qui utilise la méthode présentée dans la publication *IEEE ICTAI 2018* [Gut+18b] pour transformer un problème qui à la base est non contextuel, en problème contextuel. Ceci est notamment possible via un démarrage avec des bandits-manchots non contextuel combinés à notre méthode *ICE* et une poursuite du processus de recommandation via des bandits-manchots contextuels traditionnels.

Les étoiles rouges numérotées 1 et 3 quant à elles font références à des publications sur le contexte et se situent de ce fait totalement dans l'aire « Contexte » (aire rouge). La contribution numéro 1 fait référence à la fois à notre article de revue [Gut+17] publié dans le journal *From*

3. combinaisons d'une contribution à la fois pour le contexte (aire rouge) et pour les bandits-manchots (aire verte)

Data To Decision, ISTE Ltd.) et à notre chapitre de livre [Gut+18d]. Ces publications contribuent à la fois sur la partie modélisation (via la proposition d'une représentation du contexte pour des utilisateurs mobiles), sur la partie acquisition (via la capture de journaux Wi-Fi et leur traitement), et sur le raisonnement contextuel (via la modélisation du problème sous forme de chaînes de Markov dont le traitement a pour objectif la prédiction de déplacement). L'étoile numéro 3 est proche de la partie raisonnement de l'aire de contexte car fait référence à l'article de l'atelier *EXCES* de la conférence *SAGEO 2018* [Gut+18c] dont les travaux portent sur le géo-partitionnement (*geo-clustering*) (c.-à-d., contribution au raisonnement contextuel via une technique d'apprentissage non supervisée).

L'étoile bleue numérotée 7 fait référence à l'une de nos deux publications à *IEEE ICTAI 2019* [Gut+19b] portant sur une approche porte-folio d'algorithmes de bandits-manchots non contextuels et contextuels. Elle a pour objectifs de maximiser à la fois la précision globale et la diversité pour améliorer la précision individuelle. Cette méthode est une alternative à notre contribution correspondante à l'étoile 6, voire peut se substituer à tout autre méthode de bandits-manchots pourvu qu'elle l'intègre dans son porte-feuille. Elle permet ainsi d'aborder le problème de recommandation sous une forme plus générique et d'avoir une meilleure maîtrise des critères de performance.

Enfin, l'étoile marron numérotée 8 est une dernière contribution faisant référence à notre seconde publication à *IEEE ICTAI 2019* [Gut+19c] portant sur une évaluation en ligne d'algorithmes de bandits-manchots intégrés dans l'application mobile *scéno* pour la recommandation d'événements culturels. L'objectif de cette contribution est de mettre en évidence l'importance de prendre en compte le contexte spatial dans les systèmes de recommandation employant des algorithmes de bandits-manchots contextuels. Nous y évaluons quatre méthodes : 1) Un algorithme de recommandation aléatoire ; 2) Un algorithme de bandit-manchot non contextuel intitulé ε-*Greedy* ; 3) Un algorithme de bandit-manchot contextuel *LinUCB* opérant sur un contexte dépourvu d'information spatiale ; 4) Un algorithme de bandit-manchot contextuel *LinUCB* opérant sur un contexte enrichit des informations spatiales déduites par notre algorithme de partitionnement spectral (étoile n° 3).

Organisation du mémoire

Liens vers les sites web et applications mobiles

Ce mémoire fait référence à un certain nombre de sites web et d'applications mobiles.

Les liens de ces sites et applications sont donnés « *directs* » dans le corps du mémoire, c'est-à-dire selon leur accessibilité en l'an 2019. En revanche, afin de pérenniser leur consultation, chacun de ces liens est répertorié « *archivé* » dans l'Annexe E. Ainsi, les liens se rapportant à notre site `http://wifilib-clustering.info/` sont disponibles dans le Tableau E.1, les liens concernant les applications mobiles sont consultables dans le Tableau E.3, et enfin les liens concernant les sites de référence sont répertoriés dans le Tableau E.2.

Organisation du manuscrit

Le mémoire est organisé de la façon suivante :

1. Partie I : **État de l'art**
   Cette partie composée de trois chapitres établit l'état de l'art correspondant aux trois grands axes de recherche abordés dans cette thèse : 1) les systèmes de recommandations ; 2) les algorithmes de bandits-manchots pour la recommandation ; 3) le contexte.
   — Chapitre 1 : **Les systèmes de recommandation**
   Ce chapitre dresse un état de l'art des principales approches employées par les systèmes de recommandation et décrit une analyse comparative ciblée sur nos besoins. Ainsi, dans ce chapitre nous y abordons les systèmes de recommandation : s'appuyant sur le filtrage collaboratif ; basés sur le contenu ; basés sur les connaissances ; employés pour des groupes d'utilisateurs ; et enfin ceux s'appuyant sur le contexte. Enfin, nous comparerons ces différentes approches afin de déduire celles qui correspondraient le mieux à notre cas de recommandation à des utilisateurs mobiles ;
   — Chapitre 2 : **Les bandits-manchots pour la recommandation**
   Ce chapitre est consacré à un état de l'art sur les algorithmes de bandits-manchots pour la recommandation. Nous y définissons le problème pour le cas contextuel et non contextuel, et rappelons les principaux algorithmes de la littérature permettant de le résoudre. Enfin, nous établissons une étude comparative de chacun de ces algorithmes selon des critères de précision et de personnalisation, d'applicabilité et de contraintes de temps-réel, et de résistance à des environnements non stationnaires ;
   — Chapitre 3 : **Le contexte**
   Ce chapitre est dédié à l'état de l'art sur la modélisation, l'acquisition et le raisonnement contextuel. Nous y rappelons les différentes modélisations proposées pour le contexte en ce qui concerne les applications *MCSC*. Nous revenons sur les notions d'acquisition du contexte et les différentes manières de le capturer. Enfin, comme il est incontournable de fournir du contexte facilement exploitable par des systèmes de recommandation nous y décrivons les principales méthodes de raisonnement contextuel.

2. Partie II : **Contributions aux algorithmes de bandits-manchots pour la recommandation**
   Cette partie composée de deux chapitres correspond à nos contributions sur les algorithmes de bandits-manchots et plus particulièrement l'étude de la performance de ces algorithmes selon trois critères : la précision globale ; la diversité des recommandations ; la précision individuelle.
   — Chapitre 4 : **Algorithmes de bandits-manchots : une histoire de précision**
   Ce chapitre décrit notre étude préliminaire de sept algorithmes de la littérature sur douze jeux de données et selon les critères de précision globale, de diversité, et de précision individuelle. La première contribution de cette étude est la mise en place

d'un nouveau mode de calcul de la diversité des recommandations et de l'analyse d'un nouvel indicateur : la précision individuelle. La seconde contribution de cette étude correspond à l'analyse en elle-même qui démontre que chaque algorithme obtient des résultats plus ou moins différents en fonction du jeu de données sur lequel il est évalué, et ce sur les trois critères. Ce chapitre étend les publications effectuées en conférences françaises et internationales [Gut+18a ; Gut+19a] ;

— Chapitre 5 : **Améliorer la précision individuelle : diversifier les recommandations**

Ce chapitre est consacré à deux contributions présentant chacune à leur manière comment mieux diversifier les recommandations faites aux utilisateurs et comment cela a un impact positif sur la précision individuelle. Dans la première contribution nous opérons une modification sur un algorithme de bandits-manchots contextuels des plus populaires dans la littérature : *LinUCB*. Nous lui appliquons un mécanisme de diversification de ses recommandations par l'usage d'une fenêtre glissante pénalisant les éléments qui ont été précédemment trop recommandés. Cette nouvelle méthode *SW-LinUCB* permet à la fois de mieux diversifier mais aussi de diminuer les utilisateurs pour lesquels la précision des recommandations qui leurs sont faites est très faible. Cette contribution fait référence aux publications en conférences françaises et internationales [Gut+18a ; Gut+19a]. La seconde contribution : *Gorthaur*, est une approche de type porte-folio d'algorithmes qui sélectionne les algorithmes de bandits-manchots maximisant à la fois la précision globale et la diversité des recommandations. L'avantage de cette méthode est qu'elle nous émancipe de devoir choisir manuellement le bon algorithme en amont du projet sans connaissances a priori d'un problème donné. Cette contribution fait référence à l'une de nos deux publications à *IEEE ICTAI 2019* [Gut+19b].

3. Partie III : **Contributions à l'élaboration du contexte pour les algorithmes de recommandation**

Cette partie composée de deux chapitres correspond à nos contributions sur l'élaboration du contexte pour les algorithmes de recommandation. Elle décrit dans un premier temps comment nous modélisons le contexte, et comment nous l'acquérons sous sa forme brute. Dans un second temps, nous détaillons comment nous raisonnons sur ce contexte brut afin de pouvoir le fournir à nos algorithmes de recommandation contextuelle sous une forme structurée et exploitable.

— Chapitre 6 : **Modélisation, capture et analyse du contexte pour la recommandation**

Ce chapitre est dédié à une étude préliminaire du contexte dans la ville d'Angers et a donné lieu à des contributions sur la modélisation, la capture du contexte et son analyse [Gut+17 ; Gut+18d ; Gut+18e]. Après avoir rappelé le contexte technologique de *Mobile Crowd Sensing and Computing (MCSC)* dans lequel s'inscrit ce type d'application, nous proposons une modélisation du contexte dans le cadre de la recommandation dans la ville intelligente, et nous analysons les informations contex-

tuelles capturées à partir de journaux de connexions au Wi-Fi urbain de la ville d'Angers. Nous décrivons également le composant mobile de capture de contexte de notre application *scéno*. Nous expliquons enfin comment le contexte capturé peut être exploité par les systèmes de recommandation ;

— Chapitre 7 : **Raisonnements contextuels et application aux systèmes de recommandation**
Ce chapitre décrit comment il est possible de raisonner sur des données contextuelles brutes issues des journaux de connexions au Wi-Fi urbain. Nous y présentons une analyse des trajectoires, une méthode de prédiction de la mobilité, et une méthode de déduction de quartiers dans la ville. Pour ce dernier cas, nous employons un algorithme de partitionnement spectral [Cra+12] et nous décrivons concrètement dans ce chapitre comment nous utilisons les quartiers inférés en tant que données de contexte directement dans le système de recommandation de l'application mobile *scéno*. Nous y présentons les résultats de précision globale obtenus par le système de recommandation qui semble ainsi tirer parti des quartiers inférés au préalable. Ces résultats bien que positifs sont très récents et n'ont pas encore été présentés dans le cadre d'une communication scientifique. En revanche les trois méthodes de raisonnement contextuel présentées dans ce chapitre ont été revues et acceptées, pour certaines, en atelier de conférences francophones [Gut+18c], d'autres en article de journal [Gut+17] étendu en chapitre de livre (international) [Gut+18d]. La contribution sur le géo-partitionnement (*geo-clustering*) fait également référence à notre seconde publication à *IEEE ICTAI 2019* [Gut+19c].

4. Annexes : ce mémoire comporte 5 annexes permettant de mieux préciser ou détailler certains chapitres.
   — l'Annexe A permet de consulter différents pseudo-codes des algorithmes de bandits-manchots présentés dans le Chapitre 2 de la partie état de l'art ;
   — l'Annexe B présente une analyse détaillée de l'étude préliminaire résumée dans le Chapitre 4 ;
   — l'Annexe C correspond à l'ensemble des tableaux de résultats de l'étude préliminaire présentée au Chapitre 4 ;
   — l'Annexe D propose différents compléments d'information sur les données *Wifilib* et l'application *scéno* abordées dans les Chapitres 6 et 7 ;
   — l'Annexe E répertorie les liens archivés de l'ensemble des sites web cités dans ce mémoire.

PREMIÈRE PARTIE

État de l'art

Introduction de la partie I

Le sujet de cette thèse porte plus particulièrement sur les algorithmes de bandits-manchots pour de la recommandation à des utilisateurs mobiles dans la ville intelligente. Cette première partie sera donc l'opportunité de faire référence à de nombreux travaux connexes portant sur les trois dimensions abordées en référence à ce sujet :

1. Les systèmes de recommandation ;
2. Les algorithmes de bandits-manchots non contextuels et contextuels ;
3. Le contexte.

Chacune de ces dimensions représente un vaste domaine de recherche en particulier.

Concernant les systèmes de recommandation, à partir des années 1990 où la première approche de filtrage collaboratif vit le jour [Gol+92 ; Res+94], de nombreuses autres méthodes se sont développées [RRS15] et nécessitent aujourd'hui la prise en compte d'un environnement utilisateur toujours plus dynamique. Ainsi, dans le premier chapitre, après y avoir rappelé l'histoire et les enjeux des systèmes de recommandation, nous effectuons un état de l'art sur les principales méthodes employés dans ces systèmes. Nous effectuons également une étude comparative de ces approches afin de sélectionner celle correspondant le mieux à nos besoins applicatifs.

En ce qui concerne les algorithmes de bandits-manchots, de nombreuses études ont déjà été menées dont certaines datant même de 1933 [Tho33] ou encore 1952 [Rob52]. Néanmoins, ce n'est que dans les années 1980 [LR85], 1990 [SB98] et plus particulièrement les années 2000/2010 [AG13 ; Aue02 ; Bou+17 ; Gre+17 ; LZ08 ; Li+10] que les algorithmes de bandits-manchots prennent véritablement leur essor tant les puissances de calcul se sont accrus. Ainsi, dans le second chapitre, après avoir défini le problème du bandit-manchot pour la recommandation dans un cadre contextuel et non contextuel, nous rappelons les principaux algorithmes permettant de le résoudre. De plus, nous effectuons une étude comparative théorique de ces différents algorithmes afin de sélectionner ceux qui correspondraient le mieux à notre cas d'étude.

Enfin, dans le cadre de la ville intelligente, il est nécessaire de prendre en considération le contexte. Celui-ci, très clairement défini par [Dey01] en 2001 et précisément formalisé par [ZLO07] en 2007, peut être considéré comme une pièce maîtresse dans le problème de recommandation tant il en contraint la résolution [Bré99]. Ainsi, dans le troisième chapitre, nous aborderons la notion de contexte, qui, dans les systèmes de recommandation et pour les algorithmes employés dans ces systèmes représente une source incontournable d'informations qu'il faut savoir capturer, transformer et exploiter à bon escient.

CHAPITRE 1

LES SYSTÈMES DE RECOMMANDATION

About Recommender Systems...

« *The term now has a broader connotation, describing any system that produces individualized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options.* » [Bur02]

Robin Burke - 2002

Sommaire

1.1 Introduction

De nos jours, les systèmes de recommandation sont de plus en plus utilisés pour suggérer p. ex., des produits aux clients de sites du commerce électronique, du contenu web, des films ou encore des informations de voyage personnalisées à des touristes [Bao+15 ; Gav+14 ; Lev+12 ; Li+10 ; LKG16]. Ils aident ainsi les utilisateurs à faire leur choix lorsqu'ils sont confrontés à un trop grand volume d'information et ce dans des domaines variés.

Vu sous l'angle du e-Commerce : « *Recommendation Systems are software agents that elicit the interests and preferences of individual consumers [...] and make recommendations accordingly. They have the potential to support and improve the quality of the decisions consumers make while searching for and selecting products online.* » [XB07]

Littéralement : *Les systèmes de recommandation sont des agents logiciels qui extraient individuellement les intérêts et les préférences de consommateurs [...] et produisent des recommandations en conséquence. Ils ont vocation à soutenir et à améliorer la qualité de la décision prise par les consommateurs lors de la recherche et de la sélection de produits en ligne.*

À la lumière de ces considérations, nous représentons les bases du fonctionnement général des systèmes de recommandation à travers la Figure 1.1. Cette figure a été élaborée entre autres en considérant les explications de [JF13].

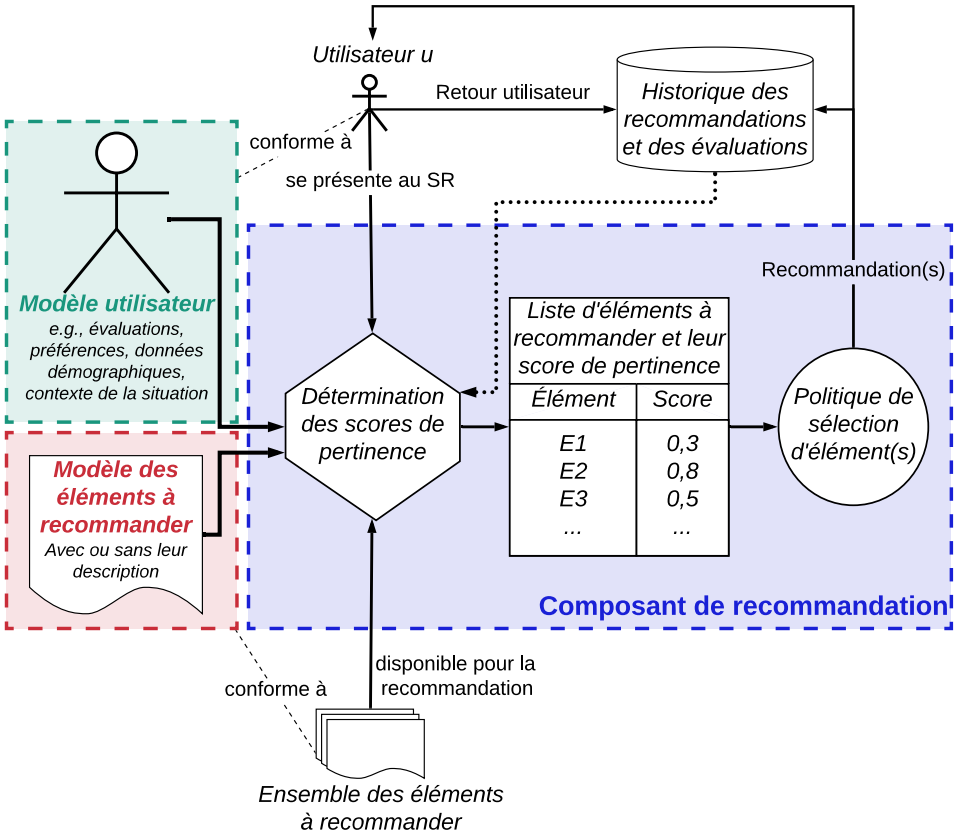


FIGURE 1.1 – Principe général de fonctionnement d'un Système de Recommandation (SR)

Ainsi, d'après [JF13] et [XB07], on peut définir qu'un système de recommandation est un composant logiciel classé dans la catégorie des outils d'aide à la décision. Celui-ci peut être vu comme une fonction ayant pour but de recommander un ou plusieurs éléments à un ou plusieurs utilisateurs (individu ou groupe). L'objectif principal d'un système de recommandation est de proposer les recommandations les plus pertinentes possibles.

Pour ce faire, un système de recommandation peut tirer partie de différentes données en entrée (p. ex., profil utilisateur, préférences, paramètres de contexte) afin de produire une ou plusieurs recommandations pertinentes en sortie.

En entrée d'un système de recommandation on peut retrouver (Voir Figure 1.1) :

— les données inhérentes à l'utilisateur défini selon un *modèle utilisateur* (p. ex., évaluations données aux recommandations précédentes, préférences, données démographiques et contexte de la situation) ;
— les données inhérentes aux éléments qu'on souhaite recommander c.-à-d., les noms des éléments à recommander complétés éventuellement par la description de leurs caractéristiques.

En sortie d'un système de recommandation on peut retrouver (Voir Figure 1.1) : le ou les éléments à recommander (p. ex., services et événement culturels) que le système considère comme étant le ou les plus pertinents à proposer (voir Section 2.4).

Les systèmes de recommandation peuvent principalement différer selon trois points [JF13 ; Lu+15 ; Neg15 ; RRS15] :

— la méthode qu'ils emploient pour la fonction d'évaluation de pertinence des éléments à recommander c.-à-d., le calcul du score de pertinence ;
— la politique de sélection du ou des éléments à recommander ;
— les données qu'ils utilisent en tant que paramètres d'entrée de la fonction d'évaluation.

Dans tous les cas, afin d'évaluer la précision des recommandations effectuées par un système, il sera nécessaire de garder un historique des recommandations qui ont été réalisées ainsi que les retours utilisateurs (p. ex., clics, évaluations) vis à vis de ces recommandations.

De plus, selon l'approche de recommandation employée, il sera possible de considérer d'autres informations comme [JF13 ; RRS15] :

— l'historique des retours utilisateurs qui peut avoir un impact sur les scores de pertinence calculés. Cet impact peut être *direct* p. ex., le retour utilisateur peut être pris en compte directement dans le calcul de l'espérance de pertinence de chaque recommandation à effectuer comme c'est le cas pour les systèmes de recommandation utilisant des algorithmes de bandits-manchots contextuels [Li+10]. Cet impact peut être également *indirect* p. ex., le retour utilisateur peut être pris en compte dans la déduction des profils utilisateurs comme c'est le cas pour les systèmes de recommandation dont l'approche est basée sur le contenu [RRS15] ;
— les données des communautés d'utilisateurs utilisant le système. Ces données sont entre autres utilisées dans le cadre du filtrage collaboratif [RRS15] (voir Section 1.4 et Figure 1.3) ;
— des modèles de connaissances. Ces modèles sont entre autres utilisés dans le cadre de l'approche basée sur les connaissances [RRS15] (voir Section 1.6.2 et Figure 1.6).

De même, selon l'approche de recommandation utilisée on pourra s'appuyer sur des heuristiques, ou sur des modèles via l'usage de techniques d'intelligence computationnelle *(Computational Intelligence)*. L'intelligence computationnelle est définie par [Kon06] qui rappelle les techniques qu'elle englobe. Dans ce chapitre dédié aux systèmes de recommandation, le terme d'intelligence computationnelle fera référence à toutes les techniques qui peuvent être employées par les systèmes de recommandation s'appuyant sur les modèles. L'Intelligence Computationnelle et ses techniques seront détaillées dans la Section 1.10.

Pour conclure, il est important de rappeler que la pertinence des recommandations produites en sortie peut dépendre d'autres facteurs [JF13] dont :

— le contexte de l'entité p. ex., le temps, la localisation, les relations, l'individualité et l'activité ;
— la diversité des éléments à recommander. La diversité peut prendre une part importante dans la perception de la précision du système par les utilisateurs p. ex., éviter les recommandations redondantes, réussir une recommandation par sérendipité.

Dans ce chapitre, après avoir rappelé un bref historique des systèmes de recommandation, nous énumérerons les différentes approches existantes et leurs principes généraux à la Section 1.3. Ensuite, nous évoquerons en détail ce qui les rapproche, ce qui les différencie et comment certaines peuvent être combinées. Pour cela, la Figure 1.1 générique sera complétée selon les spécificités de chacune de ces approches. Enfin, nous terminerons ce chapitre par un rappel de différentes techniques d'intelligence computationnelle utilisées pour les systèmes de recommandation.

1.2 Historique

L'origine exacte des premiers travaux sur les systèmes de recommandation est sujet à discussion.

D'aucuns reconnaissent le système de [Ric79] comme le premier système de recommandation automatique de l'histoire [Ben17]. Ce système avait pour objectif de recommander des livres d'une bibliothèque. Pour ce faire, il collectait des informations sur les utilisateurs sous forme d'interviews, puis classait les utilisateurs en *stéréotypes*. Le système produisait ensuite des recommandations à partir de ces *stéréotypes*.

Néanmoins, bien que ce système primitif ait pu ouvrir la voie à un domaine aujourd'hui riche et varié, le véritable essor des systèmes de recommandation est reconnu comme datant du début des années 1990. À cette époque, l'utilisation croissante du web en tant que support pour les transactions électroniques et commerciales a été un véritable vecteur de développement des systèmes de recommandation.

La première approche employée pour ces systèmes fût le filtrage collaboratif [Gol+92], utilisé pour de la recommandation de documents. Cette approche est toujours l'une des plus utilisée et suscite de nombreuses recherches que nous décrivons dans la Section 1.3. Cette même année 1992 vit la création du laboratoire de recherche *GroupLens*[1] qui reste incontournable dans le domaine. Ils ont publié une architecture utilisant le filtrage collaboratif afin de recommander des articles (de type *news*) sur Internet [Res+94]. Ce système se basait sur les notes et actions passées d'une communauté d'utilisateurs du site web *Usenet* afin de fournir des résultats personnalisés en s'appuyant sur des similarités entre utilisateurs. Ceci correspond au fondement du filtrage collaboratif.

1. Département d'informatique et d'ingénierie de l'Université du Minnesota, États-Unis.

Dans les années qui ont suivi, de nombreux systèmes se sont développés sur cette base que ce soit pour de la recommandation de musique [SM95] ou de vidéos [Hil+95].

De nos jours, de nombreuses autres approches concernant les systèmes de recommandation ont été adaptées à différentes problématiques et à différents domaines métiers. Dans tous les cas, ce qui reste indéniable, c'est que désormais les systèmes de recommandation sont devenus incontournables dans tout domaine d'application que ce soit web ou mobile.

Dans la section suivante, nous rappellerons les différentes approches pour les systèmes de recommandation.

1.3 Les approches

Afin de répondre aux différents enjeux de la recommandation, il existe différentes approches spécifiques dont chacune avec ses propres limites et axes d'améliorations. Néanmoins, il n'existe pas aujourd'hui un consensus permettant de les regrouper sur une même représentation. C'est pourquoi nous proposons d'illustrer sur la Figure 1.2 les principales approches que nous avons identifiées à partir de [JF13 ; Lu+15 ; Neg15 ; RRS15].

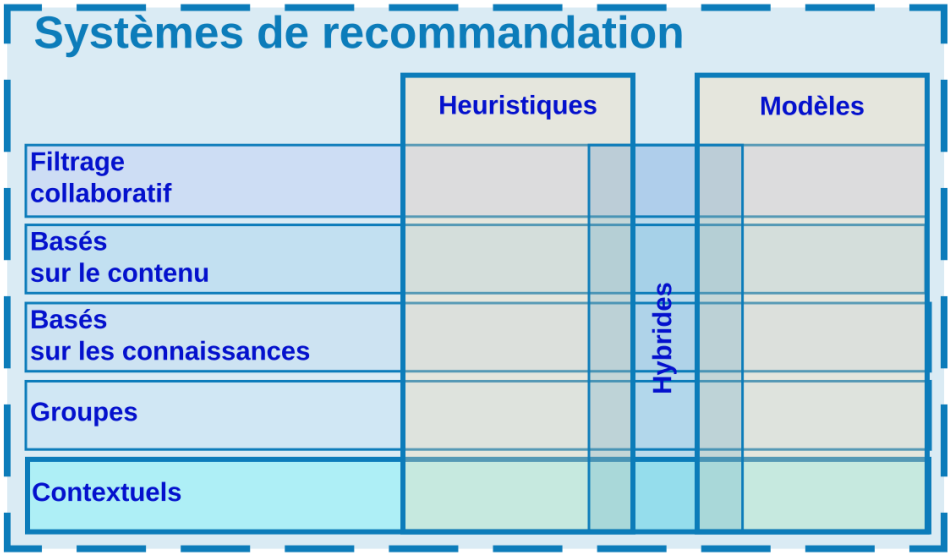


FIGURE 1.2 – Approches utilisées par les systèmes de recommandation

Les cinq principales approches identifiées à partir de ces travaux sont :

1. Les systèmes de recommandation s'appuyant sur le filtrage collaboratif ;
2. Les systèmes de recommandation basés sur le contenu *(Content-Based)* ;
3. Les systèmes de recommandation basés sur les connaissances *(Knowledge-Based)* ;
4. Les systèmes de recommandation à des groupes d'utilisateurs ;
5. Les systèmes de recommandation contextuels.

De plus, ces mêmes travaux considèrent d'autres possibilités dites « transversales ». Ils mettent notamment en lumière la possibilité de pouvoir hybrider les approches et de combiner des données d'entrée ou des méthodes. Désormais, nous considérons comme la majorité de la communauté, le fait d'hybrider comme étant une approche à part entière et la dénommons « approche hybride ».

Enfin, pour chacune des approches il existe une manière de la traiter soit via des heuristiques, soit via des modèles en utilisant des techniques d'Intelligence Computationnelle (*Computational Intelligence*). Par exemple, [RRS15] présente les approches basées sur les modèles les opposant à celles basées sur les heuristiques.

Les premiers systèmes de recommandation s'appuyaient sur des heuristiques [Res+94]. Néanmoins au vu de l'importance qu'ont gagné les systèmes basés sur les modèles, certains considèrent même que le fait d'utiliser de l'Intelligence Computationnelle est une approche à part entière [Lu+15].

Dans ce chapitre, après avoir rappelé les définitions et les différentes approches existantes pour les systèmes de recommandation, nous présenterons différentes techniques d'Intelligence Computationnelle employées dans les systèmes de recommandation. Nous illustrerons également comment l'Intelligence Computationnelle peut s'inscrire en tant que composant dans les systèmes de recommandation.

À la section suivante, nous présentons l'approche de filtrage collaboratif.

1.4 Le filtrage collaboratif

En résumé : le filtrage collaboratif (*Collaborative Filtering - CF*) permet de recommander un élément à une personne en se basant sur les opinions d'autres personnes ayant des centres d'intérêts similaires [DK04]. De ce fait, deux approches peuvent être utilisées dans le cadre du *CF* à savoir : le filtrage basé sur les utilisateurs ou le filtrage basé sur les éléments à recommander [Sar+01]. Afin de constituer ces filtrages, plusieurs types de calculs de similarité peuvent être utilisés : le calcul du coefficient de corrélation de *Pearson* [OHS05 ; Res+94], ou encore la mesure de similarité de *cosinus* [Bao+13] parfois combinée à celle de *Jaccard* [AC17]. Une approche alternative est basée sur les modèles de facteurs latents. Le principe de cette approche est de décrire les utilisateurs et les éléments à recommander par un petit ensemble de facteurs latents (facteurs non explicites) pouvant être utilisés pour prédire les notes que devrait donner chaque utilisateur à chaque élément. Il est ainsi possible d'utiliser des méthodes, telles que la factorisation matricielle [SZ16] (*aka*, *Singular Value Decomposition - SVD*), qui ont pour objectif de rechercher des facteurs latents (espace latent), en nombre relativement faible (généralement de l'ordre de 10^2), qui « expliquent » les notations des utilisateurs vis à vis des éléments (c.-à-d., matrice de données issue des notes ou choix passés). Ceci permet ainsi de prédire les entrées (c.-à-d., évaluations) manquantes.

1.4.1 Les bases de fonctionnement

Dans cette sous-section, nous présentons les bases du fonctionnement de systèmes de recommandation basés sur une approche de filtrage collaboratif. Nous les illustrons à la Figure 1.3.

La question de l'utilisateur à laquelle doit répondre un système de recommandation reposant sur une approche de filtrage collaboratif peut se résumer ainsi : « Dites-moi ce qui est populaire parmi mes pairs » [JF13].

Ainsi, en complément des composants présents sur la Figure 1.1 représentant le fonctionnement générique d'un système de recommandation, la Figure 1.3 intègre des données dites communautaires.

Au sein même du composant de recommandation, les scores de pertinence sont déterminés en calculant la similarité entre les préférences d'un utilisateur et celles d'autres utilisateurs [Neg15]. Sur la base de ces scores, il devient possible de produire des recommandations selon une politique visant la sélection du ou des éléments ayant le score le plus élevé.

Nous nous proposons de décrire plus en détail dans la sous-section suivante le mode de fonctionnement du filtrage collaboratif :

— selon deux approches possibles p. ex., méthode de voisinages ou avec des modèles à facteurs latents ;
— selon deux variantes des mesures de similarité p. ex., similarité entre utilisateurs ou entre éléments.

1.4.2 Les méthodes de filtrage collaboratif

Les méthodes de filtrage collaboratif *(CF)*, selon [RRS15], produisent des recommandations spécifiques d'éléments, basées sur des modèles de notation ou d'évaluation, ou d'utilisation (p. ex., des achats passés). Ainsi, afin de produire des recommandations, les systèmes utilisant le *CF* doivent mettre en relation deux entités différentes de base qui sont les éléments et les utilisateurs. Pour ce faire, il existe principalement deux approches :

— l'approche dite de voisinages ;
— l'approche dite de modèles à facteurs latents.

1.4.3 Les méthodes de voisinages

Les méthodes de voisinages se concentrent sur les relations entre les éléments ou, alternativement, entre les utilisateurs [DK04]. Par exemple, une approche élément-élément [Sar+01] modélise la préférence d'un utilisateur pour un élément en fonction des notations données, par ce même utilisateur, à d'autres éléments similaires. L'un des points cruciaux est donc la mesure de similarité.

La mesure de similarité s'appuie le plus souvent sur le coefficient de corrélation de *Pearson* [OHS05 ; Res+94 ; RRS15 ; Sch+07], qui peut être utilisé pour mesurer, par exemple, la corrélation $\widehat{\rho}_{ij}$ entre deux éléments. Ainsi, dans le cas d'une comparaison entre deux éléments i et

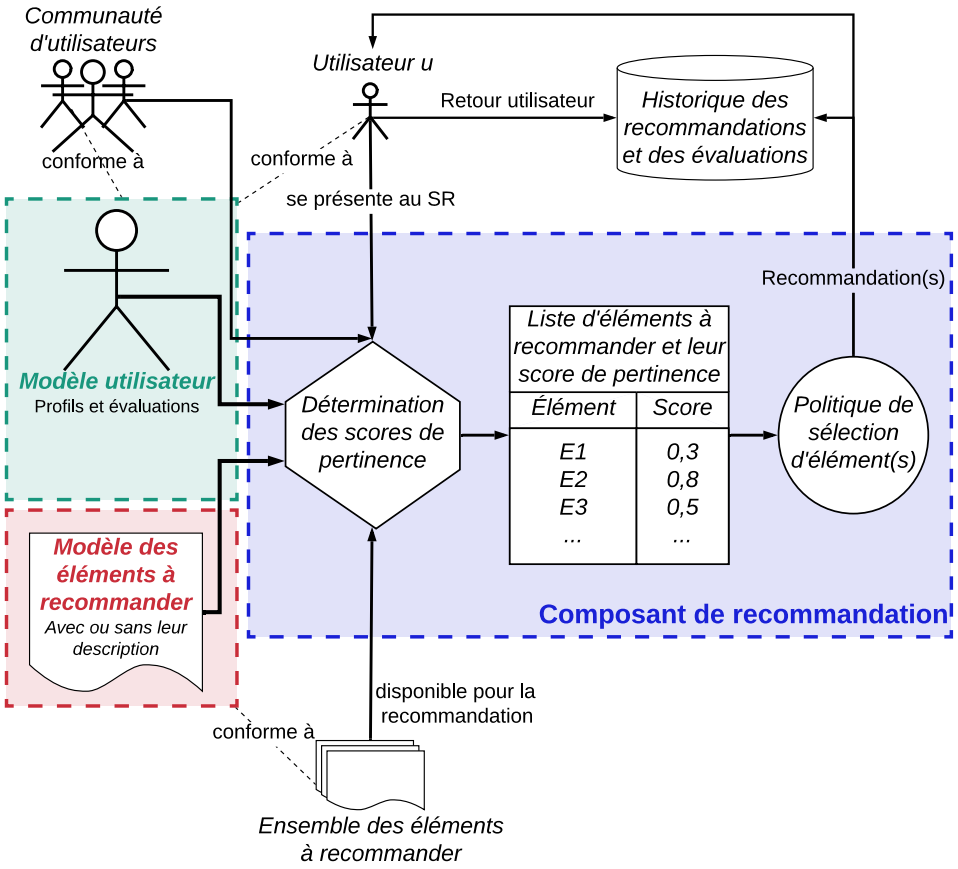


FIGURE 1.3 – Principe général du fonctionnement d'un système de recommandation utilisant les méthodes de filtrage collaboratif

j, une première définition [Ben17] de la similarité (*PCI - Pearson Correlation between Items*) est :

$$PCI(i,j) = \widehat{\rho}_{ij} = \frac{\sum_{u \in U(i,j)} (r_{ui} - \bar{r_i}) \times (r_{uj} - \bar{r_j})}{\sqrt{\sum_{u \in U(i,j)} (r_{ui} - \bar{r_i})^2 \times \sum_{u \in U(i,j)} (r_{uj} - \bar{r_j})^2}}$$

où $\bar{r_i}$ est la moyenne des notes reçues par l'item i, r_{ui} est l'évaluation donnée à i par u, $U(i,j)$ contient les utilisateurs ayant évalué les éléments i et j. Puisque les corrélations estimées basées sur un plus grand nombre d'utilisateurs sont plus fiables, il est nécessaire d'établir une mesure de similarité appropriée en fonction de ce nombre d'utilisateurs, en calculant un coefficient de corrélation rétréci (*SCCI - Shrunk Correlation Coefficient between Items*) de la forme suivante :

$$SCCI(i,j) = \frac{n_{ij} - 1}{n_{ij} - 1 + \lambda_1} \widehat{\rho}_{ij}$$

La variable $n_{ij} = |U(i,j)|$ désigne le nombre d'utilisateurs qui ont noté i et j. λ_1 est un coefficient de rétrécissement dont la valeur constante est généralement fixée à 100 [RRS15]. Ainsi,

avec ce mode de calcul, plus le nombre d'utilisateurs est grand plus on accordera de confiance à $\widehat{\rho}_{ij}$, a contrario, plus le nombre d'utilisateurs est faible, plus on atténuera la confiance que l'on accorde à $\widehat{\rho}_{ij}$.

Le calcul du coefficient de corrélation de *Pearson* peut aussi être utilisé pour déterminer le rapport entre la covariance et le produit de l'écart-type des évaluations données par deux utilisateurs u et v [Ben17] (*PCU - Pearson Correlation between Users*) :

$$PCU(u,v) = \widehat{\rho}_{uv} = \frac{\sum_{i \in I(u,v)} (r_{ui} - \bar{r_u}) \times (r_{vi} - \bar{r_v})}{\sqrt{\sum_{i \in I(u,v)} (r_{ui} - \bar{r_u})^2 \times \sum_{i \in I(u,v)} (r_{vi} - \bar{r_v})^2}}$$

où $\bar{r_u}$ est la moyenne des notes données par l'utilisateur u sur les éléments qu'il a évalué, et $I(u,v)$ contient les éléments ayant été évalués par les utilisateurs u et v.

De nombreuses évaluations peuvent être inconnues ou parcimonieuses (*sparse*), ainsi certains éléments peuvent ne partager qu'une petite quantité d'évaluateurs couramment observés. Le coefficient de corrélation $\widehat{\rho}_{ij}$, est de ce fait basé uniquement sur ces utilisateurs courants. Il devient alors important de considérer des biais d'évaluation par utilisateur et par élément dans nos calculs. On notera donc b_u le biais pour l'utilisateur u, b_i le biais pour l'article i et μ la moyenne globale des évaluations. Ainsi, selon [RRS15], dans le cas par exemple de la comparaison entre deux éléments i et j, le calcul du coefficient de corrélation approximatif $\widehat{\rho'}_{ij}$ sera donné par :

$$\widehat{\rho'}_{ij} = \frac{\sum_{u \in U(i,j)} (r_{ui} - \mu - b_u - b_i) \times (r_{uj} - \mu - b_u - b_j)}{\sqrt{\sum_{u \in U(i,j)} (r_{ui} - \mu - b_u - b_i)^2 \times \sum_{u \in U(i,j)} (r_{uj} - \mu - b_u - b_j)^2}}$$

Il existe également d'autres mesures de similarité typiquement utilisées en *CF* comme celle du calcul de similarité de *cosinus* [Bao+13] ou celle de *Jaccard* [AC17 ; Mey+11]. Pour tout objet à comparer X et Y représenté par un vecteur (respectivement $\overrightarrow{x}$ et $\overrightarrow{y}$), on calcule la similarité de cosinus comme suit [Ben17] :

$$cos(\overrightarrow{x}, \overrightarrow{y}) = \frac{\overrightarrow{x} . \overrightarrow{y}}{||\overrightarrow{x}|| \times ||\overrightarrow{y}||}$$

Ainsi, dans le cas d'une comparaison entre deux éléments i et j, représentés respectivement par un vecteur $\overrightarrow{i}$ et $\overrightarrow{j}$, la similarité de cosinus (*CI - Cosine similarity between Items*) se calcule comme suit [Ben17] :

$$CI(i,j) = cos(\overrightarrow{i}, \overrightarrow{j}) = \frac{\sum_{u \in U(i,j)} r_{ui} \times r_{uj}}{\sqrt{\sum_{u \in U(i,j)} r_{ui}^2} \times \sqrt{\sum_{u} \in U(i,j) r_{uj}^2}}$$

Dans le cas d'une comparaison entre deux utilisateurs u et v, représentés respectivement par un vecteur $\overrightarrow{u}$ et $\overrightarrow{v}$, la similarité de cosinus (*CU - Cosine similarity between Users*) se calcule

comme suit [Ben17] :

$$CU(u,v) = cos(\overrightarrow{u}, \overrightarrow{v}) \frac{\sum_{i \in I(u,v)} r_{ui} \times r_{vi}}{\sqrt{\sum_{i \in I(u,v)} r_{ui}^2} \times \sqrt{\sum_{i \in I(u,v)} r_{vi}^2}}$$

Dans le cas de la mesure de similarité de *Jaccard*, en considérant U_i l'ensemble des utilisateurs ayant noté l'élément i et U_j l'ensemble des utilisateurs ayant noté l'élément j, la similarité (*JACI - JACcard similarity between Items*) entre les deux éléments i et j se calcule comme suit [RRS15] :

$$JACI(i,j) = \frac{||U_i \cap U_j||}{||U_i \cup U_j||} = \frac{||U_i \cap U_j||}{||U_i|| + ||U_j|| - ||U_i \cap U_j||}$$

Il est intéressant de noter que parfois la similarité de *cosinus* et celle de *Jaccard* [Lu+15] peuvent être combinées. Nous considérons dans la suite de cette thèse que la mesure de similarité (Sim) pourra faire référence à l'une des définitions de PCI, $SCCI$, PCU, CI, CU, et $JACI$.

1.4.4 Les modèles à facteurs latents

Les modèles à facteurs latents constituent une approche alternative dont le principe est de transformer et considérer à la fois les éléments et les utilisateurs dans un même espace de facteurs latents [RRS15]. L'idée principale des modèles à facteurs latents est de décomposer la matrice des évaluations utilisateurs en produit de matrices ayant des propriétés spécifiques. L'espace latent est utilisé pour expliquer les évaluations utilisateurs en caractérisant à la fois les éléments et les utilisateurs en tant que facteurs, déduits automatiquement des retours (*feedback*) utilisateurs. Dans ce cadre, les éléments et les utilisateurs sont décrits par des vecteurs de même dimension d. La valeur de cette dimension est déterminée par le nombre de facteurs latents pris en considération.

Pour la description d'un élément, les composantes du vecteur correspondant sont les valeurs prises par les facteurs latents [RRS15] respectifs à cet élément.

Pour la description d'un utilisateur, les composantes du vecteur correspondant sont les contributions des facteurs latents respectifs à la note que l'utilisateur donnerait à un élément [2].

L'une des principales méthodes utilisées dans cette approche est la factorisation matricielle. Les modèles de factorisation matricielle font correspondre les utilisateurs et les éléments sur un espace de facteurs latents commun de dimension d, de sorte que les interactions utilisateur-élément soient modélisées comme des produits internes dans cet espace. Par exemple, lorsque les éléments ont trait à la restauration, les facteurs peuvent mesurer :

— des dimensions évidentes telles que par exemple la cuisine gastronomique par rapport aux fast-foods, ou l'orientation de la restauration vers les enfants ;

2. cf. Cours du Conservatoire national des arts et métiers (CNAM) intitulé : « Recherche par similarité. Application aux systèmes de recommandation » `http://cedric.cnam.fr/vertigo/Cours/RCP216/coursSimilariteRecommandation.html`

— des dimensions moins bien définies telles que par exemple l'originalité de la cuisine proposée (p. ex., cuisine moléculaire et combinaison de saveurs exotiques), la recherche de profondeur donnée à celle-ci (p. ex., revisite d'un plat ou d'une pâtisserie).

Remarquons que parfois les dimensions peuvent malheureusement être totalement ininterprétables.

La plus courante des méthodes de factorisation matricielle repose sur la décomposition en valeurs singulières (*SVD - Singular Value Decomposition*) de la matrice de données (issue des évaluations utilisateurs) M de rang d (dimension $d \times d$). Ainsi, la matrice M est définie telle que :

$$M \approx U.D.I^t$$

où $D \in \mathbb{R}^{d \times d}$ est une matrice diagonale correspondant aux poids des d facteurs. Chacune des n_i (nombre d'éléments) colonnes de $I^t \in \mathbb{R}^{d \times n_i}$ est la représentation dite « réduite » d'un élément. Nous les notons comme étant les vecteurs z_i. Chacune des n_u (nombre d'utilisateurs) lignes de $U \in \mathbb{R}^{n_u \times d}$ est la représentation dite « réduite » d'un utilisateur. Nous les notons comme étant les vecteurs x_u

Ainsi, les évaluations pourront être prédites selon la règle suivante :

$$\widehat{r}_{ui} = z_i^T \, . \, x_u$$

où le produit scalaire résultant, $z_i^T \, . \, x_u$, capture l'interaction entre l'utilisateur u et l'élément i c.-à-d., l'intérêt général de l'utilisateur pour les caractéristiques de l'élément.

Le principal inconvénient de cette méthode est qu'elle fonctionne avec des données d'évaluations complètes. Or, de nombreux jeux de données mettent à disposition une matrice d'évaluations utilisateurs-éléments très creuse ce qui aboutit à un problème sous-déterminé. Ainsi, une solution de régularisation devient nécessaire afin de chercher une approximation de rang d et ce à partir des données disponibles uniquement (c.-à-d., sans assimiler les valeurs absentes où égales à 0). Cette solution de factorisation régularisée [KBV09 ; Pat07] recherche une approximation de rang d réduit en tenant uniquement compte des valeurs disponibles dans la matrice de données M. Ainsi, afin d'apprendre les paramètres du modèle (x_u et z_i), il est nécessaire de résoudre un problème d'optimisation visant à minimiser l'erreur quadratique régularisée telle que :

$$\underset{z*,x*}{min} \sum_{Pres(u,i)} (r_{ui} - z_i^T x_u)^2 + \lambda_2(||z_i||^2 + ||x_u||^2)$$

où $Pres(u,i)$ correspond aux paires (u,i) pour lesquelles r_{ui} est connu. La constante λ_2, qui contrôle l'étendue de la régularisation (au sens *RIDGE*), est généralement déterminée par validation croisée.

Enfin, comme vu en 1.4.3, il est conseillé de prendre en considération les biais de notations b_u et b_i par rapport à μ la moyenne globale des évaluations. Ainsi, en prenant en compte ces

biais, les évaluations peuvent être prédites selon la règle suivante :

$$\widehat{r}_{ui} = \mu + b_i + b_u + z_i^T \ . \ x_u$$

Ainsi, afin d'apprendre les paramètres du modèle (b_u, b_i, x_u et z_i), le problème d'optimisation correspondant devient alors :

$$\underset{b*,z*,x*}{min} \sum_{Pres(u,i)} (r_{ui} - \mu - b_i - b_u - z_i^T \ . \ x_u)^2 + \lambda_2(b_i^2 + b_u^2 + ||z_i||^2 + ||x_u||^2)$$

Notons que la minimisation est généralement effectuée par une descente de gradient stochastique ou par une méthode des moindres carrés.

Dans cette section, nous venons de rappeler l'approche basée sur le filtrage collaboratif selon deux méthodes (basée heuristiques ou basée modèles) principalement employées dans la littérature. Dans la section suivante, nous rappelons l'approche des systèmes de recommandation basés sur le contenu.

1.5 Les recommandations basées sur le contenu

En résumé : les systèmes de recommandation basés sur le contenu (*Content-Based - CB*) disposent d'une architecture de haut niveau et recommandent des articles ou produits similaires aux éléments précédemment appréciés par un utilisateur [PB07]. Selon [Lu+15], les deux principales méthodes combinées pour réaliser ses recommandations sont les suivantes :

- des méthodes d'apprentissage automatique par la mise en place de modèles permettant d'apprendre les intérêts des utilisateurs à partir de leurs données d'historiques. Ces données constituent la base de connaissances correspondant à un jeu de données d'entraînement ;
- des méthodes de recommandations par l'usage de techniques traditionnelles de recherche d'information, telle que la mesure de similarité de cosinus.

1.5.1 Les bases de fonctionnement

Dans cette sous-section, nous présentons les bases du fonctionnement de systèmes de recommandation basés sur le contenu. Nous les illustrons dans la Figure 1.4.

La question utilisateur à laquelle doit répondre un système de recommandation basé sur le contenu peut se résumer ainsi : « Montrez-moi des choses qui ressemblent à ce que j'ai aimé » [JF13].

Dans ces systèmes, l'objectif du composant de recommandation, consiste à déterminer quels éléments de la liste, en fonction de leurs descriptions, correspondent le mieux avec les préférences déjà connues mais aussi apprises de l'utilisateur.

Ainsi, à la différence de la Figure 1.1 représentant le fonctionnement générique d'un système de recommandation, la Figure 1.3 intègre de manière systématique (obligatoire) la prise

en compte de l'historique des retours utilisateurs dans son processus de recommandation. Nous illustrons plus en détails le mode de fonctionnement du composant de recommandation et plus particulièrement la prise en compte des retours utilisateurs à la Figure 1.5.

Dans la section 1.5.2 suivante, nous décrivons le fonctionnement de l'approche basée sur le contenu en rappelant :

— son processus de recommandation ;
— l'architecture de haut niveau sur laquelle repose ce processus.

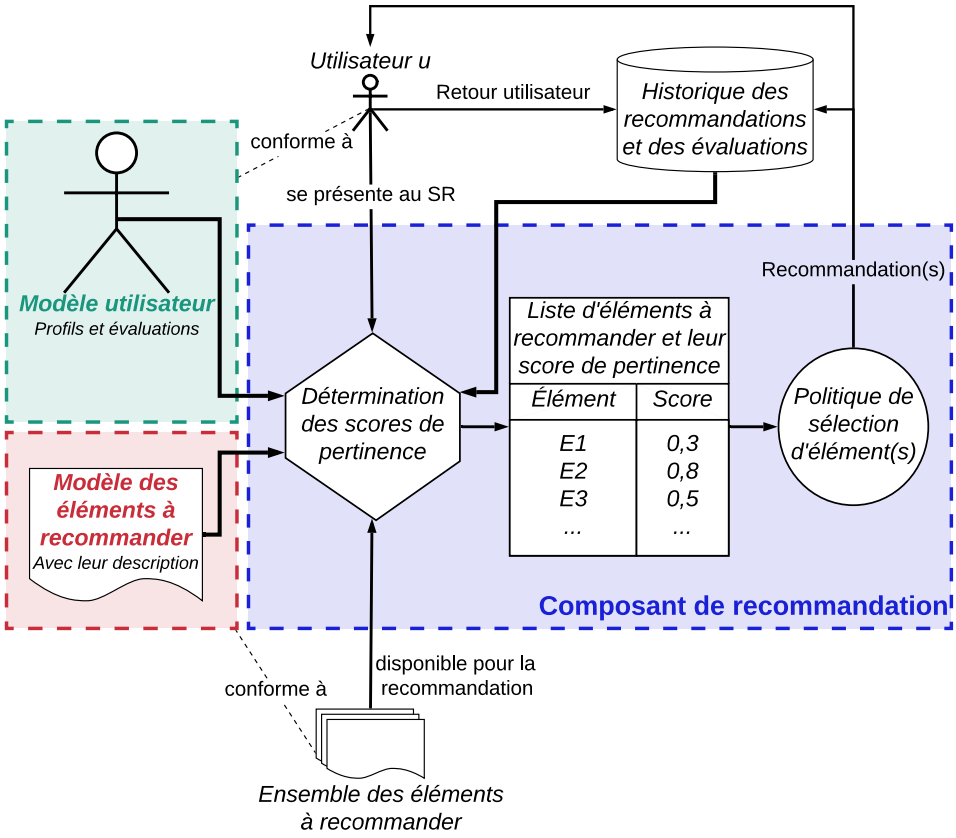


FIGURE 1.4 – Principe de fonctionnement d'un système de recommandation basé sur le contenu.

1.5.2 Le processus de recommandation

Les systèmes de recommandation basés sur le contenu analysent l'ensemble des descriptions des éléments qui ont été précédemment évalués par un utilisateur. Cette analyse permet de construire un profil utilisateur centré sur ses intérêts à partir des caractéristiques des éléments évalués par cet utilisateur. Le profil est, dans ce cas, une représentation structurée

des intérêts de l'utilisateur et permet de lui recommander de nouveaux éléments intéressants [RRS15].

Avec cette approche, le processus de recommandation consiste essentiellement à faire correspondre les attributs du profil utilisateur avec ceux d'un élément. Comme le profil inféré reflète avec précision les préférences de l'utilisateur, il constitue un avantage considérable pour l'efficacité du système de recommandation. En effet, ceci permet au système de filtrer, en déterminant si un utilisateur est intéressé par un élément spécifique ou non, et dans le cas contraire, d'éviter de lui recommander. La précision des recommandations effectuées par le système sera ensuite évaluée par la mesure de l'erreur quadratique moyenne ou par la mesure du niveau d'intérêt de l'utilisateur pour l'élément recommandé.

1.5.3 Architecture de haut niveau

Une architecture de haut niveau mettant en oeuvre, entre autres, des méthodes d'apprentissage automatique ou de calcul de similarité est nécessaire pour supporter le processus de recommandation des systèmes basés sur le contenu (voir Figure 1.5).

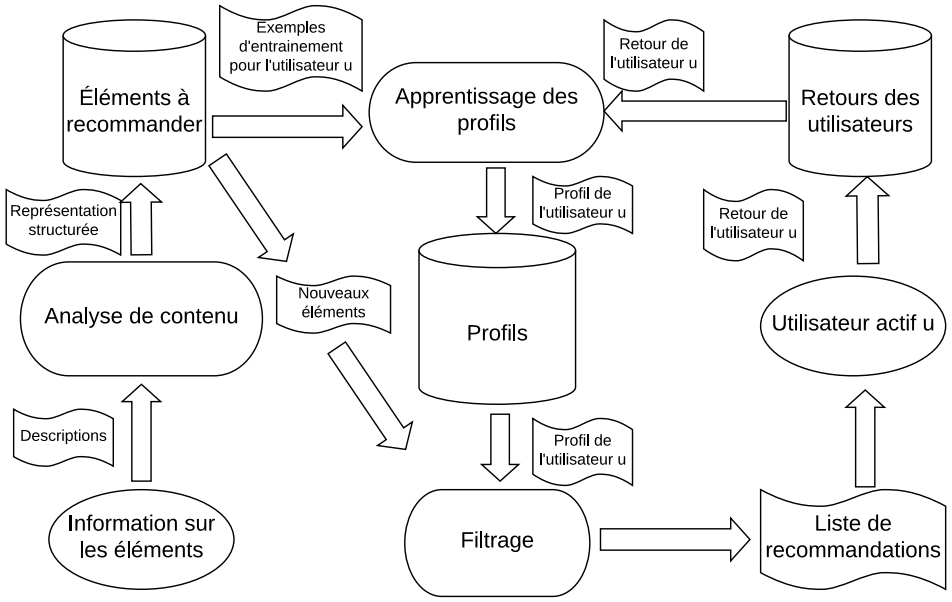


FIGURE 1.5 – Architecture d'un système de recommandation basé sur le contenu, traduit de [RRS15]

En effet, les systèmes de recommandation basés sur le contenu ont besoin de techniques appropriées pour représenter les éléments et inférer le profil d'utilisateur. Ils requièrent également la mise en place de stratégies permettant de comparer le profil d'utilisateur à la représentation d'élément. Le processus de recommandation qui en découle se déroule en trois étapes, chacune étant gérée par un composant distinct :

1. **Un composant qui analyse le contenu.** Lorsque les informations ne sont pas structurées, il est nécessaire de mettre en œuvre un traitement permettant d'extraire des informations pertinentes structurées. Le principal objectif de ce composant est de représenter le contenu des éléments à recommander (documents, pages web, actualités, descriptions de produits, etc.) sous une forme adaptée (valeurs continues ou discrètes) aux étapes de traitement qui le succèdent. La description des éléments est analysée par des techniques d'extraction de caractéristiques (*features extraction*) afin de transformer la représentation originale (source brute d'information) de l'élément, en une représentation cible utilisable par le système (p. ex., en vecteurs de valeurs continues, ou en vecteur de valeurs binaires - *one-hot*). Cette représentation calculée représente le point d'entrée des deux autres composants : celui qui apprend et construit le profil, et celui qui filtre ;
2. **Un composant qui apprend et construit le profil**[3]**.** Ce composant collecte les préférences de l'utilisateur et tente de généraliser ces données, afin de construire le profil utilisateur (centré préférences). Cette méthode de construction de profil est généralement réalisée au moyen de techniques d'apprentissage automatique [Mit97], permettant d'inférer un modèle d'intérêts des utilisateurs à partir des éléments qu'ils ont appréciés ou non par le passé (historique). Par exemple, dans le cadre de la recommandation de pages web [RRS15], notre composant pourrait implémenter une méthode axée sur la pertinence des retours utilisateurs (*feedback*) [Roc71]. Ce composant représente le profil utilisateur par un vecteur « prototype » construit à partir des retours positifs et négatifs des utilisateurs. Dans le cas de recommandation de pages web, les jeux de données d'entraînement seront alors constitués de pages web sur lesquelles un retour (*feedback*) positif ou négatif a été fourni par l'utilisateur ;
3. **Un composant de filtrage.** Ce composant exploite le profil utilisateur pour recommander des éléments pertinents en faisant correspondre la représentation du profil (calculé par le module d'analyse de contenu) à celle des éléments à recommander. La décision de recommander ou non un élément, valeur binaire (0 ou 1) ou continue (probabilité), est basée sur une mesure de la similarité [Her+04]. Dans l'exemple mentionné précédemment concernant la recommandation de pages web [RRS15], le calcul de correspondance entre les représentations du profil et des éléments est réalisé en calculant la similarité de cosinus entre le vecteur prototype et le vecteur des éléments représentés par des pages web (voir Section 1.4 sur le détail de calcul de similarité entre deux vecteurs).

Dans cette section, nous avons rappelé l'approche basée sur les contenus qui est définie par son processus de recommandation typique et appuyée de son architecture de haut niveau. Dans la section suivante, nous rappelons l'approche des systèmes de recommandation basés sur les connaissances.

3. La description de ce composant nous amènera à nous pencher plus en détail, dans la section 1.10, sur les techniques employées pour l'apprentissage de profils.

1.6 Les recommandations basées sur les connaissances

En résumé : les systèmes de recommandation basés sur les connaissances (*Knowledge-Based - KB*) permettent la recommandation d'éléments à des utilisateurs à partir de la connaissance que le système possède sur les utilisateurs, les éléments à recommander, et la relation existante entre utilisateurs et éléments. Ce type de systèmes de recommandation découle de la déduction entre les besoins d'un utilisateur et la capacité d'un élément à répondre à ces besoins. Les principales méthodes utilisées aujourd'hui dans les techniques de *KB* sont basées sur les cas [Bri+05 ; Ric+06]. En considérant ainsi un élément comme étant un cas, les recommandations sont générées en calculant les cas les plus similaires aux requêtes ou profils des utilisateurs. L'usage de techniques de recommandation *KB* peuvent également conduire à raisonner à partir d'ontologies [MDS09]. Il sera alors possible d'utiliser les ontologies d'un domaine de connaissances spécifique pour effectuer ensuite des mesures de similarité sur la sémantique entre les différents éléments à recommander et les profils utilisateurs [CBC08].

1.6.1 Les bases de fonctionnement

Dans cette sous-section, nous présentons les bases du fonctionnement de systèmes de recommandation basés sur les connaissances. Nous les illustrons dans la Figure 1.6.

La question utilisateur à laquelle doit répondre un système de recommandation basé sur les connaissances peut se résumer ainsi : « Dites-moi ce qui convient en fonction de mes besoins » [JF13].

Dans ces systèmes, le composant de recommandation utilise des connaissances extérieures pour ses calculs de scores de pertinence. Ceux-ci sont généralement basés sur des scores de similarité entre la sémantique des différents éléments à recommander et les profils utilisateurs.

Ainsi, à la différence de la Figure 1.1 représentant le fonctionnement générique d'un système de recommandation, la Figure 1.6 intègre des modèles de connaissance des éléments à recommander. Ces connaissances sont souvent représentées par des ontologies.

Dans la section suivante, nous décrivons les fondamentaux de l'approche basée sur les connaissances en rappelant sur quelles méthodes elle peut reposer, à savoir :

— des cas ;
— des contraintes.

1.6.2 Les systèmes basés sur les connaissances

Les systèmes basés sur les connaissances recommandent des éléments à partir des connaissances de leur domaine, notamment sur la manière dont certaines caractéristiques de ces éléments peuvent répondre aux besoins et aux préférences des utilisateurs (c.-à-d., à quel point l'élément est utile pour l'utilisateur). En effet, les techniques de recommandation *KB* maintiennent une base de connaissances fonctionnelles à jour décrivant comment un élément à recommander particulier répond aux besoins d'un utilisateur spécifique.

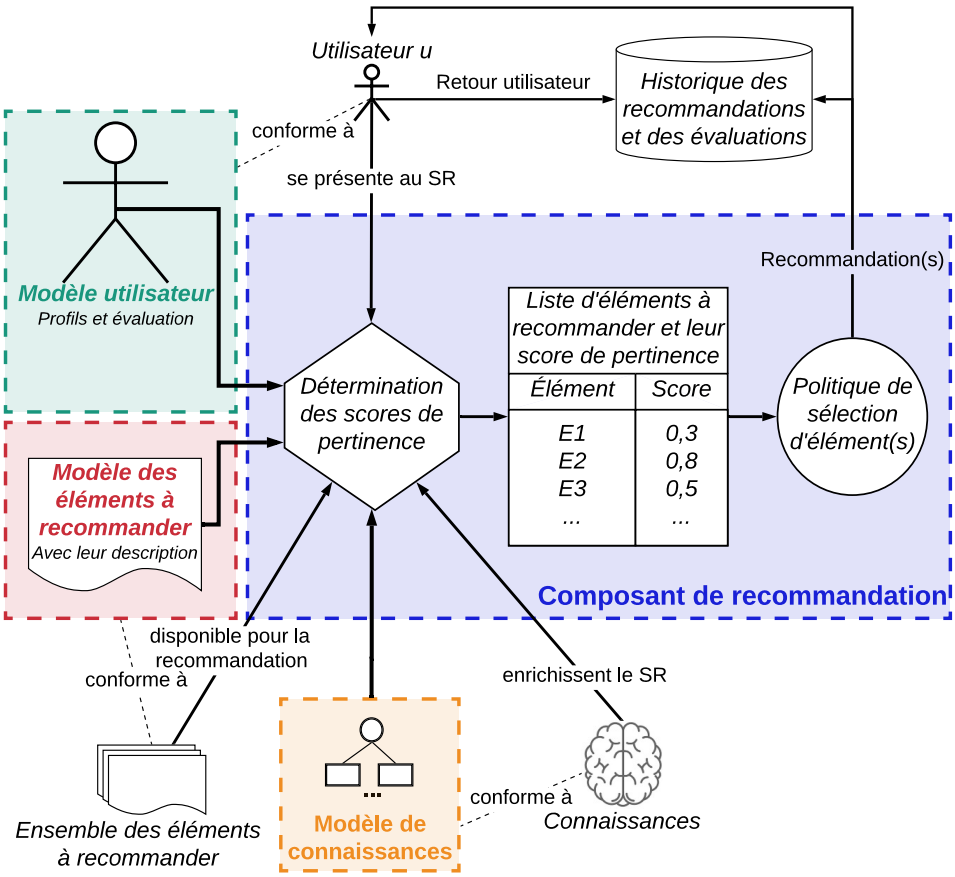


FIGURE 1.6 – Principe de fonctionnement d'un système de recommandation basé sur les connaissances

Les systèmes de recommandation basés sur les connaissances reposent principalement sur :

1. **Des cas** [Bri+05 ; Ric+06]. Ce type de système recommande des éléments similaires à ceux que les utilisateurs ont jugés intéressants. Un élément est traité comme un cas comportant plusieurs caractéristiques. Au même titre que décrit précédemment sur les systèmes basés sur le contenu, les techniques basées sur les cas sont utilisées pour analyser les caractéristiques des éléments disponibles et les préférences déclarées d'un utilisateur, afin d'identifier une ou plusieurs possibilités de recommandations pertinentes. Selon [RRS15], dans ces systèmes, le calcul de similarité permet d'estimer dans quelle mesure les besoins de l'utilisateur (description du problème) correspondent aux recommandations (solutions du problème). Dans ce cas, la mesure de similarité peut être directement interprétée comme étant l'« utilité » de la recommandation pour l'utilisateur. Les ontologies [MDS09], en tant que méthodes de représentation formelle des connaissances, permettent de définir les concepts d'un domaine et les relations

existantes entre ces concepts. Les ontologies créent ainsi des descriptions de nos cas considérés comme des ressources d'apprentissage. Elles permettent de ce fait la personnalisation et l'adaptation des recommandations en fonction du profil de l'utilisateur défini par l'ontologie, grâce à des mesures de similarité entre les éléments à recommander et ces profils [CBC08]. Utiliser les ontologies présente certains avantages dont le plus important est la possibilité de réutilisation et de partage [SBZ10], notamment entre applications [SBZ10 ; Yu+07]. On retrouvera cinq critères fondamentaux de conception à prendre en compte pour les ontologies [Gru93] : la clarté, la cohérence, l'extensibilité, la déformation d'encodage minimale, et l'engagement ontologique minimal ;

2. **Des contraintes** [FB08 ; Fel+06] qu'on établit à partir de requêtes *(query)* c.-à-d., on considère qu'une requête est une conjonction de contraintes sur des caractéristiques. Alors que les systèmes reposant sur des cas déterminent les recommandations sur la base de mesures de similarité, les systèmes reposant sur des contraintes exploitent principalement des bases de connaissances prédéfinies contenant des règles explicites sur la manière de relier les exigences des clients aux caractéristiques des éléments. De ce fait, un système basé sur les contraintes sera aussi performant que sa base de connaissances est juste. Par conséquent, la base de connaissances doit être correcte, complète et à jour afin de garantir des recommandations inférées de haute qualité. Notons que le principal avantage de raisonner sur des contraintes est l'explicabilité.

Quelle que soit la méthode employée, les connaissances sont toujours exploitées de la même manière. Les besoins des utilisateurs sont collectés, des corrections concernant des exigences incohérentes sont automatiquement proposées dans les situations où aucune solution n'a pu être trouvée, et les résultats de la recommandation sont expliqués. La différence majeure réside plutôt dans la manière dont les solutions sont calculées (p. ex., quelle mesure de distance est employée).

Les systèmes basés sur les connaissances fonctionnent généralement mieux que les autres au début de leur déploiement. En revanche, s'ils ne sont pas équipés de composants d'apprentissage, ils risquent d'être dépassés par d'autres approches utilisant des techniques d'intelligence computationnelle et sachant de ce fait exploiter les historiques d'interactions entre utilisateur et système. D'autre part, un défi important à prendre en considération est qu'en raison des compétences très limitées en programmation informatique des experts du métier ou du domaine, il existe généralement un écart entre ingénieurs (techniques) et experts métiers en termes de développement de la base de connaissances et de savoir-faire en matière de maintenance [Fel+06]. Ainsi, les experts métiers sont uniquement responsables de fournir les connaissances mais pas leur formalisation dans une représentation exécutable par les systèmes de recommandation (c.-à-d., la base de connaissances des systèmes de recommandation). Ceci pose un problème de maintien de la précision des recommandations dans le temps notamment dû à des contraintes de non-stationnarité inhérentes aux problèmes de recommandation en ligne c.-à-d., un système de recommandation intégré dans une application réelle en ligne comme p. ex., un site web, une application mobile.

Dans cette section, nous venons de rappeler l'approche basée les connaissances abordée selon deux manières : les cas et les contraintes. À la section suivante, nous rappelons l'approche des systèmes de recommandation basés sur les groupes d'utilisateurs.

1.7 Les recommandation basées sur les groupes d'utilisateurs

En résumé : ce type de systèmes de recommandation est généralement utilisé pour produire des suggestions à des groupes d'utilisateurs lorsque les membres de ce groupe doivent trouver un consensus [McC+06 ; Oco+01 ; Pop13]. Cela peut être notamment le cas pour les recommandations de musiques, de films, d'événements ou encore de décisions plus complexes comme le sont les plans de voyage. Par exemple, [McC+06] a proposé un système de recommandation de groupe pour la recommandation de vacances au ski. Le système proposé ici consiste à faire collaborer des utilisateurs et utiliser leur *feedback* en regard de caractéristiques prédéfinies dans le système. Ainsi, tous les membres du groupe peuvent critiquer tant les stations dans leur ensemble que les hébergements. Ensuite, l'ensemble des commentaires est agrégé et les recommandations qui satisfont globalement le groupe sont finalement retenues et proposées. C'est pourquoi, le choix de la stratégie d'agrégation de tels systèmes prendra alors toute son importance pour une meilleure précision des recommandations et une meilleure performance [RRS15].

1.7.1 Les bases de fonctionnement

Dans cette sous-section, nous présentons les bases du fonctionnement de systèmes de recommandation à des groupes d'utilisateurs. Nous les illustrons à la Figure 1.7.

La question utilisateur à laquelle doit répondre un système de recommandation à des groupes peut se résumer ainsi : « Dites-moi quels éléments correspondent le mieux à l'ensemble des utilisateurs du groupe ».

Ainsi, en complément des composants présents à la Figure 1.1 représentant le fonctionnement générique d'un système de recommandation, la Figure 1.7 intègre la notion de groupe d'utilisateurs et de stratégie d'agrégation des préférences individuelles (souvent recensées sous forme de votes ou d'évaluations).

Le composant de recommandation utilise une stratégie d'agrégation de préférences individuelles pour effectuer ses calculs de score de pertinence. La politique de sélection consiste à choisir le ou les éléments ayant obtenu le meilleur score en correspondance à la stratégie de calcul retenue.

Dans la sous-section 1.7.2 nous décrivons différents systèmes de recommandation de groupes existant dans la littérature et mettons le focus sur les stratégies d'agrégation les plus populaires.

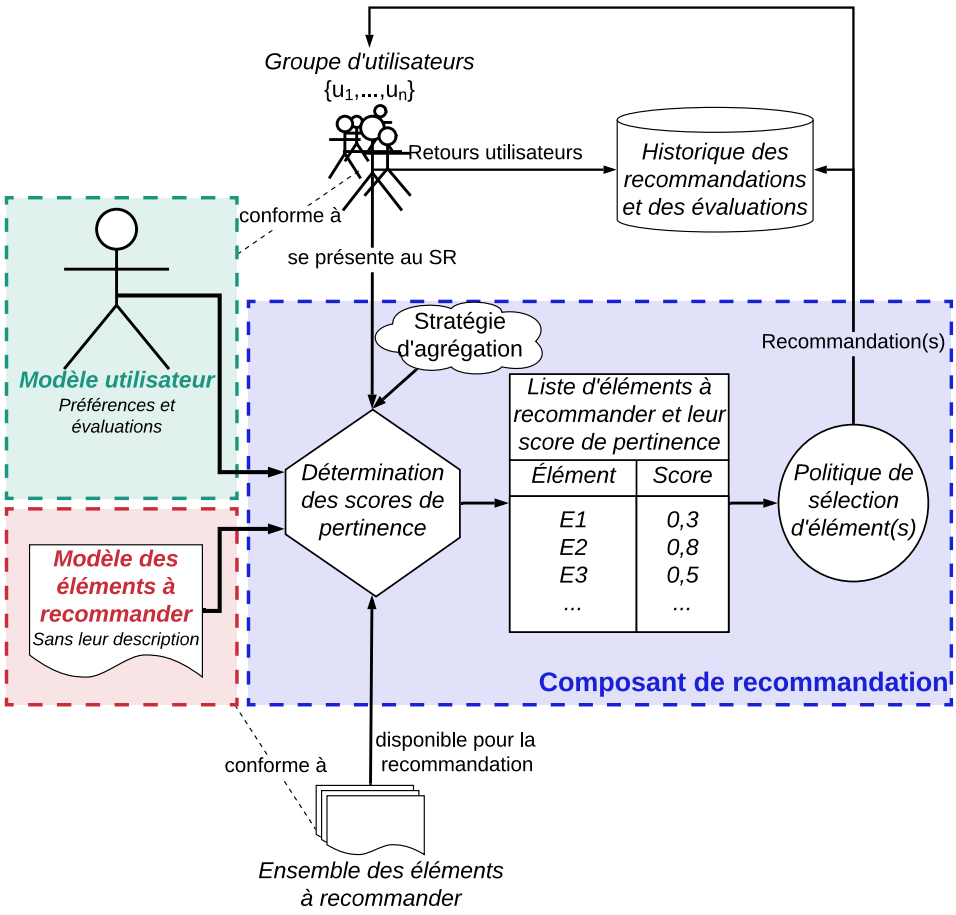


FIGURE 1.7 – Principe de fonctionnement d'un système de recommandation basé sur les groupes d'utilisateurs

1.7.2 Recommander à un groupe d'utilisateurs

Recommander à un groupe d'utilisateurs plutôt qu'à un individu, peut être utile dans de nombreuses situations, p. ex., dans le cadre de la recommandation des programmes de télévision, ou une séquence de chansons à écouter, et ceux sur la base de modèles comportant l'ensemble des membres du groupe. Il est reconnu que recommander à des groupes d'utilisateurs est encore plus compliqué que de recommander à des individus [RRS15]. En supposant que nous sachions parfaitement faire correspondre des recommandations à des utilisateurs individuels (c.-à-d., personnalisation complète des recommandations), la question reste de savoir comment combiner ces modèles pour des utilisateurs individuels au niveau d'un groupe.

Pour ce faire, il existe différents types de systèmes de recommandation basés sur les groupes qui ont été implémentés en fonction du scénario « métier » envisagé, afin de :

— **déployer une intelligence ambiante** [MA98], c.-à-d., concevoir des environnements physiques sensibles à la présence des personnes ;

— **recommander des destination de vacances ou de voyages** [McC+06], c.-à-d., aider les utilisateurs à choisir des vacances communes ;
— **recommander des programmes de télévision ou des films au cinéma** [Oco+01 ; Yu+06b], c.-à-d., proposer un programme télévisé ou un film au cinéma à un groupe de personnes ;
— **recommander des lieux dans le cadre du tourisme** [Ard+01], c.-à-d., recommander des visites à des groupes de touristes en tenant compte par exemple des caractéristiques des différents sous-groupes (N.B., les sous-groupes sont inclus dans le groupe).

De ce fait, à partir de ces scénarios ont été développés des systèmes de recommandation que nous rappelons et classifions comme suit [RRS15] :

1. Les systèmes utilisant les préférences individuelles préalablement connues en opposition à ceux qui les découvrent par inférence au fil du temps. Dans la plupart des scénarios, le système de recommandation dispose des préférences individuelles *a priori*. En revanche, si on prend l'exemple de *CATS* [McC+06], les préférences individuelles se découvrent au cours du temps ;

2. Les systèmes dont les éléments recommandés sont expérimentés par le groupe, ou ceux qui sont présentés en tant qu'options parmi une liste. Dans le cadre de la télévision interactive [Yu+06b] par exemple, le groupe expérimente les informations. De même, pour des scénarios type intelligence ambiante comme *MUSICFX* [MA98], le groupe expérimente différentes musiques. En revanche, dans d'autres scénarios, une liste de recommandations peut être présentée sous forme d'options. Par exemple, *POLYLENS* [Oco+01] présente une liste de films que le groupe peut vouloir regarder ;

3. Les systèmes dont le groupe est passif versus ceux dont le groupe est actif. Dans la plupart des cas, le groupe n'interagit pas avec la manière dont les préférences individuelles sont agrégées. Cependant, dans le cadre du système *TRAVEL DECISION FORUM* [Jam04] et de CATS [McC+06], le groupe ajuste le modèle proposé ;

4. Les systèmes qui recommandent un unique élément et ceux qui recommandent une liste d'éléments. Dans le scénario de *POLYLENS* [Oco+01] par exemple, il ne suffit de recommander qu'un élément en considérant que les personnes ne voient généralement qu'un film par soir. Il en est de même concernant les systèmes *TRAVEL DECISION FORUM* [Jam04] et *CATS* [McC+06] où les utilisateurs ne partagent qu'un seul jour de vacances ensemble. En revanche, dans le cadre de la télévision interactive [Yu+06b], il est nécessaire de recommander une séquence d'éléments, par exemple pour constituer une suite d'émissions d'actualités complète. De même, dans *INTRIGUE* [Ard+01], comme il est fort probable qu'un groupe de touristes visite plusieurs attractions au cours du voyage, il est nécessaire de leur recommander une série d'attractions touristiques à visiter. C'est le cas également dans le cadre de l'intelligence ambiante [MA98], où il est probable qu'un utilisateur entende plusieurs chansons ou visualise plusieurs éléments sur des présentoirs en magasin.

Recommander à des groupes de personnes nécessite de mettre en place une stratégie d'agrégation des retours individuels p. ex., commentaires [McC+06] ou évaluations [Oco+01 ; Pop13 ; RRS15]. Les méthodes d'agrégation sont de ce fait le point central des systèmes de recommandation à des groupes, c'est pourquoi nous consacrerons le prochain paragraphe à dresser une vue d'ensemble des principales stratégies d'agrégations existantes.

1.7.3 Agréger les retours individuels

Le principal problème que doivent résoudre les systèmes de recommandation à des groupes, est comment s'adapter à l'ensemble du groupe en fonction des informations relatives aux préférences des utilisateurs. Par conséquent, de nombreuses stratégies existent pour regrouper les retours individuels dans une seule notation de groupe (p. ex., dans le cadre d'élections). [RRS15] rappelle onze stratégies d'agrégation employées dans les systèmes de recommandation. Nous les décrivons ci-dessous :

1. **Stratégie utilisant la moyenne des évaluations**. Les systèmes utilisant cette stratégie recommandent l'élément possédant la meilleure moyenne d'évaluation à l'ensemble du groupe ;
2. **Stratégie utilisant la moyenne des évaluations en excluant le minimum**. Cette stratégie est similaire à celle cité ci-dessus à ceci près qu'elle calcule la moyenne en retirant la pire évaluation dans un groupe ;
3. **Stratégie utilisant la multiplication des évaluations**. Les systèmes utilisant cette stratégie recommandent à l'ensemble du groupe, l'élément possédant le meilleur score d'évaluation, p. ex., si trois utilisateurs ont noté un élément respectivement 1, 3 et 4, alors le score de l'élément est donc de $1 \times 3 \times 4 = 12$;
4. **Stratégie basée sur la méthode de *Borda***. Les systèmes utilisant cette stratégie comptent les points en se référant au classement ordonné des éléments dans les listes de préférences des individus, c.-à-d., le dernier élément de la liste obtient 0 point, le suivant 1 point, etc. Ils recommandent ensuite l'élément possédant le score le plus élevé calculé à partir de l'ensemble des listes de préférences (somme des points calculés) ;
5. **Stratégie basée sur la méthode de *Copeland***. Les systèmes utilisant cette stratégie comptent le nombre de fois, durant un vote, qu'un élément bat d'autres éléments (à la majorité) et y soustrait le nombre de fois qu'il perd ;
6. **Stratégie utilisant le vote plural**. Cette stratégie correspond à un système de *scrutin* uninominal majoritaire c'est à dire que l'élément qui reçoit le plus de votes est celui considéré comme optimal à la recommandation du groupe ;
7. **Stratégie utilisant le vote par approbation**. Avec cette stratégie, nous comptons les évaluations de chaque élément dans un groupe ayant uniquement obtenu des notations supérieures à un seuil fixé à l'avance. Le système recommande ensuite l'élément ayant le score d'évaluations maximal calculé ;

8. **Stratégie de *Least Misery (LM)***. Dans le cadre d'une stratégie *LM* on recommande l'élément qui a obtenu l'évaluation maximale dans un groupe ;
9. **Stratégie du *Most Pleasure (MP)***. Dans le cadre d'une stratégie *MP* on recommande l'élément qui a obtenu la moins mauvaise des évaluations les plus basses dans un groupe ;
10. **Stratégie visant l'équité (*Fairness*)**. Dans ce cas, les éléments recommandés sont tour à tour ceux ayant obtenu l'évaluation maximale pour chacun des membres du groupe ;
11. **Stratégie suivant le « *leader* »**, en d'autres termes la personne la plus influente ou la plus respectée. Dans ce cas, l'élément recommandé est celui dont l'évaluation est maximale pour le membre d'un groupe ayant été identifié comme le plus respecté.

Selon [RRS15], qui réalisa des expériences sur l'ensemble des stratégies, il semble que la plus efficace est celle de la stratégie de multiplication, bien que de nombreuses autres citées ci-dessus s'en rapprochent en termes de performances. Néanmoins, ces performances dépendent fortement du groupe et de sa connaissance des règles d'évaluation qui peuvent impliquer l'échec de certaines méthodes comme *Copeland*, *LM*, ou encore le vote plural.

Il est important de noter également que les membres du groupe d'expérimentation de [RRS15] se sont souciés de l'importance concernant une équité dans les recommandations faites au groupe.

Dans cette section, nous avons rappelé l'approche de recommandation à des groupes et ses différentes stratégies d'agrégation de préférences individuelles. À la section suivante, nous rappelons l'approche des systèmes de recommandation sensibles au contexte.

1.8 Les systèmes de recommandation contextuels

En résumé : les systèmes de recommandation sensibles au contexte peuvent prendre tout leur sens dans le cadre de la mobilité [WBE09] et de la recommandation à des utilisateurs mobiles [Guo+15]. Ils sont également un atout important dans tout type de recommandation nécessitant de capter le contexte de l'utilisateur [AT11] et ce afin de gagner en précision. L'approche de recommandation contextuelle prend généralement son essence à partir de données telles que : le profil utilisateur (p. ex., âge, sexe, préférences), le contexte associé (p. ex., environnement, équipement), et les caractéristiques des éléments à recommander. C'est notamment le cas dans de nombreuses applications de type *Mobile Crowd Sensing and Computing (MCSC)* [Guo+15]. De plus selon la méthode employée, ces systèmes peuvent dynamiquement intégrer les retours des utilisateurs sur les éléments qui leur sont recommandés. Dans certains cadres applicatifs nécessitant de prendre en compte le contexte de l'utilisateur (p. ex., application mobile), les systèmes de recommandation sensibles au contexte peuvent être plus performants que leurs homologues non-contextuels. Ces derniers, aveugles à certaines situations dépendantes du contexte, peuvent échouer à recommander de manière adéquate par manque de pertinence dans ce contexte donné (p. ex., il pleut, le système recommande une

visite touristique d'extérieur). C'est d'autant plus observable pour les applications soumises à une forte contrainte de non-stationnarité comme le sont par exemple les applications mobiles [Gre+17].

1.8.1 Les bases de fonctionnement

Dans cette sous-section, nous présentons les bases du fonctionnement de systèmes de recommandation contextuels. Nous les illustrons sur la Figure 1.8.

La question utilisateur à laquelle doit répondre un système de recommandation contextuel peut se résumer ainsi : « Montrez-moi plus de ce qui ressemble à ce que j'ai aimé dans un contexte donné ».

Ainsi, à la différence de la Figure 1.1 représentant le fonctionnement générique d'un système de recommandation, la Figure 1.8 intègre les données de contexte en entrée et la possibilité de prendre en compte l'historique des retours utilisateurs dans le processus de recommandation.

La différence principale qui existe entre les approches contextuelles et les autres approches réside dans le fait que le contexte des utilisateurs, définis en tant qu'entités, est obligatoirement pris en considération sous toutes ses formes existantes [4] [ZLO07] c.-à-d., le temps, la localisation, les relations, l'individualité et l'activité.

Ainsi, l'objectif du composant de recommandation, consiste à déterminer quels éléments de la liste correspondent le mieux au profil de l'utilisateur évoluant dans un contexte donné.

Ainsi, dans les sous-sections suivantes, nous décrivons :

— la notion de contexte. Il est important de la rappeler afin de mieux comprendre les tenants et aboutissants qu'engendre la prise en compte des informations de contexte ;
— pourquoi se diriger vers un système de recommandation contextuel, quel est le principe de fonctionnement de ces systèmes sensible au contexte, et comment les informations de contexte s'ingèrent dans ces systèmes ;
— les trois formes que peuvent prendre les méthodes se focalisant sur l'apprentissage des préférences contextuelles ;
— les techniques permettant d'aborder le problème sous l'angle de la contextualisation de la fonction de recommandation c.-à-d., en se basant sur des heuristiques ou sur des modèles.

1.8.2 La notion de contexte

Cette thèse a apporté des contributions dans le domaine de recherche sur le contexte. Pour cette raison, le Chapitre 3 sera consacré à en dresser un état de l'art et à en rappeler les notions fondamentales. Néanmoins, dans cette section, il semble important de rappeler succinctement ce que le terme de contexte peut englober dans le cadre de ces systèmes, et

4. Dans la mesure du possible, c.-à-d., ce qui est capturable, voir chapitre 3.

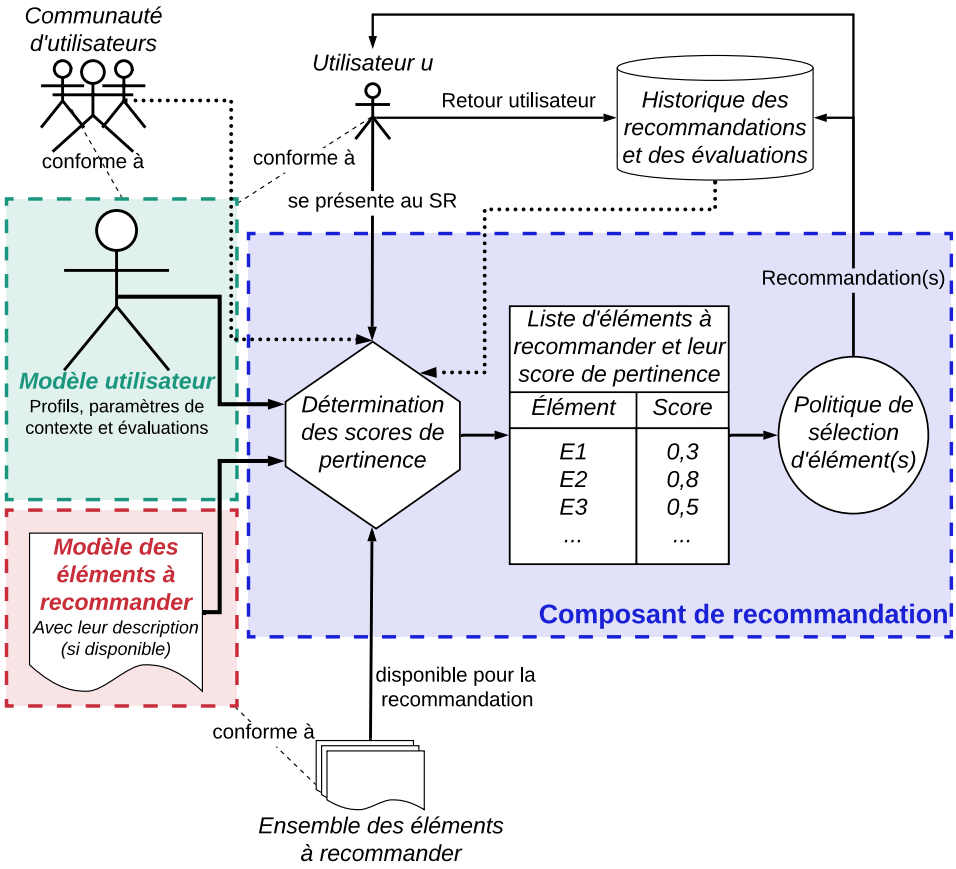


FIGURE 1.8 – Principe de fonctionnement d'un système de recommandation contextuel

ce avant d'évoquer plus en détails le principe des systèmes de recommandation tenant compte du contexte.

Le contexte correspond à *toute information pouvant être utilisée pour caractériser la situation d'une entité (personne, objet physique ou informatique)* [Dey01]. Ainsi, les données qui peuvent être collectées et utilisées dans les systèmes de recommandation contextuels (notamment mobiles), sont principalement :

— des données de profils (âge, sexe, catégories socio-professionnelles, préférences) ;
— des données inhérentes aux éléments à recommander (p. ex., catégorie, description) ;
— des données sur l'équipement de l'utilisateur (p. ex., marque, modèle, système d'exploitation) ;
— des données sur l'environnement de l'utilisateur (p. ex., localisation, date et heure, données météorologiques).

1.8.3 Vers une approche contextuelle de la recommandation

La majorité des approches existantes en matière de systèmes de recommandation se concentrent sur la recommandation des éléments les plus pertinents pour des utilisateurs individuels et ne prennent en compte aucune information contextuelle, p. ex., l'heure, le lieu et la compagnie d'autres personnes. En d'autres termes, lorsque les systèmes de recommandation traditionnels déterminent la meilleure recommandation à effectuer, ceux-ci traitent deux types d'entités : les utilisateurs et les éléments, mais en omettant de les placer dans leur contexte. Or, pour certaines applications effectuant par exemple des recommandations de santé [Gre+17], d'articles web [Li+10] ou encore de films [Bal+18], il ne suffit plus de simplement prendre en compte les utilisateurs et les éléments. Dans certains cas, il devient nécessaire d'intégrer les informations contextuelles au processus de recommandation afin de recommander des éléments aux utilisateurs selon différentes circonstances. En revanche, même si un certain nombre d'applications nécessitant l'apport d'informations contextuelles est aujourd'hui clairement identifié, il n'en reste pas moins énigmatique dans quel cas il est pertinent de raisonner avec du contexte [RRS15]. C'est pourquoi, si l'application n'est pas clairement identifiée ou si on souhaite généraliser l'approche dans le cadre de recherches portant sur les systèmes de recommandation, il est toujours important de comparer méthodes contextuelles et non-contextuelles, et ce sur plusieurs jeux de données.

1.8.4 Les systèmes de recommandation sensibles au contexte

Plus couramment dénommés *CARS (Context-Aware Recommmendation System)* [ZMB15] par la communauté, les systèmes de recommandation sensibles au contexte démontrent que, selon le domaine d'application et les données disponibles, certaines informations contextuelles peuvent être utiles afin de gagner en précision dans les recommandations. Bien que de nombreuses recherches aient été effectuées dans le domaine des systèmes de recommandation, la grande majorité des approches existantes se concentre uniquement sur la recommandation d'éléments à des utilisateurs, mais ne prenne en compte aucune information contextuelle. Les *CARS* quant à eux, traitent de la modélisation, et de la prédiction des goûts et des préférences des utilisateurs en intégrant les informations contextuelles disponibles. Dans leur processus de recommandation, ces systèmes utilisent ainsi le contexte en tant que données complémentaires, c.-à-d., sous forme notamment de caractéristiques explicites et structurées [BHB18 ; Bou14 ; Li+10 ; Zhe15]. Ces préférences et ces goûts à long terme sont généralement exprimés sous forme d'évaluations et sont modélisés comme une fonction prenant en compte trois paramètres : les éléments, les utilisateurs, et le contexte. De ce fait, les algorithmes de recommandation contextuelle qui en découlent, tentent d'estimer la fonction d'évaluation R suivante :

$$R : U \times I \times X \rightarrow Rating$$

où U correspond au domaine des utilisateurs, I le domaine des éléments, X correspond au contexte (c.-à-d., informations contextuelles associées à l'application), et $Rating$ correspond

au domaine des évaluations (c.-à-d., $r_{ui} \in Rating$ est l'évaluation donnée à l'élément $i \in I$ par l'utilisateur $u \in U$ — voir la Section 1.4 traitant du filtrage collaboratif).

1.8.5 Intégrer du contexte dans les systèmes de recommandation

L'idée d'utiliser des informations contextuelles dans les systèmes de recommandation a émergé à partir des travaux de [HK01]. Ces derniers ont émis l'hypothèse que l'intégration de connaissances sur l'utilisateur dans un algorithme de recommandation pouvait conduire à de meilleures recommandations dans certaines applications. Selon [RRS15], les approches utilisant les informations contextuelles dans le processus de recommandation peuvent être classées en deux groupes :

1. **Les recommandations contextuelles produites via des requêtes ou de la recherche d'information**. Ce type d'approche utilise des informations contextuelles obtenues directement de l'utilisateur (p. ex., son humeur ou ses intérêts courants), de l'environnement (p. ex., l'heure locale, la météo, la localisation) afin de filtrer et interroger un référentiel restreint d'éléments à recommander (p. ex., des restaurants ou des bars). Ensuite le système présente les éléments correspondant le mieux (p. ex., les restaurants à proximité et qui sont actuellement ouverts) à l'utilisateur. Ce type d'approche a notamment été utilisée par une grande variété de systèmes de recommandation [Abo+97 ; Cen+06 ; VPK04] ;
2. **Les recommandations contextuelles produites via l'estimation de préférences contextuelles.** Cette approche [Ado+05 ; Gre+17 ; Li+10 ; Oku+06 ; Pan+09 ; Yu+06a] tente de modéliser et d'apprendre les préférences utilisateurs : soit en observant les interactions de ces derniers avec le système (p. ex., derniers achats, dernières consultations) ; soit en obtenant des informations sur les préférences de l'utilisateur concernant les divers éléments précédemment recommandés (c.-à-d., retours utilisateurs sur les recommandations effectuées). Selon [RRS15], pour modéliser les préférences contextuelles des utilisateurs et générer des recommandations, cette approche adopte généralement des méthodes issues par exemple du filtrage collaboratif (*CF*), des systèmes basés sur le contenu, ou encore des systèmes hybrides. Elle peut également utiliser diverses techniques d'intelligence computationnelle pouvant aller de méthodes d'analyse de données de type « *fouille de données* » (*Data Mining*) [Ado+05] à des méthodes d'apprentissage automatique (*Machine Learning*) [Agi+06] (p. ex., classificateurs Bayésiens [DLT04 ; TV03], machines à vecteurs de support *SVM* [Oku+06]).

Cette seconde approche représente un enjeu plus récent que la première et peut être abordée selon trois méthodes distinctes [RRS15] :

1. **Le pré-filtrage contextuel.** Dans ce cas, les informations de contexte entrent en jeu en amont du processus de recommandation, c.-à-d., le contexte est utilisée pour sélectionner ou construire le jeu de données fourni en entrée. Le processus de recommandation traditionnel est ensuite mis en oeuvre selon l'approche choisie (p. ex., filtrage collaboratif ou approche basée contenu) ;

2. **Le post-filtrage contextuel.** Dans ce cas, les informations de contexte sont initialement ignorées. Le processus de recommandation traditionnel est tout d'abord mis en œuvre selon l'approche choisie. C'est seulement par la suite que l'ensemble de recommandations résultant de ce processus est ajusté (contextualisé) pour chaque utilisateur à l'aide des informations de contexte ;
3. **La modélisation contextuelle ou contextualisation de la fonction de recommandation.** Dans ce cas, les informations de contexte sont directement utilisées dans le modèle d'estimation de l'évaluation (p. ex., score de pertinence ou espérance). C'est le cas, entre autres, des systèmes de recommandation utilisant les bandits-manchots contextuels [AG13 ; Li+10] qui supposent une dépendance linéaire entre les récompenses obtenues (évaluations utilisateurs) et le contexte observé qui est sous la forme de vecteur de caractéristiques.

Dans le cadre de l'approche d'estimation des préférences contextuelles, la modélisation contextuelle correspond à la méthode adoptée dans cette thèse. C'est pourquoi nous établissons un état de l'art plus complet de cette méthode dans la sous-section suivante.

1.8.6 Modélisation contextuelle

Cette approche met en oeuvre des informations contextuelles directement dans la fonction de recommandation. Ces informations sont utilisées en tant que prédicteurs explicites des évaluations des utilisateurs pour chaque élément. Alors que les approches de pré-filtrage et de post-filtrage contextuelles utilisent des fonctions de recommandation traditionnelles (c.-à-d., recommandations 2D ($R : U \times I \rightarrow Rating$) [Ado+05 ; RRS15]), l'approche de modélisation contextuelle donne lieu quant à elle à des fonctions de recommandation considérées comme étant réellement multidimensionnelles. Ces fonctions représentent essentiellement des modèles prédictifs. Ces modèles peuvent être construit à l'aide de diverses techniques dites d'*Intelligence Computationnelle* (*IC*) [Lu+15] comme p. ex., des arbres de décision ou des modèles probabilistes. Il est également possible d'aborder la modélisation contextuelle via une approche heuristique qui incorporent des informations de contexte (p. ex., environnement ou activité) en plus des données de l'utilisateur (p. ex., profil) et de l'élément (p. ex., description).

Ci-dessous, nous rappellerons les principales techniques employées dans l'approche basée sur les heuristiques et dans l'approche basée sur les modèles.

1.8.6.1 Approches basées sur les heuristiques

L'approche traditionnelle de recommandation à deux dimensions (utilisateur et élément), basée entre autres sur les techniques de voisinage [BHK98 ; Sar+01], peut être étendue à des cas comprenant des informations contextuelles utilisant de multiples dimensions. Ceci est rendu possible en utilisant des mesures de distance ou de similarité à d dimensions au lieu de se limiter uniquement à des approches utilisateur-utilisateur ou élément-élément utilisées

traditionnellement dans de telles techniques (voir la Section 1.4 traitant des techniques de voisinage de l'approche de filtrage collaboratif).

Prenons comme exemple, un espace de recommandation traditionnel ($U \times I$) auquel nous rajoutons la dimension temporelle T. Suivant la méthode heuristique traditionnelle du plus proche voisin basée sur la somme pondérée des évaluations pertinentes, la prédiction d'une notation spécifique $r_{u,i,t}$ dans cet exemple peut être exprimée comme suit [Ado+05 ; RRS15] :

$$r_{u,i,t} = k \sum_{(u',i',t') \neq (u,i,t)} W((u,i,t),(u',i',t')) \times r_{u',i',t'}$$

où k est un facteur de normalisation, et $W((u,i,t),(u',i',t'))$ décrit la pondération que porte $r_{u',i',t'}$ dans la prédiction de $r_{u,i,t}$. W est généralement inversement proportionnel à la distance $dist[(u,i,t),(u',i',t')]$). En d'autres termes, plus les deux points sont similaires (ou plus la distance qui les sépare est faible), plus le poids de $r_{u',i',t'}$ est important dans la somme pondérée. Un exemple d'une telle relation serait $W((u,i,t),(u',i',t')) = 1/dist[(u,i,t),(u',i',t')]$, mais de nombreuses alternatives sont également possibles [RRS15]. À noter que à nouveau le choix de la méthode de calcul de distance ($dist$) dépend de l'application. Néanmoins, l'une des méthodes les plus simples permettant de définir une fonction de distance ($dist$) multidimensionnelle consiste à utiliser une méthode de réduction, en ne prenant compte que des points comportant les mêmes informations contextuelles, c'est à dire [RRS15] :

$$dist[(u,i,t),(u',i',t')] = \begin{cases} dist[(u,i),(u',i')] & \text{si} \quad t = t' \\ +\infty & \text{sinon} \end{cases}$$

Avec cette fonction de distance, $r_{u,i,t}$ ne dépend plus que des évaluations ayant les mêmes valeurs de temps t. Par conséquent, ce cas est réduit à une estimation en deux dimensions des évaluations ayant le même contexte de temps t. Il existe plusieurs méthodes de calcul employées dans la fonction de distance [RRS15], comme la distance de Manhattan :

$$dist[(u,i,t),(u',i',t')] = w_1\ d_1(u,u') + w_2\ d_2(i,i') + w_3\ d_3(t,t')$$

ou la distance Euclidienne :

$$dist[(u,i,t),(u',i',t')] = \sqrt{w_1\ d_1^2(u,u') + w_2\ d_2^2(i,i') + w_3\ d_3^2(t,t')}$$

où d_1, d_2 et d_3 sont des fonctions de distance définies respectivement pour les dimensions U, I, et T, et w_1, w_2 et w_3 sont les pondérations attribuées à chacune de ces dimensions en fonction de leur importance. Ainsi, la fonction de distance $dist[(u,i,t),(u',i',t')]$ peut être définie de différentes manières. En revanche, bien que cette fonction soit généralement calculée pour les évaluations d'un même utilisateur envers un même élément, l'un des enjeux actuels reste d'identifier différentes manières plus générales de définir la distance et de les comparer en termes de performances [RRS15].

1.8.6.2 Approches basées sur les modèles

Plusieurs techniques de recommandation basées sur les modèles ont été proposées dans la littérature notamment pour la recommandation à deux dimensions ($U \times I$) [AT05]. Certaines de ces méthodes peuvent être directement étendues à des problèmes multidimensionnels (utilisateur, élément, et contexte avec l'ensemble de ses dimensions), telles que la méthode proposée par [AEK00] dont les résultats surpassent le filtrage collaboratif.

Une autre méthode proposée par [AEK00] combine les informations sur les utilisateurs et les éléments dans un modèle de préférences, de type *bayésien*. Ce modèle est basé sur la régression hiérarchique et utilise des méthodes de *Monte-Carlo par chaînes de Markov (c.-à-d., Monte Carlo Markov Chain - MCMC)* afin d'estimer ses paramètres. Ce type de méthode se rapprochant de celles employées dans le cadre de cette thèse (bandits-manchots contextuels), nous détaillons son principe au paragraphe suivant.

***Monte-Carlo Markov Chain (MCMC)* :**

- soit U un ensemble de n utilisateurs tels que $U = \{u_1, ..., u_n\}$, où chaque utilisateur $u \in U$ est caractérisé par son vecteur x_u (c.-à-d., attributs de l'utilisateur u, comme par exemple le sexe, l'âge, les préférences) ;
- soit un ensemble de k éléments à recommander défini par $I = \{i_1, ..., i_k\}$, où chaque élément $i \in I$ est caractérisé par son vecteur z_i (c.-à-d., attributs de l'élément i, comme par exemple la catégorie, le prix, le poids, la taille) ;
- soit r_{ui} l'évaluation faite par l'utilisateur u et portant sur l'élément i. L'évaluation peut prendre différentes valeurs selon les applications possibles : $r_{ui} \in \mathbb{R}$ [AEK00], ou $r_{ui} \in \mathbb{N}$ [ZMB15], ou encore $r_{ui} \in \{0, 1\}$ [Li+10] ;
- les évaluations r_{ui} ne sont connues que pour certains sous-ensembles de $U \times I$.

Ainsi, le problème d'estimation de l'évaluation est défini comme suit :

$$\widehat{r}_{ui} = x_u \, \gamma_i + z_i \, \lambda_u + w_{ui} \, \theta + \varepsilon_{ui}, \;\; \varepsilon_{ui} \sim \mathcal{N}(0, \sigma^2), \lambda_u \sim \mathcal{N}(0, B_u), \gamma_i \sim \mathcal{N}(0, B_i)$$

Dans cette équation de régression, θ correspond à la pente (inconnue) de la droite de régression représentée sous forme d'un vecteur de coefficients inconnus (restant à estimer). Le terme γ_i représente les coefficients de pondération propres à l'élément i (c.-à-d., tout ce qui n'est pas explicitement intégré dans le « profil » de l'élément comme par exemple la mise en scène, la musique et le jeu des acteurs pour les films), et λ_u représente les coefficients de pondération spécifiques à l'utilisateur u. L'erreur est représentée par ε_{ui}. Elle suit une distribution normale centrée sur une moyenne de 0 et un écart-type σ. De plus, on considère un modèle hiérarchique en supposant ainsi que les paramètres de régression γ_u et λ_i sont normalement distribués de moyenne centrée sur 0 avec $\lambda_u \sim \mathcal{N}(0, B_u)$ et $\gamma_i \sim \mathcal{N}(0, B_i)$ où B_u et B_i sont des matrices de covariance inconnues. Enfin $w_{ui} = x_u \otimes z_i$ représente le produit de *Kronecker* entre les vecteurs de caractéristiques x_u et z_i résultant ainsi en un long vecteur contenant toutes les combinaisons possibles entre chacune des dimensions de x_u et z_i.

Limites des *MCMC* : Les paramètres du modèle à apprendre sont θ, σ^2, B_u, et B_i. Ils sont estimés à partir des données contenant les évaluations utilisateurs déjà connues en appliquant les méthodes de *MCMC* décrites dans [AEK00]. Par conséquent pour obtenir une meilleure estimation de r_{ui} et de ce fait pour converger vers une meilleure précision des recommandations effectuées, nous noterons l'importance des descripteurs x_u et z_i qui servent de base au modèle. En effet, posséder une description complète et suffisamment pertinente du contexte représenté par ces descripteurs est un avantage indéniable pour obtenir une bonne précision des recommandations en fonction de chacune des situations contextuelles rencontrées. Néanmoins, les avantages de ce genre de méthode sont généralement aussi leurs inconvénients, c'est-à-dire que plus les informations des utilisateurs et des éléments fournies au modèle seront incomplètes (c.-à-d., jeux de données parcimonieux - *sparse*), ou manqueront de pertinence, moins l'estimation calculée sera juste et plus les recommandations produites deviendront imprécises. D'autre part, un trop grand nombre de dimensions des descripteurs peut aussi entraîner un phénomène connu sous le nom de *malédiction de la dimension* (*curse of dimensionality*) [Bel15]. En effet, plus le nombre de dimensions est grand, plus le nombre d'observations nécessaires est important pour que la méthode soit valide statistiquement et afin de ne pas se retrouver avec des données dites « parcimonieuses » (*sparse*). De plus, un trop grand nombre d'observations rend inopérantes ces méthodes puisque la complexité en temps devient trop importante pour résoudre le problème.

Ces limites sont également celles des approches utilisant les bandits-manchots contextuelles. Elles ont constitué un verrou scientifique sur lequel nous avons travaillé dans le cadre de cette thèse. De ce fait, nous avons dédié deux chapitres, référant à deux de nos contributions sur l'amélioration des informations de contexte fournies au modèle.

Dans cette section, nous avons rappelé l'approche des systèmes de recommandation sensibles au contexte et les différentes manières d'aborder et de résoudre le problème. À la section suivante, nous rappelons l'approche des systèmes hybrides pouvant intégrer, entres autres, le contexte.

1.9 Les approches hybrides pour la recommandation

En résumé : afin de pallier les défauts que chaque technique possède individuellement, certaines approches proposent des techniques dites hybrides [Bur02]. Selon la littérature, une combinaison de plusieurs solutions apporte des améliorations significatives de performances par rapport aux approches individuelles [Bur02 ; Bur07 ; Kor08 ; Pen+00 ; RRS15]. La pratique la plus commune aujourd'hui est de combiner le *CF* avec d'autres techniques, notamment celle faisant appel au contexte [Dey01 ; RRS15], afin d'éviter les problèmes de démarrage à froid (*Cold-Start*) [LKH14], de montée à l'échelle (*Sclability*) [Sar01] ou encore des jeux de données parcimonieux (*sparse*) [Bel+13].

1.9.1 Les bases de fonctionnement

Hybrider deux approches implique que chacune d'entre elles puisse apporter un complément en données d'entrée ou de méthodes [JF13].

De ce fait, la question utilisateur qui se posera dans un système hybride dépendra des approches combinées et des données prises en compte.

De même, si nous devions représenter un système de recommandation hybride sur une figure, celle-ci ne serait tout simplement qu'une combinaison possible de toutes les approches existantes.

Il serait peu pertinent d'évoquer toutes les opportunités d'hybridation possibles, c'est pourquoi à la Section 1.9.2 suivante, nous rappellerons les principales identifiées dans la littérature.

1.9.2 Les systèmes de recommandation utilisant une approche hybride

Les systèmes de recommandation utilisant une approche hybride sont basés sur la combinaison des techniques de base des systèmes de recommandation que nous avons décrites dans les sections précédentes. De nombreuses méthodes ont été proposées pour combiner ces techniques afin de créer un nouveau système qualifié d'hybride [Bur02 ; Bur07 ; RRS15].

Ainsi, un système hybride qui combinerait deux techniques distinctes T_1 et T_2, essaierait d'utiliser les avantages de T_1 pour pallier les inconvénients de T_2 [RRS15].

Par exemple, les méthodes de *CF* souffrent de problèmes liés aux nouveaux éléments à recommander, plus connu sous le nom de démarrage à froid (*Cold-Start*) [LKH14], c'est-à-dire qu'elles peinent, voire ne parviennent pas, à recommander correctement des éléments sans évaluations réalisées a priori par des utilisateurs. En revanche, cela n'a pas d'impacts par exemple sur les approches basées sur le contenu, puisque les prédictions calculées pour les nouveaux éléments sont basées sur leur description est en général facilement disponible. Ainsi, on trouverait un avantage à combiner ces deux méthodes [RRS15].

De nombreuses autres propositions d'approches combinées sont également rendues possibles grâce à l'usage du contexte. En effet, le contexte de l'utilisateur lorsqu'il demande une recommandation peut être utilisé pour mieux personnaliser la sortie du système. Par exemple, si on prend en considération le temporalité dans la recommandation (contexte temporel) : les recommandations de vacances en hiver devraient être très différentes de celles proposées en été, comme il en serait de même pour les recommandations de restaurant le samedi soir entre amis qui devraient être différentes de celles pour un déjeuner de travail entre collègues [RRS15].

Dans la sous-section suivante, nous décrirons plus en détail comment il est possible de combiner une approche contextuelle avec du filtrage collaboratif.

1.9.3 Combiner l'approche contextuelle et le filtrage collaboratif

Les systèmes de recommandation sensibles au contexte peuvent offrir plusieurs possibilités d'utilisation d'approches combinées avec le *CF*. L'une de ces possibilités part du fait

que les informations contextuelles complexes peuvent être divisées en plusieurs composants. Par exemple, [Ado+05] a développé une technique combinant des informations provenant de plusieurs *pré-filtres* contextuels différents. Avoir un certain nombre de pré-filtres différents permet d'obtenir plusieurs généralisations différentes (et potentiellement pertinentes) d'un même contexte spécifique. Afin d'illustrer cette méthode, imaginons un contexte observé $c = (Amie, Restaurant, Samedi)$, celui-ci peut être généralisé par $c_1 = (Amie, Quelque\ Part, Samedi)$, $c_2 = (Avec\ quelqu'un, Restaurant, N'importe\ quand)$, et un certain nombre d'autres contextes possibles. Suivant cette idée, [Ado+05] utilisent alors des pré-filtres basés sur le nombre de contextes possibles pour chaque évaluation, puis combinent ainsi les recommandations issues de chaque *pré-filtre* contextuel, c.-à-d., il est possible d'utiliser un ensemble des pré-filtres agrégés ou alors le meilleur des pré-filtres.

On peut résumer le fonctionnement de ce type d'hybridation en trois étapes :

1. Utiliser les évaluations a priori des utilisateurs (données d'entraînement), et déterminer quel(s) pré-filtre(s) surpasse(nt) la méthode de *CF* traditionnelle ;

2. Choisir le meilleur pré-filtre pour un contexte donné afin de prédire l'évaluation spécifique que donnera l'utilisateur dans ce contexte particulier ;

3. Utiliser un algorithme de recommandation (2D) [Ado+05 ; RRS15] permettant d'estimer la fonction d'évaluation R telle que :

   $$R : U \times I \rightarrow Rating$$

   où U correspond au domaine des utilisateurs, I le domaine des éléments, et $Rating$ correspond au domaine des évaluations (c.-à-d., $r_{ui} \in Rating$ est l'évaluation donnée à l'élément $i \in I$ par l'utilisateur $u \in U$. Voir la Section 1.4 traitant du filtrage collaboratif).

Une autre possibilité de combinaison se base sur l'utilité de chaque information contextuelle qui peut être différente selon qu'elle est utilisée dans les étapes de pré-filtrage, de post-filtrage ou de modélisation [RRS15]. Par exemple, s'il est pertinent d'utiliser le pré-filtrage pour des données de contexte temporel [Ado+05], comme le jour de la semaine ou du week-end, il sera préférable d'utiliser du post-filtrage par exemple pour des données de type météorologique [RRS15].

Ainsi, déterminer l'utilité de différentes informations contextuelles par rapport à différents *paradigmes* de systèmes de recommandation utilisant le contexte représente aujourd'hui une perspective de recherche importante.

Dans cette section, nous venons de rappeler l'approche de recommandation hybdride et différentes combinaisons possibles. À la section suivante, nous détaillerons les différentes techniques d'Intelligence Computationnelle. Celles-ci peuvent être employées dans toutes les approches rappelées précédemment selon un angle de résolution basé sur les modèles.

1.10 Les techniques d'Intelligence Computationnelle (IC)

En résumé : les systèmes de recommandation utilisant des techniques dites d'Intelligence Computationnelle - IC (*Computational Intelligence - CI*) [Kon06] offrent des gains en termes de qualité de recommandation [Wan+14]. Cependant, ces techniques sont coûteuses en terme de complexité en temps dans la phase de mise à jour par rapport à des techniques heuristiques traditionnelles [RRS15]. Les techniques considérées comme étant de l'intelligence computationnelle incluent principalement : les réseaux bayésiens [AP15 ; BHK98], les algorithmes de classification (clustering) [GP14 ; KA08 ; LK03 ; Mer99 ; SK12], les réseaux de neurones [BP98 ; CVS07 ; Hsu+07 ; Hu17] , la *SVD* (*Singular Value Decomposition*. Voir sous-section 1.4.4) [Gol+01 ; Sar+00], ou encore les bandits-manchots [AG13 ; Bou+17 ; LZ08 ; Li+10]. Certaines de ces techniques peuvent être par exemple utilisées en filtrage collaboratif basé sur un modèle. Les systèmes de recommandation traditionnels (notamment collaboratifs) possèdent leur propre limites et résistent peu à la non-stationnarité, aux jeux de données parcimonieux, ou encore au démarrage à froid. Ces problèmes restent des freins qui affectent la qualité de la prédiction. De ce fait, au cours des vingts dernières années, l'usage de l'intelligence computationnelle dans les systèmes de recommandation a suscité beaucoup d'intérêt en raison de l'amélioration possible des performances et de la capacité de résolution de ces problèmes [Wan+14].

Dans cette section nous rappellerons différentes méthodes d'intelligence computationnelle qu'il est possible d'utiliser afin de mettre en œuvre un système de recommandation.

1.10.1 L'Intelligence Computationnelle : un composant complémentaire

Nous reprenons la Figure 1.1 générique et montrons où l'Intelligence Computationnelle (IC) peut intervenir dans le processus de recommandation (voir Figure 1.9).

En effet, elle peut intervenir à trois niveaux différents du processus de recommandation et peut intégrer :

— une stratégie de calcul des scores de pertinence sous forme d'une espérance estimée. Celle-ci peut se déterminer en correspondance à des évaluations ou retours passés vis à vis des données du problème (p. ex., profil, contexte ou description des éléments). Ces scores peuvent être calculés pour différents critères, p. ex., précision, diversité, nouveauté ou sérendipité. Si on prend l'exemple des algorithmes de bandits-manchots contextuels basés sur un modèle linéaire [AG13 ; Li+10], l'espérance calculée correspond au produit scalaire entre le vecteur de coefficients θ_i, pour chaque élément à recommander $i \in I$, et le vecteur de contexte utilisateur x_u observé ;

— une méthode de sélection des éléments à recommander qui peut être mono-objectif (p. ex., uniquement basée sur la précision), bi-objectifs (p. ex., basé à la fois sur la précision et la diversité) ou multi-objectifs (p. ex., basée à la fois sur la précision, la diversité, et la nouveauté). Si on prend de nouveau l'exemple des algorithmes de bandits-manchots contextuels basés sur un modèle linéaire, dans le cas mono-objectif la politique de sé-

lection des éléments à recommander est de choisir l'élément qui possède l'espérance calculée la plus élevée ;

— un traitement des retours utilisateurs afin de se baser sur un historique à jour. Ceci est la pierre angulaire de tout apprentissage dynamique notamment en environnement non-stationnaire. Dans l'exemple des algorithmes de bandits-manchots contextuels basés sur un modèle linéaire, ces retours utilisateurs sont traités de telle manière qu'ils sont historisés via un vecteur de coefficients θ_i correspondant à l'estimateur pour chaque élément $i \in I$ à recommander.

Les composants d'IC permettent le bon fonctionnement de certaines approches basées sur les modèles. C'est par exemple le cas du composant d'apprentissage de profil de l'approche basée sur le contenu ou encore du filtrage collaboratif basé sur les modèles. De plus, dans le cadre des systèmes de recommandation contextuels nous verrons comment il devient possible d'utiliser l'IC quand on cherche à contextualiser la fonction de recommandation. Les informations de contexte dans ce cas sont utilisées dans le modèle d'estimation du score de pertinence comme évoqué précédemment dans l'exemple des bandits-manchots contextuels.

Dans la sous-section suivante nous rappelons brièvement quelques méthodes d'IC existantes dans la littérature et qui ont été utilisées pour les systèmes de recommandation.

1.10.2 Revue de techniques existantes

Comme expliqué dans la section 1.3 et rappelé à la sous-section précédente, les techniques d'IC peuvent être employées en tant que composants pour les approches basées sur les modèles. En effet, les techniques d'IC sont largement utilisées pour élaborer des modèles de recommandation et un certain nombre d'entre-elles ont déjà été employées dans différents types de systèmes de recommandation.

[Kon06] classe l'intelligence computationnelle en 4 familles : le *Granular Computing* (p. ex., logique floue, raisonnement probabiliste), le *Neuro-Computing* (p. ex., apprentissages supervisé, non supervisé, et par renforcement), l'*Evolutionary Computing* (p. ex., algorithmes génétiques) et l'*Artificial Life* (p. ex., système immunitaire artificiel). Ci-dessous nous rappelons brièvement certaines des techniques d'IC pour la recommandation :

1.10.2.1 Les approches bayésiennes

À travers une méthodologie probabiliste, les approches bayésiennes [AP15] permettent de résoudre des problèmes de classifications. Pour les systèmes de recommandation [ON08], dans un réseau bayésien construit en appliquant un algorithme d'apprentissage comme celui décrit par [CHM97], les nœuds correspondent aux éléments à recommander et les états pour chaque nœud correspondent aux valeurs d'évaluation possible. Ce réseau peut également considérer un état correspondant à l'absence d'évaluation en cas de données manquantes. Ainsi, les résultats de ce réseau [ON08] se présentent sous forme d'arbres de décision représentant chaque table de probabilité conditionnelle (espérance d'évaluation) pour chaque nœud.

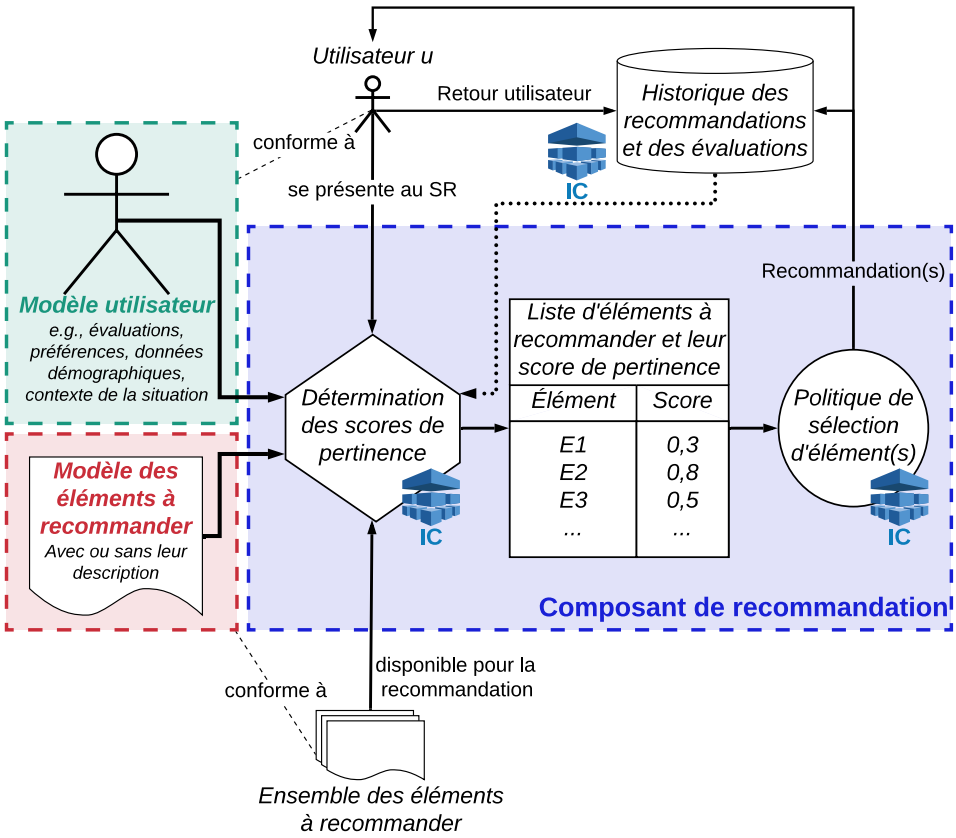


FIGURE 1.9 – Principe général de fonctionnement d'un système de recommandation utilisant des techniques d'IC

1.10.2.2 Les réseaux de neurones artificiels

Les réseaux de neurones artificiels peuvent être aussi élaborés pour résoudre des problématiques liées aux systèmes de recommandation [CVS07 ; Hsu+07 ; Hu17]. Ceux-ci représentent un ensemble de nœuds inter-connectés appelés *neurones*, inspirés par les neurones biologiques qu'on retrouve dans un cerveau. Dans le cas des systèmes de recommandation, il existe de nombreuses modélisations possibles pour résoudre le problème. L'un des modèles les plus populaires est celui de la recommandation de vidéos Youtube [CAS16] et se présente comme suit : les dimensions observées du contexte sont les paramètres d'entrée du réseau de neurones (c.-à-d., profil utilisateur, caractéristiques des vidéos déjà vues, et recherchées), les évaluations des recommandations de vidéos effectuées par rapport à ces paramètres correspondent aux sorties (visualisation(s) complète(s) ou partiel(les) des vidéos recommandées). Enfin, au sein de ces réseaux de neurones résident des couches cachées entièrement connectées (*fully connected*), qui possèdent un ensemble de poids correspondant à chacune des variables étudiées en entrée (dimensions du contexte) par rapport à chaque recommandation envisagée. Durant la phase d'entraînement le réseau re-calcule de manière itérative la valeur

de ses poids entre autres par rétro-propagation du gradient (c.-à-d., calcul du gradient de l'erreur pour chaque neurone de la dernière couche du réseau vers la première). En fonction du jeu de données, l'entraînement peut nécessiter plusieurs itérations de plusieurs époques (tout en évitant le sur-apprentissage, problème de biais-variance). Au fur et à mesure de l'entraînement, ces poids se révèlent en corrélation, plus ou moins forte, des résultats observés en sortie (réussite des recommandations pour chaque élément). Ceci permet à terme, de réaliser des prédictions sur la réussite des recommandations à des utilisateurs évoluant dans un contexte donné, en sélectionnant celles ayant le meilleur score de prédiction.

1.10.2.3 Les méthodes de partitionnement

Les techniques de partitionnement (*clustering*) classifient les différents éléments à recommander dans des groupes en fonction de leur similarité les uns par rapport aux autres [GP14 ; SK12]. En effet, des éléments appartenant à un même groupe sont nécessairement plus proches en termes de similarité que ceux appartenant à un groupe différent. Les méthodes principalement utilisées sont celles des *k plus proches voisins (k-NN)* ou encore le partitionnement en *K-moyennes (K-means)*.

1.10.2.4 La logique floue

Les approches utilisant la logique floue sont également utilisées dans les systèmes de recommandation [ZN09]. Elles permettent une meilleure gestion de l'incertitude non-stochastique et sont par conséquent bien adaptées à la manipulation d'informations imprécises, à l'absence de connaissance des situations ou du contexte, et à l'évolution continuelle des profils et des préférences utilisateurs [Cor+07 ; Yag03].

1.10.2.5 Les méthodes évolutionnaires

En tant que technique de recherche stochastique généralement adaptée pour les problèmes d'optimisation, les algorithmes génétiques [Bob+11] sont particulièrement utilisés dans les systèmes de recommandation selon deux aspects : le clustering (*GA-based K-means clustering* [KA08]) et les modèles utilisateurs hybrides [AB08].

1.10.2.6 Les bandits-manchots

Les algorithmes permettant de résoudre le problème de bandits-manchots ont été de nombreuses fois adaptés et expérimentés dans le cadre des systèmes de recommandation [Bou14 ; Gal15 ; Li+10 ; LKG16]. Les algorithmes de bandits-manchots utilisent les observations passées pour optimiser leurs choix futurs dans l'unique objectif de maximiser leurs gains. Lorsque un algorithme de bandits-manchots effectue une recommandation à un utilisateur, il reçoit en retour une évaluation (note) en rapport à cette recommandation. L'évaluation est prise en compte sous forme de récompense et la somme cumulée de ces récompenses correspond

aux gains obtenus par l'algorithme. La précision globale des recommandations effectuées aux utilisateurs dépend donc de la capacité de ces algorithmes à maximiser leurs gains.

Comme évoqué en introduction, nous avons apporté plusieurs contributions au problème des bandits-manchots dans le cadre de cette thèse. Le Chapitre 2 leur est dédié et décrit en détails les différents algorithmes qui permettent de le résoudre.

1.11 Synthèse

Dans ce chapitre nous avons abordé l'ensemble des approches existantes dans le cadre des systèmes de recommandation.

Cette thèse portant sur les systèmes de recommandation à des utilisateurs mobiles dans la ville, dans cette section, nous allons comparer ces différentes approches afin de justifier notre choix.

1.11.1 Critères

Afin d'effectuer notre choix, nous avons retenu trois critères auquels ce type de systèmes doit satisfaire :

1. **L'explicabilité** c.-à-d., pouvoir expliquer les phénomènes observés notamment via des données de contexte (entité, temps, individualité, localisation, activités, relations) ;
2. **La généralisation** c.-à-d., avoir des garanties théoriques comme le sont les preuves de convergence ;
3. **La dynamicité** c.-à-d., résister à des contraintes de non-stationnarité inhérentes à la mobilité et à l'évolution des goûts, des souhaits et des besoins utilisateurs ; pouvoir fonctionner en quasi temps-réel ; être robuste à la parcimonie de données ou encore au démarrage à froid.

À ces critères s'ajoutent également les contraintes des **données** dont nous disposons. En effet, dans le cadre de cette thèse nous ne bénéficions pas :

— d'une communauté d'utilisateurs pour lesquels nous possédons des connaissances a priori (évaluations passées, préférences ou consultations passées) ;
— d'informations inhérentes aux réseaux sociaux.

1.11.2 Exclusions

Dans le tableau comparatif ci-dessous, nous excluons toutes les méthodes basées sur une heuristique car elles ne répondent pas pleinement au critère de dynamicité et de généralisation. Aussi, nous focaliserons nos comparaisons sur les approches utilisant des méthodes basées sur les modèles. De plus, la recommandation à des utilisateurs mobiles s'effectue indépendamment à un seul individu à la fois et vise de ce fait la personnalisation. C'est pourquoi

nous écartons la possibilité d'utiliser une approche de recommandation à des groupes et ne l'intégrons pas non plus dans notre tableau de synthèse.

1.11.3 Évaluations

Dans le tableau 1.1 nous évaluons chaque approche avec un score de pertinence pour chacun des critères comme suit :
$+$: Approprié à notre sujet de recommandation de services à des utilisateurs mobiles
$+-$: Plus ou moins approprié à notre sujet de recommandation de services à des utilisateurs mobiles
$-$: Pas approprié à notre sujet de recommandation de services à des utilisateurs mobiles

TABLE 1.1 – Tableau comparatif des approches de systèmes basés sur des modèles

Approches	**Explicabilité**	**Généralisation**	**Dynamicité**	**Données**
Filtrage Collaboratif	$+-$	$+$	$-$	$-$
Basée sur le contenu	$+-$	$+$	$+$	$+$
Basée sur les connaissances	$+-$	$+$	$-$	$+-$
Contextuelle	$+$	$+$	$+$	$+$

1.11.4 Analyse

En se rapportant au tableau 1.1, nous remarquons par ordre de pertinence croissant que :
— le **filtrage collaboratif** bien qu'étant l'une des approches les plus populaire et offrant de nombreux avantages, ne peut correspondre totalement aux exigences requises pour notre sujet. En effet, les points bloquants se situent sur les données que nous avons à disposition mais aussi les limites de cette approche sur des questions de dynamicité (manque de résistance à la non stationnarité, problème de démarrage à froid) ;
— l'**approche basée sur les connaissances** souffre de dynamicité puisqu'un tel système nécessite de maintenir ces connaissances à jour afin d'être efficace. Or, nous souhaitons que notre système de recommandation à des utilisateurs mobiles soit le plus autonome possible et nécessite le moins de maintenance humaine ;
— l'**approche basée sur le contenu** quant à elle est pertinente mais n'inclut pas explicitement l'utilisation du contexte dans l'apprentissage des préférences utilisateurs. Or notre système de recommandation à des utilisateurs mobiles sera soumis à de fortes contraintes contextuelles des utilisateurs et sera de ce fait très sensible aux cinq catégories du contexte (voir Chapitre 3) ;
— l'**approche tenant compte du contexte** est considéré selon nos critères d'évaluation comme étant la plus pertinente pour notre sujet. En effet, cette approche intègre à la fois

l'ensemble des paramètres de contexte requis permettant à la fois d'améliorer l'explicabilité des algorithmes mais aussi la résolution du problème de recommandation à des utilisateurs mobiles. De plus, cette approche a également l'avantage de répondre à des contraintes de dynamicité (temps-réel, résistance à la non stationnarité). Enfin, selon la méthode d'approche contextuelle choisie, il n'est pas toujours nécessaire de posséder des données a priori sur la communauté utilisateur.

1.12 Bilan

Dans ce chapitre nous avons rappelé les différentes approches de recommandations existantes dans la littérature.

Nous avons ensuite effectué une analyse afin de déterminer laquelle de ces approches pourrait le mieux correspondre à notre sujet de recommandation de services à des utilisateurs mobiles dans la ville.

Selon notre analyse à la sous-section 1.11.4, nous avons donc opté pour une approche contextuelle de la recommandation basée sur les modèles.

Néanmoins, dans le cadre d'une approche contextuelle basée sur les modèles, il existe de nombreuses méthodes faisant appel à des techniques d'Intelligence Computationnelle que nous avons rappelées à la section 1.10. La question qui se pose désormais est le choix de la technique à employer dans notre cas.

Parmi les familles d'IC rappelées dans la sous-section 1.10.2, nous avons choisi celle du *Neuro-Computing* qui garantit entre autres des fondements théoriques solides. De plus, les techniques issues de cette famille ont été largement employées dans le cadre des systèmes de recommandations et ont été jugées pertinentes (Voir références de la Section 1.10).

Parmi les approches possibles dans la famille *Neuro-Computing*, nous avons choisi celles du domaine de l'apprentissage par renforcement qui possède l'avantage, entre autres, de ne nécessiter aucune connaissances a priori c.-à-d., nul besoin de données a priori d'une communauté d'utilisateurs.

Enfin, parmi les techniques existantes en apprentissage par renforcement, nous avons choisi d'utiliser celles résolvant le problème du bandits-manchots pour la recommandation. C'est un problème de décision séquentielle qui se prête bien aux contraintes de notre sujet en termes notamment de dynamicité.

Le problème des bandits-manchots pour la recommandation étant une thématique de recherche à part entière, nous consacrerons le chapitre 2 à énoncer le problème et à décrire les algorithmes existants pour le résoudre. À la fin du chapitre 2 nous effectuerons une analyse comparative de ces algorithmes afin de retenir ceux qui correspondent le mieux à nos besoins.

CHAPITRE 2

LES BANDITS-MANCHOTS POUR LA RECOMMANDATION

Some aspects of the sequential design of experiments

« *Nevertheless, enough is visible to justify a prediction that future results in the theory of sequential design will be of the greatest importance to mathematical statistics and to science as a whole.* » [Rob52]

Herbert Robbins - 1952

Sommaire

2.1 Introduction

Le problème du bandit-manchot *(Multi-Armed Bandit - MAB)* est un sujet qui a suscité de nombreuses recherches depuis sa première formalisation en 1952 [Rob52]. De nombreuses formulations ont pu être proposées : stochastiques [Aue02 ; BF16 ; LR85], ou encore bayésiennes [AG12]. Plus précisément, le défi pour tout problème de *MAB* consiste à construire une stratégie visant à tirer le bras [1] optimal sans connaissance préalable de la rentabilité de chacun des bras disponibles. La résolution de ce problème consiste à trouver un compromis entre l'exploration de l'ensemble des bras pour en déduire leurs rentabilités et l'exploitation de ce qui a été inféré pour favoriser la sélection des bras optimaux. Une version étendue de ce problème prend en compte le contexte. Il s'agit du problème de bandits-manchots contextuels *(Contextual Multi-Armed Bandit - CMAB)* [AG13 ; LZ08]. Ainsi, dans une approche *CMAB*, le

1. levier du bandit manchot en référence aux machines à sous.

défi visant à déterminer le bras optimal reste le même que pour un problème de *MAB* mais le choix du bras doit tenir compte du contexte des utilisateurs.

Dans ce chapitre nous rappellerons le problème du bandit-manchot contextuel et non-contextuel, et certains algorithmes permettant de le résoudre. Nous étendrons également la définition du problème de bandits à celui de la recommandation et ce pour chaque cas (contextuel et non-contextuel). De plus, nous évoquerons les contraintes associées en termes de non-stationnarité en fonction de l'algorithme employé pour résoudre le problème. Enfin, nous dresserons une synthèse sur l'ensemble des algorithmes que nous avons rappelés dans ce chapitre afin de justifier les choix effectués pour notre sujet de recommandation à des utilisateurs mobiles.

2.2 Le problème du bandit-manchot

Avant de décrire le problème du bandit-manchot contextuel dans le cadre de la recommandation, il convient de rappeler en amont le problème classique de bandits-manchots stochastiques et des algorithmes pour les résoudre. De plus, tout au long de nos expérimentations, nous confrontons les algorithmes de *MAB* et *CMAB* face à notre problématique concrète qu'est la recommandation à des utilisateurs mobiles.

Notons, que dans le cadre de cette thèse, nous nous intéresserons plus particulièrement au problème de bandits-manchots de type Bernoulli où les récompenses retournées par l'utilisateur seront : 0 si l'utilisateur ne valide pas la recommandation qui lui a été faite (c.-à-d., recommandation non pertinente) ; 1 si l'utilisateur valide la recommandation qui lui a été faite (c.-à-d., recommandation pertinente).

2.2.1 Définitions générales du problème des bandits-manchots

Selon [Rob52], nous pouvons définir formellement le problème de bandits-manchots contextuels (voir Définition 1), les algorithmes pour le résoudre (Définition 2) et le regret (Définition 3).

Définition 1 ***. Problème du bandit-manchot (MAB)****. Dans un problème de MAB, il existe une distribution de récompenses D_r de moyenne $\mu_a \in [0,1]$, et de support $r \in [0,1]$ (c.-à-d., r correspondant aux récompenses retournées), telle que $D_r = (\mu_{a_1}, ..., \mu_{a_k}) \in [0,1]^k$, où $A = \{a_1, ..., a_k\}$ un ensemble donné de k bras indépendants à tirer et μ_{a_i} la probabilité de récompense d'un bras $a_i,\ i \in [1,k] \subseteq \mathbb{N}$. Le problème est séquentiel : à chaque itération, un échantillon $(r_{a_1}, ..., r_{a_k}), r_{a_i} \in \{0,1\}$ est tiré de D_r, puis un bras $a_i \in A$ est choisi par l'agent, et sa récompense r_{a_i} est alors révélée.*

Définition 2 ***. Algorithme de bandits-manchots (MAB)****. Un algorithme $\mathcal{A}$ de MAB sélectionne un bras $a_i \in A,\ A = \{a_1, ..., a_k\}$ à chaque itération t, en se basant sur la séquence des précédentes observations $(a_{i,1}, r_{a_i,1}), ..., (a_{i,t-1}, r_{a_i,t-1})$ et on conserve $\vec{r_t}$ le vecteur de récompenses.*

L'objectif est de maximiser la récompense totale attendue $\sum_{t=1}^{T} \mathbb{E}_{\vec{r_t} \sim D[r_{a,t}]}$. Une politique optimale a la connaissance des récompenses moyennes de chaque bras et joue a^* le bras ayant la plus haute récompense moyenne, c.-à-d., $a^* = \arg\max_{a \in A}(\mu_a)$. Ainsi, afin de mesurer la performance d'un algorithme $\mathcal{A}$ de *MAB*, on peut mesurer le regret cumulé obtenu $\rho_T(\mathcal{A})$ (T l'horizon) par rapport à celui d'une politique optimale. On peut donc donner la définition suivante du regret (Définition 3).

Définition 3 ***. Le Regret****. Le gain d'une politique optimale à l'horizon T est $g_T^* = \sum_{t=1}^{T} r_{a^*,t}$. De ce fait, l'espérance du gain de la politique optimale est $\mathbb{E}[g_T^*] = T\ \mu^*$. Considérons $\mathcal{A}$ tout algorithme de bandit-manchot. Le regret cumulé de l'algorithme ayant joué la séquence de bras $a_{i,1}, ..., a_{i,T}$ est donc $\rho_T = g_T^* - \sum_{t=1}^{T} r_{a_i,t}$.*

2.2.2 Énoncé du problème de bandits-manchots pour la recommandation

Un problème de bandits-manchots stochastiques est composé d'un ensemble de k bras indépendants $A = \{a_1, ..., a_k\}$. Lorsqu'on pose le problème du bandit-manchot pour la recommandation à des utilisateurs $u \in U$, alors à chaque bras $a \in A$ est associé un élément à recommander, c'est à dire dans notre cas un service.

Chaque bras de A (c.-à-d., élément à recommander) possède une probabilité cachée de récompense qu'on assumera *constante* dans un premier temps $\mu_a \in [0, 1]$ traduit par une distribution de récompense probable dans l'espace des bras $D_r = (\mu_{a_1}, ..., \mu_{a_k}) \in [0, 1]^k$. À chaque itération $t \in [1, T]$, T étant l'horizon, un utilisateur $u \in U$ se présente pour recevoir une recommandation. Alors, un agent actionne un bras $a_t \in A_t$ et observe la récompense obtenue $r_{t,a}$ tirée depuis D_{μ_a}. Dans le cadre de la recommandation, cette récompense obtenue correspond au retour de l'utilisateur (p. ex., évaluation, retour positif ou négatif). Un bras a est dit *optimal* si celui-ci permet de gagner, en moyenne, la récompense la plus élevée :

$$a^* = \arg\max_{a \in A}(\mu_a)$$

La récompense moyenne de a^* est de ce fait :

$$\arg\max_{a \in A}(\mu_a) = \mu_a^*$$

L'objectif revient donc à trouver une stratégie garantissant de minimiser le regret ρ_T à l'horizon T tel que :

$$\rho_T = T\mu^* - \sum_{t=1}^{T} \widehat{r_t}$$

ou en d'autres termes permettant de tirer le bras optimal à chaque itération t tel que :

$$a_t^* = \arg\max_{a \in A}(\mu_{a,t})$$

Notons que les bandits-manchots stochastiques reposent sur l'hypothèse de stationnarité des distributions [All16] où la distribution D_r reste inchangée tout au long des itérations. Dans ce cas, on peut considérer le problème de bandits-manchots comme un processus de *Markov* à état-unique.

2.2.3 Algorithmes de bandits-manchots pour la recommandation

À chaque itération t, un algorithme résolvant le problème de bandits-manchots pour la recommandation doit donc sélectionner le meilleur élément à recommander (c.-à-d., le bras optimal) pour un utilisateur, en tenant compte de ses précédents retours (c.-à-d., récompenses). De nombreux algorithmes ont été proposés pour résoudre le problème de bandits-manchots, parmi lesquels *Thompson Sampling (TS)* défini par [AG12 ; Tho33], ε*-Greedy* [SB98 ; Wat89], ε*-first* analysé par [EMM02 ; MT04], *Upper Confidence Bounds (UCB)* proposés par [Aue02], ou encore des algorithmes de type *Softmax* [Luc12] comme *EXP3* [Aue+95].

Aussi, à la sous-section suivante nous rappelons ces algorithmes, parmi les plus populaires de la littérature.

2.2.4 Les algorithmes de bandits-manchots

Dans cette sous-section, nous rappellerons différents algorithmes de bandits-manchots *(MAB)* pouvant être employés pour la recommandation.

Il s'agira notamment d'algorithmes :

— de type *glouton* avec un mécanisme d'exploration aléatoire : ε*-Greedy* [Wat89] et ε*-First* [EMM02] ;
— de type *glouton* à bornes de confiance supérieure *UCB1* [Aue02] et *UCB2* [ACF02] ;
— de type *glouton* à bornes de confiance supérieure avec un mécanisme de diversification de type fenêtre glissante *SW-UCB* [GM11] ;
— de type probabiliste *Thompson Sampling* [AG12] ;
— avec adversaire *Softmax* [Luc12] et *EXP3* [Aue+02].

Par soucis de clarté, pour chaque algorithme de bandits-manchots pour la recommandation, le flux de données séquentiel d'utilisateurs se présentant pour recevoir une recommandation est simulé par une sélection aléatoire parmi les instances de l'ensemble du jeu de données jusqu'à un horizon T défini dès le début. Le protocole de simulation sera précisé au Chapitre 4.

2.2.4.1 ε*-Greedy*

En premier lieu, à l'instar des algorithmes dits de *gloutons*, le mécanisme d'ε*-Greedy* [Wat89] suit une stratégie de sélection systématique du bras ayant la meilleure valeur d'action estimée. Cette stratégie de sélection *gloutonne* se traduit par la sélection du bras considéré comme optimal c'est à dire possédant la meilleure espérance de récompense mesurée :

$$\arg\max_{a \in A}(\widehat{\mu}_{a,t})$$

où $\widehat{\mu}_{a,t}$ est la moyenne représentant la valeur d'action moyenne estimée. ε-*Greedy* se base certes sur le fonctionnement d'une stratégie *gloutonne* mais il incorpore en plus une proportion constante ou décroissante ε d'exploration aléatoire [SB98 ; Wat89]. De ce fait, le principe de l'algorithme ε -*Greedy* sera de choisir, à chaque itération, un bras (c.-à-d., élément à recommander) au hasard avec une probabilité $\varepsilon \in [0, 1]$ et le bras possédant la meilleure valeur d'action estimée avec une probabilité de $1 - \varepsilon$.

Il existe deux principales implémentations de l'algorithme ε-*Greedy* :

1. Avec $\varepsilon \in [0, 1]$ fixe où si un ensemble de bras contient k bras, alors la borne supérieure du regret [SB98] sera en :
$$\mathcal{O}\left(\frac{\varepsilon}{k} \sum_{a \in A} \Delta_a\right)$$
où $\Delta_a = \mu^* - \mu_a$;
2. Avec ε décroissant où si un ensemble de bras contient k bras, alors la borne supérieure du regret [SB98] sera en :
$$\mathcal{O}\left(\frac{k \ln T}{\Delta_{min}^2}\right)$$
où $\Delta_{min} = \min_{a \in A \setminus \{a^*\}} (\mu^* - \mu_a)$.

Ainsi, un algorithme de recommandation utilisant ε-*Greedy* sélectionne les éléments à recommander selon l'algorithme 5 (voir Annexe A.1).

Bien que la vitesse de convergence soit inversement proportionnelle à ε, l'intérêt d'une telle stratégie est qu'elle permet une meilleure exploration des solutions possibles. Néanmoins, dans un cas de flux stationnaire seule une stratégie gloutonne peut garantir la convergence des valeurs d'action estimées vers celles réelles. C'est pourquoi dans ce cadre, il peut devenir intéressant de faire diminuer la valeur d'ε au fur et à mesure des itérations afin de ré-adopter une stratégie purement gloutonne [SB98 ; TP11]. En revanche, dans le cas de flux non stationnaires, la stratégie ε-*Greedy* classique (sans diminution d'ε) restera intéressante puisqu'elle permettra l'exploration continue nécessaire à la recherche de la nouvelle meilleure valeur d'action estimée.

2.2.4.2 ε-*First*

L'une des variantes les plus simples d'ε-*Greedy* est ε-*first* [EMM02]. La stratégie ε-*first* consiste à effectuer la phase d'exploration en un seul tenant au début durant une période prédéfinie sur l'ensemble des itérations $T \in \mathbb{N}$. Ainsi, durant les εT premiers tours, les bras sont sélectionnés au hasard. Puis, une fois cette phase d'exploration pure terminée, suit la phase d'exploitation pure durant $(1 - \varepsilon)T$ itérations pendant lesquelles l'algorithme sélectionne le bras ayant obtenu la moyenne estimée la plus élevée. À noter que comme pour ε-*Greedy*, $\varepsilon \in [0, 1]$.

Les analyses de [EMM02 ; MT04] ont démontré que si un ensemble de bras contient k bras, alors la borne supérieure du regret de l'algorithme ε-*first* sera en :

$$\mathcal{O}\left(1 + \frac{k \ln(2kT)}{\Delta_{min}^2}\right)$$

Un algorithme de recommandation utilisant ε-*first* sélectionne les éléments à recommander selon l'algorithme 6 (voir Annexe A.2).

2.2.4.3 *UCB1* et *UCB2*

UCB1 pour – *Upper Confidence Bounds* – [Aue02] se fonde sur les travaux de [Agr95 ; Rob52]. *UCB1* est un algorithme basé sur la notion d'optimisme face à l'incertitude et de ce fait fonctionne sur le principe des stratégies à bornes de confiance. Il utilise la borne de *Chernoff-Hoeffding* [Che+52 ; Hoe63] afin de calculer un intervalle de confiance sur la récompense moyenne μ_a de chaque bras a. Le mécanisme de base d'*UCB1* est de suivre une stratégie de sélection systématique du bras ayant la meilleure valeur d'action estimée. En revanche, en plus de considérer ces valeurs d'action estimées, *UCB1* prendra également en compte la confiance que l'on a en chacune de ces valeurs. De ce fait la stratégie d'*UCB1* dans notre cas d'étude consistera en la sélection d'un élément à recommander (c.-à-d., un bras) comme suit :

$$\arg\max_{a \in A}\left(\widehat{\mu}_{a,t} + \sqrt{\frac{\alpha \ln t}{t_a}}\right)$$

où t représente l'itération courante ($t \in [1, T]$), t_a le nombre de fois que le bras a a été recommandé à l'itération t, et $\alpha \in \mathbb{R}^+$ est un facteur multiplicateur où généralement $\alpha = 2$. Le *bonus* $\sqrt{\frac{\alpha \ln t}{t_a}}$ représente quant à lui la valeur de l'intervalle de confiance cité précédemment.

Ainsi, il a été démontré par [Aue02] en utilisant l'inégalité de *Chernoff-Hoeffding* que :

$$\mathbb{P}\left[\widehat{\mu_a} + \sqrt{\frac{\alpha \ln T}{t_a}} \leq \mu_a\right] \leq \exp\left(-2t_a\left(\frac{\alpha \ln(T)}{t_a}\right)\right) = \frac{1}{T^{2\alpha}}$$

L'analyse de l'espérance du regret d'*UCB1* [Aue02] a démontré que :

$$\mathbb{E}[\rho_T] \leq 8 \sum_{a:\Delta_a > 0} \frac{\ln T}{\Delta_a} + \left(1 + \frac{\pi^2}{3}\right) \sum \Delta_a$$

où $\Delta_a = \mu^* - \mu_a$. Ainsi pour un ensemble de k bras, alors la borne supérieure du regret [Aue02] sera en :

$$\mathcal{O}\left(\frac{k \ln T}{\Delta_{min}}\right)$$

où $\Delta_{min} = \min_{a \in A \setminus \{a^*\}} (\mu^* - \mu_a)$.

Un algorithme de recommandation utilisant *UCB1* sélectionne les éléments à recommander selon l'algorithme 7 (voir Annexe A.3).

L'un des inconvénients majeurs d'*UCB1* est qu'il dépend fortement des conditions initiales. Ainsi, l'une des améliorations intéressantes d'*UCB1* est *UCB2*. Il permet de réduire la fraction de temps pendant laquelle un bras sous-optimal est sélectionné, réduisant ainsi le regret global, au prix d'un algorithme légèrement plus compliqué.

UCB2 [ACF02], est une amélioration dite itérative d'*UCB1* basée sur des époques dont la longueur augmente de façon exponentielle au cours du temps [BLL15].

La stratégie d'*UCB2* consiste en la sélection d'un bras a (c.-à-d., élément à recommander) comme suit :

$$\arg\max_{a \in A} \left(\widehat{\mu}_a + \sqrt{\frac{(1+\alpha)(\ln e\ t_a/(1+\alpha)^{j_a})}{2+(1+\alpha)^{j_a}}} \right)$$

durant $[(1+\alpha)^{j_a+1} - (1+\alpha)^{j_a}]$ fois avant de terminer l'époque et de choisir un nouveau bras. Notons que j_a est un compteur indiquant combien d'époques le bras $a \in A$ a été sélectionné et $0 < \alpha < 1$ est un paramètre qui influence le taux d'apprentissage.

Selon l'analyse de [ACF02], si un ensemble de bras contient k bras, alors la borne supérieure du regret d'*UCB2* pour $t \geq \max_{a:\mu_a<\mu^*} \frac{1}{2\Delta_a^2}$ sera en :

$$\mathcal{O}\left(\sum_{a:\mu_a<\mu^*} \frac{(1+\alpha)(1+4\alpha)\ln(2e\Delta_a^2 t)}{2\Delta_a} + \frac{c_\alpha}{\Delta_a} \right)$$

où e est la constante d'*Euler* et comme prouvé par [ACF02] :

$$c_\alpha = 1 + \frac{(1+\alpha\ e)}{\alpha^2} + \frac{(1+\alpha)}{\alpha^{(1+\alpha)}}(1 + \frac{11(1+\alpha)}{5\alpha^2 \ln(1+\alpha)})$$

2.2.4.4 *Thompson Sampling (TS)*

Poursuivons par la plus ancienne des politiques datant de 1933 : l'échantillonnage de Thompson (ou *Thompson Sampling* - TS) [Tho33].

TS offre des performances équivalentes à celles d'autres algorithmes tels que ε-*Greedy* ou de type *UCB*, et donne même de meilleurs résultats dans plusieurs cas d'utilisation tels que la recommandation d'articles d'actualités ou de publicités [CL11].

À l'instar d'ε-*Greedy*, il s'agit d'une politique randomisée [Lou+15], à la différence près que *Thompson Sampling* utilise l'aléatoire d'une autre manière, c.-à-d., d'inspiration bayésienne. Ainsi, *Thompson Sampling* (*TS*) [AG12 ; Tho33] est l'une des approches classiques du problème de bandits-manchots, connue comme une méthode bayésienne d'échantillonnage a posteriori.

Concrètement, *TS* consiste à tirer à chaque itération t un échantillon $\gamma_{a,t}$ de la loi a posteriori courante $Pr(\tilde{\mu}_{t,a})$, où $\mu_{t,a}$ est la moyenne de récompense (ou la probabilité de succès) du bras a

à l'itération t, et à sélectionner le bras ayant conduit à l'échantillon correspondant à la moyenne la plus élevée, c.-à-d., $a_t^* = \arg\max \gamma_{a,t}$. Ainsi, avec *TS*, on suppose que la récompense $r_{t,a}$ obtenue après la sélection d'un bras a à l'itération t suit la distribution $Pr(r_{t,a}|\tilde{\mu}_{t,a})$.

La plupart des travaux, dont ceux menés dans cette thèse, considère le problème de bandit-manchot sous sa forme de *Bernoulli* avec des récompenses binaires, c.-à-d., $r_{t,a} \in \{0, 1\}$. Par conséquent, on considère $S_{t,a}$ le nombre de succès observés ($r_{t,a} = 1$), et $F_{t,a}$ le nombre d'échecs observés ($r_{t,a} = 0$). Dans *TS*, comme la distribution a priori $Pr(\tilde{\mu}_{t,a})$ est donnée par $Beta(S_{t,a} + 1, F_{t,a} + 1)$, alors la distribution a posteriori $Pr(\tilde{\mu}_{t,a}|r_{t,a}) \propto Pr(r_{t,a}|\tilde{\mu}_{t,a})Pr(\tilde{\mu}_{t,a})$ à l'itération $t + 1$ est calculée en utilisant le théorème de Bayes comme suit : $Beta(S_{t+1,a} + 1, F_{t+1,a} + 1)$.
Plus précisément, *TS* procède comme suit : à l'initialisation, *TS* suppose que chaque bras $a \in A$ possède une distribution bêta antérieure $Beta(1, 1)$ sur μ_a. Puis, à chaque itération t, l'algorithme de *TS* génère des échantillons indépendants $\gamma_{t,a}$ à partir de la distribution a posteriori $\tilde{\mu}_{t,a}$, et sélectionne le bras obtenant la plus grande valeur obtenue de l'échantillon généré (c.-à-d., $a_t = \arg\max_{a \in A} \gamma_{t,a}$). Finalement l'algorithme incrémente $S_{t,a}$ et $F_{t,a}$ selon la récompense obtenue suite à la recommandation du bras a (c.-à-d., recommandation de l'élément à l'utilisateur).

Selon l'analyse de [AG12], si un ensemble de bras contient k bras, alors la borne supérieure du regret de *TS* sera en :

$$\mathcal{O}\left((\sum_{a:\Delta_a>0} \frac{1}{\Delta_a})^2 \ln T \right)$$

Ainsi, un algorithme de recommandation utilisant *TS* [AG12] sélectionne les éléments (c.-à-d., bras) à recommander selon l'algorithme 8 (voir Annexe A.4).

2.2.4.5 *Softmax*

Originellement, la stratégie *Softmax* [Luc12] consiste en un choix aléatoire selon une distribution de *Gibbs* [Luc05 ; Luc]. Ainsi, dans le cadre de l'utilisation de *Softmax* pour un problème de bandits-manchots, le bras a (c.-à-d., l'élément à recommander) est choisi avec une probabilité p_a définie comme suit :

$$p_a = \frac{e^{\frac{\widehat{\mu_a}}{\tau}}}{\sum_{i=1}^{k} e^{\frac{\widehat{\mu_a}}{\tau}}}$$

où $\tau \in \mathbb{R}^+$ est un paramètre appelé *température* tel que plus $\tau \to \infty$ alors plus on explore, et plus $\tau \to 0$ plus on exploite. Il est alors possible de paramétrer les valeurs de τ pour favoriser soit l'exploration soit l'exploitation.

Dans le cadre de l'utilisation de *Softmax* selon une distribution de *Gibbs*, alors τ est choisi dès de le départ et reste constant [VM05]. A contrario, une autre version de la stratégie de *Softmax* consiste à travailler sur une distribution de *Boltzmann* [CF98] où la valeur de τ est décroissante au fur et à mesure des itérations [CF98]. À l'origine, ces distributions furent comparées en rapport à la seconde loi de la thermodynamique où la distribution de *Boltzmann*

semblait plus adaptée puisqu'elle tient compte de l'entropie d'un système avec une valeur de τ décroissante, au contraire de la distribution de *Gibbs* [Jay65]. En paramétrant un τ décroissant au fur et à mesure des itérations, une approche selon la distribution de *Boltzmann* semble donc plus adaptée dans le cas d'une distribution stationnaire, puisqu'elle permet de favoriser une phase exploratoire en premier lieu, puis de passer au fur et à mesure sur une phase d'exploitation de la meilleure solution identifiée.

C'est le cas de l'algorithme de *Metropolis-Hastings* qui est une méthode de *Monte-Carlo* par chaînes de *Markov*. Il a été développé pour décrire l'évolution d'un système thermodynamique. Cet algorithme permet de calculer une distribution de probabilité en construisant la chaîne de *Markov* correspondant à la distribution D_r.

Softmax Annealing est l'un des algorithmes adapté de *Metropolis-Hastings* utilisé dans le cadre de résolution du problème de bandit-manchot et qui utilise une décroissance de la température τ telle que $\tau = \frac{1}{\ln t}$.

Ainsi, pour nos expérimentations nous avons implémenté l'algorithme *Softmax Annealing* afin qu'il sélectionne un bras a (c.-à-d., l'élément à recommander), à chaque itération t, avec une probabilité $P_s(t)$ définie comme suit :

$$P_{a,t} = \frac{e^{\widehat{\mu_a} \ln(t)}}{\sum_{i=1}^{k} e^{\widehat{\mu_a} \ln(t)}}$$

Selon l'analyse de [Aue+95], si un ensemble de bras contient k bras, alors la borne supérieure du regret de *Softmax Annealing* sera en :

$$\mathcal{O}\left(\sqrt{2T \ln k}\right)$$

Ainsi, un algorithme de recommandation utilisant *Softmax Annealing* sélectionne les éléments à recommander selon l'algorithme 9 (voir Annexe A.5).

2.2.5 Non-stationnarité dans les bandits-manchots

Dans certains cadres applicatifs, comme c'est le cas pour les systèmes de recommandation, la distribution des probabilités D_r peut évoluer au cours du temps. Dans le cadre des bandits-manchots, on peut modéliser la non-stationnarité existante soit en la définissant *par parties*, soit de manière *continue*.

Afin de simuler une stationnarité par partie, un adversaire défini à l'avance $m-1$ points de ruptures permettant de définir des périodes T_x de stationnarité où x est l'indice de la période. La simulation est de ce fait décomposée en $n \leq m$ périodes stationnaires égales. Par exemple, si on considère qu'une période de recommandation correspond à 5000 itérations, alors nous avons des périodes T_x de 5000 itérations correspondant à $[T_x, T_{x+1}[$. Une période T_x débute quand $\arg\max_{a \in A} \mu_{a,T_x} \neq \arg\max_{a \in A} \mu_{a,T_{x-1}}$. De ce fait, on considère le bras optimal a_x^* sur la période

T_x comme étant :

$$a_x^* = \arg\max_{a \in A} \mu_{a,T_x}$$

Il devient alors possible de considérer le problème en le modélisant via une chaîne de *Markov* dont la probabilité d'apparition d'un événement dépend uniquement de l'état courant [All16]. Cette modélisation peut être approfondie avec le modèle des *restless bandits* [GLS16]. Dans celui-ci, les probabilités de transition peuvent être différentes suivant si un bras a été joué ou non. La politique optimale des algorithmes de *restless bandits* maximise la récompense obtenue en tirant parti des probabilités de transition. Ainsi, selon [GLS16], dans un problème de bandit-manchot, $\forall a \in A$, nous avons :

- un ensemble d'états E_a ;
- un ensemble de probabilités $\{P_a = \varepsilon^a_{j \to n} | j, n \in E_a\}$ tel que $\varepsilon^a_{j \to n}$ est la probabilité d'être à l'état n si le bras a est sélectionné à l'état j ;
- un ensemble de gains $G_a = \{g^a_j | j \in E_a\}$ où g^a_j est le gain obtenu quand le bras a est sélectionné à l'état j.

2.2.5.1 Résoudre les problèmes de non-stationnarité

La littérature, présente un certain nombre d'exemples de problèmes non-stationnaires et des techniques permettant de les résoudre [All16].

Parmi ces techniques, nous pouvons relever celles mettant en œuvre une forme de pénalisation des solutions les plus utilisées. Cette pénalisation prend forme au fur et à mesure des itérations via l'utilisation :

- d'un facteur γ décroissant au cours du temps pondérant le calcul de la moyenne des récompenses [KS06]. Plus γ est proche de 1 plus l'algorithme fonctionne comme son modèle d'origine, a contrario plus γ tend vers 0 plus on force l'exploration de nouvelles solutions en donnant moins d'importance aux solutions déjà sélectionnées ;
- d'une fenêtre glissante permettant de prendre en compte un historique plus ou moins important [GM11 ; GLS16]. Une fenêtre glissante de taille $wsize$ a pour principal objectif de pénaliser les solutions qui ont été le plus sollicitées en se basant sur l'historique des récompenses obtenues pour chaque solution au cours des $wsize$ dernières itérations.

De façon générale, ces techniques permettent de forcer l'exploration d'autres possibilités jusque là considérées comme sous optimales. C'est le cas pour des algorithmes de bandits-manchots *(MAB)* comme *Discount-UCB* (D-UCB) [KS06] et *Sliding Window-UCB* (SW-UCB) [GM11], ou encore des algorithmes de recherche d'opérateurs en environnement non stationnaire [GLS16].

D-UCB par exemple permet d'estimer l'espérance des récompenses d'un bras à un instant donné en utilisant un facteur γ diminuant au cours du temps. Ce facteur biaise ainsi la moyenne des récompenses des bras trop sélectionnés en faveur d'observations plus récentes, ce qui revient à donner plus d'importance aux exemples récents plutôt qu'aux exemples anciens.

SW-UCB quant à lui utilise une fenêtre d'historique de taille H permettant d'obtenir, au même titre que *D-UCB*, une borne supérieure du regret cumulé en $\mathcal{O}(\sqrt{mT \ln T})$. La borne

inférieure du regret (sur flux non stationnaire) étant $\Omega(\sqrt{T})$, la borne supérieure pourrait paraître éloignée, néanmoins au niveau des algorithmes de bandits, il n'existe pourtant pas aujourd'hui de bornes plus proches [GM11].

Enfin, il existe des algorithmes de bandits-manchots avec adversaire comme *EXP3* [Aue+02]. Ce type d'algorithme est basé sur l'hypothèse qu'il existe un adversaire conscient ou inconscient c.-à-d., on suppose que les récompenses obtenues pour la séquence générée par le processus markovien sont définies par un adversaire. Quand l'adversaire est inconscient, les récompenses sont générées à l'avance et ne changent pas quelle que soit la nature des interactions avec le système [All16]. A contrario, lorsque l'adversaire est conscient, il définit les récompenses de la séquence en ayant conscience des interactions.

Parmi ces différents algorithmes permettant de résoudre le problème de bandit-manchot non stationnaire, nous décrirons deux des plus populaires : *SW-UCB* [GM11] et *EXP3* [Aue+02].

2.2.5.2 *SW-UCB* : Algorithme de bandit-manchot avec fenêtre glissante

SW-UCB [GM11] est une extension de *D-UCB* [KS06] qui utilise une fenêtre glissante plutôt qu'un facteur de pénalisation continue. *SW-UCB* utilise le même processus de maximisation qu'*UCB1* mais pour une fenêtre de période H, ce qui résulte en une stratégie qui consiste à sélectionner un élément à recommander (un bras) comme suit :

$$\underset{a\in A}{\arg\max}\left(\overline{\mu}_{a,t}(H)+\sqrt{\frac{\alpha\ln\left(t\wedge H\right)}{t_a(H)}}\right)$$

où dans ce cas

$$\overline{\mu}_{a,t}(H)=\frac{1}{t_a(H)}\sum_{s=t-H+1}^{t}r_{a,s}\,\mathbb{1}_{\{a_s=a\}}$$

$$t_a(H)=\sum_{s=t-H+1}^{t}\mathbb{1}_{\{a_s=a\}}$$

Selon les analyses de [KS06] pour*D-UCB* et [GM11] pour *SW-UCB*, si nous avons m points de ruptures dans la stationnarité, alors la borne supérieure du regret de ces algorithmes sera en :

$$\mathcal{O}(\sqrt{mT\ln(T)})$$

Un algorithme de recommandation utilisant *SW-UCB* sélectionne les bras (c.-à-d., les éléments à recommander) selon l'algorithme 10 (voir Annexe A.6).

2.2.5.3 *EXP3* : Algorithme de bandit-manchot avec adversaire

EXP3 : exponential-weight algorithm for exploration and exploitation [AC98 ; Aue+95 ; Aue+02] est un algorithme de bandits manchots avec adversaire que nous pouvons considérer comme une variante plus compliquée de l'algorithme *SoftMax*. L'avantage d'*EXP3* dans le cas de la non stationnarité de la distribution des récompenses c'est qu'il possède, à l'instar d'ε-*Greedy*,

un mécanisme continue d'exploration qu'il est possible de paramétrer via un facteur dit *d'égalitarisme* $\eta \in [0, 1]$. Dans le cas d'*EXP3*, la probabilité de choisir le bras a à l'itération t est définie par :

$$P_{a,t} = (1 - \eta)\frac{w_{a,t}}{\sum_{i=1}^{k} w_{a_i,t}} + \frac{\eta}{k}$$

où $w_{a,t+1} = w_{a,t} \exp\left(\eta \frac{r_{a,t}}{p_{a,t}k}\right)$ si le bras a a été sélectionné à l'itération t avec $r_{a,t}$ étant la récompense observée, sinon $w_{a,t+1} = w_{a,t}$.

Si un ensemble de bras contient k bras, il a été démontré par [Aue+02], qu'un regret en $c\sqrt{kT\ln(k)}$ est obtenu pour une constante c fixée.

Ainsi, un algorithme de recommandation utilisant *EXP3* sélectionne les éléments à recommander selon l'algorithme 11 (voir Annexe A.7).

2.3 Le problème du bandit-manchot contextuel

De nos jours, les bandits-manchots contextuels (*Contextual Multi-Armed Bandit - CMAB*) sont très largement considérés par de nombreuses applications se heurtant à des problèmes de décision séquentielle p. ex., les systèmes de recommandation [Li+10], les essais cliniques [VBW15], ou les applications mobiles en santé [Gre+17]. À chaque itération, les algorithmes d'apprentissage de *CMAB* ont pour objectif de choisir une action optimale (sélectionner le bras optimal) parmi un ensemble de possibilités, en tenant compte du contexte donné et des récompenses obtenues pour les actions passées.

Dans le cadre de cette thèse portant sur la recommandation contextuelle de service à des utilisateurs mobiles, nous nous sommes orientés vers l'étude et l'utilisation d'algorithmes de *CMAB*. En effet, ces algorithmes savent tirer parties des informations contextuelles pertinentes pour atteindre une parfaite personnalisation des recommandations qu'ils produisent aux utilisateurs évoluant dans un contexte donné [ZB16].

Il existe deux approches concernant le problème de bandits-manchots contextuels :

1. Une première approche est basée sur la création de modèles prédictifs. Les algorithmes qui utilisent ces modèles cherchent à capturer la dépendance entre les contextes, les bras et les récompenses. Parmi les algorithmes qui en découlent, il existe *Epoch-Greedy* [LZ08] qui permet de convertir tout algorithme de classification en ligne (online) en algorithme de bandit-manchot contextuel. Il sera important de considérer *LinUCB* [Li+10] et *Contextual Thompson Sampling* [AG13] qui sont deux algorithmes des plus populaires dans la littérature. Ces algorithmes supposent une dépendance linéaire entre la récompense attendue d'une action et son contexte. Ils adaptent ainsi des modèles linéaires aux problèmes des bandits contextuels ;
2. Une seconde approche du problème de bandits-manchots contextuels est proposée quant à elle sous l'angle de la sélection de politiques. Dans ce cadre, l'algorithme prend en compte un ensemble de politiques Π et cherche la politique optimale parmi cet ensemble. Les quatre principaux algorithmes permettant de résoudre le problème

de *CMAB* selon cette approche sont : *EXP4* et *EXP4.P* [Aue+95] qui souffrent d'une complexité algorithmique élevée [All16] (c.-à-d., linéaire en $k|\Pi|$, Π étant l'ensemble des politiques et k le nombre de bras) pour le cas où un ensemble d'experts est disponible ; *RandomizedUCB* [Dud+11] et ILOVETOCONBANDITS [Aga+14] qui sont à la fois longs et complexes.

Les algorithmes résolvant le problème de bandit-manchot contextuel selon la seconde approche possèdent plusieurs faiblesses [All16] entravant leur utilisation pratique de par :

1. Leurs complexités algorithmiques qui sont sur-linéaires, allant à l'encontre des contraintes de temps de réponse des algorithmes en ligne ;
2. La complexité en mémoire qui est linéaire en $|\Pi|$ (nombre de politiques) si chaque politique doit être instanciée ;
3. La nécessité de construire une stratégie discrétisant l'espace des politiques afin de pouvoir les dénombrer (si l'algorithme considère une classe de politiques).

Pour l'ensemble de ces raisons et celle que nous évoquerons en tableau de synthèse à la fin de ce chapitre, parmi les algorithmes de *CMAB* nous avons privilégié dans cette thèse ceux de la première approche, c.-à-d., *LinUCB* et *Contextual Thompson Sampling*. Néanmoins, nous évoquerons tout de même le fonctionnement des principaux algorithmes de la seconde approche : pour *RandomizedUCB* et ILOVETOCONBANDITS sans entrer dans les détails, et d'une manière plus précise les plus populaires d'entre eux à savoir *EXP4* et *EXP4.P*.

Enfin, l'une des perspectives concernant les travaux de cette thèse sera la recommandation d'événements culturels à des utilisateurs mobiles. Ainsi, nous conclurons ce chapitre par une analyse appuyée d'un tableau de synthèse afin d'apporter quelques lumières sur nos choix en termes d'approches et d'implémentations.

2.3.1 Définitions générales du problème de Bandit-Manchot Contextuel basé sur les modèles

Selon [LZ08], nous pouvons formellement définir le problème de bandits-manchots contextuels (voir Définition 4), les algorithmes pour le résoudre (Définition 5) et le regret (Définition 6).

Définition 4 ***. Problème du bandit-manchot Contextuel (CMAB)****. Dans un problème de CMAB, il existe une distribution conjointe $D_{x,r}$ entre les contextes x et les récompenses r, telle que $D_{x,r} = (x, r_{a_1}, \ldots, r_{a_k})$, où $x \in X \subseteq \mathbb{R}^d$ est un contexte, $A = \{a_1, ..., a_k\}$ un ensemble donné de k bras indépendants à tirer et $r_{a_i} \in \{0, 1\}$ la récompense d'un bras $a_i,\ i \in [1, k] \subseteq \mathbb{N}$. Le problème est séquentiel : à chaque itération, un échantillon $(x, r_{a_1}, \ldots, r_{a_k})$ est tiré de $D_{x,r}$, le contexte x est présenté, observé, puis un bras a est choisi par l'agent, et sa récompense r_a est alors révélée.*

Définition 5 ***. Algorithme de bandit-manchot contextuel (CMAB)****. Un algorithme $\mathcal{A}$ de CMAB sélectionne un bras $a_i \in A,\ A = \{a_1, ..., a_k\}$ à chaque itération t, en se basant sur la séquence*

des précédentes observations $(x_1, a_{i,1}, r_{a_i,1}), \ldots, (x_{t-1}, a_{i,t-1}, r_{a_i,t-1})$ *et le contexte courant observé* x_t *et on conserve* $\vec{r_t}$ *le vecteur de récompenses.*

L'objectif est de maximiser la récompense totale attendue $\sum_{t=1}^{T} \mathbb{E}_{x,\vec{r_t} \sim D[r_{a,t}]}$. Soit $\Pi : X \to A$ l'ensemble des politiques possibles où la politique optimale devant être déterminée est $\pi^* = \arg\max_{\pi \in \Pi} \mathbb{E}_{r,x}[r_{t,\pi(x)}]$. Ainsi, afin de mesurer la performance d'un algorithme $\mathcal{A}$) de *CMAB*, on peut mesurer le regret cumulé obtenu $\rho_T(\mathcal{A})$ par rapport à celui d'une politique optimale. On peut donc définir le regret (Définition 6).

Définition 6 ***. Regret****. L'espérance de récompense pour une politique* $\pi \in \Pi$ *est :*

$$R(\pi) = \mathbb{E}_{(x,\vec{r}) \sim D}[r_{\pi(x)}]$$

Considérons $\mathcal{A}$ *tout algorithme de bandit-manchot contextuel. Soit* $Z^T = \{(x_1, \vec{r_1}), ..., (x_T, \vec{r_T})\}$ *et le regret attendu de* $\mathcal{A}$ *par rapport à une politique* π *défini comme suit :*

$$\Delta\rho(\mathcal{A}, \pi, T) = T\ R(\pi) - \mathbb{E}_{Z^T \sim D^T} \sum_{t=1}^{T} r_{\mathcal{A}(x),t}$$

Le regret attendu de $\mathcal{A}$ *jusqu'à l'horizon* T *par rapport à l'espace des politiques* Π *est alors défini comme :*

$$\Delta\rho(\mathcal{A}, \Pi, T) = \sup_{\pi \in \Pi} \rho(\mathcal{A}, \pi, T)$$

2.3.2 Énoncé du problème de bandit-manchot contextuel pour la recommandation

Les approches contextuelles du problème de bandits-manchots *(CMAB)* [LZ08] ont été très largement étudiées via des méthodes telles que *LinUCB* [Li+10], Contextual Thompson Sampling *CTS* [AG13] ou encore Neural Bandit [AFB14]. À noter que ces dernières méthodes résolvent le problème de *CMAB* en supposant une dépendance linéaire entre la récompense attendue d'une action et son contexte.

Selon les travaux de Langford [LZ08], le problème de *CMAB* peut être défini comme suit : Soit $A = \{a_1, ..., a_k\}$ un ensemble donné de k bras indépendants. À l'instar du problème de bandits-manchots non contextuels, lorsqu'on pose le problème du bandit-manchot contextuel pour la recommandation, alors à chaque bras $a \in A$ est associé un élément à recommander à l'utilisateur, c'est à dire dans notre cas un service. Soit $X \subseteq \mathbb{R}^d$ l'ensemble de vecteurs de contexte de dimension d caractérisant à la fois un utilisateur et son environnement p. ex., $x \in X$ est un vecteur binaire codant les caractéristiques telles que : l'âge, le sexe, le métier, les préférences, la localisation ou encore les caractéristiques des bras eux-mêmes. Soit l'horizon $T \in \mathbb{N}^*$, à chaque itération $t \in [1, T]$, le contexte x_t incluant l'utilisateur, est pris en considération afin de permettre la sélection du bras optimal compte tenu des récompenses obtenues lors des itérations précédentes. Pour chaque itération t, soit $r_t = (r_{t,a_1}, ..., r_{t,a_k})$ le vecteur de

récompense où r_{t,a_i} correspond à la récompense obtenue après avoir sélectionné le bras a_i. Dans le cadre de cette thèse où les récompenses sont tirées depuis des distributions de Bernoulli $r_{t,a_i} \in \{0,1\}$. Soit $\mathcal{D}_{x,r}$ la distribution conjointe entre les contextes x et les récompenses r. Ainsi, soit $\Pi : X \rightarrow A$ l'ensemble des politiques possibles où la politique optimale devant être déterminée est $\pi^* = \arg\max_{\pi \in \Pi} \mathbb{E}_{r,x}[r_{t,\pi(x)}]$. Alors, soit $\pi_t \in \Pi$ la politique empruntée par un algorithme de *CMAB* $\mathcal{A}$ à l'itération t. Par conséquent, dans le cadre d'un environnement stationnaire où $\mathcal{D}_{x,r}$ ne varie pas, le pseudo-regret instantané à l'itération t peut alors être défini tel que $\rho_t(\mathcal{A}) = \mathbb{E}_{r,x}[r_{t,\pi^*(x_t)} - r_{t,\pi(x_t)}]$ et le pseudo-regret cumulé tel que $\rho(\mathcal{A}) = \sum_{t=1}^{T} \rho_t(\mathcal{A})$.

2.3.3 Algorithmes de bandits-manchots contextuels pour la recommandation

Un système de recommandation utilisant un algorithme de bandits-manchots contextuels fonctionne selon la description de l'algorithme 1 [All16].

Algorithme 1 : Algorithme de bandit-manchot contextuel

Données : L'ensemble des k éléments à recommander (c.-à-d., les bras) $a \in A$ disponibles, l'horizon T, et l'ensemble des n contextes fixes disponibles X.

1 **pour** $t = 1, ..., T$ **faire**
2 Considérer $x_t \in X$: un utilisateur et son contexte;
3 Sélectionner l'élément $a \in A$ estimé comme étant optimal selon la politique $\pi_t(x_t)$ et le recommander à l'utilisateur u;
4 Observer la récompense $r_{a,t}$ retournée par l'utilisateur;
5 L'agent modifie sa politique de choix de bras;

2.3.4 Généralisation du problème de classification en ligne : *Epoch-Greedy*

Epoch-Greedy [LZ08] est un algorithme considéré comme générique [All16] et similaire aux algorithmes de type ε-*Greedy* [SB98]. En effet, à l'instar d'ε-*Greedy*, *Epoch-Greedy* alterne entre phases d'exploration (uniforme) et phases d'exploitation. Comme défini par [LZ08] et reformulé dans la thèse de [All16], avec *Epoch-Greedy* les exemples d'apprentissage sont collectés durant l'exploration (uniforme) afin d'apprendre un modèle prédictif (hors ligne). Ce modèle est ensuite employé afin de sélectionner les bras les plus optimaux selon la meilleure espérance de récompense découlant du modèle courant, et ce jusqu'à la prochaine phase d'exploration. Dans la plupart des cas pratiques, l'horizon T n'est pas connu. De ce fait, il est compliqué d'utiliser des algorithmes se basant uniquement sur une seule grande phase d'exploration initiale et qui se basent ensuite sur un seul modèle consolidé et appris durant la phase d'exploitation avec un nombre d'exemples suffisants. Ainsi, dans *Epoch-Greedy*, la principale raison qui motive l'alternance successive entre phase d'exploration et phase d'exploitation est de pouvoir obtenir un algorithme pouvant être appliqué sans connaissance de l'horizon T. Comme tout algorithme de bandit-manchot, *Epoch-Greedy* repose sur des fondements théo-

riques prouvés des bornes du regret espéré. Sa borne supérieure du regret a donc été prouvée en $\tilde{\mathcal{O}}\left(T^{2/3}\right)$dans le meilleur des cas [Li+10], ou $\tilde{\mathcal{O}}\left(T^{2/3}\right)$ dans le pire des cas [All16 ; LZ08].

Ainsi, un système de recommandation utilisant *Epoch-Greedy* sélectionne les éléments à recommander selon l'algorithme 12 (Annexe Section A.8) défini par [LZ08] et re-décrit dans [All16].

2.3.5 Approches supposant un modèle linéaire

Dans ce cas spécifique, nous supposons que la récompense attendue d'un bras a à l'itération t est une fonction linéaire du vecteur de contexte x_t de dimension d tel que $\mathbb{E}[r_{t,a}|x_t] = \widehat{\theta_a}^\top x_t$ où $\widehat{\theta}_a$ de dimension d représente le vecteur de coefficients estimé associé au bras a.

Les algorithmes tels que *LinUCB* [Li+10] ou *Contextual Thompson Sampling* (*CTS*) [AG13] ont été modélisés et largement étudiés afin de résoudre ce problème de *CMAB* avec modèle linéaire. Aussi, à la sous-section suivante nous rappelons ces deux algorithmes, parmi les plus populaires dans la littérature.

2.3.5.1 *LinUCB*

LinUCB [Li+10] est un algorithme contextuel à bornes supérieures de confiance qui renforce rapidement la sélection des bras optimaux en ajoutant un *bonus* (l'écart de la récompense) au gain total calculé.

À chaque itération t, *LinUCB* sélectionne le bras $a \in A$ avec l'espérance optimiste calculée $p_{t,a}$ maximum parmi l'ensemble des bras disponibles. L'espérance $p_{t,a}$ est construite à partir d'une combinaison linéaire du coefficient $\theta_{t,a}$ et du vecteur de caractéristiques x_t auxquels vient s'ajouter l'écart de récompense qui représente la valeur d'action optimiste du gain obtenu. Le vecteur de coefficient $\widehat{\theta_a}$ est construit à partir de la matrice D_a de dimension $n \times d$ (n recommandations en correspondance de d caractéristiques), et $b_a \in \mathbb{R}^d$ représente le vecteur de réponse correspondant, dont les poids pour chaque dimension sont fonction des récompenses obtenues. Plus précisément,

$$\widehat{\theta_a} = (D_a^\top D_a + I_d)^{-1} b_a$$

où I_d représente la matrice identité de dimension $d \times d$.

Par conséquent, à chaque itération t, *LinUCB* sélectionne le bras a_t tel que $a_t = \arg\max_{a \in A} p_{t,a}$ où

$$p_{t,a} = \widehat{\theta_a}^\top x_t + \alpha\sqrt{x_t^\top (D_a^\top D_a + I_d)^{-1} x_t}$$

Ainsi, $\widehat{\theta_a}^\top x_t$ représente l'espérance de récompense et $\alpha\sqrt{x_t^\top (D_a^\top D_a + I_d)^{-1} x_t}$ l'écart de récompense où α est un paramètre pouvant être considéré comme un critère de robustesse face au bruit.

De plus, selon [Wal+09], il existe une probabilité d'au moins $1 - \delta$ que :

$$| \widehat{\theta}_a^\top x_t - \mathbb{E}[r_{t,a}|x_t] | \leq \alpha\sqrt{x_t^\top (D_a^\top D_a + I_d)^{-1} x_t}$$

avec $\alpha = 1 + \sqrt{ln(2/\delta)/2}$. Si on considère un ensemble de k bras, alors la borne supérieure du regret sera en $\tilde{\mathcal{O}}\left(d\sqrt{T\ln((1+T)/\delta)}\right)$.

Ainsi, un système de recommandation utilisant *LinUCB* sélectionne les éléments à recommander selon l'algorithme 13 [Li+10] (Annexe Section A.9).

2.3.5.2 *Contextual Thompson Sampling*

Contextual Thompson Sampling (CTS) [AG13] ou encore appelé *LinTS*, est un algorithme de type bayésien permettant de résoudre le problème de bandits-manchots contextuels (*CMAB*) et qui provient de la plus ancienne heuristique connue du problème de bandits-manchots : *Thompson Sampling* [Tho33]. Le principe de base de *CTS* est basé sur la fonction de probabilité gaussienne et sur les estimations gaussiennes a priori.

C'est à dire qu'avec *CTS*, comme la distribution a priori $Pr(\tilde{\theta}|r_{t,a})$ est donnée par $\mathcal{N}(\widehat{\theta}_{t,a}, v^2 B_{t,a}^{-1})$, alors la distribution a posteriori $Pr(\tilde{\theta}|r_{t,a}) \propto Pr(r_{t,a}|\tilde{\theta}_{t,a})Pr(\tilde{\theta}_{t,a})$ à l'itération $t+1$ est calculée en utilisant le théorème de Bayes comme suit :

$$\mathcal{N}(\widehat{\theta}_{t+1,a}, v^2 B_{t+1,a}^{-1})$$

Les paramètres de la distribution gaussienne multivariée sont pré-calculés à chaque itération t comme suit :

— le vecteur de coefficients :

$$\widehat{\theta}_{t,a} = B_{t,a}^{-1}\left(\sum_{t=1}^{t-1} r_{t,a}\right)$$

— la matrice de covariance :

$$B_{t,a} = I_d + \sum_{t=1}^{t-1} x_{t,a} x_{t,a}^{\top}$$

— le paramètre v :

$$v = \sigma\sqrt{\frac{24}{\varepsilon} d\, ln(\frac{1}{\delta})}$$

où $\varepsilon = \frac{1}{ln(T)}$, et $\delta \in (0,1)$ est défini tel que $1-\delta$ est la probabilité d'obtenir :

$$\mathbb{E}[\rho_T] = \tilde{\mathcal{O}}\left(d^{3/2}\sqrt{T}\right)$$

De plus, notons que pour une constante $\sigma \geq 0$ nous avons $r_{t,a} - x_{t,a}^{\top}\,\theta_{t,a}$ qui est conditionnellement σ-sous-gaussien.

Ainsi, à chaque itération t, *CTS* tirera un échantillon $\tilde{\theta}_{t,a}$ de la fonction de probabilité de la distribution gaussienne multivariée $\mathcal{N}(\widehat{\theta}_{t,a}, v^2 B_{t,a}^{-1})$, et sélectionnera le bras $a \in A$ ayant conduit à l'échantillon correspondant à l'espérance calculée la plus élevée, c'est à dire :

$$a_t^* = \arg\max x_{t,a}^{\top} \tilde{\theta}_{t,a}$$

Ainsi, un système de recommandation utilisant *CTS* sélectionne les éléments à recommander selon l'algorithme 14 [AG13] (Annexe Section A.10) .

2.3.6 Approches basées sur la sélection de politiques

L'algorithme *EXP4* [Aue+02] et son extension *EXP4.P* [Bey+11], ainsi que l'algorithme *RandomizedUCB* [Dud+11] et son contemporain *ILOVETOCONBANDITS* [Aga+14] sont des algorithmes reposant sur une approche basée sur la sélection de politiques.

Dans les sous-sections suivantes, nous rappellerons brièvement le fonctionnement *RandomizedUCB* et *ILOVETOCONBANDITS* en évoquant pourquoi leur utilisation dans notre application semble inadéquate. Nous nous pencherons ensuite plus en détail sur *EXP4* et *EXP4.P*.

2.3.6.1 *RandomizedUCB*

RandomizedUCB utilise un oracle résolvant un problème d'optimisation convexe [All16]. Celui-ci fournit une distribution de probabilité sur les politiques. La borne du regret de *RandomizedUCB* [Dud+11] peut être considéré comme optimale $\mathcal{O}(\sqrt{Tk\ln\frac{\Pi}{\delta}})$ mais sa complexité polynomiale (entre $\mathcal{O}(T^5)$ et $\mathcal{O}(T^6)$) rend cet algorithme inutilisable pour des applications nécessitant des réponses en temps-réel [All16].

2.3.6.2 *ILOVETOCONBANDITS*

ILOVETOCONBANDITS est une extension de *RandomizedUCB* et en reprend de ce fait les principes. Sa principale amélioration est qu'il réduit le nombre d'appels à l'Oracle en ne l'utilisant que durant des pas de temps spécifiques [All16]. Cependant, la complexité de *ILOVETOCONBANDITS* [Aga+14] reste de l'ordre de $\mathcal{O}((kT)^{3/2})$.

2.3.6.3 *EXP4* et *EXP4.P*

EXP4 est l'une des plus populaires et des plus anciennes méthodes parmi les approches de sélection de politiques. L'algorithme *EXP4* [Aue+02] est dénommé ainsi car est une extension contextuelle d'*EXP3* avec experts (c.-à-d., *EXP4 : EXP3 with expert advice*).

Ainsi, *EXP4* [Aue+02] à l'instar de *EXP3*, est un algorithme de bandit-manchot avec adversaire sauf que cette fois la probabilité de sélectionner un bras se définie via un ensemble de N vecteurs ξ issus du contexte (c.-à-d., l'algorithme *EXP4* décompose les informations de contexte en N experts) et la fonction de poids est identiquement remplacée pour prendre en compte ces vecteurs du contexte [BLL15]. De plus, *EXP4* peut s'appliquer à des données non stationnaires puisque les récompenses sont choisies par un adversaire.

Ainsi, *EXP4* possède un mécanisme qui sélectionne le meilleur élément à recommander (c.-à-d., bras) en se basant sur l'avis de ses N experts. Notons que les poids w sont désormais calculés par vecteur $\xi^i \in \mathbb{R}^k$. Ainsi, un vecteur ξ peut être considéré comme un *expert* indiquant un coefficient de sélection pour chacun des k bras $a \in A$ [BLL15]. De ce fait, la recherche du

lien entre les récompenses r des bras et le contexte x défini par $D_{r,x}$ est ainsi déléguée aux N *experts* ξ [All16].

Ainsi, dans le cas d'*EXP4*, la probabilité de choisir le bras a à l'itération t est définie par :

$$P_{a,t} = (1-\eta)\sum_{i=1}^{N} \frac{w_{i,t}\ \xi_{a,t}^{i}}{\sum_{j=1}^{N}\ w_{j,t}} + \frac{\eta}{k}$$

avec $\eta \in [0,1]$ et où à chaque itération

$$\forall a \in A,\ \widehat{r}_{a,t} = \begin{cases} r_{a,t}/p_{a,t} & \text{si} \quad a = a_t \\ 0 & \text{sinon} \end{cases}$$

$$\forall i \in \{1,..,N\},\ \widehat{y}_{i,t} = \xi^{i}\ .\ \widehat{r}_{t}$$

$$\forall i \in \{1,..,N\},\ w_{i,t+1} = w_{i,t} \exp\left(\eta\ \frac{\widehat{y}_{i,t}}{k}\right)$$

L'analyse de [Aue+02] a démontré que si un ensemble de bras contient k bras et qu'il y a N experts, alors la borne supérieure du regret d'*EXP4* sera en :

$$\mathcal{O}(\sqrt{Tk\ln(N)})$$

Ainsi, un système de recommandation utilisant *EXP4* sélectionne les éléments à recommander selon l'algorithme 15 [Aue+02] (voir Annexe A.11).

La borne du regret d'*EXP4* est nuancée par une variance trop importante. Ainsi, une extension d'*EXP4*, dénommée *EXP4.P* [Bey+11], obtient cette même borne mais avec une probabilité plus forte.

EXP4.P [Bey+11] par rapport à *EXP4* [Aue+02] apporte une probabilité plus élevée du regret espéré. Ainsi, même si *EXP4.P* obtient un regret moyen inférieur à *EXP4*, sa variance en revanche est plus faible, ce qui le rend plus robuste pour une utilisation dans un cadre réel. Principalement, *EXP4.P* exploite la technique standard utilisée pour prouver l'inégalité de *Bernstein* [Fre75] en matière de martingales. La différence est que la borne a été prouvée pour toute estimation fixe de la variance plutôt que toute borne sur la variance.

Dans le cas d'*EXP4.P*, la probabilité de choisir le bras a à l'itération t est définie par :

$$P_{a,t} = (1-k\ p_{min})\sum_{i=1}^{N} \frac{w_{i,t}\ \xi_{a,t}^{i}}{\sum_{j=1}^{k}\ w_{a_j,t}} + p_{min}$$

avec $p_{min} = \frac{\ln N}{KT}$ et où à chaque itération

$$\forall a \in A,\ \widehat{r}_{a,t} = \begin{cases} r_{a,t}/p_{a,t} & \text{si} \quad a = a_t \\ 0 & \text{sinon} \end{cases}$$

$$\forall i \in \{1, .., N\},\ \widehat{y}_{i,t} = \xi^i \ .\ \widehat{r}_t$$

$$\underline{\widehat{v}_{i,t} = \sum_j = 1^k\ \xi^i_{a_j,t}}$$

$$p_{a_j,t}$$

et

$$\forall i \in \{1, .., N\}, w_{a,t+1} = w_{a,t} \exp\left(\frac{p_{min}}{2}\ \left(\widehat{y}_{i,t} + \widehat{v}_{i,t}\ \sqrt{\frac{\ln(N/\eta)}{kT}}\right)\right)$$

2.3.7 Non-stationnarité dans les bandits-manchots contextuels

Comme nous l'avons décrit pour le problème du bandit-manchot non contextuel, il est important de rappeler que la non stationnarité dans les bandits-manchots contextuels peut être due à trois types de variations de l'environnement [All16 ; CMB18] :

1. La « *dérive conceptuelle* » (*Concept-drift*), c'est à dire dans notre cas que les probabilités de correspondance entre éléments à recommander (c.-à-d., bras) et utilisateurs (récompenses qu'ils retournent via leurs évaluations) évoluent sans modification des variables de contexte observées. La dérive conceptuelle a un impact de ce fait sur la distribution de $D_{r|x}$ où seule la distribution des récompenses change [All16 ; CMB18] ;
2. Le « *déplacement covarié* » (*Covariate-shift*), où cette fois ce sont une ou plusieurs variables de contexte observées qui changent. Le *déplacement covarié* a un impact sur la distribution D_x. « *Ce type de non-stationnarité peut par exemple être observé lorsque l'environnement représenté par certaines variables de contexte change (un capteur qui est déplacé).* » [All16 ; CMB18] ;
3. Un mixte des deux à la fois, *dérive conceptuelle* et *déplacement covarié*.

Dans la littérature, il existe un certain nombre d'exemples concernant les problèmes de non-stationnarité et des techniques permettant de les résoudre [All16 ; Gre+17 ; WIW18].

Prenant en compte les types de non-stationnarité existants dans le problème du bandit-manchot contextuel, il est possible d'y remédier selon deux principales approches :

— en utilisant une approche basée sur la sélection de politiques comme c'est le cas pour l'algorithme *EXP4* [Aue+02] et *EXP4.P* [Bey+11] que nous avons décrit à la sous-section 2.3.6 ;
— soit en proposant des évolutions ou extensions concernant les algorithmes basés sur les modèles [Gre+17 ; KST08] par exemple comme c'est le cas pour les plus récents d'entre eux : 1) *NeuralBandit* [AFB14] qui consiste à garder un facteur d'exploration γ constant au cours du temps, et qui permet de poursuivre la mise à jour des modèles en cas de non-stationnarité des données ; 2) *dLinUCB* [WIW18] qui est un algorithme de bandit hiérarchique, dans lequel un modèle de bandit-manchot maître opère sur un ensemble de modèles de bandits contextuels *esclaves* pour interagir avec l'environnement non stationnaire.

2.4 Critères d'évaluation de la performance des algorithmes de bandits-manchots

2.4.1 Cumul des récompenses et précision globale

Dans la littérature, la précision globale est le critère principal d'étude concernant les systèmes de recommandation à base de bandits-manchots [Bou+17 ; Li+11 ; LKG16]. Ce critère de performance représente la capacité d'un système à maximiser les correspondances entre les utilisateurs et les éléments qui leur sont recommandés. C'est une mesure globale principalement caractérisée par la somme des récompenses obtenues par un système de recommandation. Comme nous l'avons mentionné précédemment, une récompense correspond à un retour utilisateur face à une recommandation, qui se traduit, la plupart du temps, par une évaluation positive ou nulle. Une évaluation positive (généralement égale à 1) est un retour positif de l'utilisateur face à la recommandation et sera considérée comme étant un gain. De ce fait, la représentation de la performance qui en découle peut s'effectuer par le calcul d'une moyenne ou un cumul des gains obtenus par un système [Zen+16]. Le cumul des récompenses ou gains $g(T)$ d'un système de recommandation peut être exprimé tel que :

$$g(T) = \sum_{t=1}^{T} r_t$$

où t représente le nombre d'itérations, $t \in [1, T]$ et r_t la récompense obtenues à l'itération t. Dans le cadre de cette thèse nous rappelons que les récompenses sont binaires : $r_t = \{0, 1\}$, et nous travaillerons de ce fait avec un bandit de *Bernoulli*.

De ce fait, la précision globale moyenne de recommandation $Acc(T) \in [0, 1]$ peut être exprimée telle que :

$$Acc(T) = \frac{g(T)}{T}$$

Il est alors possible de représenter graphiquement ces deux critères comme suit :

— l'évolution de la moyenne des performances globales de recommandation $Acc(t)$ en fonction de t ;

— l'évolution du cumul des gains $g(t)$ en fonction de t.

Enfin, il est également possible de calculer cette performance en n'observant que les cent dernières recommandations afin d'obtenir une mesure plus instantanée de la performance. Cette visualisation est représentée par l'évolution de la moyenne des performances sur les 100 dernières recommandations en fonction de t.

Notre utilisation du critère de précision globale en résumé : la précision globale est un critère de performance basé sur le total des récompenses positives cumulées à l'horizon T. De ce fait, pour obtenir la précision globale, nous calculons le gain c'est à dire le nombre total de récompenses positives $g(T)$ puis nous calculons enfin la précision (Accuracy : Acc) tel que : $Acc(T) = \frac{g(T)}{T}$ où $g(T) = \sum_{t=1}^{T} r_t$.

2.4.2 Regret

Ci-dessus, nous avons défini qu'un retour positif de l'utilisateur face à la recommandation sera considéré comme étant un gain. Ainsi, a contrario, une évaluation nulle caractérise un échec de la recommandation et sera considérée comme étant un regret. La représentation de la performance qui en découle peut ainsi s'effectuer par le calcul du cumul des regrets observés [All16].

Le cumul des regrets ρ_T à l'horizon T d'un système de recommandation peut être exprimé tel que :

$$\rho_T = \sum_{t=1}^{T}(1 - r_t)$$

où t représente le nombre d'itérations, $t \in [1, T]$ et $r_=\{0, 1\}$ la récompense obtenue à l'itération t. Notons qu'il est aussi possible d'exprimer le cumul des regrets obtenus à l'horizon T par la somme totale des regrets (récompenses nulles).

Dans le cadre de la recommandation contextuelle en environnement stationnaire, le pseudo-regret instantané d'un algorithme $\mathcal{A}$ à l'itération t peut alors être défini tel que :

$$\rho_t(\mathcal{A}) = \mathbb{E}_{r,x}[r_{t,\pi^*(x_t)} - r_{t,\pi(x_t)}]$$

et le pseudo-regret cumulé tel que :

$$\rho(\mathcal{A}) = \sum_{t=1}^{T} \rho_t(\mathcal{A})$$

Il est alors possible de représenter graphiquement le regret en traçant l'évolution du cumul des regrets $\rho(t)$ en fonction de t.

Notre utilisation du critère de Regret en résumé : le regret est un critère de performance basé sur le total des récompenses négatives cumulées à l'horizon T. De ce fait, pour obtenir le cumul de regret, nous calculons le nombre de récompenses négatives obtenues c.-à-d., $r_t = 0$ tel que : $\rho_T = \sum_{t=1}^{T} \rho_t$.

2.5 Améliorer la performance des systèmes de recommandation à base de bandits-manchots

Le critère le plus fréquemment observé pour mesurer la performance d'un algorithme de bandits-manchots reste la précision globale c.-à-d., le nombre de fois qu'une récompense positive a été obtenue en tirant les différents bras [AG13 ; Aue02 ; BF16 ; LR85 ; Li+10]. Néanmoins, en fonction du domaine dans lequel les bandits sont appliqués, l'évaluation de leur performance peut nécessiter de s'ouvrir à d'autres critères. En effet, comme c'est tout particulièrement le cas pour les systèmes de recommandation, il a été observé dans certaines études que les mesures de précision ne sont pas suffisamment adaptées et pourraient être préjudiciables et nuire à la

satisfaction des utilisateurs [MRK06]. De ce fait, même si des algorithmes de *CMAB* tels que *LinUCB* [Li+10] ou encore *Contextual Thompson Sampling* [AG13] permettent à terme une personnalisation complète auprès de chaque utilisateur, une autre étude soutient en revanche que ceux-ci nécessitent un si grand nombre d'itérations pour atteindre cette personnalisation qu'ils risquent de causer la frustration des utilisateurs avant d'y parvenir [ZB16].

De telles constats ont conduit des recherches sur les *CMAB* et les systèmes de recommandations vers deux directions : 1) Tenter de réduire le nombre d'itérations nécessaires pour atteindre la personnalisation pour chaque utilisateur [Bou+17 ; ZB16], 2) Prendre en considération d'autres critères d'évaluation de la performance comme : la qualité [CHM15], la diversité [CBB14] et la nouveauté [Lac15], la couverture et la sérendipité [GDJ10], ou encore la satisfaction utilisateur [WGX15].

Ainsi, dans le cadre de cette thèse, outre la précision globale et les regrets cumulés qui restent le coeur des évaluations à formuler, nous avons jugé pertinent dans le cadre des systèmes de recommandation de réfléchir à la performance des algorithmes de bandits-manchots selon deux autres critères :

— la diversité des recommandations. Ce critère de performance prend tout son sens dans le cadre des systèmes de recommandation bi-objectifs [Lac15 ; Lac17 ; TT17 ; YDM14] dans lesquels il faut à la fois être précis dans les recommandations faites auprès des utilisateurs mais également satisfaire le fournisseur d'éléments à recommander (p. ex., publicités, coupon de réduction d'une enseigne, ou événements culturels divers) ;

— La précision obtenue pour chaque utilisateur (c.-à-d., précision individuelle). Ceci devient une nécessité vitale dans le cadre de systèmes de recommandation visant des applications en santé par exemple. Aucune étude n'a été réalisée sur ce point à notre connaissance, ni aucune proposition de critère d'évaluation.

Il est alors légitime de se poser la question de comment améliorer les approches au regard de ces différents critères de performances tout en nous penchant sur l'amélioration de la précision globale. En effet, ce critère reste principal et incontournable car il est basé sur des fondements théoriques fiables sur lesquels l'évaluation de toute nouvelle approche doit reposer.

2.5.1 Améliorer la précision globale et les regrets cumulés

Il est possible d'améliorer la précision globale selon trois approches :

1. Créer un nouvel algorithme surpassant les approches existantes en ce qui concerne la borne des regrets comme ce fût le cas lorsque *LinUCB* [Li+10] ou *Contextual Thompson Sampling* [AG13] furent déterminés ;
2. Étendre les approches existantes en améliorant leur borne et/ou la variance (p. ex., *EXP4.P* par rapport à *EXP4*, ε-*Greedy* avec ε paramétré constant ou décroissant, ou encore *UCB2* par rapport à *UCB1*) ;
3. En travaillant sur les données de contexte afin de les conserver les plus à jour et les plus pertinentes possibles, voire en ajoutant des caractéristiques pertinentes par rai-

sonnement, p. ex., en extrayant une partie restreinte de ce contexte afin de garder les dimensions les plus pertinentes [Bou+17], en combinant des techniques d'apprentissage non supervisé pour extraire du contexte supplémentaire pertinent [Cra+12], ou encore en collectant de nouvelles données à jour via le principe du MCSC [Guo+15].

Les principales contributions de cette thèse sur l'amélioration de la précision globale ont été abordées selon la troisième approche à savoir : opérer sur le contexte afin d'apporter des données plus pertinentes à considérer par le modèle.

Nous consacrerons donc une partie de cet état de l'art par la suite à cette troisième approche.

2.5.1.1 Opérer sur le contexte

Dans le domaine de l'apprentissage automatique, le contexte et le problème à résoudre sont étroitement liés, de sorte que le contexte contraint la résolution du problème sans y intervenir explicitement [Bré99]. Ainsi, dans le cas de bandits-manchots contextuels basés sur des modèles linéaires, si le modèle linéaire est « vérifié », et la distribution $D_{r,x}$ stationnaire, nous pouvons obtenir des garanties de précision très élevées [Gre+17]. Néanmoins, cela ne serait pas le cas si le contexte fourni est inutile ou restreint (parcimonieux - *sparse*). En effet, si à chaque instant le contexte fourni n'est pas pertinent ou si le modèle varie trop dans le temps comme c'est le cas pour la recommandation à des utilisateurs mobiles, alors la relation entre composantes du contexte et récompenses peut devenir non linéaire [Gre+17]. Par conséquent, ne pas fournir un contexte assez pertinent ou complet (variables manquantes et donc non observées), à différents instants de la prise de décision automatique, peut amener les participants à apprendre localement ou à régresser vers des stratégies non contextuelles [SKS16].

Dans le cadre de problèmes de bandits-manchots contextuels, le changement de distribution correspondant à des variables de contexte non observées peut être considéré comme une problématique de flux non stationnaire (dérive conceptuelle) [All16]. En effet dans ce cas, le changement de distribution de récompenses observé sur des variables de contexte pourrait avoir été implicitement provoqué par une autre variable inconnue et non observée par l'algorithme. Ces variables non observées peuvent être souvent à l'origine de ce qui est identifié comme étant des biais de confusions [BLL15] et nuisent fortement à la performance du système de recommandation, p. ex., ne pas observer la température et continuer d'un jour sur l'autre à recommander de la glace au chocolat à un enfant alors que les températures ont chuté de 10℃ du jour au lendemain. Dans cet exemple, pour recommander de manière pertinente c'est bien entendu la variable de catégorie d'âge ($\hat{\text{a}}ge = \{\text{bébé}, enfant, jeune, adulte, \text{aîné}\}$) qu'il faut prendre en compte mais également la variable température. Observer l'une sans l'autre peut générer de la non stationnarité car un poids trop important a été accordé à une seule de ces deux variables.

Malheureusement, dans la plupart des cas réels tels que les systèmes de recommandation, les contextes peuvent involontairement ou volontairement manquer d'informations et rester incomplets pour différentes raisons telles que p. ex., un manque d'information sur les caracté-

ristiques des éléments à recommander, une mauvaise modélisation du contexte (c'est-à-dire un contexte spécifié de manière incomplète), des restrictions dues à des problématiques de confidentialité et de protection de la vie privée, un profil mal renseigné, des informations manquantes sur l'environnement de l'utilisateur (par exemple, une localisation temporairement indisponible). Ce contexte doit être complété par (ou modélisé à partir de) nouvelles observations pertinentes [Lu+15] afin de mieux correspondre aux cas du métier et améliorer la précision.

2.5.1.2 Nos contributions à l'amélioration du contexte

Étant donné que le contexte est une information clé à fournir pour améliorer la précision de tels problèmes, de nombreux travaux sont axés sur des méthodes d'amélioration du contexte telles que p. ex., la modélisation du contexte [BHB18], la détection de contexte [Guo+15] ou la réduction de dimensions [Bou+17] afin d'améliorer respectivement la précision [Sli14] ou les problèmes de complexité en temps de calcul [Bou+17].

Dans le cadre de cette thèse, nous avons donc travaillé à plusieurs contributions (voir Figure 2 dans la partie *Introduction*) visant l'amélioration du contexte afin de gagner en précision globale et en regret cumulé total :

1. Méthode de modélisation et d'acquisition de contexte. Ces contributions sont identifiées sur la Figure 2 comme étant : *ISTE Ltd.2017* [Gut+17] et *Elsevier 2018* [Gut+18d] ainsi que *Inforsid 2018 (Data Intelligence)* [Gut+18e] ;
2. Méthode de raisonnement contextuel. Ces contributions sont identifiées sur la Figure 2 comme étant : *ISTE Ltd.2017, Elsevier 2018* [Gut+17 ; Gut+18d], *SAGEO 2018 (EXCES)* [Gut+18c] et l'une de nos dernières soumissions internationales 2019 (Étoile n° 8) ;
3. Méthode d'enrichissement et/ou création de contexte. Ces contributions sont identifiées sur la Figure 2 comme étant : *IEEE ICTAI 2018* [Gut+18b] et *IEEE RIVF 2019* [Gut+19d].

2.5.2 Améliorer la diversité dans les recommandations

Une seconde problématique tout aussi importante concerne la diversité des recommandations effectuées. La diversité représente la capacité d'un système à proposer une plus ou moins grande variété de services [VC11 ; Zho+10]. La majorité des travaux tendent à montrer que la diversification serait l'un des points-clé pour améliorer la satisfaction utilisateur. Par exemple cela permettrait de mieux répondre aux besoins éphémères des utilisateurs [Ash+15], de les aider à découvrir de nouveaux éléments [CHM15], ou d'éviter les recommandations redondantes [Hu+17].

Aujourd'hui, les travaux concernant la diversité se focalisent principalement sur la façon dont le système peut faire varier sa liste d'éléments à recommander au regard d'un ensemble global d'éléments disponibles [KP17].

De ce fait, il existe différentes mesures possibles de la diversité telles que la similarité intra-liste – Intra-List Similarity (ILS) – et la similarité relative [CBB14]. Celles-ci se basent sur un calcul de similarité (Sim) entre les différents éléments disponibles à la recommandation. En

effet, l'ILS d'une liste $\mathcal{L}$ d'items i à recommander définie telle que $\mathcal{L} = \{i_1, ..., i_m\}$, et $m = |\mathcal{L}|$, peut s'exprimer comme suit :

$$ILS(\mathcal{L}) = \frac{\sum_{j=1}^{m-1} \sum_{k=j+1}^{m} Sim(i_j, i_k)}{\frac{m(m-1)}{2}}$$

Il en découle une mesure de la diversité de $\mathcal{L}$ telle que :

$$Diversité(\mathcal{L}) = \frac{\sum_{j=1}^{m-1} \sum_{k=j+1}^{m} (1 - Sim(i_j, i_k))}{\frac{m(m-1)}{2}}$$

En partant du principe que chaque élément i d'une liste $\mathcal{L}$ corresponde à une catégorie distincte, il est alors possible de considérer la totalité des recommandations réalisées à l'itération $t = T$ comme étant $\mathcal{R}_T$ l'ensemble des éléments i proposés une ou plusieurs fois par le système aux utilisateurs. Nous pouvons ensuite calculer $ILS(\mathcal{R}_T)$ ou $Diversité(\mathcal{R}_T)$ afin de déterminer la diversité des recommandations effectuées. Ceci peut s'exprimer comme suit :

$$Diversité(\mathcal{R}_T) = \frac{\sum_{t_j=1}^{T-1} \sum_{t_k=t_j+1}^{T} (1 - Sim(i_{t_j}, i_{t_k}))}{\frac{T(T-1)}{2}}$$

Sachant que $Diversité(\mathcal{R}_T) \in [0; 1]$, plus $Diversité(\mathcal{R}_T) \to 0$ plus la diversité des recommandations est faible, tandis que plus $Diversité(\mathcal{R}_T) \to 1$ plus la diversité des recommandation est importante.

Dans les systèmes de recommandation à base de bandits-manchots, la diversité peut devenir pertinente pour la satisfaction de l'utilisateur [Ash+15 ; CHM15 ; Hu+17]. La diversification peut aussi trouver son intérêt dans le cadre d'environnements non-stationnaires afin de permettre à l'algorithme de rester à jour et de favoriser les observations les plus récentes. À l'aide d'une fenêtre glissante, des algorithmes tels que *SW-UCB* [GM11], ou encore *Windows Thompson Sampling with Restricted Context (Windows TSRC)* [Bou+17] permettent d'atténuer les effets résultant de la non-stationnarité.

2.5.2.1 Nos contributions à l'amélioration de la diversité

L'une des contributions de cette thèse a été de proposer une nouvelle mesure de la diversité basée entre autres sur le coefficient de variation, et d'améliorer, via ce nouveau critère, la diversité des recommandations effectuées par un algorithme de *CMAB*. Ces contributions sont identifiées dans la Figure 2 comme étant : *PFIA APIA 2018* [Gut+18a], *ACM SAC 2019* [Gut+19a] et l'une de nos dernières soumissions internationales 2019 (Étoile n° 7).

2.5.3 Améliorer la précision individuelle

À notre connaissance, aucune proposition de mesure sur l'individu n'a été concrètement formulée dans l'état de l'art outre celle de la satisfaction utilisateur (c.-à-d., *Hit Rate - HR*)

[WGX15]. Or, la satisfaction utilisateur intervient après expérience concrète de la recommandation (p. ex., on recommande un film, l'utilisateur clique sur cette recommandation qui l'intéresse donc, mais l'évalue ensuite après avoir visualisé le film). Dans notre cas, nous proposons de mesurer la précision individuelle des recommandations uniquement. De même, à notre connaissance aucune étude n'a été effectuée pour aller dans le sens de la mesure de précision individuelle dans les systèmes de recommandation à base de bandits-manchots.

Les mesures utilisées aujourd'hui pour évaluer les systèmes de recommandation sont considérées par certaines études comme pouvant aller jusqu'à nuire à ces systèmes [MRK06]. Aussi, même si la métrique de la précision individuelle n'est pas encore établie, elle semble malgré tout une variable importante à prendre en considération dans l'évaluation de la performance. Certaines approches prétendent que la diversification des recommandations peut en partie pallier le manque de personnalisation [Ash+15 ; CHM15 ; Hu+17].

2.5.3.1 Nos contributions à l'amélioration de la précision individuelle

À notre connaissance, aucune approche n'aborde spécifiquement le problème de la recherche d'un compromis entre précision individuelle et précision globale pour les *CMAB* à travers l'usage de techniques de diversification. Outre l'étude de la précision globale, et individuelle obtenues par les algorithmes de bandits-manchots, la recherche de ce compromis s'est constitué comme l'une des contributions de cette thèse. Ces contributions sont identifiées à la Figure 2 comme étant : *PFIA APIA 2018* [Gut+18a] et *ACM SAC 2019* [Gut+19a].

2.6 Synthèse

Dans cette section nous justifions nos choix d'algorithmes. Les algorithmes que nous sélectionnerons ici entreront ensuite dans nos simulations visant notre cas d'étude de recommandation de services à des utilisateurs mobiles dans la ville.

2.6.1 Critères de choix

Afin de justifier nos choix, nous nous basons sur trois grands critères :

1. Le niveau de précision et de personnalisation. Le niveau de précision correspond plus particulièrement à la borne des regrets cumulés espérés à l'horizon T, ou à la précision globale finale espérée. La personnalisation correspond à la capacité de l'algorithme à faire correspondre différents éléments à différents types de profil utilisateur. Cette personnalisation peut passer soit par la contextualisation de la fonction de recommandation, soit par des mécanismes de diversification des recommandations, soit les deux ;
2. L'applicabilité à notre cas d'étude et la capacité de traitement en temps réel. L'applicabilité et les besoins en temps réel sont étroitement liés et correspondent à la capacité de

l'algorithme à pouvoir s'ancrer à nos contraintes applicatives c.-à-d., mobilité, dynamicité, immédiateté. Ces éléments font principalement référence aux contraintes d'interactivités, aux temps de réponse requis et à aux temps total de calcul pour atteindre la précision globale ;

3. La résistance à la non-stationnarité. Ce dernier critère fait référence à la capacité de l'algorithme à résister aux changements brusques de préférences utilisateurs, ou à d'autres éléments pouvant avoir un impact sur la distribution $D_{x,r}$. Ce critère est étroitement lié au niveau d'exploration de l'algorithme et si celui-ci est intrinsèquement robuste, de par sa conception, pour résister à un environnement non-stationnaire.

Nous considérons qu'un algorithme possédant au moins une double évaluation négative (c.-à-d., $--$) sur l'un des trois critères devient de ce fait non pertinent ou non utilisable dans notre cas d'étude.

Enfin, nous essaierons de représenter, autant que faire se peut, chaque grande catégorie d'algorithmes par au moins un algorithme de cette catégorie (p. ex., *LinUCB* pour la catégorie des bandits-manchots contextuels basé sur un modèle linéaire ou encore *EXP4.P* pour la catégorie des bandits-manchots contextuels basés sur la sélection de politique).

Dans le tableau 2.1 nous évaluons chaque algorithme avec un score Sc de pertinence pour chacun des critères comme suit :
$++$: Très approprié à la recommandation de services à des utilisateurs mobiles
$+$: Approprié à la recommandation de services à des utilisateurs mobiles
$+-$: Moyennement approprié à la recommandation de services à des utilisateurs mobiles
$-$: Peu approprié à la recommandation de services à des utilisateurs mobiles
$--$: Très peu approprié à la recommandation de services à des utilisateurs mobiles.

2.6.2 Analyse

Dans le tableau 2.1, nous pouvons observer deux grandes familles d'algorithmes :
— les algorithmes de bandits-manchots (*Multi-Armed Bandit - MAB*) ;
— les algorithmes de bandits-manchots contextuels (*Contextual Multi-Armed Bandit - CMAB*).
Parmi ces deux grandes familles, il existe plusieurs catégories.

Pour les *MAB* :
— les algorithmes basés sur une stratégie *gloutonne* avec une phase d'exploration limitée. Dans le tableau 2.1 il s'agit d'*UCB1* ;
— les algorithmes basé sur une stratégie *gloutonne* avec un mécanisme d'exploration. Dans le tableau 2.1 il s'agit d'*UCB2*, de ε-*First*, de ε-*Greedy* decreasing (ε-*Greedy* dec.), ε-*Greedy* fixe, de *D-UCB* et *SW-UCB* ou encore de *Thompson Sampling (TS)* ;
— les algorithmes avec adversaires. Dans le tableau 2.1 il s'agit de *Softmax Annealing (Softmax A.)* ou encore d'*EXP3*.

Pour les *CMAB* :

— les algorithmes dont l'approche de recommandation est basée sur la création de modèles prédictifs. Dans le tableau 2.1 il s'agit d'*Epoch-Greedy*, *LinUCB*, *Contextual Thompson Sampling (CTS)* ;

— les algorithmes basé sur une stratégie de sélection de politiques. Dans le tableau 2.1 il s'agit d'*EXP4* et *EXP4.P*, de *RandomizedUCB* et *ILOVETOCONBANDIT*.

Au regard des différents scores attribués dans le tableau 2.1 pour chacun des critères, nous exprimons l'analyse suivante :

Parmi les algorithmes de *MAB* :

— ceux basés sur une stratégie *gloutonne* avec une phase d'exploration limitée (*UCB1* semblent inappropriés pour notre cas d'étude car bien qu'ayant une borne des regrets optimale, ils dépendent fortement des conditions initiales et sont peu robustes à la non-stationnarité. Nous les excluons donc de notre cas d'étude ;

— ceux basées sur une stratégie *gloutonne* avec un mécanisme d'exploration possèdent une borne optimale du regret mais personnalisent difficilement leur recommandation en fonction du profil utilisateur. En effet, n'étant pas sensibles au contexte, ils accordent mal les éléments à des changements de contexte et observent ainsi des regrets contextuels inévitables. En revanche, certains algorithmes via un mécanisme de diversification (exploration) continue, permettent en partie de rattraper cet inconvénient. En ce qui concerne l'applicabilité et la résistance à la non-stationnarité, certains de ces algorithmes ont l'avantage d'être peu complexes et de bénéficier d'un système d'exploration adéquate pour résister aux changements de distribution. Dans notre cas d'études nous retiendrons donc *UCB2*, ε*-Greedy* fixe, et *Thompson Sampling (TS)* ;

— ceux avec adversaires sont très intéressants car en plus des algorithmes ci-dessus, ils offrent une meilleure garantie de résistance à la non stationnarité. En revanche, ils souffrent toujours de regrets contextuels. Dans notre cas d'études nous retiendrons donc *EXP3*.

Parmi les algorithmes de *CMAB* :

— ceux dont l'approche de recommandation est basée sur la création de modèles prédictifs sont particulièrement intéressants puisqu'ils offrent une garantie tant en termes de précision globale que de personnalisation via l'usage de paramètres contextuels. Des points de vigilance résident néanmoins sur leur résistance à la non-stationnarité qui reste limitée. Dans notre cas d'études nous retiendrons donc *LinUCB* et *CTS* que nous favorisons par apport à *Epoch-Greedy* dont la borne est non optimale ;

— ceux basées sur une stratégie de sélection de politiques offrent une garantie de regret optimal. Pour certains, *EXP4* et *EXP4.P*, ils offrent même une meilleure garantie de résistance à la non stationnarité et une application possible pour notre cas d'études. En revanche, même si pour *RandomizedUCB* et *ILOVETOCONBANDIT* leur regret est optimal, ils n'offrent pas de garanties supérieures de résistances à la non-stationnarité par rapport aux autres algorithmes. De plus, leurs complexités algorithmiques sont surlinéaires et vont à l'encontre des contraintes de temps de réponse requis dans notre cas

de recommandation à des utilisateurs mobiles. Ceci les rend de ce fait inapplicables. C'est pourquoi, dans notre cas d'études nous retiendrons *EXP4.P* qui est une version plus stable de l'algorithme *EXP4* (c.-à-d., *EXP4* possède une variance trop élevée).

2.7 Bilan

Dans ce chapitre nous avons rappelé les différents algorithmes de bandits-manchots pour la recommandation.

Nous avons ensuite effectué une analyse afin de déterminer lesquels de ces algorithmes pourraient le mieux correspondre à notre sujet de recommandation de services à des utilisateurs mobiles dans la ville.

Selon notre analyse à la sous-section 2.6.2, nous en avons retenu un certain nombre que nous utiliserons dans nos prochaines simulations hors ligne.

Parmi ces algorithmes, nous avons principalement retenu ceux résolvant le problème de bandits-manchots contextuels : *LinUCB*, *CTS* et *EXP4.P*. Nous les confronterons en simulation à des algorithmes de bandits-manchots ne prenant pas en compte le contexte, et ce à des fins de comparaisons : *UCB2*, ε*-Greedy* avec ε fixe, *TS*, et *EXP3*. Cette comparaison permettra, entre autres, de vérifier qu'il est bien pertinent de prendre en compte le contexte dans notre système.

D'autre part, l'une de nos contributions porte sur l'utilisation de bandits-manchots non contextuels afin de créer du contexte. Ce contexte est ensuite utilisable par des algorithmes de bandits-manchots contextuels. C'est également à ce titre, que l'étude des algorithmes de bandits-manchots non contextuels prendra tout son sens.

Enfin, les algorithmes de bandits-manchots contextuels prenant en compte des données de contexte sous forme de représentation structurée en tant que paramètres d'entrée, il convient désormais de clarifier la notion de contexte à travers un chapitre qui lui est consacré. Ainsi, le chapitre 3 décrira un état de l'art du contexte en rappelant : les modélisations possibles du contexte, l'acquisition du contexte, et le principe du raisonnement contextuel.

TABLE 2.1 – Comparaison des principaux algorithmes de bandits-manchots selon trois méta-critères

	Précision & Personnalisation			**Applicabilité & Temps-réel**			**Résistance à la non-stationnarité**		
Algorithme	**Regret**	**Contexte**	Sc	**Complexité**	**Besoin en CPU**	Sc	**Adversaire**	**Exploration**	Sc
ε-*First*	$\mathcal{O}(1+(2k\ln 2kT/(\Delta_{min}^2)))$	non	$--$	$\mathcal{O}(k)$	faible	$++$	non	initiale	$-$
UCB1	$\mathcal{O}(k\ln T/(\Delta_{min}))$	non	$--$	$\mathcal{O}(k)$	faible	$++$	non	très limitée	$--$
UCB2	$\mathcal{O}(\sum \frac{(1+\alpha)(1+4\alpha)\ln(2e\Delta_a^2 t)}{2\Delta_a} + \frac{c_\alpha}{\Delta_a})$	non	$-$	$\mathcal{O}(k)$	faible	$++$	non	décroissante	$+-$
D-UCB	$\mathcal{O}(\sqrt{mT\ln(T)})$	non	$-$	$\mathcal{O}(k)$	faible	$++$	non	décroissante	$+-$
SW-UCB	$\mathcal{O}(\sqrt{mT\ln(T)})$	non	$-$	$\mathcal{O}(k)$	faible	$++$	non	continue	$+$
ε-*Greedy* fixe	$\mathcal{O}(\varepsilon/k\sum_{a\in A}\Delta_a)$	non	$-$	$\mathcal{O}(k)$	faible	$++$	non	continue	$+$
ε-*Greedy* dec.	$\mathcal{O}(k\ln T/(\Delta_{min}^2))$	non	$-$	$\mathcal{O}(k)$	faible	$++$	non	décroissante	$+-$
Softmax A.	$\mathcal{O}(\sqrt{Tk\ln k})$	non	$-$	$\mathcal{O}(k)$	faible	$++$	non	décroissante	$+-$
EXP3	$\mathcal{O}(\sqrt{Tk\ln k})$	non	$-$	$\mathcal{O}(k)$	faible	$++$	oui	continue	$++$
TS	$\mathcal{O}((\sum_{a:\Delta_a>0}\frac{1}{\Delta_a})^2\ln T)$	non	$-$	$\mathcal{O}(k)$	faible	$++$	non	décroissante	$+-$
EXP4	$\mathcal{O}(\sqrt{Tk\ln N})$	oui	$++$	$\mathcal{O}(N)$	moyen	$+-$	oui	continue	$++$
EXP4.P	$\mathcal{O}(\sqrt{Tk\ln N/\eta})$	oui	$++$	$\mathcal{O}(N)$	moyen	$+-$	oui	continue	$++$
Epoch-Greedy	$\mathcal{O}((k\ln N/\eta)^{1/3}T^{2/3})$	oui	$-$	$\mathcal{O}(\ln N)$	moyen	$+-$	non	continue	$++$
RandomizedUCB	$\mathcal{O}(\sqrt{Tk\ln N/\eta})$	oui	$++$	$\approx\mathcal{O}(T^6)$	fort	$--$	non	très limitée	$--$
ILOVETOCONBANDIT	$\mathcal{O}(\sqrt{Tk/\ln N/\eta})$	oui	$++$	$\approx\mathcal{O}(kT^{3/2})$	fort	$--$	non	limitée	$-$
LinUCB	$\mathcal{O}\left(d\sqrt{T\ln((1+T)/\delta)}\right)$	oui	$++$	$\mathcal{O}(d^3)$	moyen	$+$	non	limitée	$-$
CTS	$\mathcal{O}\left(\frac{d^2}{\varepsilon}\sqrt{T^{1+\varepsilon}\ln(Td)\ln\frac{1}{\varepsilon}}\right)$	oui	$++$	$\mathcal{O}(d^3)$	fort	$+-$	non	décroissante	$+-$

CHAPITRE 3

LE CONTEXTE

Context is key

« *Context is not simply the state of a predefined environment with a fixed set of interaction resources. It's part of a process of interacting with an ever-changing environment composed of reconfigurable, migratory, distributed, and multiscale resources.* [CC05]
Joëlle Coutaz & James Crowley - 2005 »

Sommaire

3.1 Introduction

Depuis le début des années 1960, la notion de contexte a été modélisée et exploitée dans de nombreux domaines de l'informatique. La communauté scientifique discute des définitions et des utilisations depuis de nombreuses années sans parvenir à un consensus clair [Dou04].

Ainsi, avant d'évoquer plus en détail les principes de modélisation, d'acquisition, et de raisonnement contextuel, il semble important de rappeler deux des principales définitions du contexte qui ont été données.

En informatique dite ubiquitaire, le contexte correspond à « *toute information pouvant être utilisée pour caractériser la situation d'une entité (personne, objet physique ou informatique). Et plus généralement tout élément pouvant influencer le comportement d'une application* » [Dey01]. En Intelligence Artificielle (IA), le contexte est considéré comme étant *ce qui n'intervient pas directement dans la résolution d'un problème mais contraint sa résolution* [Bré02]. Si on prend l'exemple des systèmes de recommandation, selon leur disponibilité et le cas d'application, les caractéristiques qui composeront le contexte pourront être : l'utilisateur lui-même (p. ex., profil, préférences), son équipement (p. ex., marque, modèle), le contexte spatio-temporel (p. ex., localisation, date) et les informations relatives aux éléments à recommander (p. ex., catégorie, description).

Dans ce chapitre, nous rappellerons quelles modélisations du contexte ont pu être proposées dans la littérature. Nous aborderons ensuite par quels moyens il est possible d'acquérir du contexte, afin de pouvoir alors raisonner sur ce contexte et en extraire une essence, pertinente et exploitable.

3.2 Modélisation du contexte

La modélisation du contexte correspond à une représentation permettant d'aider à la compréhension des propriétés, des relations et des détails du contexte. Cette modélisation varie en fonction du domaine et des caractéristiques du contexte.

Dans le cadre de cette thèse, nous nous sommes intéressés à la modélisation du contexte pour la recommandation de services à des utilisateurs mobiles évoluant dans la ville intelligente. Ainsi dans cette section, nous rappellerons en premier lieu différentes modélisations générales du contexte présentées dans la littérature. Ensuite nous nous focaliserons sur la modélisation spécifique pour les systèmes de recommandation à des utilisateurs mobiles [WSW07], p. ex., recommandation de visites de musées [Ben15], recommandation de sites culturels [Ben17].

3.2.1 Types de modélisation du contexte

Avant d'évoquer la modélisation concrète du contexte, il convient de rappeler très rapidement quels types de modélisation existent pour le représenter.

Le type de modélisation du contexte peut varier en fonction p. ex., de la richesse et de la qualité de l'information, de la mobilité, des dépendances et des relations, de l'hétérogénéité, du niveau d'ambiguïté, du type de raisonnement envisagé [Bet+10 ; SDO18 ; SL04]. Ainsi, la classification la plus courante de la modélisation de contexte est définie selon les types suivants [Bel+12 ; Bet+10 ; Li+15 ; Per+14 ; SDO18 ; SL04] :

— modélisation clé-valeur [SAW94 ; SL04] ;
— modélisation basée sur les langages de balisage [Ait11] ;
— modélisation graphique [HIR03 ; SL04] ;
— modélisation orientée objets [CMD99 ; SL04] ;
— modélisation basée sur la logique [MB97 ; SL04] ;
— modélisation basée sur les ontologies et les règles [Ait11] ;
— modélisation spatiale [Fra01] ;
— modélisation basée sur l'incertitude [GS01a ; RC03] ;
— modélisation hybride [ABR09 ; HLI04].

Dans notre cas, nous n'avons pas, à proprement parlé, utilisé une modélisation spécifique pour représenter le contexte. Néanmoins, une grande partie de nos contributions a visé, à terme, l'acquisition et le raisonnement contextuels de données géo-spatiales. Pour supporter cette acquisition et ce raisonnement, la modélisation qui se serait le plus rapprochée de nos travaux serait la modélisation spatiale [Fra01].

3.2.2 Modélisation générale du contexte

Il est possible de tirer une forme générique de la modélisation du contexte que nous représentons Figure 3.1 selon [ZLO07].

Cette représentation générique du contexte propose cinq grandes catégories d'informations de contexte centrées sur l'entité. Le terme entité fait référence ici à la définition de [Dey01] que nous avons rappelée en introduction du chapitre et qui de ce fait peut représenter une personne, un objet physique ou informatique.

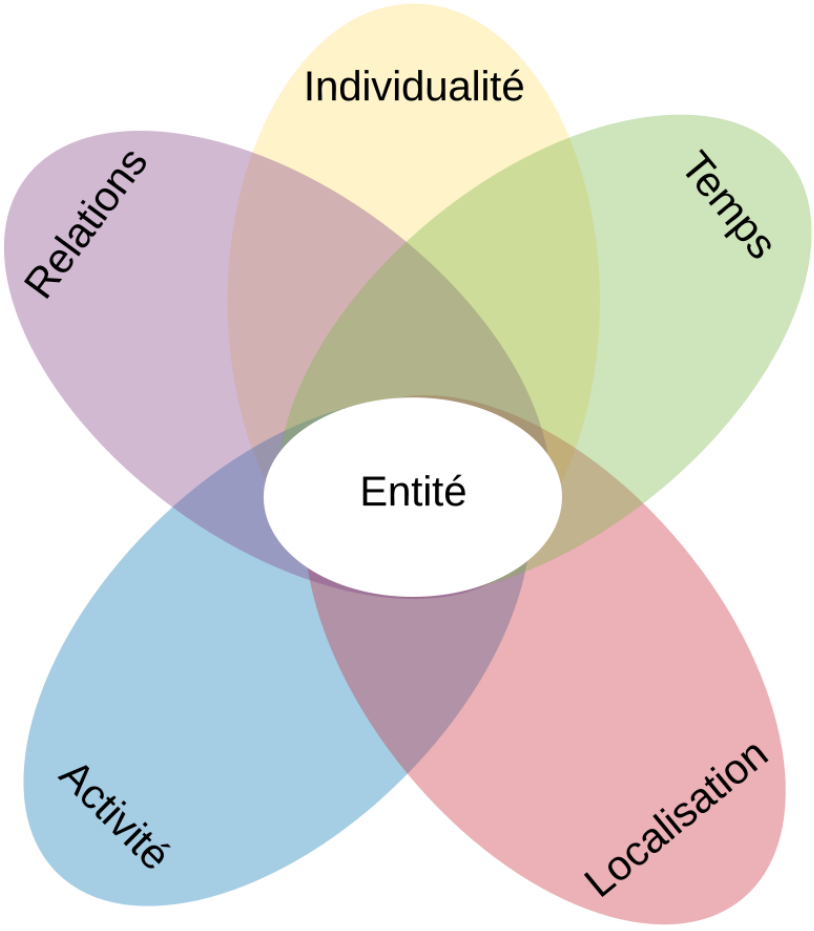


FIGURE 3.1 – Les cinq catégories fondamentales pour l'information de contexte [ZLO07].

3.2.2.1 Individualité du contexte

Cette dimension correspond aux informations contextuelles sur l'entité à laquelle le contexte est lié.

Selon [ZLO07], les informations contextuelles sur l'entité comprennent tout ce qui peut être observé sur une entité, à savoir généralement son état. Une entité (personne ou objet) peut

agir différemment dans un système sensible au contexte. Afin de représenter les informations de contexte d'individualité, il existe quatre types d'entités : naturelles, humaines, artificielles et de groupes. Nous les décrivons ci-dessous.

Les entités naturelles représentent tout ce qui se produit naturellement et ne résultent d'aucune activité ou intervention humaine. Plus généralement, on définit les entités naturelles comme étant le contexte produit par l'environnement naturel [ZLO07] (p. ex., arbres, roches, et autres éléments liées à la nature, sans aucun ajout artificiel de la part d'êtres humains), mais aussi le produit de l'interaction entre la nature et l'être humain.

Les entités humaines représentent tout ce qui caractérise l'être humain [ZLO07]. Dans les systèmes de recommandation contextuels par exemple, comme dans toute autre application sensible au contexte pour lesquelles les caractéristiques des individus peuvent influencer la prise de décision, il est incontournable de prendre en considération ces caractéristiques afin de réaliser des adaptations automatiques et répondre aux besoins de l'utilisateur. En effet, ce système adaptatif doit fonder ses décisions (p. ex., recommandations) sur l'évaluation du comportement de l'utilisateur et de ses caractéristiques. Par exemple ces caractéristiques peuvent être des préférences spécifiques à un domaines particulier p. ex., dans la recommandation d'événements culturels il peut s'agir des préférences sur les catégories d'événements (musique, théâtre, conférences, ...), ainsi que des données de profils (âge, sexe, catégorie socio-professionnelle), ou encore des préférences sur la langue utilisée (français, anglais, espagnol, russe, chinois, etc.).

Les entités artificielles représentent tout phénomène ou produit résultant des actions humaines [ZLO07]. Plus généralement, on définit les entités artificielles comme étant le contexte d'objets construits par l'être humain p. ex., les matériels informatiques et de télécommunications (p. ex., téléphones mobiles, ordinateurs fixes ou portables, périphériques informatiques) ou tout objet connecté (Smartphones inclus) possédant des capteurs physiques ou chimiques (p. ex., température, humidité, pression, son, champ magnétique, accélération, altitude), les bâtiments (p. ex., maisons, appartements, usines, bureaux), les véhicules (p. ex., vélos, voitures, tramways, trains, bus). Les informations de contexte de ces entités comprennent les descriptions de ces objets. Par exemple pour un téléphone mobile la description correspondrait à des propriétés telles que p. ex., le système d'exploitation (p. ex., Android, iOS, WPhone), la taille de l'écran, le niveau de connexion (p. ex., sans données, 3G, 4G, Wi-Fi) et la fiabilité de cette connexion réseau accessible.

Les entités de groupes représentent un ensemble d'entités qui partagent des aspects communs du contexte, interagissent entre elles ou ont établi certaines relations entre elles [ZLO07]. Il peut devenir utile d'aborder les entités sous forme de groupes pour structurer et capter des caractéristiques qui n'apparaîtraient pas si nous prenions chaque entité individuellement. Si on

prend l'exemple des êtres humains, les entités d'un même groupe peuvent avoir en commun p. ex., des intérêts, des compétences, des liens sociaux ou familiaux. Si on prend l'exemple cette fois du matériel informatique les entités d'un même groupe peuvent avoir en commun (p. ex., la puissance de calcul, le niveau de connexion réseau, la taille d'affichage à l'écran). Dans le cadre des entités de groupes, chaque membre du groupe partagera ainsi une identité commune à ce groupe. La relation de cardinalité entre les membres et les groupes est de l'ordre de (0..n) c.-à-d., un membre peut soit n'appartenir à aucun groupe, soit appartenir à un ou plusieurs groupes. La détermination de l'appartenance d'un membre à un groupe peut être soit effectuée a priori (connue à l'avance via des relations existantes), soit de manière dynamique à travers des observations.

3.2.2.2 Les relations

Cette catégorie d'informations de contexte a pour objectif de capturer les relations qu'une entité a établi avec d'autres entités [ZLO07], permettant ainsi de faire des regroupements. Ces relations peuvent être établies entre tout type d'entité individuelle (personne et objet physique ou virtuel).

Les relations peuvent être subdivisées en trois catégories :

— relations sociales ;
— relations fonctionnelles ;
— relations de composition.

Les relations sociales représentent les aspects sociaux du contexte de l'entité [ZLO07]. De ce fait, les relations entre deux personnes ou plus peuvent être considérées comme des associations ou affiliations sociales p. ex., amis, ennemis, personnes neutres, voisins, collègues ou encore des proches. Il est important de considérer le rôle que joue la personne au sein de ces relations sociales (p. ex., niveau d'intimité, de partage, de leadership). Ainsi, les informations issues des caractéristiques partagées ou non avec d'autres personnes (calculs de similarité ou de distance) peuvent également enrichir les caractéristiques décrivant une personne en individuel. De ce fait, il est aussi possible de déduire des modèles de comportement, ou alors des groupes de personnes partageant les mêmes intérêts, objectifs ou niveaux de connaissance.

Les relations fonctionnelles indiquent qu'une entité utilise une autre entité dans un but précis [ZLO07]. Par exemple, si un utilisateur utilise un ordinateur portable, de bureau ou son téléphone mobile, est assis sur une chaise ou se déplace. Les relations fonctionnelles intègrent également des propriétés de communication ou d'interactions (p. ex., clavier, caméra, microphone).

Les relations de composition expriment la relation existante entre une entité et les parties (autres entités) qui la composent [ZLO07]. Par exemple, une entité utilisateur est composée de bras, de jambes, d'une tête etc. Une entité téléphone mobile est composée de capteurs, d'un

écran, d'un processeur, etc. Les relations de cardinalité de chacune des parties vis à vis de l'entité sont de l'ordre de (1..1) car une partie ne peut appartenir qu'à une seule entité qu'elle compose.

Une autre forme de relation identifiable est l'**association**. Celle-ci est plus faible que la composition car elle n'implique pas l'appartenance d'une partie à une seule entité. Les relations de cardinalité de chacune des parties vis à vis de l'entité sont de l'ordre de (1..n) car une partie peut appartenir à une ou plusieurs entités à laquelle elle est associée. Par exemple, une photocopieuse en entreprise peut appartenir à différentes personnes d'un département mais aussi à différents départements.

3.2.2.3 Le temps

La catégorie temporelle est incontournable puisque l'être humain organise toute sa vie autour de cette dimension [GS01b]. Elle doit donc impérativement être prise en considération dans un système sensible au contexte. Il existe plusieurs manières de prendre en compte le temps dont la donnée brute est inhérente à la date c.-à-d., l'heure (courante et fuseau horaire), le jour, le mois et l'année. En fonction du domaine auquel on souhaite utiliser la catégorie du temps, il existe différentes modélisations. Certains domaines, par exemple la restauration ou le commerce, utilisent des intervalles de temps via des échelles dites catégorielles telles que p. ex., les heures de travail, les week-ends, les saisons [ZLO07]. Ainsi, dans la modélisation du contexte, il devient absolument nécessaire de représenter des intervalles de temps clairement identifiables et utilisables par toute application sensible au contexte. De même, il devient intéressant d'identifier des éléments temporels récurrents (p. ex., le dimanche, Noël, Pâques) et de les combiner à ces intervalles de temps afin d'enrichir la modélisation des caractéristiques de l'utilisateur. De plus, historiser ces situations ou contextes crée une véritable archive d'informations contextuelles constituant la base pour accéder au contexte passé [ZLO07]. Ceci permet l'analyse a posteriori de cet historique, et permet de déduire les habitudes d'utilisation des utilisateurs afin de prédire leur comportement futur. L'un des avantages de ce principe d'analyse et de déduction de *futurs* contextes et qu'il bénéficie de l'historique nécessaire à l'extrapolation des informations permettant de pallier les problèmes d'accès aux données de contextes incomplètes ou imprécises.

3.2.2.4 L'activité

La catégorie de l'activité englobe toutes les activités dans lesquelles l'entité (personne ou objet) est engagée dans le présent mais également le sera dans le futur. Cette catégorie répond à la question « Que veut l'entité et comment ? » [ZLO07]. Cette catégorie peut être décrite sous forme d'objectifs, de tâches, et d'actions p. ex., une entité peut être engagée dans une tâche qui de ce fait détermine les objectifs des activités qui y sont effectuées [Bru96]. Une tâche peut être décrite comme étant une unité à exécuter et qui a un objectif spécifique [Kle02]. Une tâche est composée d'un ensemble de sous-tâches ou opérations ayant un objectif déterminé. Il se dessine ainsi une organisation composée de macro-objectifs (objectifs de haut niveau)

inhérents aux tâches, et des objectifs opérationnels (objectifs de bas niveau) inhérents aux opérations à effectuer dans le cadre d'une tâche. Par exemple, dans la tâche aller au cinéma il existe une séquence d'opérations à effectuer comme p. ex., choisir le film, choisir le cinéma où on souhaite visualiser le film, choisir son mode de transport, réserver la séance (en avance via internet ou sur place). Il s'avère que si l'objectif de haut niveau qui est de se rendre au cinéma pour visualiser un film est plus stable et de ce fait plus cohérent, ceux des objectifs de bas niveau peuvent changer assez souvent p. ex., en ce qui concerne le fait de choisir un film, on peut décider de modifier sa sélection au dernier moment (ou pas) en fonction de conditions (plus de places libres, changement de préférences d'un individu, modification de décision de groupe, etc.). C'est pourquoi il est important de faire la différence entre les objectifs de haut niveau et ceux de bas niveau [ZLO07]. En conséquence, le contexte d'activité peut être représenté par des modèles de tâches (spécifiques à un domaine) structurant les tâches en hiérarchies de sous-tâches (opérations), qui constituent la représentation la plus avancée des objectifs utilisateurs possibles [Vas96]. L'objectif peut donc être déterminé soit par l'entité elle-même, soit selon un choix automatique parmi l'ensemble des objectifs existants. Cet objectif obtenu via un choix automatique peut être révélé via une politique sélectionnant par exemple l'objectif possédant la probabilité la plus élevée [ZLO07].

3.2.2.5 La localisation

Avec l'émergence ces deux dernières décennies de la téléphonie mobile et de son utilisation à grande échelle, l'utilisation de la géo-localisation dans les systèmes sensibles au contexte est devenue non seulement possible mais également incontournable [Guo+15]. Si on prend l'exemple d'un utilisateur mobile dont l'emplacement brut est capturé, celui-ci peut être décrit soit comme un emplacement absolu, signifiant de ce fait l'emplacement exact de l'utilisateur [Zhe11], soit comme un emplacement relatif, signifiant l'emplacement de l'utilisateur par rapport à autre chose [Cra+12] (p. ex., un quartier, un commerce, toute zone délimitée géographiquement et déduit d'un raisonnement contextuel). Ainsi, les modèles d'emplacements physiques peuvent être divisés en deux types : les modèles d'emplacements quantitatifs c.-à-d., géométriques (p. ex., coordonnées GPS) ; et les modèles d'emplacements qualitatifs i.e, symboliques (p. ex., quartiers, bâtiments, rues, régions, pays) [CD04 ; ZLO07]. Ces derniers, à plusieurs niveaux, introduisent une notion de granularité spatiale. Les modèles de superposition permettent une interprétation des informations spatiales quantitatives en les transformant en informations qualitatives appropriées [Cra+12 ; Guo+15]. Cette transformation est importante entre autre dans le cadre des systèmes de recommandation qui nécessitent l'apport de contexte pertinent et facilement utilisable pour alimenter leur calcul c.-à-d., fournir une représentation structurée de l'information contextuelle. En général, une entité possède toujours un emplacement physique qualitatif, qui peut être représenté par différents emplacements quantitatifs [ZLO07].

3.2.3 Modélisation du contexte pour les systèmes de recommandation à des utilisateurs mobiles

La plupart des modélisations du contexte proposées pour les systèmes récents [Ben15 ; Ben17 ; Fab+18 ; Guo+15] se rapprochent voire se basent sur la définition du contexte de [Dey01] et la modélisation de [ZLO07] c.-à-d., exploitent tout ou partie des cinq catégories du contexte. Depuis la définition de la catégorie de localisation [ZLO07], l'importance de celle-ci a augmenté en raison du volume croissant d'informations disponibles en temps-réel c.-à-d., informations de localisation issues des téléphones mobiles (p. ex., GPS [Cra+12], localisation via les connexions 3G/4G [RB11]) et autres objets connectés et communiquants (p. ex., bornes d'accès Wi-Fi urbain [ME12], capteurs de mouvements de piétons [Adi15]). De plus, les usages ont évolué et l'utilisation d'appareils mobiles est devenu commun tant pour la recherche d'informations spécifiques [BTB09] que l'usage d'applications diverses rentrant dans le cadre de la Ville Intelligente [Fab+18] et des usages identifiés dans le paradigme du *Mobile Crowd Sensing and Computing* [Guo+15] (p. ex., l'application mobile *Vivre à Angers*[1] ou encore *Angers Map*[2]).

3.2.4 Notre contribution à la modélisation du contexte

Dans notre cas, nous avons travaillé sur les journaux de connexions aux points d'accès réseau Wi-Fi urbain dans la ville d'Angers. Dans le cadre d'un projet RFI - Atlanstic 2020 *(Event-AI)*, nous avons également développé une application mobile permettant la visualisation et la recommandation d'événements culturels (*scéno*).

Afin de supporter les phases de capture de contexte et de raisonnement contextuel, nous avons proposé une représentation du contexte se rapprochant de [Ben17 ; BTB09 ; ZLO07].

Ces contribution sont identifiées à la Figure 2 comme étant : *ISTE Ltd.2017 et Elsevier 2018* [Gut+17] ainsi que *Inforsid 2018 (Data Intelligence)* [Gut+18d].

3.3 Acquisition du contexte

Alors que l'*Internet des Objets* (c.-à-d., *Internet Of Things - IoT*) a pris place dans le paysage technologique quotidien, il est suivi par le déploiement dans le monde d'un nombre de capteurs plus important [Per+14]. De ces quelques 2,5 milliards d'octets le volume de données générées chaque jour [GCC17] en résulte une croissance rapide du stockage et du traitement de données.

1. Application disponible sur Google Play (`https://play.google.com/store/apps/details?id=fr.angers.app`) regroupant l'ensemble de 15 sources de données Open Data de la ville d'Angers (`https://data.angers.fr/pages/home/`) au service du citoyen.

2. Application disponible sur Google Play (`https://play.google.com/store/apps/details?id=fr.patrickrgn.angersmap`) affichant le déplacement du réseau de transport Angevin en temps-réel, ainsi que d'autres informations comme la disponibilité des parkings, les travaux en cours, etc.

Pour les systèmes sensibles au contexte, comme peuvent l'être les systèmes de recommandation, il devient alors crucial d'ajouter de la valeur à ces données collectées de plusieurs sources ou capteurs. Selon [Per+14 ; SDO18], cette valorisation s'opère via la modélisation, l'acquisition, le raisonnement, et la distribution du contexte.

Dans cette section, nous traiterons de l'acquisition du contexte en rappelant les cinq facteurs à prendre en compte dans le cadre du développement de systèmes sensibles au contexte [Per+14] :

1. Responsabilité ;
2. Fréquence ;
3. Source du contexte ;
4. Type de capteur ;
5. Processus d'acquisition.

3.3.1 Les systèmes basés sur la responsabilité

L'acquisition de contexte (par exemple des données du capteur GPS de Smartphones) est principalement réalisée selon deux méthodes [Pie+08] bien connues : la transmission de l'information c.-à-d., méthode de « push », ou l'extraction de l'information c.-à-d., méthode de « pull ».

Dans la méthode de « push », le composant applicatif qui est chargé d'acquérir les informations du ou des capteurs, effectue périodiquement (par intervalles de temps) ou instantanément, une requête auprès du matériel possédant le capteur afin d'acquérir ses données [Per+14].

Dans la méthode de « pull », c'est le capteur (physique ou virtuel) qui renvoie les données au composant applicatif chargé d'acquérir les informations du capteur et ce également de manière périodique ou instantanée [Per+14].

3.3.2 Les systèmes basés sur la fréquence

L'acquisition du contexte peut se réaliser selon deux types d'événements différents : les événements dits instantanés et les événements selon un intervalle (périodique).

Les événements qui surviennent instantanément ne s'étendent pas sur certaines périodes [Per+14] p. ex., lancer son application mobile pour y consulter de l'information, passer devant un lampadaire connecté dans la ville intelligente, rentrer dans un parking connecté. Afin de détecter ce type d'événement, les données du capteur doivent être acquises lorsque l'événement se produit p. ex., détecter les entrées/sorties de voitures (parkings, stationnements), la connexion d'un utilisateur à son application.

Les événements qui surviennent sur une période de temps p. ex., les événements météorologiques comme les orages, la pluie, la neige ; les saisons comme l'été ou l'hiver, sont considérés comme des événements dits « intermittents » [Per+14]. Afin de détecter ce type

d'événement, les données du capteur doivent être acquises périodiquement (p. ex., détecter et envoyer des données à l'application toutes les 30 secondes).

Dans les deux cas, les méthodes de « push » et de « pull » décrites ci-dessus peuvent être employées.

3.3.3 Les systèmes basés sur la source du contexte

Les méthodes d'acquisition de contexte peuvent être classées en trois catégories [Che+04] en fonction de l'origine du contexte :

- **acquérir directement à partir du matériel du capteur** [Per+14]. Dans ce cas le contexte est acquis directement du capteur en communiquant avec le matériel du capteur et les API associées. Les pilotes logiciels et les bibliothèques doivent être installés localement ;
- **acquérir via une infrastructure middleware** [Per+14]. Dans ce cas les données de capteur (contexte) sont acquises par des solutions middleware telles que par exemple GSN (Solution middleware de traitement de flux de données de capteurs) [Sal10]. Les applications extraient les données du capteur à partir du middleware et non pas directement à partir du matériel du capteur ;
- **acquérir à partir de serveurs de contexte** [Per+14]. Dans le cas où le contexte est acquis à partir de plusieurs sources de stockage de contexte (bases de données, flux RSS, ou services web) via différents mécanismes comme les appels aux services web. Ce mécanisme est utile lorsque le dispositif d'hébergement de l'application sensible au contexte dispose de ressources informatiques limitées (p. ex., application mobile).

3.3.4 Les systèmes basés sur le type de capteur

Il existe différents types de capteurs pouvant être utilisés pour acquérir du contexte [Per+14]. Même si en règle générale, le terme de «capteur» désigne des dispositifs matériels, dans la littérature les capteurs peuvent aussi caractériser les différentes sources de données pouvant fournir des informations pertinentes de contexte [Per+14]. Ainsi, les capteurs sont divisés en trois catégories [IS03] : physique, virtuel et logique (ou logiciel).

Les capteurs physiques [SV01] sont soit intégrés physiquement dans un matériel spécifique (p. ex., smartphone, réfrigérateur connecté, machine à laver connectée, voiture), soit restent des objets connectés et communiquants autonomes (p. ex., thermomètre numérique, hygromètre numérique, capteur de poussières, capteur de présence). Tout ces capteurs génèrent des données par eux mêmes. Les capteurs physiques sont les plus couramment utilisés et c'est d'autant plus le cas à notre époque qui est pleinement entrée et ancrée dans le paradigme de l'Internet des Objets. En effet, la plupart des matériels numériques (*IoT*) que nous utilisons aujourd'hui sont équipés de capteurs (température, pression, accéléromètre, magnétomètre, microphone, GPS, etc.).

Les capteurs virtuels ne génèrent pas nécessairement des données de capteur par eux-mêmes [Per+14]. Les données issues de capteurs virtuels proviennent de sources multiples

et variées (p. ex., réseaux sociaux, calendrier, applications de type IRC [3]). Ces capteurs n'ont aucune existence physique et utilisent les technologies de services web (web services) pour envoyer et recevoir des données [Per+14].

Les capteurs logiciels combinent l'utilisation de capteurs physiques et de capteurs virtuels afin de produire des informations plus pertinentes [Per+14]. Un service web dédié à la fourniture d'informations météorologiques peut être considéré comme une capteur logique (p. ex., *API OpenWeatherMap* [4] que nous avons utilisé entre autres dans notre projet *Event-AI*). Les stations météorologiques représentent un très bon exemple puisque les informations météorologiques sont produites en combinant des capteurs physiques et virtuels. En effet, elles utilisent des milliers de capteurs physiques pour collecter des informations météorologiques mais collectent également des informations à partir de capteurs virtuels tels que des cartes géographiques, des calendriers et des données d'historique. Un autre exemple que de nombreuses applications mobiles emploient et celui des systèmes d'exploitation mobile *Android* et *iOS*. Ces systèmes à travers les capteurs physiques des téléphones mobiles mais également à travers des capteurs virtuels via des appels à des API externes, disposent de ce fait d'un certain nombre de capteurs logiciels tels que p. ex., la pluviométrie ou l'altitude.

3.3.5 Les systèmes basés sur le processus d'acquisition

Il y a trois types d'acquisition des données de contexte : données détectées, données dérivées, et données fournies manuellement [Per+14].

En ce qui concerne les données détectées, celle-ci peuvent être détectées par des capteurs (p. ex., récupérer la température à partir d'un capteur physique) ou issues de données stockées dans une base de données (p. ex., extraire les détails de rendez-vous à partir d'un calendrier) [Per+14].

En ce qui concerne les données dérivées, les informations sont générées en effectuant des opérations de calcul sur les données du capteur p. ex., calculer la distance entre deux bornes Wi-Fi (faisant rôle ici de capteur) dans la ville en utilisant leurs coordonnées GPS [Cra+12]. C'est à ce titre qu'on peut également employer des techniques de raisonnement numérique ou logique afin d'effectuer des calculs plus complexes (voir section 7).

En ce qui concerne les données fournies manuellement, les utilisateurs fournissent des informations de contexte par exemple en renseignement leur profil, leurs préférences, leurs habitudes. Cette méthode peut être utilisée pour récupérer tout type d'information jugé pertinent pour le système.

3. Internet Relay Chat
4. `https://openweathermap.org/api`

3.3.6 Synthèse

Nous pouvons regrouper les cinq facteurs à prendre en compte pour l'acquisition du contexte sur une figure récapitulative (Voir Figure 3.2). Tout système permettant l'acquisition du contexte pourra ainsi se définir en référence à cette figure.

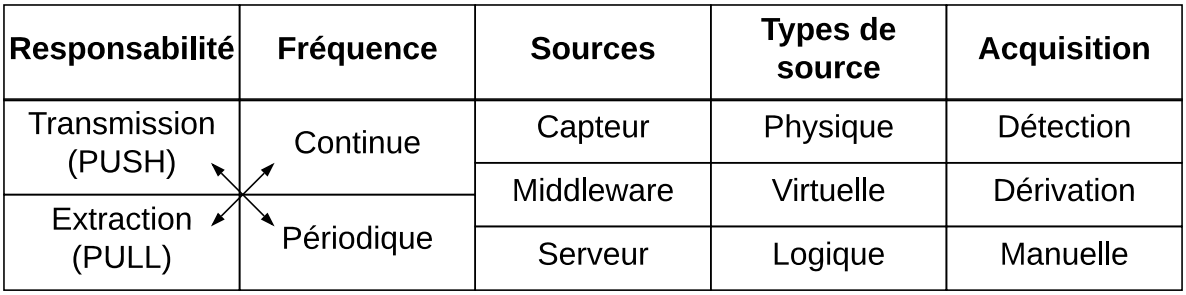


FIGURE 3.2 – Les cinq facteurs de l'acquisition du contexte [Per+14].

3.3.7 Notre contribution à l'acquisition du contexte

Dans le cadre de l'utilisation du contexte, nous avons réalisé trois différentes collectes de données afin d'acquérir du contexte selon trois besoins différents :

— la première collecte de données nous a permis de constituer un jeu de données d'entraînement hors ligne pour de la recommandation de services à des utilisateurs mobiles. Pour ce faire, nous avons réalisé un sondage pour lequel 1076 personnes ont répondu dans deux contextes différents. En effet, dans ce sondage, nous avons mis en situation deux contextes possibles selon lesquels les sondés pouvaient donner leur appréciation vis à vis de 18 recommandations de services possibles dans la ville intelligente. Ci-dessous les deux mises en situations que nous avons présentées :

1. Météo favorable au printemps : « *Il fait beau, nous sommes un samedi du mois de mai à 16h, température : 20 °C. Vous êtes dans le centre ville et vous avez une application mobile qui peut vous proposer des services personnalisés en temps réel. Parmi les services ci-dessous, lesquels vous intéresserait-il de recevoir ?* » ;
2. Météo défavorable en automne : « *Il pleut, nous sommes un samedi du mois de novembre à 16h, température : 8 °C. Vous êtes dans le centre ville et vous avez une application mobile qui peut vous proposer des services personnalisés en temps réel. Parmi les services ci-dessous, lesquels vous intéresserait-il de recevoir ?* ».

En plus des réponses à ces questions, nous avons demandé au préalable de renseigner des informations de profils comme l'âge, le sexe, la ville, le pays, le métier, la catégorie socio-professionnelle, la spécialité (p. ex., scientifique, littéraire, droit), le niveau d'études et les centres d'intérêts. Cette première acquisition de contexte nous a donc permis

de réaliser des simulations hors ligne de nos algorithmes de bandits-manchots pour la recommandation. Le jeu de données constitué est disponible sur Kaggle et se nomme : *Recommendation System for Angers Smart City (RS-ASM)*[5] ;

— la seconde collecte de données a été effectuée par la société Afone un opérateur virtuel basé à Angers, et qui mène un projet nommé *Wifilib*[6]. *Wifilib* est un réseau WiFi urbain continu mis à disposition dans les zones les plus fréquentées de différentes villes en France. Les données collectées se présentent sous la forme de journaux de connexions Wi-Fi (au format *JSON*) des différents utilisateurs du réseau et ce pour chacune des bornes Wi-Fi disponibles dans 15 villes en France durant une année (2015 à 2017). Cette collecte de données nous a permis d'acquérir du contexte brut que nous avons pu raffiner ensuite à l'aide notamment de techniques de raisonnements contextuels : l'utilisation des chaînes de *Markov* pour la prédiction de la mobilité ; et le partitionnement spectral pour la déduction de quartier dans la ville ;

— la troisième et dernière collecte de données a été réalisée à l'aide de notre application mobile de recommandation d'événements culturels : *scéno*, du projet *Event-AI*. Grâce à cette collecte de données nous avons pu acquérir du contexte de manière séquentielle et dynamique, ce qui nous a permis d'alimenter en contexte les algorithmes de recommandation de notre application en vue d'une évaluation en ligne.

En nous référant à la Figure 3.2 représentant une synthèse de l'état de l'art sur l'acquisition du contexte, nous pouvons positionner les trois acquisitions employées dans le cadre de cette thèse au tableau 3.1.

TABLE 3.1 – Positionnement de nos acquisitions de contexte par rapport à l'état de l'art

	Responsabilité	**Sources**	**Type de source**	**Acquisition**
RS-ASM	N/A	Serveur	Virtuelle	Manuelle
Wifilib	Pull/Push	Serveur	Physique	Détection
Event-AI	Push	Serveur	Logique	Dérivation + Manuelle

Nous rentrerons plus particulièrement dans le détail de ces acquisitions de contexte dans le chapitre ayant trait à nos contributions sur le sujet (voir Chapitre 6).

Ces contributions sont identifiées à la Figure 2 comme étant : *ISTE Ltd.2017, Elsevier 2018* [Gut+17 ; Gut+18d], et *Inforsid 2018 (Data Intelligence)* [Gut+18e].

3.4 Raisonnement contextuel

Le raisonnement contextuel peut être décrit comme étant l'extraction de nouvelles connaissances ou l'extraction de plusieurs ensembles de contextes à partir d'un contexte brut et ce afin d'obtenir une meilleure compréhension de celui-ci [Bik+07 ; Gua+07 ; SDO18].

5. https://www.kaggle.com/assopavic/recommendation-system-for-angers-smart-city
6. https://www.wifilib.com/index.html

Si on prend l'exemple de la capture d'informations géo-localisées d'utilisateurs mobiles, le contexte brut correspondrait alors aux coordonnées GPS collectées vis à vis d'une action donnée (p. ex., partage sur les réseaux sociaux, « like », connexions Wi-Fi sur des points d'accès). Ce contexte brut resterait inexploitable sans un raisonnement contextuel permettant d'extraire des connaissances compréhensibles ou lisibles comme c'est par exemple le cas dans le projet *Livehoods* mettant en évidence des quartiers dans la ville de *Pittsburg, USA* via des techniques de partitionnement spectral [Cra+12].

En résumé, l'incertitude et l'imperfection des données dites brutes du contexte posent la question de son exploitabilité. C'est à ce titre que le raisonnement contextuel rentre en jeu afin de proposer des solutions remédiant à cette incertitude et à cette imperfection. Le raisonnement contextuel comporte trois principales étapes : le pré-traitement du contexte, le croisement de données multi-sources, et l'inférence de contexte [Li+15 ; NF04 ; SDO18].

3.4.1 Les différentes étapes du raisonnement contextuel

Pré-traitement. Dans la phase de pré-traitement du contexte [SDO18], les données sont nettoyées et re-construites en définissant les attributs de contexte pertinents (p. ex., réduction de dimensions, induction du sous-ensemble de caractéristiques), en remplissant les données manquantes, en validant le contexte, en supprimant les valeurs aberrantes et en utilisant des techniques de *Data Mining*.

Croisement multi-sources. Dans l'étape de croisement de données émanant de plusieurs sources [SDO18], les différentes données issues de plusieurs sources de données sont combinées pour produire des données plus fiables, plus précises, et plus complètes, qui ne pourraient être fournies par des données d'une source unique.

Inférence de contexte. Dans la phase d'inférence de contexte [SDO18], l'objectif est d'identifier/calculer le nouveau contexte pertinent, et de le faire correspondre au contexte brut (raisonnement logique et probabiliste).

L'un des points centraux du raisonnement contextuel, reste le choix du modèle qui est employé pour inférer le nouveau contexte pertinent. C'est pourquoi nous décrirons plus précisément ci-dessous les différents modèles existant dans la littérature.

3.4.2 Modèles de raisonnement contextuel

Le raisonnement contextuel a pu être classé en différentes catégories [SDO18] en fonction de l'approche choisie. Ainsi, selon [Per+14], il existe des modèles utilisant des approches basées sur :

— la logique floue [HMP12 ; Hag+04 ; PLZ08] ;
— la logique probabiliste [KRW09 ; LL05 ; Zha+09] ;

— les ontologies [Chi13 ; Gyr15 ; PB14], pouvant être combinées à l'usage de règles [BDG11 ; Gyr15 ; Per+13] ;
— l'apprentissage supervisé [BCR09 ; KK10 ; POC11] ;
— l'apprentissage non supervisé [LL05 ; SJ08 ; VAL01].

3.4.2.1 Logique floue.

La logique floue est différente de la logique traditionnelle où tout est représenté par des valeurs de vérité $vrai, faux$ [SDO18]. En effet, dans la logique floue, une vérité *partielle* déterminée à l'aide de probabilités est également possible. De cette manière, la représentation du monde réel à travers la logique floue est plus réaliste que l'utilisation de la logique dite traditionnelle. L'un des principaux intérêts d'utiliser la logique floue est la personnalisation. Par exemple, [HMP12] propose un modèle basé sur la logique floue pour la représentation des préférences contextuelles d'utilisateurs. Leur objectif est principalement de déterminer les préférences les plus pertinentes afin de personnaliser la requête de l'utilisateur en fonction des informations contextuelles disponibles.

Il est à noter que les techniques de raisonnement en logique floue sont fréquemment utilisées avec d'autres techniques de raisonnement notamment le raisonnement ontologique, probabiliste et basé sur des règles.

3.4.2.2 Logique probabiliste

Avec cette technique, les décisions sont basées sur le calcul de probabilités d'événements et de faits [SDO18] où différentes données issues de différents capteurs sont ainsi combinées à une logique probabiliste. Les modèles de *Dempster-Shafer* [Wu+02] et de *Markov* cachés [SDO18] sont notamment utilisés comme raisonnement probabiliste pour prédire le prochain événement, reconnaître les activités et prévoir les incertitudes. Le modèle de *Dempster-Shafer* utilise le croisement de données de plusieurs capteurs pour calculer la probabilité d'événements. Les modèles de *Markov cachés* quant à eux fournissent une vision du prochain état en utilisant l'état courant mesuré.

Plus précisément, les modèles de *Markov* cachés sont des réseaux bayésiens dynamiques basés sur un modèle de *Markov* statistique [SDO18] (modèle mathématique probabiliste qui définit les états futurs en fonction de l'état courant). Par exemple [ME12] utilise des modèles de *Markov Cachés* en se basant sur les traces de connexions Wi-Fi, afin d'estimer les trajectoires des utilisateurs dans la ville. Ces trajectoires peuvent à ce titre être considérées comme du contexte pertinent à prendre en considération dans des applications réelles comme entre autres les systèmes de recommandation.

3.4.2.3 Les ontologies et l'usage de règles.

La logique basée sur une ontologie dépend de la logique de description [SDO18]. De ce fait, le raisonnement peut être réalisé avec des données modélisées ontologiquement. Il existe

des langages de web sémantiques tels que *RDF*, *RDFS* et *OWL*, qui sont utilisés pour implémenter un raisonnement basé sur une ontologie. Ces langages peuvent être combinés avec la modélisation d'ontologies, ce qui donne un certain avantage [SDO18]. Par exemple [SCD15] ont défini une ontologie de réseau de capteurs sémantiques pour les maisons intelligentes et ont mis en œuvre un simulateur de réseau de capteurs sémantiques pour la domotique.

Cependant, le principal inconvénient du raisonnement basé sur des ontologies est qu'il reste incapable de fournir les valeurs manquantes (en cas de parcimonie) et de détecter d'éventuelles ambiguïtés [SDO18]. De ce fait, il devient nécessaire de le combiner avec un raisonnement basé sur des règles. Avec le raisonnement basé sur les règles, le contexte raisonné peut être acquis avec une structure de type *If-Else* [SDO18]. Par exemple, il devient possible d'utiliser les préférences utilisateur et la détection d'événements, modélisés avec des règles à utiliser dans les applications de l'*Internet des Objets* [BDG11].

3.4.2.4 L'apprentissage supervisé.

En apprentissage supervisé, les données sont tout d'abord collectées puis étiquetées afin de pouvoir effectuer la phase d'entraînement. Ensuite, différentes techniques et algorithmes sont élaborés en fonction des résultats attendus et appliqués à toutes les données disponibles.

Réseaux de neurones. Parmi les techniques d'apprentissage supervisé, on peut citer les réseaux de neurones qui sont utilisés dans le raisonnement contextuel pour mettre en exergue des modèles complexes existant entre des entrées (données captées) et des sorties (résultats, actions observés) [SDO18]. Par exemple [SME14] ont utilisé des capteurs de *smartphone* (accéléromètre, gyroscope, GPS, magnétomètre et température) pour améliorer la navigation personnelle sur téléphone mobile (*UX*) via l'utilisation de réseaux de neurones pour la partie apprentissage du système.

Réseaux bayésiens. Une autre technique employée est celle des réseaux bayésiens dans le cadre d'un raisonnement probabiliste. Par exemple, les classificateurs bayésiens ont déjà été utilisés dans le domaines de la santé p. ex., surveillance et classification du taux de respiration, nombre de pas [Jat+08].

Arbres de décision. Il est également possible d'utiliser des arbres de décision pour de la classification de données. Par exemple, dans [EPC08], les activités humaines (marcher, courir, s'asseoir, etc.) sont déterminées à l'aide de réseaux de capteurs portatifs (accéléromètre, podomètre, etc.) et de systèmes embarqués.

SVM. Enfin, les techniques de *SVM* (Machines à Vecteurs de Support) peuvent aussi être utilisées. Par exemple [Kha+14] a proposé une méthode permettant de classer les données de *streaming* émanant de différents objets connectés en utilisant entre autres une technique de *SVM*.

3.4.2.5 L'apprentissage non supervisé.

Le but de l'apprentissage non supervisé est de regrouper (partitionner) des données non étiquetées [SDO18]. Les résultats sont retournés généralement plus rapidement qu'avec une approche d'apprentissage supervisé. En outre, avec l'apprentissage non supervisé, un grand volume de données hétérogènes est divisé en plusieurs sous ensembles homogènes plus faciles à interpréter et à gérer. Il existe différentes approches d'apprentissage non supervisé parmi lesquelles : les réseaux de neurones non supervisés, l'apprentissage de règles d'association, et le partitionnement (*clustering*).

Clustering. Le partitionnement (*clustering*) est utilisée pour extraire des résultats significatifs à partir de données non étiquetées comme c'est le cas pour la méthode de partitionnement des *k* plus proche voisins. Il existe différents algorithmes de *clustering* dans les techniques d'apprentissage non supervisées. Parmi ces techniques on peut citer l'algorithme de *k-means* [Kan+02 ; MRF03], le *fuzzy-clustering* [Pha01], *DBSCAN* (*clustering* spatial) [Maj+13], ou encore l'algorithme *OPTICS* [Ank+99]. Le *k-means* est l'un des algorithmes de classification le plus fréquemment utilisé. Il fournit la distance minimale entre des données similaires et la différence maximale entre les clusters.

Apprentissage de règles d'association. Il est aussi possible d'utiliser une méthode d'apprentissage de règles d'association. Le but de cette méthode est de trouver des relations (associations) intéressantes entre les variables. Pour se faire il existe différents algorithmes d'apprentissage de règles d'association, tels que *apriori* [GL12], *eclat* [Lun+17] et *FP-growth* [HGN00].

Réseaux de neurones. Il existe également la technique de *KSOM (Kohonen Self-Organization Map)* [Van01] (c.-à-d., technique de réseau neuronal non supervisé) utilisée pour la classification des données réelles dans le cadre d'applications sensibles au contexte, telles que la détection de bruit et de valeurs aberrantes [Per+14].

3.4.3 Le géo-partitionnement (*geo-clustering*) : un exemple de raisonnement contextuel.

Les applications dans la ville intelligente utilisent très souvent des données relatives à la mobilité des utilisateurs. Par exemple, les applications de type *LBSN* (*Location-Based Social Networks*) [Bao+13] utilisent non seulement des données relatives aux usagers, mais également à leurs habitudes de déplacement. Les applications de *MCSC* (*Mobile Crowd Sensing and Computing*) utilisent quant à elles des données locales générées par les dispositifs mobiles de leurs utilisateurs pour en déduire de l'information plus globale sur par exemple le niveau de bruit, le trafic routier, le contexte ambiant etc. [Guo+15]. La prise en compte de la localisation des utilisateurs est donc indispensable pour les applications dans la ville. Pour cela, elles ont à

leur disposition un très grand nombre de données brutes (coordonnées GPS, connexions aux points d'accès Wi-Fi et antennes GSM environnantes, . . .).

Toute la question qui se pose au regard de ce géo-contexte brut est celle de son exploitabilité. C'est pourquoi, il est nécessaire d'utiliser des algorithmes d'apprentissage dont l'objectif est de trouver des lieux qui ont un sens pour les utilisateurs. Ils se basent soit sur des données géographiques [SDO18], soit directement sur des mesures physiques (*fingerprint algorithms*) [Kim+09]. Dans les approches se basant sur les données géographiques, les données sont la plupart du temps recueillies par des capteurs ou des applications embarquées sur les mobiles [AT05]. Ces différentes données de localisation peuvent être regroupées grâce à des algorithmes de partitionnement en zones accueillant des utilisateurs partageant des similarités quant aux lieux qu'ils fréquentent. Ces méthodes de partitionnement reposent principalement sur la méthode des *k-moyennes* et ses variantes [Kan+02]. Le principe est de regrouper par itération les données qui sont les plus similaires possibles. Ces méthodes sont très sensibles notamment en présence de données aberrantes, et ne permettent d'obtenir que des formes de regroupement sphériques. Il existe des algorithmes de classification comme CURE [GRS98], *BIRCH* [ZRL96] ou *CHAMELEON* [KHK99] permettant de pallier ces inconvénients.

Une autre approche employée en apprentissage non supervisée est le partitionnement spectral. Elle a été largement étudiée et obtient de bons résultats empiriques lorsqu'elle est bien paramétrée [NJW02 ; Nou+11 ; Ryu+17]. Cette méthode a notamment été utilisée pour le projet *Livehoods*[7] [Cra+12] qui, à partir de données de *Foursquare* (*LBSN*), a pu inférer des dynamiques urbaines et sociales se traduisant visuellement par le découpage de zones dans les grandes villes Nord Américaines. Les résultats obtenus via partitionnement spectral par le projet *Livehoods* sur *Foursquare* ont été évalués et sont pertinents.

3.4.4 Nos contributions au raisonnement contextuel

Dans le cadre de cette thèse, nous avons travaillé sur plusieurs contributions sur le raisonnement contextuel (voir Figure 2 dans la partie *Introduction*).

La première de ces contributions, identifiée dans la Figure 2 comme étant *ISTE Ltd.2017 et Elsevier 2018* [Gut+17 ; Gut+18d], utilise des chaînes de *Markov* cachées pour prédire la prochaine connexion des utilisateurs sur un réseaux Wi-Fi urbains à grande échelle.

La seconde de ces contributions, identifiée dans la Figure 2 comme étant : *SAGEO 2018 (EXCES)* [Gut+18c], utilise un algorithme de partitionnement spectral pour déduire des quartiers dans la ville. En effet, pour cette seconde contribution, à notre connaissance, sur les connexions aux réseaux Wi-Fi urbains à grande échelle, aucune étude n'a ni proposé de méthode de partitionnement spectral, ni mis en évidence les performances d'une telle approche. Ainsi, l'algorithme que nous avons utilisé est une variante de celui proposé par [Cra+12]. Plus précisément, la méthode *spectral_clustering* de la bibliothèque *Scikit-learn*[8] que nous avons utilisée pour mettre en oeuvre notre algorithme est issue de techniques de partitionnement

7. `http://livehoods.org/`
8. Bibliothèque de *Python* dédiée à l'apprentissage automatique : `http://scikit-learn.org/`

spectral telles que définies entre autres par [Von07]. Par la suite, nous avons étendu cette contribution en réalisant une soumission internationale (Étoile n° 8) proposant d'utiliser les clusters inférés dans un système de recommandation d'événements culturels à des utilisateurs mobiles. Ces travaux font l'objet notamment d'une évaluation en ligne effectuée dans l'application mobile *scéno*.

Ainsi dans notre première contribution nous effectuons un raisonnement contextuel via l'usage d'une technique de logique probabiliste alors que dans nos deux autres contributions nous raisonnons via l'usage d'une technique d'apprentissage non-supervisée.

3.5 Bilan

Dans ce chapitre nous avons rappelé les trois points cruciaux à prendre en compte lorsqu'on souhaite mettre en jeu un système de recommandation contextuel : la modélisation du contexte, son acquisition et le raisonnement contextuel.

L'état de l'art sur les systèmes de recommandation, les bandits-manchots et sur le contexte ayant été rappelé, nous consacrerons les prochains chapitres à la description de nos contributions à savoir :

1. Nos contributions aux Algorithmes de bandits-manchots pour la recommandation ;
2. Nos contributions à l'élaboration du contexte pour les algorithmes de recommandation.

DEUXIÈME PARTIE

Contributions aux algorithmes de bandits-manchots pour la recommandation

Introduction de la partie II

Aujourd'hui, les algorithmes de bandits-manchots reposent sur des bases théoriques solides sur lesquelles il est possible de s'appuyer. Cette promesse forte d'une performance prouvée et explicable, a été un élément clé de décision d'utilisation des algorithmes de bandits-manchots dans le cadre de cette thèse.

Néanmoins, même si ces garanties théoriques restent une valeur sûre et incontournable, il est utile d'évaluer les méthodes dans un cadre réel c.-à-d., en ligne dans des applications, ou hors ligne sur des jeux de données du monde réel.

Dans cette partie, nous proposons ainsi au Chapitre 4 une étude préliminaire où nous évaluons la performance de sept algorithmes de bandits-manchots pour la recommandation sur douze jeux de données différents. En plus du critère de précision globale, nous proposons d'évaluer la performance selon deux nouvelles métriques : la diversité et la précision individuelle.

À la lumière des analyses de notre étude préliminaire sur les trois critères de précision globale, diversité et précision individuelle, au Chapitre 5 nous proposons deux contributions portant sur l'impact de la diversification des recommandations sur la précision individuelle. Pour ce faire nous proposons à la Section 5.2 un premier mécanisme de diversification des recommandations directement intégré dans un algorithme de bandit-manchot contextuel : *LinUCB*. Ce mécanisme utilise le principe de fenêtre glissante et permet de pénaliser les éléments qui sont recommandés trop fréquemment aux utilisateurs. Ensuite, nous proposons à la Section 5.3 une seconde méthode permettant de diversifier les recommandations via l'emploi d'une approche de type porte-folio d'algorithmes de bandits-manchots contextuels et non contextuels. Enfin, nous comparons ces deux méthodes afin d'identifier laquelle est la plus pertinente pour nos besoins applicatifs.

CHAPITRE 4

ALGORITHMES DE BANDITS-MANCHOTS : UNE HISTOIRE DE PRÉCISION

Being accurate is not enough...

« *Most research up to this point has focused on improving the accuracy of recommender systems. We believe that not only has this narrow focus been misguided, but has even been detrimental to the field. The recommendations that are most accurate according to the standard metrics are sometimes not the recommendations that are most useful to users.* [MRK06]

Sean M. McNee, John Riedl & Joseph A. Konstan - 2006 »

Sommaire

4.1 Introduction

Ce chapitre fait référence en partie à nos contributions [Gut+18b] (étoile n° 4), [Gut+19d] (étoile n° 5), [Gut+18a] et [Gut+19a] (étoile n° 6) ainsi que [Gut+19b] (étoile n° 7) présentées sur la Figure 2 en introduction de ce mémoire. En effet, dans ces articles, nous nous référons notamment à notre étude préliminaire [1] qui constitue le point d'entrée des méthodes que nous avons élaborées.

Ce chapitre est dédié à l'extension et la description de cette étude préliminaire, par conséquent elle est plus exhaustive et plus détaillée que ce que nous avions mis à disposition sur *github*. De ce fait, dans ce chapitre :

1. Nous rappellerons les algorithmes décrits au Chapitre 2 et que nous avons comparés dans cette étude ;

1. Disponible sur *github* : `https://github.com/mabresearchstudy/mabaccuracy`

2. Nous préciserons les critères de performance qui ont servi à l'évaluation. Ainsi, nous rappellerons les deux principaux critères employés dans la littérature et ceux que nous avons mis en place afin d'évaluer la diversité des recommandations et la précision individuelle. Ces nouveaux indicateurs représentent en eux-mêmes une partie de nos contributions [Gut+18a ; Gut+18b ; Gut+19a ; Gut+19d] ;
3. Nous décrirons le protocole expérimental défini pour les simulations à savoir : les jeux de données étudiés et le processus de simulation (c.-à-d., construction du vecteur de contexte, simulation du flux de données d'utilisateurs et des recommandations, choix des horizons, évaluations des performances et tests statistiques) ;
4. Nous observerons et analyserons les résultats que nous avons obtenus. Ces résultats nous permettrons de déduire les performances de chacun des algorithmes sur différents jeux de données du monde réel selon les critères définis pour leur évaluation. Ils seront le point d'entrée des contributions qui suivront au Chapitre 5 [Gut+18a ; Gut+19a] et au Chapitre 7 [Gut+18b ; Gut+19d]. L'ensemble des analyses détaillées, des tableaux et des figures de cette étude préliminaire a été placé dans les annexes (Voir Annexe B pour les analyses et figures, et voir Annexe C pour les tableaux). Néanmoins, nous résumerons les analyses dans ce chapitre afin d'en faciliter la lecture.

4.2 Algorithmes comparés

Dans ce premier chapitre concernant nos contributions sur les bandits-manchots pour la recommandation, nous en étudierons différents existants et que nous avons rappelé au Chapitre 2.

Ainsi, nous nous pencherons sur les performances des algorithmes suivants :

— **bandits-manchots (*Multi-Armed Bandits - MAB*) :** *UCB2*, ε*-Greedy* avec ε fixe, *Thompson Sampling* (*TS*) et *EXP3* ;
— **bandits-manchots contextuels (*Contextual Multi-Armed Bandits - CMAB*) :** *LinUCB*, *Contextual Thompson Sampling* (*CTS*) et *EXP4.P*.

Nous définirons plus précisément ce que nous entendons par *performance* des algorithmes de bandits-manchots dans la Section 4.3.

4.3 Évaluation de la performance

Afin d'évaluer la performance des algorithmes de bandits-manchots, nous avons employé l'un des deux principaux critères définis dans la littérature : la précision globale (voir Section 2.4).

De plus, dans le cadre de cette thèse, nous avons défini deux nouveaux critères :

— la précision individuelle, qui représente la précisions des recommandations obtenues pour chaque individu (p. ex., utilisateurs) ;

— la diversité dans le cas où le nombre de bras k reste fixe tout au long de l'expérience. Puisque le nombre de bras est fixe, les calculs pour déterminer la diversité sont différents de ceux effectués pour la similarité intra-liste (*Intra-List Similarity - ILS*) rappelée en Section 2.4 et qui fait référence à de la diversité d'éléments à recommander au sein d'une liste. En effet, le calcul d'*ILS* suppose : d'une part que nous ayons la description des éléments permettant de calculer leur distance les uns par rapport aux autres ; d'autres parts que la liste d'éléments ne soit pas fixe et que nous puissions l'enrichir avec d'autres éléments extérieurs.

4.3.1 La notion de contexte dans nos simulations

Dans nos simulations évaluant la performance, nous représentons le contexte [Bou+17 ; Bré99 ; Dey01 ; Li+10] sous une forme structurée à travers un vecteur de caractéristiques. Selon leur disponibilité et le cas d'application sur lequel nous travaillons, les caractéristiques peuvent être différentes, par exemple : l'utilisateur lui-même (profil, préférences), le contexte spatio-temporel, et les caractéristiques des éléments à recommander. Comme dans les applications du monde réel, les utilisateurs évoluant dans un contexte donné peuvent être des utilisateurs caractérisés comme « *abonnés* », alors nous considérons qu'ils peuvent être régulièrement identifiés. C'est le cas par exemple lors de l'utilisation d'une application mobile ou d'abonnement à des journaux en ligne.

Par conséquent, nous considérons qu'un même utilisateur u, peut rencontrer à plusieurs reprises des situations représentées par le même vecteur de contexte x. Nous représentons ceci par la notion de paire utilisateur-contexte (u, x).

4.3.2 Précision globale et Cumul des regrets (Rappel)

La précision globale est un critère de performance basé sur le total des récompenses positives cumulées à l'horizon T. De ce fait, pour obtenir la précision globale, nous calculons le gain c'est à dire le nombre total de récompenses positives $g(T)$ puis nous calculons enfin la précision (*Accuracy* : Acc) tel que :

$$Acc(T) = \frac{g(T)}{T}$$

où $g(T) = \sum_{t=1}^{T} r_t$ et $r_t \in \{0, 1\}$.

Le cumul des regrets à l'Horizon T quant à lui correspond à $\sum_{t=1}^{T} \rho_t$ ou encore $\sum_{t=1}^{T}(1 - r_t)$.

4.3.3 Diversité

Dans le cadre des systèmes de recommandation à des utilisateurs, il a été observé dans certaines études que les mesures de précision ne sont pas suffisamment adaptées et pourraient être préjudiciables et nuire à la satisfaction des utilisateurs [MRK06]. Ainsi, la précision seule ne suffit pas à garantir qu'un système soit performant [GDJ10]. Il est en effet important

de prendre en compte les facteurs humains, qui jouent un rôle important dans les processus de décision [CBB13].

Par conséquent, l'un des critères important à prendre en compte est la diversité des recommandations (voir Section 2.4). Néanmoins, dans le cas où la liste des éléments à recommander reste fixe ou possède des catégories, soit non distinctes d'éléments, soit sur-représentées par un trop grand nombre d'éléments, il serait alors plus pertinent de considérer la diversité des recommandations à travers une mesure de dispersion telle que l'écart-type. En effet, plus un ensemble de données, de même échelle a un écart-type élevé, plus sa dispersion est importante.

Ainsi, soit n_{i_j} le nombre de recommandations effectuées pour chacun des k éléments $i \in I$ tel que : $\forall j \in \{1..k\}, \forall n_{i_j} \in \mathbb{N}^*, N = \{n_{i_1}, ..., n_{i_k}\}$, n_{i_j} correspond au nombre de recommandations effectuées pour l'élément i_j.

Alors, la moyenne des recommandations $\overline{N}$ est calculée comme suit :

$$\overline{N} = \frac{1}{k} \sum_{j=1}^{k} n_{i_j}$$

La variance de ces recommandations $V(N)$ est donc :

$$V(N) = \frac{1}{k} \sum_{j=1}^{k} (n_{i_j} - \overline{N})^2$$

On en déduit l'écart-type $\sigma(N)$ tel que :

$$\sigma(N) = \sqrt{V(N)}$$

Néanmoins, pour interpréter correctement l'écart-type il est important de pouvoir le normaliser à une échelle qui en permette la lecture et le rende indépendant du nombre total de recommandations effectuées. C'est pourquoi le coefficient de variation qui découle de l'écart-type semble être une mesure statistique appropriée. À partir de $\sigma(N)$ et $\overline{N}$ il est donc possible de calculer le coefficient de variation $c_\nu(N)$ comme suit :

$$c_\nu(N) = \frac{\sigma(N)}{\overline{N}}$$

On peut ensuite qualifier la dispersion des recommandations telle que quand $c_\nu(N) \to 0$ alors la dispersion de sélection des bras (éléments à recommander) tend vers son maximum. A contrario, la dispersion des recommandations tend vers son minimum quand $c_\nu(N) \to \sqrt{k}$ [KK57].

Le calcul du niveau de diversité $Div(N)$ (croissant de 0 à 1) peut donc être défini comme suit :

$$Div(N) = 1 - \frac{c_\nu(N)}{\sqrt{k}}$$

Dans nos expériences, nous utiliserons ce calcul de la diversité.

4.3.4 Précision individuelle

La plupart des systèmes sont évalués à travers une mesure de précision globale e.g., dans le cadre des bandits-manchots : la récompense moyenne, le cumul des récompenses, ou le nombre de regrets total [AG13 ; BF16 ; Li+10] (voir Chapitre 2). Néanmoins, de telles métriques semblent inadéquates pour déterminer la précision à associer à chaque contexte [MRK06]. Ainsi, si nous prenons l'exemple des systèmes de recommandation pour lesquels les utilisateurs peuvent être des visiteurs réguliers, ou des abonnés (exemple : applications mobiles), il semble essentiel de tenir compte de leurs retours individuels en regard des recommandations qui leur sont faites.

Afin de prendre en compte les retours individuels des utilisateurs, nous proposons un nouveau critère que nous nommons précision individuelle [Gut+18a ; Gut+18b ; Gut+19a ; Gut+19d].

La précision individuelle par utilisateur $Acc_u(T)$ peut être définie comme étant

$$\forall u \in \mathcal{U}, Acc_u(T) = \frac{\sum_{t=1}^{T} r_{t,u}}{T_u}$$

où T_u représente le nombre de fois qu'un utilisateur avec son contexte a été sélectionné à l'horizon T, et $r_{t,u}$ correspond à la récompense retournée par u à l'itération t.

Les mesures de précision individuelle de chaque utilisateur peuvent être représentées par une fonction de distribution cumulative (FDC). La FDC nous permet ainsi d'observer la distribution de la précision individuelle sur l'ensemble des utilisateurs $\mathcal{U}$ avec leur contexte à l'horizon T.

Afin d'illustrer la lecture d'un tracé de FDC, nous proposons de décrire succinctement la Figure 4.1. Nous détaillerons son analyse de manière plus précise dans la section ayant trait à l'analyse des résultats (Section 4.5). Ainsi, prenons deux tracés de la Figure 4.1 : la FDC de *LinUCB* et celle d'*UCB2*. Concernant *LinUCB* on observe que 100% de la précision est répartie dans le dernier intervalle de précision individuelle (entre $0,9$ et 1 de précision). C'est à dire que 100% de la population est satisfaite pour plus de neuf recommandations sur dix qui lui ont été faites. Dans le cas d'*UCB2*, on constate que la FDC croît en tout début (entre 0 et $0,1$ de précision) jusqu'à atteindre un cumul représentant 75% de la population. La FDC croît ensuite en toute fin (entre $0,9$ et 1 de précision) concernant les 25% de la population restante. Cela signifie, que pour 75% de la population, l'algorithme obtient une précision des recommandations inférieure à $0,1$ et que pour 25% de la population l'algorithme obtient une précision des recommandations proche de 1.

De ce fait, grâce à cette mesure de la précision individuelle, il est possible de visualiser comment la précision est répartie au sein des individus. Nous remarquerons plus tard, dans la section présentant les résultats (Section 4.5), que pour une même précision globale il existe une répartition de la précision individuelle totalement différente. Ces courbes nous permettent d'observer que différents algorithmes obtiennent des résultats plus ou moins équilibrés en termes de précision individuelle de recommandation au sein de la population. En d'autres termes, nous pourrions considérer qu'ils obtiennent des résultats plus ou moins « équitables ».

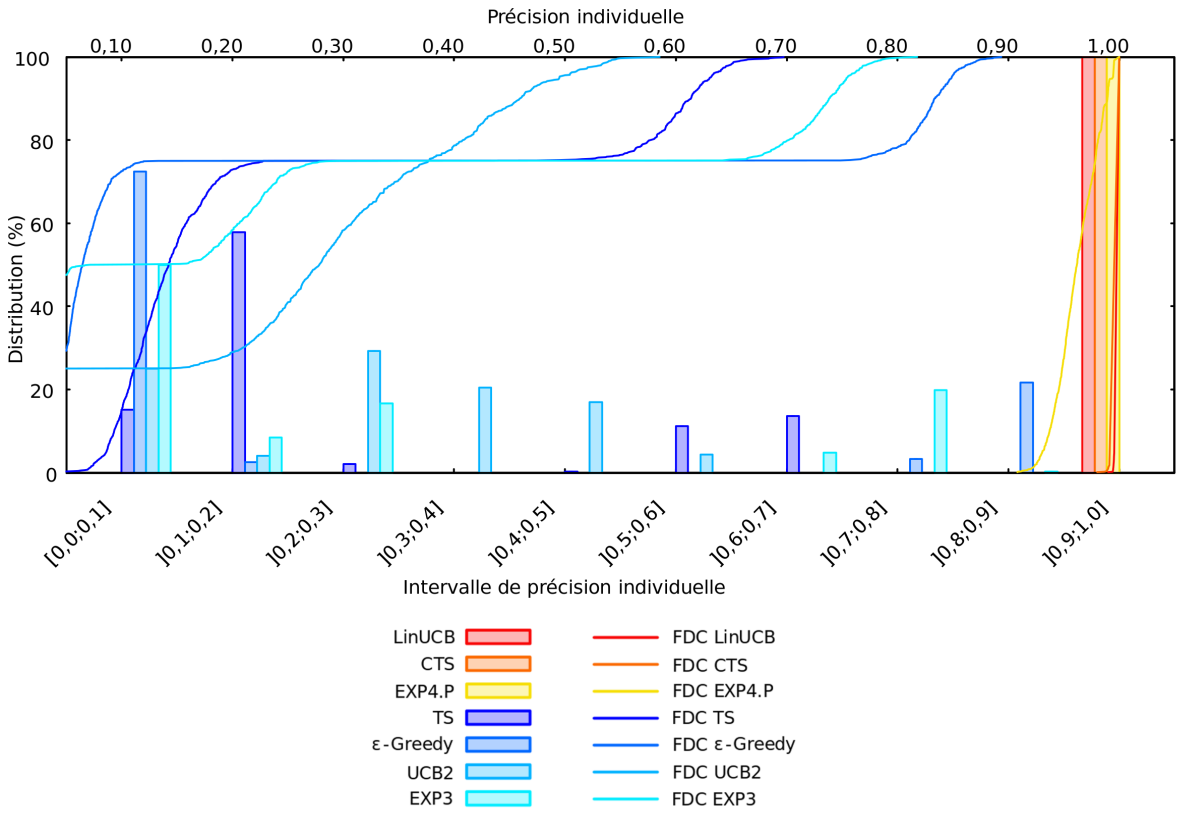


FIGURE 4.1 – FDC de différents algorithmes de *MAB* et *CMAB* sur le jeu de données de *Contrôle* (Vecteur complet optimal)

4.4 Simulations

Avant d'observer et d'analyser les résultats, il est important de rappeler les jeux de données que nous avons sélectionnés pour notre étude et le processus de simulation.

4.4.1 Sélection des jeux de données

Avant de déployer les algorithmes de bandits-manchots en condition réelle, il est nécessaire d'en réaliser une évaluation hors-ligne à partir de données collectées au préalable.

En effet, même si les algorithmes de bandits-manchots contextuels et non contextuels possèdent des garanties théoriques fortes de convergence, nous supposons que leurs performances peuvent différer selon la nature du jeu de données employé (nombre de bras, nombre d'instances, niveau de description du vecteur de contexte, répartition des probabilités de récompense, etc.), comme cela pourra être le cas dans l'application finale que nous visons c.-à-d., recommandation contextuelle de services dans la ville intelligente via l'application mobile

scéno[2]. C'est pourquoi, dans cette étude préliminaire, nous observerons et analyserons la performance des différents algorithmes de bandits-manchots sur différents jeux de données de différentes natures en termes de : nombre d'instances (contextes observables), dimensions contextuelles (de 0 à 80 dimensions catégorielles impliquant de 0 à 270 dimensions binaires), et de bras (de 0 à 100 classes).

Même si notre travail final se focalise sur les systèmes de recommandation, il est pertinent d'éprouver nos solutions sur divers champs d'application afin de pouvoir les valider. Ainsi, nous avons choisi douze jeux de données de domaines métiers variées issus de la classification supervisée. Deux de ces jeux de données ont été générés de manière artificielle (*Contrôle* et *YSE*). Ils seront donc pris en considération à des fins de simulation pour de la recommandation.

Après avoir rappelé comment nous préparons le contexte pour chaque jeu de données, ainsi que la procédure que nous employons pour créer des cas de restriction sur le contexte [Bou+17] (c.-à-d. en tronquant une partie du contexte), nous décrivons les jeux de données que nous avons employés.

4.4.1.1 Préparation du contexte des jeux de données

Notons que, le contexte issu des jeux de données a été préparé afin d'être fourni aux algorithmes en une représentation structurée de variables binaires (encodage *one-hot*). Ce pré-traitement a consisté à transformer les variables continues des jeux de données originaux en variables catégorielles en les divisant suivant leur plage de valeurs selon leurs quantiles, c.-à-d., suivant la pertinence : en déciles, quintiles ou quartiles. Ensuite, nous avons binarisé ($\{0, 1\}$) les variables catégorielles afin d'obtenir un vecteur de type « *one-hot* ». Notons que ce type de traitement est habituel dans le domaine de l'apprentissage automatique et plus particulièrement dans les bandits-manchots [All16 ; Li+10].

4.4.1.2 Créer de la restriction sur le contexte dans certains jeux de données

Concernant certaines expériences nous simulerons 2 cas différents : 1) Avec le vecteur de contexte complet (*vc*) présent dans le jeu de données d'origine, 2) Avec une partie tronquée du vecteur de contexte d'origine (*vt*) et ce afin d'observer l'impact d'un accès restreint aux informations de contexte. Dans ces cas de *vt*, nous rappellerons les modifications qui ont été apportées aux jeux de données spécifiquement dans leur description.

4.4.1.3 Description des jeux de données

Chacun des douze jeux de données considéré est constitué d'un nombre d'instances, s'appuie sur un contexte d'une dimension d donnée (variables binaires) et propose un nombre défini de bras (voir Tableaux 4.1 et 4.2).

Nous décrivons ci-dessous les jeux de données présentés en version *vc* dans le Tableau 4.1 et en version *vt* dans le Tableau 4.2 :

2. `https://play.google.com/store/apps/details?id=sceno.android.pfe.com.sceno&hl=fr`

— ***Contrôle*** est un jeu de données généré artificiellement afin d'obtenir un vecteur de contexte optimal x^* et une distribution équiprobable des récompenses entre les quatre bras. Il servira de jeu de données de contrôle. Nous avons également créé ce jeu de données en version *vt*, c'est à dire en tronquant le vecteur de contexte (c.-à-d., en retirant la dernière dimension). Les versions *vc* et *vt* du jeu de données *Contrôle* doivent être considérées comme des références : la version *vc* illustre le fonctionnement optimal d'un algorithme (tenant compte du contexte) avec un x^* (contexte optimal), tandis que la version avec vecteur de contexte tronqué (*vt*) illustre l'effet d'une restriction (ou parcimonie) du contexte sur les algorithmes de *CMAB* ;

— ***YSE*** est un jeu de données que nous avons spécifiquement créé pour illustrer l'effet *Yule-Simpson* [Sim51 ; Yul03] sur les algorithmes de *CMABs*. Dans la version (*vc*) présenté dans le tableau 4.1, nous proposons en premier lieu un jeu de données considérant l'ensemble des dimensions, ce qui permet d'éviter l'effet *Yule-Simpson*. Nous créons ensuite une version (*vt*) qui quant à elle, permet de provoquer l'effet *Yule-Simpson* puisqu'on ne possède plus la dimension du vecteur permettant de le pallier. Ce jeu de données est composé de 1000 instances d'hommes et de femmes travaillant dans deux filiales d'une société quelconque, une en France et une en Inde. Cette société garantit l'équité des salaires entre les deux sexes dans chacun des pays. Plus précisément, ce jeu de données comporte deux classes de salaire ($40\ 000$ euros pour les H/F travaillant en France et $20\ 000$ euros pour les H/F travaillant en Inde). Afin de créer un déséquilibre amenant à un effet *Yule-Simpson*, nous créons 300 hommes et 200 femmes en France, et inversons cette tendance pour l'Inde (i.e, 200 hommes et 300 femmes en Inde) si bien que cela aboutit à une moyenne de salaire pour les femmes inférieure à celle des hommes si on ne prend pas en compte la dimension correspondante au pays. L'objectif ici est d'observer les résultats obtenus par les algorithmes de *CMABs* dans le cas où on ne leur fournit pas la dimension « pays » en tant que donnée du problème. Dans ce jeu de données, le contexte de la version (*vc*) correspond alors au genre (H/F) et au pays (France/Inde) alors que dans la version (*vt*) nous y retirons la dimension du pays ;

— ***Recommendation System for Angers Smart City (RS-ASM)*** est un jeu de données spécifique à la **recommandation** que nous avons mis à disposition sur *Kaggle*[3] et que nous utilisons dans [Gut+18a ; Gut+18b ; Gut+19a ; Gut+19d]. À l'aide d'un questionnaire en ligne, nous avons collecté des données de 1076 personnes dans deux contextes environnementaux différents (été/hiver). Elles ont renseigné leurs données de profil (âge, sexe, catégorie socioprofessionnelle, niveau de diplôme, spécialités, préférences, ville). Ainsi, pour chaque utilisateur nous disposons de ses caractéristiques de contexte (profil et contexte environnemental inclus). Ces personnes sondées ont ensuite sélectionné, parmi un ensemble de 18 services, ceux pour lesquelles elles souhaiteraient recevoir des recommandations de type *notification* sur leur mobile et ce dans deux différents contextes donnés (été/hiver). Cela nous a permis de constituer une base de connaissances indiquant l'évaluation donnée aux différents services par chaque utilisa-

3. Voir description sur `https://www.kaggle.com/`

teur pour chaque contexte. Nous évaluons ce jeu de données à la fois avec le vecteur complet (*vc*) et avec le vecteur tronqué (*vt*). Afin d'obtenir un vecteur tronqué (*vt*) du contexte, nous avons retiré du contexte les préférences et les centres d'intérêts des utilisateurs. Cela permettra de comparer l'impact de la perte d'information contextuelle induite entre ce jeu de données issu du monde réel (*RS-ASM (vt)*) et nos jeux de données artificielles (*Contrôle (vt)* et *YSE (vt)*). D'autre part, nous évaluerons également ce jeu de données en version non-stationnaire (*ns*). Toutes les $50\ 000$ itérations nous modifions la distribution des récompenses pour chaque bras et ce en basculant de la version « *été* » à la version « *hiver* » du jeu de données sans donner accès aux dimensions de « *saison* » dans le vecteur de contexte ;

— ***Food***[4] est un jeu de données spécifiquement employé pour les systèmes de **recommandation**, utilisé et distribué par Hideki Asoh [Ono+09]. *Food* est plus particulièrement appliqué à de la recommandation de restaurant à des utilisateurs. Ce jeu de données et originalement fourni avec de la parcimonie dans les retours utilisateurs c.-à-d., chaque utilisateur n'a pas fait de retour sur l'ensemble des restaurants existants. Dans cette étude préliminaire, pour des raisons de simulation et d'observation sur l'exhaustivité du couple utilisateur-contexte et de son évaluation vis à vis de l'ensemble des classes (restaurants), nous avons comblé ces informations manquantes via des techniques de filtrage collaboratif et une mesure de similarité de cosinus utilisateur-utilisateur (voir le détail de cette méthode à la Section 1.4) ;

— ***Covertype*** et ***Poker Hand*** provenant d'*UCI Machine Learning Repository*[5] sont quant à eux des jeux de données permettant de monter à l'**échelle** dans nos expériences tant en termes de nombre d'instances que de nombre de variables de contexte. ***Covertype*** associe des types de plantes et plus particulièrement des arbres à des variables cartographiques. En ce qui concerne les systèmes de recommandation, nous pourrions considérer qu'il s'agit ici de recommander un type d'arbres à un emplacement selon ses caractéristiques cartographiques. ***Poker Hand*** quant à lui est un jeu de données associant des caractéristiques de mains de poker à la prédiction de la valeur de ces mains (c.à.d., paire, brelan, suite, full, etc.). Les mains de poker sont décrites selon les attributs des cartes qui la compose c.à.d., l'enseigne de chacune des 5 cartes (coeur, pique, carreau, trèfle) et la valeur de chacune des 5 cartes (as, roi, dame, valet, 10, etc.). Il s'agit donc ici de recommander (prédire) la valeur d'une main de poker en fonction des attributs des cartes qui la composent ;

— ***Jester*** provenant d'*UC Berkeley*[6] est un jeu de données pour de la recommandation de blagues et qui représente le cas non contextuel dans notre étude c.-à-d., aucune information de contexte n'est disponible ;

— Enfin, les jeux de données ***Mushroom***, ***Statlog***, ***Yeast***, ***Adult***, et ***Students Academics Perf.*** provenant d'*UCI Machine Learning Repository* seront considérés pour de

4. AIST context-aware food preference dataset
5. Voir descriptions sur `http://archive.ics.uci.edu/ml/`
6. Voir description sur `http://eigentaste.berkeley.edu/dataset/`

la recommandation de type **conseil** à un tiers humain. Ces jeux de données sont intéressants à prendre en compte car la recommandation ne se fait pas directement à l'utilisateur mais plutôt indirectement. Ainsi, dans ***Mushroom*** il pourrait être question de recommander à un pharmacien si un champignon est vénéneux ou comestible. Dans ***Statlog*** il serait possible de recommander à un médecin si un patient a une prédisposition ou non de pathologie cardiaque. Dans ***Yeast***, il serait question de recommander à un biologiste le site de localisation cellulaire spécifique d'une protéine selon ses attributs. Dans ***Adult*** nous pourrions imaginer recommander de potentiels prospects à une entreprise de courtage recherchant des clients supposés, touchant un salaire suffisant en considération d'un seuil pré-défini (ici $50\ 000$ euros par an). Enfin dans ***Students Academics Perf.*** il serait possible de recommander un étudiant à une école en fonction de ses résultats et de la prédiction du niveau de réussite qu'il aurait dans cette école.

Jeux de données	Nombre d'instances	Variables catégorielles	Variables binaires	Nombre de bras	Source des données	Type de jeu de données
Contrôle	1 000	4	4	4	Généré	Artificiel
YSE	1 000	2	4	2	Généré	Artificiel
RS-ASM	2 152	8	56	18	*Kaggle*	Recommandation
Food	424	80	270	20	*AIST*	Recommandation
Yeast	1 484	9	28	10	*UCI M.L.*	Conseil
Adult	48 842	14	121	2	*UCI M.L.*	Conseil
Mushroom	8 124	22	126	2	*UCI M.L.*	Conseil
Statlog	270	13	46	2	*UCI M.L.*	Conseil
Students Academics Perf.	131	22	73	3	*UCI M.L.*	Conseil
Covertype	581 012	54	95	7	*UCI M.L.*	Échelle
Poker Hand	1 025 010	11	85	9	*UCI M.L.*	Échelle
Jester	24 983	0	0	100	*UC Berkeley*	Recommandation

TABLE 4.1 – Jeux de données de nos simulations (Versions *vc*)

Jeux de données	Nombre d'instances	Variables Catégorielles	Variables Binaires	Nombre de bras	Source des données
Contrôle	1 000	3	3	4	Artificiel
YSE	1 000	1	2	2	Artificiel
RS-ASM	2 152	5	25	18	*Kaggle*

TABLE 4.2 – Jeux de données de nos simulations (Versions *vt*)

4.4.2 Processus de simulation

Dans cette sous-section, nous décrivons notre processus de simulation en termes de : paramétrage d'un horizon T fini ; gestion des retours utilisateurs sous forme de récompenses ;

de choix des tests statistiques que nous avons réalisés afin de vérifier la significativité des résultats expérimentaux obtenus.

4.4.2.1 Convergences et horizons

Dans le cadre des bandits-manchots c'est la preuve et à défaut la mesure de convergence qui sont recherchés [All16 ; Bou+17]. C'est pourquoi les expérimentations qui en sont issues bouclent sur les jeux de données après les avoir mélangés. Ainsi, pour simuler un flux de données d'utilisateurs avec leur contexte se présentant pour recevoir une recommandation, nous sélectionnons séquentiellement et aléatoirement les instances disponibles dans l'ensemble du jeu de données et ce jusqu'à un horizon T défini dès le début de la simulation.

Comme le nombre d'instances de chaque jeu de données est plus ou moins important, nous devons mettre à l'échelle l'horizon T pour chacun d'entre eux afin d'obtenir une mesure suffisante, notamment, de précision individuelle. Nous travaillons en considérant un horizon T fini. Aujourd'hui, il n'y a pas de consensus sur la valeur à donner à T dans le cadre des bandits-manchots. Celle-ci dépend fortement de ce qu'on souhaite mettre en lumière en termes d'expérience (p. ex., impact de la non stationnarité, mesure de la précision individuelle). Néanmoins, en nous basant sur certaines expérimentations de la littérature proche de notre protocole de simulation [Bou+17 ; Li+10], nous proposons ci-dessous un paramétrage spécifique de T en fonction du nombre d'instances de chaque jeux de données nous permettant d'obtenir des valeurs précises pour les trois critères que nous avons défini : précision globale, diversité, et précision individuelle.

Ainsi, pour chaque algorithme, nous simulons 10 expériences de :

— $100\ 000$ itérations concernant les jeux de données possédant très peu d'instances : *Food*, *Statlog*, et *Students Academics Performance* ;
— $200\ 000$ itérations concernant les jeux de données possédant un nombre d'instances moyen (c.-à-d. dans notre cas entre 1000 et 10000 instances) : *RS-ASM* (*vc* et *vt*), *Yeast* ; *Mushroom*, et les jeux de données artificiels (*Contrôle* (*vc* et *vt*) et *YSE* (*vc* et *vt*)) ;
— $400\ 000$ itérations concernant les jeux de données possédant un grand nombre d'instances : *Jester*, et *Adult* ;
— $10\ 000\ 000$ itérations pour les jeux de données possédant un très grand nombre d'instances : *Poker Hand* et *Covertype*.

4.4.2.2 Attribution des récompenses

Pour chaque algorithme, nous nous sommes focalisés sur le bandit de type Bernoulli si bien que pour tout jeu de données la récompense r retournée à chaque itération sera soit égale à 1 si la classe prédite est la bonne, ou 0 sinon.

4.4.2.3 Évaluations et tests statistiques

Pour chaque expérience, nous mesurons et calculons les critères d'évaluation choisis et rappelés précédemment à la Section 4.3, à savoir :

1. La précision globale (Acc) (voir Section 2.4) ;
2. La diversité (Div) (voir Section 4.3.3) ;
3. La précision individuelle (via le calcul de déciles et quartiles, le calcul des *FDC* et histogrammes) (voir Section 4.3.4).

À la lumière des mesures effectuées pour chaque expérience, nous proposerons une analyse de ces résultats dans la Section 4.5.

Plus précisément, nous déterminons les moyennes et écart-types de précisions globales (Acc), de diversité de sélection des bras (Div) sur un ensemble de 10 simulations. De plus, nous calculerons la précision individuelle et déduirons sa *FDC*[7] dont les données sont consultables en annexe dans les Tableaux C.2 à C.5,via le calcul de deux déciles (10% et 90%), de la médiane et de deux quartiles (Q_1 et Q_3). Les mesures de précision individuelle seront également représentées sur les Figure B.1 à B.16 (Annexe B) via les *FDC* et les histogrammes.

Enfin, pour la comparaison des différents algorithmes sur les critères de précision globale et de diversité, nous effectuons des tests statistiques afin d'évaluer la significativité sur l'ensemble des 10 simulations effectuées.

Ainsi, nous réalisons en premier lieu un test de *Kruskal-Wallis* (*KW*) afin de mettre en évidence les inégalités entre les résultats obtenus par chacun des algorithmes, c.-à-d., nous testons l'hypothèse *nulle* H_0 : « *Il n'y a pas de différence significative entre les résultats des algorithmes (médianes)* ».

Si le test de *KW* indique qu'il existe des différences entre les résultats, il sera alors nécessaire de réaliser des tests de rang signés de *Wilcoxon* deux à deux sur la précision globale, c'est-à-dire que nous testons l'*hypothèse nulle* H_0 : « *Il n'y a pas de différence significative entre les résultats entre chaque paire d'algorithmes* ».

Par la suite, nous indiquerons donc : si l'hypothèse nulle est rejetée ou non, et la valeur de p correspondante pour chaque comparaison que nous observerons.

4.4.3 Les différents cas étudiés

Dans le cadre de nos simulations nous étudierons précisément quatre cas :

1. **Cas avec vecteur de contexte complet (*vc*)**, c'est à dire avec la totalité du contexte original du jeu de données. Ces expériences nous permettront de mettre en lumière le fonctionnement « *nominal* » des algorithmes sur plusieurs jeux de données du monde réel, c.-à-d. sans modification de notre part. Nous observerons également les résultats sur plusieurs jeux de données de type : artificiel, spécifique à la recommandation, conseil à un tiers humain, et de montée à l'échelle ;

7. Fonction de Distribution Cumulative (*Cumulative Distribution Function*) c.-à-d., la fonction de répartition de la précision

2. **Cas avec vecteur tronqué *(vt)***, c'est à dire avec une restriction sur le contexte original du jeu de données (voir le détail dans le Tableau 4.2). Ces expériences nous permettront d'étudier l'impact d'une restriction sur le contexte, c'est à dire lorsque des informations contextuelles sont manquantes ;
3. **Cas non stationnaire *(ns)***, c'est à dire avec une évolution dynamique des probabilités de récompenses des bras par période (voir description à la Sous-section 2.2.5). Ces expériences nous permettront de confronter les algorithmes sur un cas réel de non stationnarité appliqué à la recommandation ;
4. **Cas non contextuel**, c'est à dire sans contexte. Ces expériences nous permettront d'envisager le cas où aucune information contextuelle ne serait disponible dans le cadre des recommandations.

4.5 Performance des algorithmes de bandits-manchots pour la recommandation

Dans cette section, nous évaluerons la performance de chaque algorithme de bandits-manchots contextuels et non contextuels sur l'ensemble des jeux de données pour chacun des trois critères de précision globale, diversité et précision individuelle.

4.5.1 Analyses basées sur la précision globale

Le critère de performance de précision globale est celui qui est principalement employé dans toute évaluation d'algorithmes de *MAB* et de *CMAB*.

Dans cette sous-section nous analysons ainsi les résultats de précision globale que nous avons obtenus pour les sept algorithmes de *MAB*/*CMAB* expérimentés (ε-*Greedy*,*UCB2*, *TS*, *EXP3*, *EXP4*, *LinUCB*, *CTS*) et ce pour l'ensemble des jeux de données étudiés dans les différents cas cités précédemment (voir Sous-section 4.4.3).

Afin de faciliter la lecture, nous résumons les analyses sur la précision globale pour chacun des cas et reportons les analyses détaillées en annexe de ce mémoire (Annexe B). De même, nous y reportons également l'ensemble des tableaux de résultats (Annexe C).

4.5.1.1 *MABs* versus *CMABs* sur jeux de données avec vecteur complet

Les résultats de ces expérimentations sont disponibles en annexe au Tableau C.2 et au Tableau C.1. Les analyses détaillées de ces expérimentations sont consultables dans les annexes à la Section B.1. Nous résumons la conclusion de ces analyses ci-après.

Résumé des expérimentions se focalisant sur la précision globale concernant le cas avec vecteur de contexte complet. De manière générale, nous remarquons l'avantage significatif d'utiliser un algorithme de *CMAB* basé sur un modèle linéaire plutôt qu'une méthode de MAB ou de sélection de politiques. Notons que *EXP4.P* dans certains cas ne supportent

pas la montée à l'échelle en termes de nombres d'actions et d'experts (en désaccord). Plus spécifiquement sur les jeux de données de montée à l'échelle, dans le cas de Covertype, nous remarquons également un avantage significatif à utiliser un algorithme de *CMAB* basé sur un modèle linéaire plutôt qu'une méthode de MAB ou de sélection de politiques. Notons néanmoins qu'*EXP4.P* obtient une précision globale supérieure à celles des algorithmes de *MAB*. En revanche, dans le cas de Poker Hand, même si *LinUCB* et *CTS* obtiennent sensiblement de meilleurs résultats qu'*EXP4.P* et que l'ensemble des algorithmes de *MAB*, ces différences ne sont pas statistiquement significatives.

4.5.1.2 *MABs* versus *CMABs* sur jeux de données avec vecteur de contexte tronqué

Pour expérimenter l'effet d'une restriction sur le contexte c.-à-d., avec le vecteur d'origine tronqué, nous avons utilisé trois jeux de données : *Contrôle (vt)*, *YSE (vt)* et *RS-ASM (vt)*. Rappelons que *YSE (vt)* a été construit de telle manière à provoquer un effet *Yule-Simpson* lorsque nous retirons la dimension « pays ». Ceci permet ainsi d'observer les conséquences sur les résultats obtenus pour les algorithmes de *CMAB*.

La restriction de contexte dans les algorithmes de *CMAB* a donné lieu à une expérimentation récente [Bou+17] permettant la sélection intelligente et dynamique de dimensions pertinentes du contexte. Ces travaux concluent entre autres sur l'impact de la restriction sur le contexte en termes de performance si celle-ci est trop importante notamment en environnement non stationnaire : « *Ignoring the context can be better then even considering a small random subset of it.* » [Bou+17]

Dans nos expérimentations, nous observons ainsi l'impact de la restriction de contexte sur les algorithmes de *CMAB*. Plus le contexte fourni en entrée s'appauvrit (devient parcimonieux), plus le cumul des regrets observé est important jusqu'à un seuil où la performance des *CMAB* devient moins importante que celle des *MAB*.

Les résultats de ces expérimentations sont disponibles en annexe au Tableau C.2 et au Tableau C.3. Les analyses détaillées de ces expérimentations sont consultables dans les annexes à la Section B.2. Nous résumons la conclusion de ces analyses ci-après.

Résumé des expérimentions se focalisant sur la précision globale concernant le cas avec vecteur de contexte tronqué. De manière générale, nous remarquons l'avantage significatif d'utiliser un algorithme de *CMAB* basé sur un modèle linéaire plutôt qu'une méthode de *MAB* ou de sélection de politiques. Notons néanmoins qu'*EXP4.P* obtient une précision globale supérieure à celles des algorithmes de *MAB*. En revanche, ces résultats restent à nuancer du fait que le niveau de restriction (troncature) que nous avons appliquée au contexte reste relativement peu important. Plus le vecteur sera restreint, plus la précision globale des algorithmes de *CMAB* diminuera jusqu'à passer sous le seuil de performance des algorithmes de *MAB* [Bou+17].

4.5.1.3 *MABs* versus *CMABs* dans le cas non-stationnaire

Le jeu de données de notre expérimentation sur flux non-stationnaire est *RS-ASM (ns)* où durant l'expérience de $200\ 000$ itérations, nous modifions la distribution des récompenses pour chaque bras toutes les $50\ 000$ itérations (Voir Section 2.2.5). Il y aura donc 4 périodes de stationnarités différentes durant la simulation.

Les résultats de ces expérimentations sont disponibles en annexe au Tableau C.4. Les analyses détaillées de ces expérimentations sont consultables dans les annexes à la Section B.3. Nous résumons la conclusion de ces analyses ci-après.

Résumé des expérimentions se focalisant sur la précision globale, menées sur le jeu de données non stationnaire *RS-ASM (ns)*. Nous remarquons que la non stationnarité a un impact sur la précision globale sur l'ensemble des algorithmes de *CMABs* et ε-*Greedy*. Les algorithmes *EXP3*, *UCB2* et *TS* quant à eux parviennent à contrer la non-stationnarité. Néanmoins, *LinUCB* et *CTS* restent malgré tout plus performants que ces algorithmes sur *RS-ASM (ns)*. Même si *EXP4.P* est un algorithme théoriquement connu pour sa capacité à contrer la non-stationnarité, sur ce jeu de données spécifiquement, il est impossible de le vérifier dû aux mêmes limites qu'évoquées sur la version (*vc*) du jeu de données (nombre d'experts en désaccord et nombre d'actions trop élevées).

4.5.1.4 *MABs* versus *CMABs* sur jeux de données dépourvu de contexte

Le jeu de données de notre expérimentation non contextuelle est *Jester*.

Nous avons voulu illustrer le pire cas pour un algorithme de *CMAB* en observant ce qui se passerait avec 100% de contexte tronqué, c.-à-d., sans information de contexte sur laquelle s'appuyer. Bien naturellement nous nous attendons à ce que les algorithmes de *MAB* soient supérieurs en performance. Néanmoins, nous effectuons cette observation préliminaire car dans la suite de ce mémoire, nous proposerons une contribution à la création de contexte ex-nihilo (voir Chapitre 7 et contribution [Gut+19d]).

Les résultats de ces expérimentations sont disponibles en annexe au Tableau C.5. Les analyses détaillées de ces expérimentations sont consultables dans les annexes à la Section B.4. Nous résumons la conclusion de ces analyses ci-après.

Résumé des expérimentions se focalisant sur la précision globale, menées sur le jeu de données non contextuel *Jester*. Sans surprise, nous remarquons l'avantage significatif d'utiliser un algorithme de *MAB* plutôt qu'une méthode de *CMAB*. L'algorithme qui offre de meilleures garanties expérimentales sur ce jeu de données non contextuel est *TS* suivi d'*UCB2*.

4.5.1.5 Conclusion sur les analyses concernant la précision globale

Lorsque que nous disposons d'informations de contexte suffisamment pertinentes, nous observons que les algorithmes de *CMAB* obtiennent une précision globale significativement supérieure aux algorithmes de *MAB*. En revanche, dans le cas où ces informations de contexte sont trop restreintes, il peut alors devenir plus pertinent d'employer des algorithmes de *MAB*.

Parmi les algorithmes de *CMAB*, on observe que *LinUCB* obtient de meilleurs résultats dans la majorité des cas. Il en est de même pour *TS* pour les algorithmes de *MAB*.

Ces considérations posent la question de choisir le bon algorithme pour le bon jeu de données. Il est d'autant plus important de considérer cette question dans le cadre d'évaluations en ligne. En effet, dans ces applications réelles, le choix du bon algorithme est encore plus difficile à prévoir puisque nous ne possédons aucune connaissance a priori, de la pertinence des informations dont on dispose au regard du problème de recommandation.

De ce fait, la question d'une sélection automatique des algorithmes les plus pertinents prend tout son sens. L'une des contributions de cette thèse repose notamment sur l'idée d'utiliser un porte-feuille d'algorithmes et de sélectionner, par proportion, celui ou ceux qui correspondent le mieux au problème (voir Section 5.3).

4.5.2 Analyses basées sur la diversité

Le critère de diversité n'est pas le plus fréquemment employé dans le cadre d'évaluation d'algorithmes de *MAB* et de *CMAB* en général. En revanche, concernant les systèmes de recommandation, ce critère est l'un des éléments important à prendre en compte afin d'éviter les redondances auprès des utilisateurs [Hu+17], ou encore afin de satisfaire le fournisseur d'élément à recommander (p.ex., dans le cadre de recommandations publicitaires).

Dans cette sous-section nous analysons ainsi les résultats de diversité que nous avons obtenus pour les sept algorithmes de *MAB*/*CMAB* expérimentés (ε-*Greedy*,*UCB2*, *TS*, *EXP3*, *EXP4*, *LinUCB*, *CTS*) et ce l'ensemble des jeux de données étudiés pour les différents cas cités précédemment (voir Sous-section 4.4.3). Notons que pour chaque jeu de données, le test *KW* indique qu'il existe des différences significatives entre les résultats de diversité (Tableaux C.2 et C.1) des différents algorithmes et ce pour chacun des jeux de données ($p < 0,01$, H_0 est rejeté).

Afin de faciliter la lecture, nous résumons les analyses sur la diversité pour chacun des cas et reportons les analyses détaillées en annexe de ce mémoire (Annexe B). De même, nous y reportons également l'ensemble des tableaux de résultats (Annexe C).

4.5.2.1 *MABs* versus *CMABs* sur jeux de données avec vecteur complet

Nous analysons les résultats de diversité que nous avons obtenus pour les sept algorithmes de *MAB*/*CMAB* expérimentés (ε-*Greedy*,*UCB2*, *TS*, *EXP3*, *EXP4.P*, *LinUCB*, *CTS*) et ce sur les douze jeux de données étudiés. Les résultats de ces expérimentations sont disponibles au Tableau C.2 et au Tableau C.1. Les analyses détaillées de ces expérimentations sont consultables dans les annexes à la Section B.5. Nous résumons la conclusion de ces analyses ci-après.

Résumé des expérimentions se focalisant sur la diversité concernant le cas avec vecteur de contexte complet. De manière générale, nous remarquons l'avantage significatif d'utiliser un algorithme de *CMAB* basé sur un modèle linéaire plutôt qu'une méthode de MAB ou de sélection de politiques quand le contexte fourni est suffisamment pertinent ou optimal.

Sans contexte permettant de différencier chaque action à effectuer (c.-à-d., éléments à recommander) selon une situation donnée, les algorithmes de *MAB* peinent à diversifier et restent incapables de recommander un autre élément que celui calculé comme étant le plus optimal pour l'ensemble de la population d'utilisateurs. Néanmoins, si nous devions choisir parmi les algorithmes de *MAB*, ε*-Greedy* offrirait une diversification plus satisfaisante que ses compétiteurs dans la plupart des cas. Ainsi concernant la capacité de diversification des recommandations de chacun des algorithmes, notons que :

— si les caractéristiques du contexte fournies en entrée sont suffisamment pertinentes pour que les algorithmes de *CMAB* basés sur un modèle linéaire puissent diversifier, alors ceux-ci auront un net avantage face à leurs homologues non contextuels ;
— si les caractéristiques du contexte fournies en entrée ne sont pas pertinentes, un algorithme de *MAB* possédant un mécanisme d'exploration aléatoire comme ε*-Greedy* sera plus avantageux afin de diversifier. Notons néanmoins qu'une diversification aléatoire peut coûter lourdement à la précision globale des recommandations.

Enfin, notons que *EXP4.P* bien que supérieur en termes de diversité aux algorithmes de *MAB*, reste dans la majorité des cas nettement inférieur face aux *CMABs* basés sur un modèle linéaire.

4.5.2.2 *MABs* versus *CMABs* sur jeux de données avec vecteur tronqué

L'objectif de ces expérimentations est d'observer l'effet de la restriction de contexte sur la diversité des recommandations produites par les algorithmes de *CMAB*. De même, nous observerons ce que produit l'effet *Yule-Simpson* sur la diversité via le jeu de données *YSE (vt)*.

Les résultats de ces expérimentations sont disponibles en annexe au Tableau C.2 et au Tableau C.3. Les analyses détaillées de ces expérimentations sont consultables dans les annexes à la Section B.6. Nous résumons la conclusion de ces analyses ci-après.

Résumé des expérimentions se focalisant sur la précision globale concernant le cas avec vecteur de contexte tronqué. Nous remarquons de nouveau l'avantage significatif d'utiliser un algorithme de *CMAB* basé sur un modèle linéaire plutôt qu'une méthode de *MAB* ou de sélection de politiques. Sur le critère de diversité, ceux-ci résistent mieux aux contraintes de restriction de contexte et à l'effet *Yule-Simpson*. En revanche, au même titre que pour la précision globale ces résultats restent à nuancer du fait d'un niveau de restriction (troncature) relativement peu important. Plus le vecteur de contexte sera tronqué, plus il y aura un impact négatif sur la diversité des recommandations produites par les algorithmes de *CMAB*.

4.5.2.3 *MABs* versus *CMABs* dans le cas non-stationnaire

Les résultats de ces expérimentations sont disponibles en annexe au Tableau C.4. Les analyses détaillées de ces expérimentations sont consultables dans les annexes à la Section B.7. Nous résumons la conclusion de ces analyses ci-après.

Résumé des expérimentions se focalisant sur la diversité, menées sur le jeu de données non stationnaire *RS-ASM-ns*. Nous remarquons que la non stationnarité n'a aucun

impact sur la diversité pour les algorithmes de *CMAB*. En revanche la non stationnarité a un impact en termes de diversité sur les algorithmes de *MAB* qui doivent ré-explorer et ainsi modifier les bras optimaux qu'ils exploitent entre chaque période stationnaire. Ceci résulte en une plus grande diversification des recommandations effectuées par les algorithmes de *MAB* en environnement non stationnaire. Malgré tout, les algorithmes de *CMAB* restent plus performants en termes de diversité que les algorithmes de *MAB*.

4.5.2.4 *MABs* versus *CMABs* sur jeux de données dépourvu de contexte

Les résultats de ces expérimentations sont disponibles en annexe au Tableau C.5. Les analyses détaillées de ces expérimentations sont consultables dans les annexes à la Section B.8. Nous résumons la conclusion de ces analyses ci-après.

Résumé des expérimentions se focalisant sur la diversité, menées sur le jeu de données non contextuel *Jester*. Nous remarquons que les algorithmes de *CMAB* restent contre-performants lorsqu'ils n'ont plus aucune information de contexte sur laquelle s'appuyer. Concernant les algorithmes de *MAB*, nous relevons qu'*EXP3* permet de mieux diversifier les recommandations par rapport aux autres algorithmes de sa catégorie.

4.5.2.5 Conclusion sur les analyses concernant la diversité

Lorsque que nous disposons d'informations de contexte suffisamment pertinentes, nous observons que les algorithmes de *CMAB* obtiennent une diversité significativement supérieure aux algorithmes de *MAB*. En revanche, dans le cas où ces informations de contexte sont trop restreintes, il peut alors devenir plus pertinent d'employer des algorithmes de *MAB* possédant un mécanisme d'exploration adapté. De plus, *EXP4.P* bien que de même niveau que *LinUCB* et *CTS* sur des jeux de données de artificiels, ne semble pas passer à l'échelle sur des jeux de données du monde réel.

Parmi les algorithmes de *CMAB*, on observe que *LinUCB* et *CTS* obtiennent de meilleurs résultats dans la majorité des cas. Il en est de même pour ε-*Greedy* voire *EXP3* pour les algorithmes de *MAB*.

Ainsi, dans un cadre applicatif, choisir entre *LinUCB* et *CTS* serait cornélien. En effet, *LinUCB* surpasse significativement *CTS* dans un seul cas : RS-ASM (vc). De même *CTS* ne surpasse significativement *LinUCB* que dans un seul cas également : RS-ASM (vt). Dans les autres expérimentations, même s'il y a un avantage non significatif à utiliser *LinUCB*, les deux algorithmes obtiennent des résultats similaires. Notons que pour certains jeux de données – *Food* et *Poker Hand* – *EXP4.P* et ε-*Greedy* dépassent *LinUCB* et *CTS*.

Sur cet aspect de la diversité, ces considérations posent de nouveau la question de choisir le bon algorithme pour le bon jeu de données. Il est d'autant plus important de considérer cette question dans le cadre d'évaluations en ligne où nous ferons face à de véritables utilisateurs pour lesquels la diversité des recommandation prendra tout son sens.

De ce fait, la question d'apporter un mécanisme de diversification aux meilleures algorithmes (comme *LinUCB*) se pose, au même titre que d'utiliser une sélection automatique des

algorithmes les plus pertinents sur les deux critères de précision globale et de diversité. Nous traiterons ces questions aux chapitres 5.

4.5.3 Analyses basées sur la précision individuelle

La précision globale est le critère d'évaluation le plus largement employé afin d'évaluer la performance des algorithmes de recommandation. D'autres critères tout aussi importants ont également été mis en place comme la diversité, la nouveauté ou la sérendipité. Néanmoins à notre connaissance, la précision individuelle n'a jamais fait l'objet d'une étude précise.

L'objectif sous-jacent à l'analyse de la précision individuelle, est d'observer et comprendre comment la précision globale obtenue par chaque algorithme est distribuée au sein des individus.

Dans les systèmes de recommandation où nous faisons face à de véritables utilisateurs, il est d'autant plus pertinent de prendre en compte la précision individuelle.

Dans un système, plusieurs problématiques peuvent survenir lorsqu'on ne prend pas en compte les utilisateurs les plus insatisfaits p. ex., impact image sur les réseaux sociaux, mauvaise évaluation de l'application mobile sur *Google Play*. Ce sont en effet les utilisateurs les plus mécontents qui se manifestent souvent le plus. Partant de ce constat, imaginons un traitement médical pour lequel un algorithme de recommandation obtiendrait 90% de précision globale. Si ce « *bon* » résultat est obtenu au dépend d'une population pour laquelle le traitement n'agit absolument pas, il y a une forte probabilité pour que l'image de ce traitement en soit fortement dégradé tant les individus « *insatisfaits* » le communiqueront publiquement. C'est déjà le cas de nos jours concernant les campagnes « *Antivax* » qui s'appuient sur des résultats d'inefficacité exceptionnelle d'un vaccin, pour en faire une généralité p. ex., campagne de désinformation sur le vaccin de la rougeole qui dépasse pourtant 97% d'efficacité après deux doses.

Cet exemple nous amène à prendre en considération la précision individuelle, et à déterminer une méthode visant à diminuer le nombre de personnes fortement « *insatisfaites* » c.-à-d., pour lesquelles l'algorithme obtient des précisions proches de $0,00$.

Ci-dessus, nous avons évoqué les termes de « satisfaction » et d'« insatisfaction ». Ainsi, afin de faciliter la lecture des analyses sur la précision individuelle, nous décrivons ces deux opposés comme suit :

— les individus obtenant une précision individuelle de plus de $0,75$ seront qualifiés de « satisfaits » c.-à-d., trois recommandations effectuées sur quatre ont obtenu une récompense de la part de l'utilisateur ;
— a contrario, les individus pour lesquels les recommandations effectuées ne correspondent que très rarement, c.-à-d., une précision individuelle inférieur à $0,25$, seront qualifiés d'« insatisfaits » c.-à-d., trois recommandations effectuées sur quatre n'ont obtenu aucune récompense de la part de l'utilisateur.

Bien entendu, pour chaque utilisateur donné, plus la précision individuelle tend vers 0 plus les recommandations qui lui sont faites seront jugées insatisfaisantes. De même, plus la précision

individuelle tend vers 1 plus les recommandations qui lui sont faites seront jugées satisfaisantes.

Avant d'analyser la précision individuelle et de déterminer une méthode permettant de l'améliorer, il est d'abord important de pouvoir et savoir la mesurer. C'est pourquoi, nous avons mis en place une mesure de la précision individuelle que nous avons décrit en Section 2.4. Dans cette section, nous évaluons la précision individuelle pour chaque jeu de données et chaque algorithme à l'aide d'histogrammes et de fonctions de répartition cumulative (FDC). Notons de plus, que ces observations de FDC sont en correspondance avec les valeurs des déciles et quartiles calculées dans les Tableaux C.2 et C.1.

Nous détaillons et analysons plus précisément les résultats obtenus par chaque algorithme pour chaque jeu de données dans l'Annexe B. Afin de faciliter la compréhension des figures représentant la FDC et des tableaux, nous décrivons un mémento d'aide à la lecture des valeurs et des figures en annexe dans la Section B.9.

4.5.3.1 *MABs* versus *CMABs* sur jeux de données avec vecteur complet

Les résultats de ces expérimentations sont disponibles en annexe au Tableau C.1 et au Tableau C.2. Les analyses détaillées de ces expérimentations ainsi que les figures qui les illustrent sont consultables dans les annexes à la Section B.10. Nous résumons la conclusion de ces analyses ci-après.

Résumé des expérimentions se focalisant sur la précision individuelle concernant le cas avec vecteur de contexte complet. Nous remarquons qu'il apparaît avantageux d'utiliser un algorithme de *CMAB* quand le contexte fourni est pertinent ou optimal plutôt qu'une stratégie non contextuelle telle qu'un algorithme de *MAB* qui obtient aux dépens d'une petite proportion d'utilisateurs satisfaits, une très grande proportion d'insatisfaits. Ainsi, nous pouvons résumer nos analyses en trois points :

— si le contexte fourni en entrée est suffisamment pertinent, alors les algorithmes de *CMAB* basés sur un modèle linéaire, sélectionneront les bras optimaux en bonne adéquation avec la situation rencontrée en s'appuyant sur les caractéristiques pertinentes du vecteur de contexte. Ceux-ci auront donc un net avantage face aux méthodes non contextuelles. En effet, les algorithmes de *MAB* ne peuvent se contenter que d'exploiter un bras optimal pour l'ensemble de la population et non pour chaque classe identifiée de la population. *EXP4.P* quant à lui diffère selon le jeu de données : soit il obtient une FDC mieux répartie sur l'ensemble des intervalles de précision ; soit il est totalement contre-performant ;

— si le contexte fourni en entrée n'est pas suffisamment pertinent, alors il n'y aura pas d'avantage à utiliser l'une ou l'autre des méthodes de *MAB* ou de *CMAB*. De plus, si aucune donnée contextuelle fournie en entrée n'est pertinente, alors les algorithmes de *MAB* obtiennent de meilleurs résultats de précision individuelle que les algorithmes de *CMAB* ;

— parmi les algorithmes de *MAB*, il ne semble pas qu'un seul algorithme domine tous les autres en termes de résultats de précision individuelle. Ainsi, ε-*Greedy* est le plus performant sur les jeux de données *RS-ASM (vc)* ou encore *Food*. On préférera *TS* ou *UCB2* sur les jeux de données *Covertype*, *Jester* ou encore *Poker Hand* et enfin *EXP3* sera particulièrement plus performant sur le jeu de données *Yeast*.

4.5.3.2 *MABs* versus *CMABs* sur jeux de données avec vecteur de contexte tronqué

Les résultats de ces expérimentations sont disponibles en annexe au Tableau C.3 et au Tableau C.2. Les analyses détaillées de ces expérimentations ainsi que les figures qui les illustrent sont consultables dans les annexes à la Section B.11. Nous résumons la conclusion de ces analyses ci-après.

Résumé des expérimentions se focalisant sur la précision individuelle concernant le cas avec vecteur de contexte tronqué. Nous remarquons de nouveau l'avantage significatif d'utiliser un algorithme de *CMAB* basé sur un modèle linéaire (voire pour certains cas, basé sur la sélection de politique) plutôt qu'une méthode de MAB. Sur le critère de précision individuelle, ceux-ci résistent mieux aux contraintes de restriction de contexte et à l'effet *Yule-Simpson*. En revanche, au même titre que pour la précision globale et la diversité, ces résultats restent à nuancer du fait d'un niveau de restriction (troncature) relativement peu important. Plus le vecteur sera tronqué, plus il y aura un impact négatif sur la précision individuelle des algorithmes de *CMAB*.

4.5.3.3 *MABs* versus *CMABs* dans le cas non-stationnaire

Les résultats de ces expérimentations sont disponibles en annexe au Tableau C.4. Les analyses détaillées de ces expérimentations ainsi que les figures qui les illustrent sont consultables dans les annexes à la Section B.12. Nous résumons la conclusion de ces analyses ci-après.

Résumé des expérimentions se focalisant sur la précision individuelle, menées sur le jeux de données non-stationnaire *RS-ASM (ns)*. Il sera plus avantageux d'utiliser *LinUCB* qui conserve une bonne répartition de la précision individuelle par rapport aux autres algorithmes. Néanmoins, la précision individuelle obtenue par *LinUCB* en environnement non stationnaire est inférieure à celle qu'il obtient en environnement stationnaire.

4.5.3.4 *MABs* versus *CMABs* sur jeux de données dépourvu de contexte

Les résultats de ces expérimentations sont disponibles en annexe au Tableau C.5. Les analyses détaillées de ces expérimentations ainsi que les figures qui les illustrent sont consultables dans les annexes à la Section B.13. Nous résumons la conclusion de ces analyses ci-après.

Résumé des expérimentions se focalisant sur la précision individuelle, menées sur le jeu de données dépourvu de contexte *Jester*. Nous remarquons que les algorithmes de *CMAB* ne réussissent pas à personnaliser leur recommandation du fait qu'ils n'ont aucune caractéristique de contexte sur laquelle s'appuyer pour y parvenir. En revanche, les algorithmes

de *MAB* obtiennent une meilleure précision individuelle (bien que toujours mal répartie) et sont donc plus appropriés que les algorithmes de *CMAB*. Notons également que *TS* et *UCB2* obtiennent une précision individuelle meilleure que ε-*Greedy* et *EXP3*.

4.5.3.5 Conclusion sur les analyses concernant la précision individuelle

Lorsque que nous disposons d'informations de contexte suffisamment pertinentes, nous observons que les algorithmes de *CMAB* obtiennent une meilleure précision individuelle que les algorithmes de *MAB*. En revanche, dans le cas où ces informations de contexte sont trop restreintes, il peut alors devenir plus pertinent d'employer des algorithmes de *MAB*. De plus, *EXP4.P* bien que de même niveau que *LinUCB* et *CTS* sur des jeux de données de artificiels, ne semble pas passer à l'échelle sur des jeux de données du monde réel.

Parmi les algorithmes de *CMAB*, on observe que *LinUCB* et *CTS* obtiennent de meilleurs résultats dans la majorité des cas. En revanche il n'y a pas de différences significatives entre les algorithmes de *MAB*. Néanmoins, dans certains cas il sera sensiblement plus avantageux d'utiliser ε-*Greedy* au dépend d'une précision globale plus faible. Les résultats obtenus par cet algorithme dépendent fortement du « *jeu de données* » sur lequel on l'évalue.

De plus, dans un cadre applicatif, faire un choix entre *LinUCB* et *CTS* est de nouveau cornélien. En effet, *LinUCB* surpasse *CTS* dans un seul cas : RS-ASM (vc). De même *CTS* ne surpasse significativement *LinUCB* que sur un seul cas : RS-ASM (vt). *CTS* résiste donc mieux à des cas de restriction sur le contexte. Dans les autres expérimentations les deux algorithmes obtiennent des résultats similaires.

Sur cet aspect de la précision individuelle, ces considérations posent de nouveau la question de choisir le bon algorithme pour le bon jeu de données c.-à-d., choisir un algorithme qui obtient une distribution de la précision individuelle la plus « équilibrée » pour l'ensemble de la population. Ici, la notion d'équilibre de la distribution de précision individuelle peut être de différents ordres selon l'interprétation que nous souhaitons en faire et les résultats que nous souhaitons obtenir :

1. **Définition utilitariste.** Dans le cas où nous n'obtenons pas $Acc_u(T) \approx 1,00, \forall u \in U$, on concède le fait d'avoir une proportion non négligeable d'individus très insatisfaits en contrepartie de conserver une proportion de très satisfaits importante ;
2. **Définition statistiques.** On souhaite obtenir un compromis le plus « parfait » possible c.-à-d. une satisfaction moyenne pour l'ensemble de la population inférieure au maximum possible. Ceci correspondrait à une distribution de type *Gaussienne*, centrée sur la moyenne de précision globale [8].

Concernant notre cadre applicatif, nous envisageons un compromis entre les deux définitions ci-dessus dans le cas où nous n'obtenons pas $Acc_u(T) = 1,00, \forall u \in \mathcal{U}$. Nous considérerons donc qu'une distribution est équilibrée lorsqu'on diminue la proportion de très insatisfaits

8. voir *EXP4.P* sur le jeu de données *Food* qui présente une distribution de ce type

($< 0,10$ de précision individuelle), en contrepartie d'une diminution et si possible d'une conservation, de la proportion de très satisfaits ($> 0,9$ de précision individuelle).

Afin de mieux comprendre notre raisonnement, nous étayons nos propos à l'aide d'un exemple sous l'angle de l'éthique. Prenons le cas de la recommandation d'un traitement médical à divers patients. Le système doit-il : 1) Opter pour une stratégie utilitariste c'est à dire sauver la majorité des patients au dépend de certains qu'on ne pourrait guérir ? 2) Administrer un traitement qui ne déplore aucun décès mais qui ne guérit en contrepartie que partiellement l'ensemble de la population en la maintenant en vie ?

Répondriez-vous la même chose si vous connaissiez personnellement l'un de ces patients qui ne pourrait être guéri ?

Il est d'autant plus important de considérer cette question dans notre cas de système de recommandation à des utilisateurs mobiles où nous ferons face à de véritables utilisateurs pour lesquels l'équité pourrait être importante (voir les systèmes de recommandation de groupe [RRS15]).

Ainsi, dans le chapitre 5, la solution que nous mettons en lumière est de rajouter ou favoriser la diversification dans les recommandations afin de diminuer la population de très insatisfaits par sérendipité.

4.6 Synthèse et conclusion du chapitre

Dans ce chapitre, nous avons étudié la performance de différents algorithmes de bandits-manchots contextuels (*CMAB*) et non contextuels (*MAB*) à travers le spectre de trois critères différents :

— la précision globale ;
— la diversité ;
— la précision individuelle.

Dans cette section nous concluons donc par une analyse globale que nous détaillons au Tableau 4.3.

4.6.1 Analyse globale des résultats sur les trois critères

4.6.1.1 Précision globale

Concernant le critère de **précision globale**, il en ressort dans le Tableau 4.3 que soit l'ensemble des algorithmes de *CMABs*, soit *LinUCB* et/ou *CTS* sont plus performants dans le cas où le contexte fourni est suffisamment pertinent. Si le contexte est insuffisant (c.-à-d., trop restreint ou inexistant), alors *TS* et *UCB2* sont les algorithmes les plus performants.

	Précision globale	**Diversité**	**Précision individuelle**
Contrôle (vc)	*CMABs*	*CMABs*	*CMABs*
YSE (vc)	*CMABs*	*CMABs*	*CMABs*
RS-ASM (vc)	*LinUCB*	*LinUCB*	*LinUCB*
Food	*LinUCB*, *CTS*	*EXP4.P*	*LinUCB*, *CTS*
Yeast	*LinUCB*, *CTS*	*LinUCB*, *CTS*	*LinUCB*, *CTS*
Adult	*LinUCB*, *CTS*	*LinUCB*, *CTS*	*LinUCB*, *CTS*
Mushroom	*LinUCB*, *CTS*	*LinUCB*, *CTS*	*LinUCB*, *CTS*
Statlog	*LinUCB*, *CTS*	*LinUCB*, *CTS*	*LinUCB*, *CTS*
Students Academics Perf.	*LinUCB*, *CTS*	*LinUCB*, *CTS*	*LinUCB*, *CTS*
Contrôle (vt)	*CMABs*	*LinUCB*	*LinUCB*
YSE (vt)	*LinUCB*, *CTS*	*LinUCB*, *CTS*	*LinUCB*, *CTS*
RS-ASM (vt)	*LinUCB*, *CTS*	*LinUCB*,*CTS*	*LinUCB*, *CTS*
RS-ASM (ns)	*LinUCB*	*LinUCB*	*LinUCB*
Covertype	*LinUCB*, *CTS*	*LinUCB*, *CTS*	*LinUCB*, *CTS*
Poker Hand	*LinUCB*, *CTS*	ε*-Greedy*	*CMABs* $\sim$ *MABs*
Jester	*TS*, *UCB2*	*CMABs*, *EXP3*	*TS*, *UCB2*

TABLE 4.3 – Meilleurs algorithmes pour chaque jeu de données et pour chaque critère

4.6.1.2 Diversité

Concernant le critère de **diversité**, il en ressort dans le Tableau 4.3 que *LinUCB* est dans la majorité des cas le plus performant. Néanmoins, sur certains jeux de données, c'est l'ensemble des *CMABs* qui obtiennent des résultats satisfaisants (*Contrôle (vc)* et *YSE (vc)*), ou bien *EXP4.P* seul (*Food*), ou encore *LinUCB* et *CTS* seuls (*RS-ASM (vt)*). De même, on observe pour le jeu de données *Poker Hand* qu'ε*-Greedy* reste le plus performant pour diversifier. Enfin sur le jeu de données *Jester*, on observera qu'*EXP3* est le plus adapté pour la diversification.

4.6.1.3 Précision individuelle

Concernant le critère de **précision individuelle**, il en ressort dans le Tableau 4.3 que soit l'ensemble des algorithmes de *CMAB*, soit *LinUCB* et/ou *CTS* sont plus performants dans le cas où le contexte fourni est suffisamment pertinent. Néanmoins, sur *Poker Hand*, il n'y a pas de différence notable entre les algorithmes de *CMAB* et les algorithmes de *MAB*, excepté pour ε*-Greedy* qui est contre-performant sur ce jeu de données. Par contre, si le contexte est insuffisant (c.-à-d., trop restreint ou inexistant), alors *TS* et *UCB2* sont les algorithmes les plus performants à l'instar de ce qui est observé pour le critère de précision globale.

4.6.2 Conclusion

Sur l'ensemble des trois critères, même si on observe dans la majorité des cas un avantage à utiliser *LinUCB* sur l'ensemble de ces jeux de données [9], on observe dans certains cas que d'autres algorithmes offrent un meilleur compromis. C'est notamment le cas pour *CTS*, *TS* et *UCB2*, voire *EXP4.P*, *EXP3* et ε-*Greedy* si on ne se fie qu'au critère de diversité.

Outre les évaluations hors-ligne que nous venons d'effectuer, les cas particuliers restent impossibles à prédire à l'avance dans le cadre d'une évaluation en ligne. Ainsi, choisir quel algorithme employer en amont d'une expérimentation ou d'une application à déployer en ligne en est d'autant plus complexe.

De ce fait, nous envisageons de pallier ce dilemme via deux de nos contributions :

1. Apporter un mécanisme de diversification à un algorithme de *CMAB* basé sur un modèle linéaire afin de le renforcer sur le critère de diversité mais également de précision individuelle (p. ex., en apportant de l'exploration pour gagner en sérendipité et pallier la forte insatisfaction de certains utilisateurs) ;
2. Sélectionner automatiquement les meilleurs algorithmes parmi un ensemble, en utilisant une heuristique de type approche porte-folio d'algorithmes.

Ainsi, au chapitre suivant, nous présenterons deux méthodes permettant de répondre aux enjeux que nous venons d'évoquer :

1. *SW-LinUCB* permettra d'illustrer l'approche apportant un mécanisme de diversification via fenêtre glissante à *LinUCB* ;
2. *Gorthaur* permettra d'illustrer l'approche porte-folio d'algorithmes.

9. Notons cependant que dans la majorité des cas *CTS* offre des résultats aussi compétitifs que *LinUCB*.

CHAPITRE 5

AMÉLIORER LA PRÉCISION INDIVIDUELLE : DIVERSIFIER LES RECOMMANDATIONS

When Diversity Is Needed... But Not Expected !

« *A paradox is remaining in recommender systems : most of recommender systems aim at maximizing the precision of the recommendations, but do not consider human factors, which have an important role in decision processes.* [CBB13]

Sylvain Castagnos, Armelle Brun & Anne Boyer - 2014 »

Sommaire

5.1 Introduction

Ce chapitre fait référence à nos contributions [Gut+18a] et [Gut+19a] (étoile n° 6), ainsi qu'à [Gut+19b] (étoile n° 7) présentées à la Figure 2 en introduction de ce mémoire.

Dans la littérature, la plupart des travaux sur les bandits manchots sont évalués à l'aide d'une mesure de précision globale. Concernant les bandits-manchots contextuels, les approches existantes ont pour objectif d'atteindre une personnalisation individuelle. Ainsi, leur précision globale devrait refléter la précision individuelle pour chacun des utilisateurs. Afin de mesurer le niveau de personnalisation atteint par ces approches, nous avons défini une nouvelle évaluation comparant la précision individuelle des recommandations faites à chaque utilisateur avec la précision globale (voir Chapitre 4). Sur la base de cette comparaison, démontrant des disparités entre la précision individuelle et la moyenne de précision globale, nous proposons *Sliding Window LinUCB (SW-LinUCB)*. *SW-LinUCB* est une combinaison de *LinUCB* (*CMAB*) et d'un mécanisme de diversification pénalisant les bras sélectionnés trop fréquemment. Notre approche, inspirée d'applications réelles, comme les systèmes de recommandation, vise non seulement à atteindre une bonne précision, mais doit aussi tenir compte

de la précision individuelle. Nous expérimentons et discutons nos résultats sur plusieurs jeux de données réels. Cette approche sera présentée à la Section 5.2 du présent chapitre.

Comme évoqué précédemment, les systèmes de recommandation doivent à la fois atteindre une bonne précision globale mais également savoir diversifier leurs recommandations. Malgré des garanties théoriques fortes, on observe que les algorithmes de bandits-manchots obtiennent des résultats différents selon la nature des applications du monde réel ou du jeu de données employé (voir Chapitre 4). De ce fait, avant de choisir un algorithme à appliquer selon ses besoins, il est nécessaire de réaliser une évaluation préliminaire hors-ligne sur les critères de précision globale et si nécessaire de diversité. Or, les systèmes de recommandation sont notoirement difficiles à évaluer en raison de leur nature interactive et dynamique. Pour pallier ce problème, nous avons implémenté une approche portfolio de sélection dynamique d'algorithmes de bandits-manchots pour la recommandation : *Gorthaur*. *Gorthaur* sélectionne les algorithmes avec pour objectifs de maximiser les critères de précision globale et de diversité. Au vu de nos résultats, nous remarquerons que l'intérêt d'utiliser *Gorthaur* est double : 1) Obtenir un compromis entre précision globale et diversité dans le cas où nous ne possédons aucune connaissance a priori sur la nature du jeu de données ou de l'application de recommandation que l'on souhaite déployer ; 2) Rapidement mettre en lumière un ensemble d'algorithmes optimaux. Cette approche sera présentée à la Section 5.3 du présent chapitre.

Ces deux méthodes (*SW-LinUCB* et *Gorthaur*) que nous proposons dans ce chapitre, permettent d'obtenir des résultats similaires en termes de précision individuelle mais partent d'approches de diversification différentes. Nous conclurons donc en comparant ces méthodes entre elles à la Section 5.4.

5.2 Contribution aux algorithmes de bandits-manchots : *SW-LinUCB*

Cette section fait référence à nos contributions [Gut+18a] et [Gut+19a] (étoile n° 6) présentées à la Figure 2 en introduction de ce mémoire.

Dans cette section, nous re-explicitons nos motivations à diversifier les recommandations via le rappel de mécanismes de diversification existants dans la littérature. Ensuite nous posons notre problème, et nous définissons notre nouvelle approche *SW-LinUCB*.

5.2.1 Mécanismes de diversification intra-algorithmes

Dans les systèmes de recommandation, la diversité est pertinente pour la satisfaction individuelle. La diversification peut aussi trouver son intérêt dans le cadre d'environnements non-stationnaires afin de permettre à l'algorithme de rester à jour et favoriser les observations les plus récentes. À l'aide d'une fenêtre glissante, des algorithmes tels que *SW-UCB* [GM11], ou encore *Windows Thompson Sampling with Restricted Context (Windows TSRC)* [Bou+17] permettent d'atténuer les effets résultant de la non-stationnarité. De plus, pour résoudre ces mêmes problèmes induits par la non-stationnarité et plus particulièrement dans le cadre d'une problématique de bandits de type *restless* [Whi88], il existe une approche utilisant également

une fenêtre glissante et dont l'objectif est de pénaliser les bras qui ont été tirés trop souvent [GLS16]. Cette approche intéressante a inspiré notre proposition.

Ainsi, notre méthode est basée sur la combinaison de l'algorithme original *LinUCB* [Li+10] et l'utilisation d'une fenêtre glissante inspirée de [GLS16].

5.2.2 Énoncé du problème

Soit $\mathcal{U} = \{u_1, ..., u_n\}$ l'ensemble des n agents disponibles dans un problème de bandits, et pouvant par exemple correspondre dans le cadre d'applications réelles, à des utilisateurs ou encore des patients. Inspiré par [LKG16], nous supposons pour chaque bras $a \in A$ et étant donné un contexte $x \in X \subseteq \mathbb{R}^d$, que $\mathcal{U}$ peut être partitionné en un nombre $m_a(x)$ de clusters $\mathcal{U}_{1,a}(x), \mathcal{U}_{2,a}(x), ..., \mathcal{U}_{m_a(x),a}(x)$ d'utilisateurs partageant les mêmes comportements vis à vis des récompenses qu'ils octroient à chaque bras a lorsque dans le contexte x. Faisant maintenant l'hypothèse de l'existence d'un vecteur de contexte optimal $x^* \in X^*$ qui posséderait toutes les caractéristiques pertinentes associées, avec une confiance de 100%, au bras optimal correspondant et cela pour chaque contexte disponible. Comme *LinUCB* suppose une dépendance linéaire entre la récompense attendue d'une action et son contexte tel que $\mathbb{E}[r_{t,a}|x_t] = \widehat{\theta}_a^\top x_t$, alors lorsque x^* est fourni, *LinUCB* converge vers une précision de 100% et offre une personnalisation pour chaque individu. Cela signifie que toute précision individuelle convergera également vers 100%. Cependant, dans les situations réelles, x peut manquer d'informations et rester incomplet pour différentes raisons telles que : un manque d'information sur les caractéristiques des bras, une mauvaise modélisation du contexte c'est-à-dire un contexte spécifié de manière incomplète, des restrictions dues à des problématiques de confidentialité et de protection de la vie privée, un profil mal renseigné, des informations manquantes sur l'environnement de l'utilisateur (par exemple, une localisation temporairement indisponible). Dans les cas où $x \neq x^*$, les algorithmes de *CMAB* doivent faire face à des contraintes de parcimonie dans les données ou d'incomplétude sur les caractéristiques disponibles puisque les caractéristiques de x^* manquantes dans x ne peuvent pas être prises en compte. En effet, avec un vecteur de contexte insuffisamment décrit, les clusters associés à x^* ne seront pas pris en compte par *LinUCB*, qui peut finalement être incapable de tirer le bras optimal pour différentes situations. Les utilisateurs affectés par cette parcimonie vectorielle pourraient donc se retrouver insatisfaits de la sélection des bras qui leur est proposée par *LinUCB*. Cela entraîne une diminution de la précision globale mais également de la précision individuelle ciblant ces utilisateurs. Ces problématiques nous ont conduits à construire une nouvelle approche visant, à la fois, à garder une bonne précision globale et à atténuer la diminution de la précision individuelle. La sous-section suivante présente notre méthode qui utilise un mécanisme de diversification afin de contrer le manque d'information contextuelle et favoriser la sérendipité.

5.2.3 Notre méthode : *Sliding Window LinUCB* (*SW-LinUCB*)

Notre méthode s'intitule *SW-LinUCB* pour *Sliding Window UCB* et emploie un mécanisme de diversification basé sur une fenêtre glissante permettant de pénaliser les bras qui ont été sélectionnés trop fréquemment. Nous décrivons ci-dessous comment nous élaborons la fenêtre glissante et décrivons le fonctionnement de notre algorithme.

Notre Fenêtre Glissante : dans la littérature, les méthodes existantes utilisent soit des mécanismes de type *fenêtre glissante*, soit elles appliquent un coefficient dit de *discount* pondérant ainsi les récompenses obtenues par leurs bras afin de favoriser les observations les plus récentes. Notre nouvelle approche combine *LinUCB* et l'utilisation d'un coefficient de *discount* calculé à partir d'une fenêtre glissante permettant de pénaliser la sélection des bras optimaux (tirés le plus fréquemment), afin de favoriser l'exploration des bras moins optimaux que nous appellerons ici l'*ensemble des bras sous-optimaux*. Ainsi, nous définissons un coefficient de *discount* qui pondère les récompenses cumulées obtenues pour chaque bras tel que $\sum_{t=1}^{T} \gamma_t \; r_{t,a}$ [GLS16]. Avec $\gamma_t = 1 - \frac{Occ_w(a,t)}{w}$ où w correspond à la taille de la fenêtre glissante et $Occ_w(a,t)$ représente le nombre de fois qu'un bras a a été sélectionné durant les t dernières itérations. $Occ_w(a,t) = \#_1(E_{t,a})$ où $E_{t,a} = \{0..(2^{(w+1)} - 1)\}$ représente les w dernières sélections d'un bras a donné p. ex., pour une taille de fenêtre $w = 6$, $E_{t,a} = 101001$ signifie que a a été sélectionné aux itérations $t-6$, $t-4$ et $t-1$.

Néanmoins, même s'il pourrait être intéressant de combiner une telle méthode à *LinUCB*, celle-ci reste un processus mettant en œuvre une mémoire à court-terme qui, dans notre cas, ne permettra pas de diversifier suffisamment. Dans notre cas, nous devons conserver un processus d'élimination des mauvaises solutions sur le long terme tout en diversifiant suffisamment parmi l'ensemble des bras sous-optimaux. Ainsi, nous proposons une nouvelle méthode de calcul du gain basée sur le $p_{t,a}$ originel tel que

$$p_{t,a}^{w} = \gamma_t \widehat{\theta_a}^{\top} x_t + \alpha \sqrt{x_t^{\top} M_a^{-1} x_t} \tag{5.1}$$

où $M_a = (D_a^{\top} D_a + I_d)$. Ce calcul permet à la fois de garder la confiance (élimination à long terme) grâce au *bonus*, et de diversifier suffisamment parmi l'ensemble des bras sous-optimaux en pénalisant temporairement l'espérance calculée des récompenses pour les bras sélectionnés trop fréquemment.

L'algorithme *SW-LinUCB* : l'objectif de *SW-LinUCB* est de déterminer la politique π qui maximise les récompenses cumulées à l'horizon T tandis qu'une fenêtre glissante force la diversification parmi l'ensemble des bras sous-optimaux. Notre hypothèse est la suivante : en fonction du niveau de parcimonie du vecteur de contexte ($x \neq x^*$), diversifier parmi l'ensemble des bras sous-optimaux atténuera la perte de précision individuelle pour les utilisateurs pour lesquels la méthode d'origine obtient une très faible précision. Notre méthode est décrite dans l'algorithme 2.

Algorithme 2 : *Sliding Window LinUCB (SW-LinUCB)*

Données : L'ensemble des k bras $a \in A$ disponibles, $\alpha \in \mathbb{R}^+$, l'horizon T, et l'ensemble des n contextes fixes disponibles X.

$w \leftarrow k$;
pour $t = 1$ *à* T **faire**
  Considérer $x_t \in X$: un utilisateur et son contexte;
  pour tous les $a \in A$ **faire**
    si a *n'a pas encore été sélectionné* **alors**
      $Occ_w(a,t) \leftarrow 0$;
      $M_a \leftarrow I_d$;
      $b_a \leftarrow 0_{d\times 1}$;
    $\widehat{\theta}_a \leftarrow M_a^{-1} b_a$;
    si $t > w$ **alors**
      Calculer $Occ_w(a,t) = \#_1(E_{t,a})$;
    Calculer $p^w_{t,a} = \left(1 - \frac{Occ_w(a,t)}{w}\right) \widehat{\theta_a}^\top x_t + \alpha\sqrt{x_t^\top M_a^{-1} x_t}$ comme défini à l'Équation (5.1);
  Sélectionner le bras $a_t = \arg\max_{a_t \in A}(p_{t,a})$ et observer la récompense r_t retournée par l'utilisateur u;
  $M_{a_t} \leftarrow M_{a_t} + x_t x_t^\top$;
  $b_{a_t} \leftarrow b_{a_t} + r_t x_t$;
  $\forall a \neq a_t$, mettre à jour toutes les sous-séquences $E_{t,a}$ en ajoutant un bit 0;
  Mettre à jour la sous-séquence E_{t,a_t} en ajoutant un bit 1;
  si $t > w$ **alors**
    Réaliser un décalage logique vers la gauche (Left Shift) de $E_{t,a}$ et de E_{t,a_t} ;

5.2.4 Expérimentations et Résultats

5.2.4.1 Jeux de données :

L'évaluation de notre proposition s'est faite sur quatre jeux de données que nous avons déjà employé précédemment (voir Chapitre 4) : *Contrôle*, *RS-ASM*, *Covertype*, et *Poker Hand*. Nous avons choisi d'étudier notre heuristique sur ces jeux de données pour les raisons suivantes :

— le jeu de données *Contrôle* nous permet d'observer les résultats sur un jeu de données simplifié, généré artificiellement et pour lequel nous sommes en mesure de connaître les résultats à l'avance ;
— avec *RS-ASM* nous pouvons observer les résultats sur un cas pratique pour la recommandation
— *Covertype* et *Poker Hand*, nous permettent d'observer le passage à l'échelle.

5.2.4.2 Critères :

Mesure de Précision Globale : nous mesurerons la précision globale comme défini au Chapitre 2 dans la Section 2.4.
Mesure de Diversité : nous mesurerons la diversité des recommandations comme définie au Chapitre 4 dans la Section 4.3.3.
Mesure de Précision Individuelle : nous mesurerons la précision individuelle des recommandations comme définie au Chapitre 4 dans la Section 4.3.4.

5.2.4.3 Comparaison des Algorithmes

Nous comparons notre algorithme *SW-LinUCB* avec les méthodes suivantes : *UCB2* (*MAB*) [ACF02], et *LinUCB* classique (*CMAB*) [Li+10]. Notons que *LinUCB* et *SW-LinUCB* auront la même valeur du paramètre α calculé pour $\delta = 0,1$.

5.2.4.4 Protocole Expérimental

Pour chaque algorithme et pour chaque jeu de données, nous simulons 2 cas différents : 1) Avec le vecteur de contexte complet *(vc)*, 2) Avec une partie tronquée *(vt)* du vecteur de contexte d'origine. Ici, le terme tronqué représente les caractéristiques, sélectionnées, que nous décidons de perdre au début de l'expérience (voir les détails sur les versions *vt* des jeux de données dans le Tableau 4.2 et la description disponible en 4.4.1.3).

Ainsi, pour chacun des différents cas et pour chaque algorithme, nous simulons 20 expériences de $10\ 000\ 000$ d'itérations pour Poker Hand et Covertype, et $100\ 000$ concernant RS-ASM et le jeu de données de contrôle. Comme le nombre d'instances de chaque jeu de données est plus ou moins important, nous devons mettre à l'échelle l'horizon T pour chacun d'entre aux afin d'obtenir une mesure suffisante de la précision individuelle.

De plus, pour simuler un flux de données d'utilisateurs se présentant pour recevoir une recommandation (voir ligne 3 de l'algorithme 2), nous sélectionnons séquentiellement et aléatoirement les contextes disponibles dans l'ensemble du jeu de données. Ensuite, nous déterminons les moyennes et écart-types de précision globale et de diversité de sélection des bras sur l'ensemble des 20 simulations. De plus, nous calculons la précision individuelle et déduisons sa FDC dont les données et la représentation sont représentées Figures 5.1, 5.2, 5.3, 5.4, et Tableau 5.1. Enfin, nous réalisons un test de *Kruskal-Wallis* pour vérifier l'inégalité des moyennes obtenues sur les critères observés sur l'ensemble des algorithmes, puis nous complétons ces tests par des comparaison deux à deux en réalisant des tests de rang de *Wilcoxon* pour mettre en évidence la significativité statistique de ces inégalités.

5.2.4.5 Analyse Globale

Les analyses ci-dessous s'appuient sur les résultats présentés dans la Tableau 5.1 dont les FDCs sont illustrées Figures 5.1, 5.2, 5.3 et 5.4.

		Mesures globales		Distribution de la Précision Individuelle				
		Acc	Div	10%	Q_1	Med	Q_3	90%
UCB2	*Contrôle*	$0,25_{\pm 0,001}$	$0,59_{\pm 0,03}$	$0,00_{\pm \varepsilon}$	$0,11_{\pm 002}$	$0,28_{\pm 0,03}$	$0,38_{\pm 0,02}$	$0,46_{\pm 0,03}$
	RS-ASM	$0,59_{\pm 0,04}$	$0,04_{\pm 0,004}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,50_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	Poker Hand	$0,5_{\pm \varepsilon}$	$10^{-4}{}_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,89_{\pm 0,02}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	Covertype	$0,49_{\pm 0,002}$	$10^{-4}{}_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
LinUCB (*vc*)	*Contrôle*	$1,0_{\pm \varepsilon}$	$0,997_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	RS-ASM	$0,78_{\pm \varepsilon}$	$0,86_{\pm \varepsilon}$	$0,04_{\pm 0,02}$	$0,77_{\pm 0,02}$	$0,95_{\pm \varepsilon}$	$0,99_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	Poker Hand	$0,53_{\pm \varepsilon}$	$0,06_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,90_{\pm 0,01}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	Covertype	$0,72_{\pm \varepsilon}$	$0,44_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
SW-LinUCB (*vc*)	*Contrôle*	$0,991_{\pm \varepsilon}$	$0,997_{\pm \varepsilon}$	$0,98_{\pm \varepsilon}$	$0,99_{\pm \varepsilon}$	$0,99_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	RS-ASM	$0,76_{\pm \varepsilon}$	$0,88_{\pm \varepsilon}$	$0,06_{\pm 0,02}$	$0,68_{\pm 0,01}$	$0,92_{\pm 0,01}$	$0,98_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	Poker Hand	$0,48_{\pm \varepsilon}$	$0,34_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,22_{\pm 0,02}$	$0,50_{\pm \varepsilon}$	$0,72_{\pm 0,02}$	$0,87_{\pm 0,02}$
	Covertype	$0,69_{\pm \varepsilon}$	$0,47_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,40_{\pm \varepsilon}$	$0,89_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
LinUCB (*vt*)	*Contrôle*	$0,749_{\pm \varepsilon}$	$0,88_{\pm 0,07}$	$0,35_{\pm 0,09}$	$0,52_{\pm 0,04}$	$0,88_{\pm 0,04}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	RS-ASM	$0,629_{\pm \varepsilon}$	$0,33_{\pm 0,01}$	$0,01_{\pm \varepsilon}$	$0,06_{\pm 0,01}$	$0,92_{\pm 0,01}$	$0,97_{\pm \varepsilon}$	$0,99_{\pm \varepsilon}$
	Poker Hand	$0,50_{\pm \varepsilon}$	$0,01_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,84_{\pm 0,04}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	Covertype	$0,60_{\pm \varepsilon}$	$0,35_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
SW-LinUCB (*vt*)	*Contrôle*	$0,746_{\pm \varepsilon}$	$0,96_{\pm 0,02}$	$0,43_{\pm 0,02}$	$0,50_{\pm \varepsilon}$	$0,82_{\pm 0,01}$	$0,99_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	RS-ASM	$0,567_{\pm \varepsilon}$	$0,69_{\pm \varepsilon}$	$0,08_{\pm 0,01}$	$0,33_{\pm 0,01}$	$0,61_{\pm 0,01}$	$0,85_{\pm 0,01}$	$0,95_{\pm 0,01}$
	Poker Hand	$0,48_{\pm \varepsilon}$	$0,33_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,26_{\pm 0,01}$	$0,50_{\pm \varepsilon}$	$0,67_{\pm \varepsilon}$	$0,85_{\pm \varepsilon}$
	Covertype	$0,56_{\pm \varepsilon}$	$0,41_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,25_{\pm \varepsilon}$	$0,62_{\pm \varepsilon}$	$0,88_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$

TABLE 5.1 – Résultats sur plusieurs jeux de données avec vecteur complet (*vc*) et vecteur tronqué (*vt*)
($\varepsilon = 0,0009$)

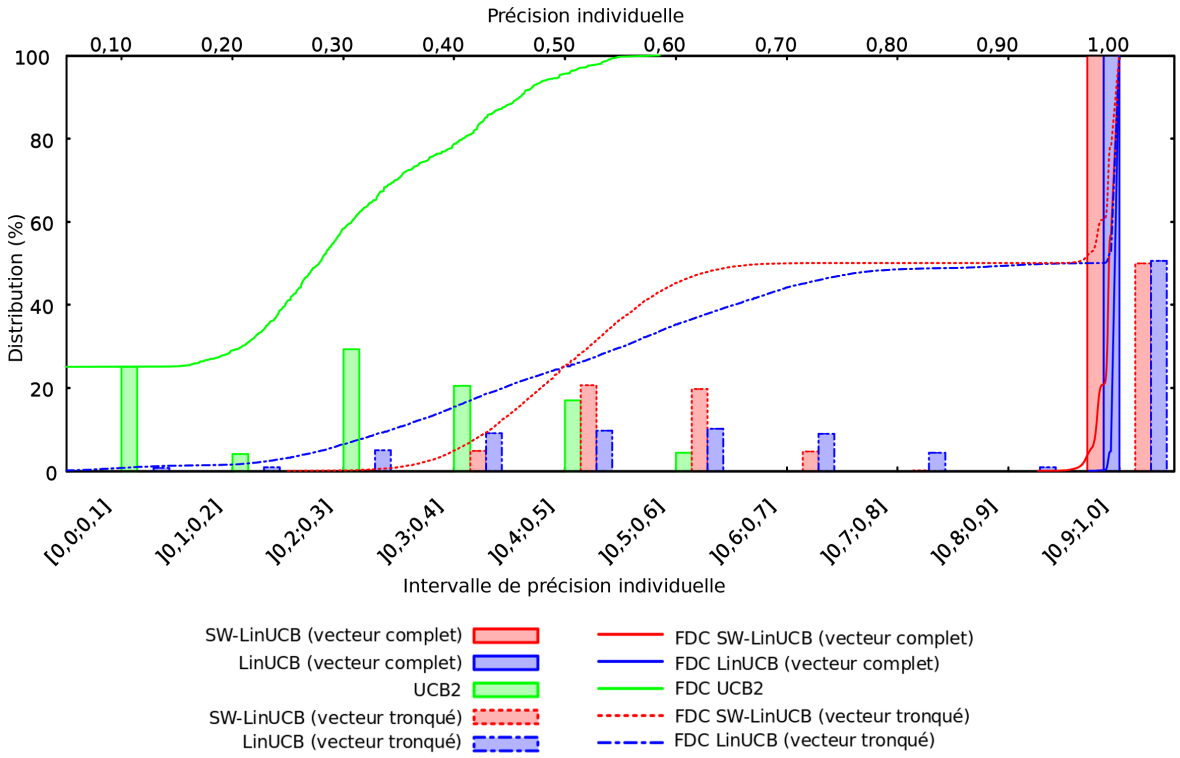


FIGURE 5.1 – Distribution de la précision individuelle pour chaque algorithme sur le jeu de données *Contrôle* (*vc* et *vt*)

Tests Statistiques : un test de *Kruskal-Wallis* (*KW*) pour chaque expérience nous indique qu'il y a une différence significative entre les mesures de précision de chacun des 3 algorithmes ($p < 0,01$). De plus, le test des rangs signés de *Wilcoxon* met en évidence une différence significative entre chaque paire d'algorithmes ($p < 0,01$).

Diversité : en ce qui concerne les expériences sur le jeu de données de contrôle nous observons comme attendu que lorsque nous fournissons un vecteur optimal x^* les deux algorithmes de *CMAB* diversifient à 100%. Néanmoins, pour chaque jeu de données, lorsque nous perdons 25% de l'information du vecteur d'origine, alors la diversité décroît pour les deux algorithmes de *CMAB*. En effet, ils ne réussissent pas à trouver la bonne politique en regard de la règle de correspondance cachée entre récompenses et dimensions du contexte puisqu'une partie pertinente de ce contexte a été tronquée. En revanche, même si *LinUCB* ne diversifie pas autant que dans le cas où nous lui fournissons un vecteur plus complet, *SW-LinUCB* quant à lui offre dans ces mêmes conditions une meilleure diversification que l'algorithme original. Enfin, comme prévu, *UCB2* agit comme un algorithme glouton à bornes supérieures de confiance : il trouve le bras optimal et continue de le tirer. Néanmoins, à la différence d'*UCB1* et grâce à son mécanisme d'exploration, *UCB2* ne tire le bras optimal que durant $[(1+\alpha)^{j_a+1} - (1+\alpha)^{j_a}]$ fois

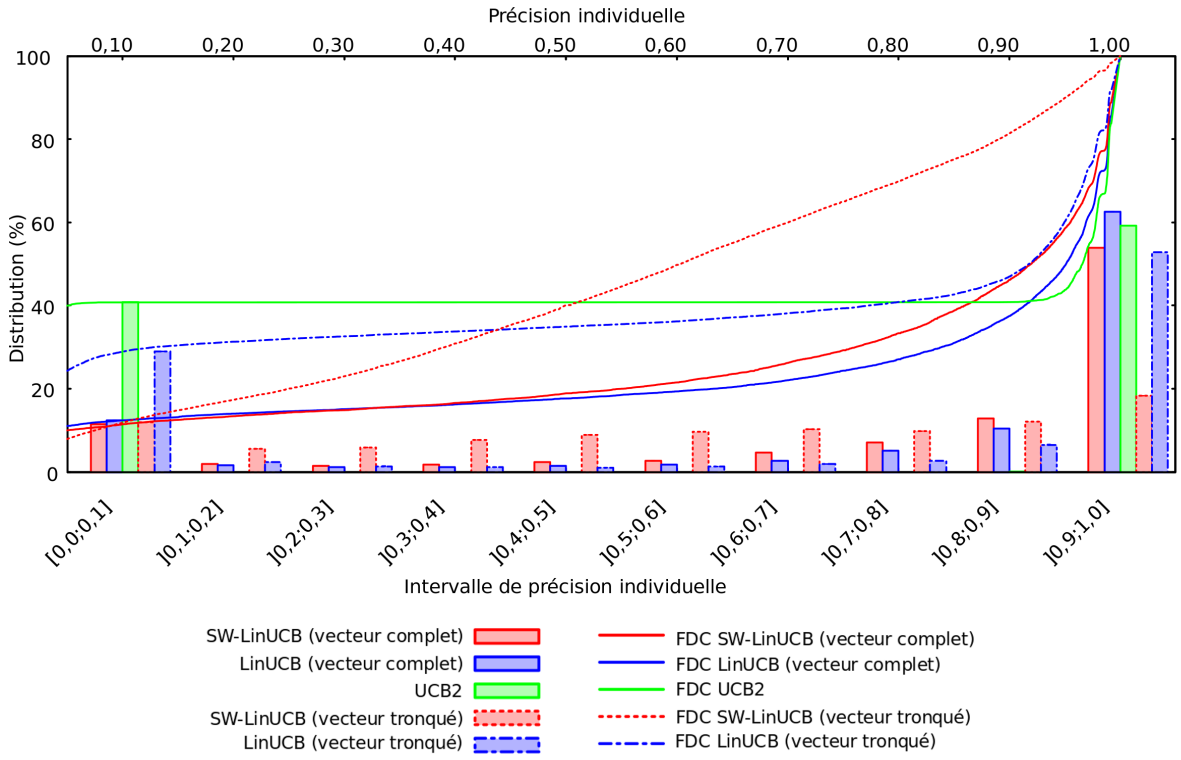


FIGURE 5.2 – Distribution de la précision individuelle pour chaque algorithme sur le jeu de données *RS-ASM* (*vc* et *vt*)

(voir Chapitre 2) avant de terminer l'époque et de choisir un nouveau bras optimal. Ceci résulte en une valeur de diversité obtenue supérieure à *UCB1* dans la majorité des cas et égale sinon. Néanmoins, la diversité obtenue par *UCB2* reste inférieure à celles de *LinUCB* et *SW-LinUCB*.

Précision Globale versus Individuelle : comme attendu, *LinUCB* obtient la meilleure performance globale dans tous les cas et pour tout jeu de données. Sans surprise, la précision globale diminue lorsque nous tronquons le vecteur (*vt*) de contexte, mais il est important de noter que même avec le niveau de restriction sur le contexte choisi dans notre expérience, les algorithmes de *CMABs* restent encore meilleurs que l'algorithme de *MAB* représenté par *UCB2*. Cependant, nous observons dans tous les cas (sauf quand $x = x^*$), que *LinUCB* crée un écart de précision individuelle très important entre les utilisateurs. D'autre part, sur l'ensemble des jeux de données et dans tous les cas (sauf quand $x = x^*$), *SW-LinUCB* perd en précision globale par rapport à *LinUCB* mais en revanche trouve, grâce à son mécanisme de diversification, un meilleur compromis en ce qui concerne la distribution de la précision individuelle. Enfin, pour tous les jeux de données, nous observons que plus l'incomplétude du vecteur de contexte est importante, plus un fossé se crée entre les différentes précisions individuelles d'où résultent distinctement une classe de précisions que l'on peut catégoriser de hautes et une classe de

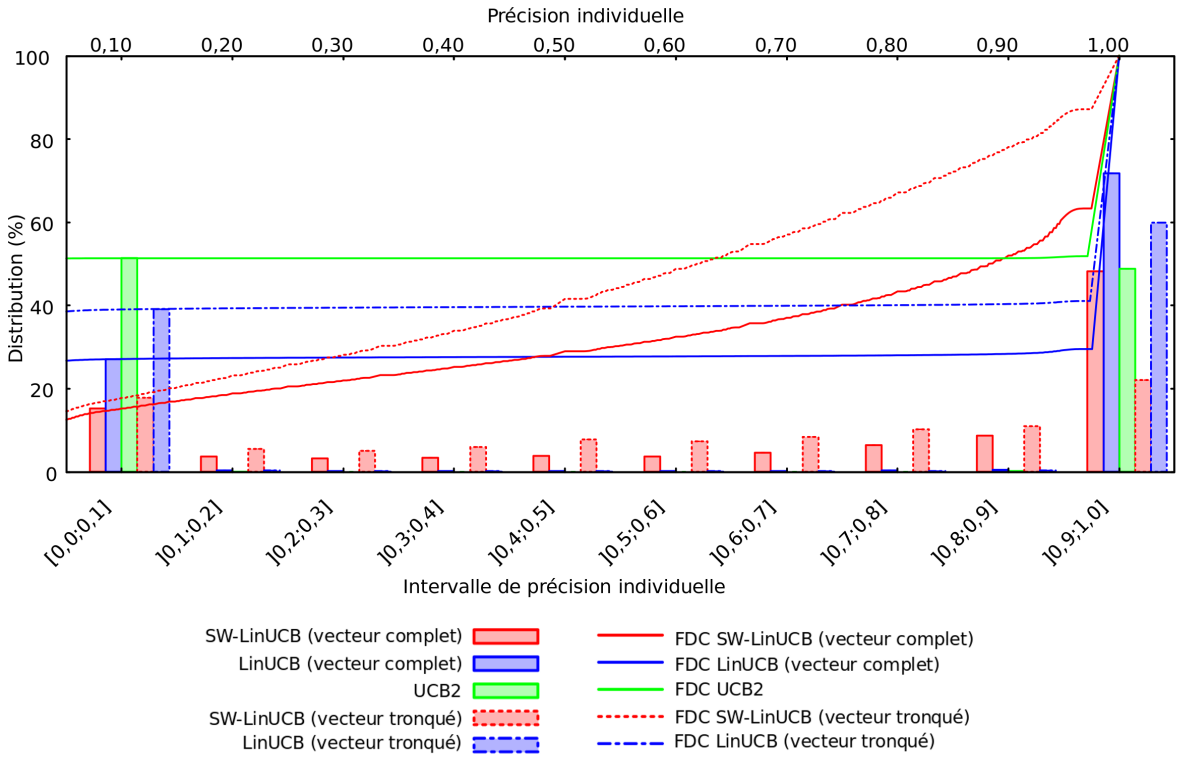


FIGURE 5.3 – Distribution de la précision individuelle pour chaque algorithme sur le jeu de données *Covertype* (*vc* et *vt*)

précisions dites basses. De la même manière, on remarque que plus x tend vers x^*, plus la distribution de la précision individuelle est uniformément répartie parmi les utilisateurs.

5.2.4.6 Analyse Spécifique sur *Covertype*

Diversité : On observe que *SW-LinUCB* diversifie plus (*vc* : $Div = 0,47$, *vt* : $Div = 0,41$) que *LinUCB* (*vc* : $Div = 0,44$, *vt* : $Div = 0,35$) alors que l'algorithme *UCB2* continue de tirer plus fréquement le même bras tout au long des itérations ($Div = 10^{-4}$) avec malgré tout une exploration qui, même si elle existe, ne permet pas de se hisser au niveau des *CMABs*. De plus, nous remarquons que lorsque x tend vers x^*, les caractéristiques fournies en tant que dimension du vecteur de contexte permettent à *LinUCB* et *SW-LinUCB* de diversifier d'avantage.
Précision Globale VS Individuelle : On observe que *LinUCB* conserve une meilleure précision globale (*vc* : $Acc = 0,72$, *vt* : $Acc = 0,60$) que *SW-LinUCB* (*vc* : $Acc = 0,69$, *vt* : $Acc = 0,56$). De plus, il est important d'observer que le niveau d'incomplétude du vecteur de contexte n'est pas encore assez important pour permettre à notre algorithme de *MAB UCB2* d'être plus précis $(0,49)$. En outre, on observe Figure 5.3 et Tableau 5.1, que *SW-LinUCB* reste le meilleur en termes de distribution de la précision individuelle (*vc* : $Q_1 = 0,40$, *vt* : $Q_1 = 0,25$) que *Li-*

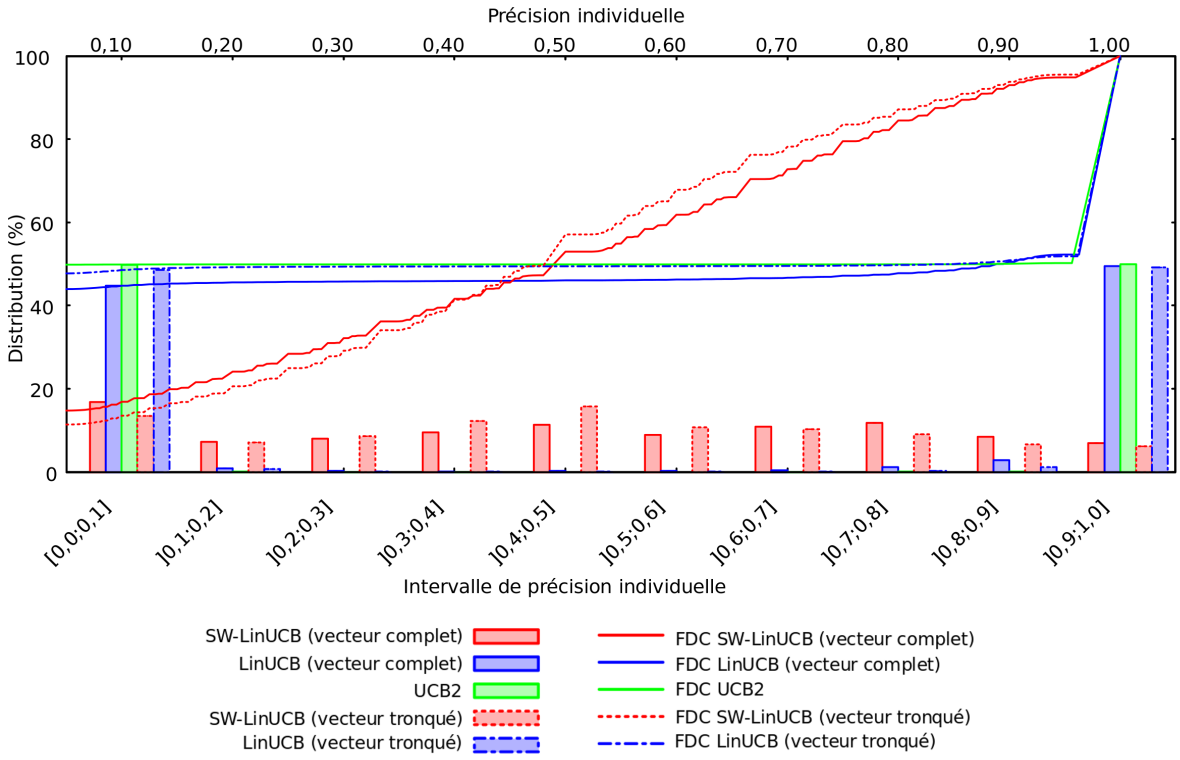


FIGURE 5.4 – Distribution de la précision individuelle pour chaque algorithme sur le jeu de données *Poker Hand* (*vc* et *vt*)

nUCB (*vc* : $Q_1 = 0,00$, *vt* : $Q_1 = 0,00$). Ces derniers résultats montrent que notre mécanisme de diversification permet d'augmenter la précision individuelle de la population obtenant habituellement les valeurs les plus faibles avec la méthode d'origine. Enfin, la comparaison entre les résultats *vc* et *vt* montre que, pour les deux méthodes de *CMAB*, les précisions globales et individuelles sont toutes deux proportionnelles au niveau d'information et de complétude du vecteur contexte.

5.2.5 Conclusion et Perspectives

Dans cette section, nous proposons une nouvelle approche adaptée de l'algorithme original *LinUCB* visant à la fois à améliorer la précision individuelle et à maintenir une bonne précision globale. Nous montrons qu'en privilégiant la diversité, notre algorithme *SW-LinUCB* offre un compromis entre précision globale et individuelle que nous pensons mieux adapté à un certain nombre d'applications du monde réel comme les systèmes de recommandation ou les essais cliniques.

À la section suivante, nous décrivons une autre méthode de diversification des recommandations par la mise en place d'une heuristique, *Gorthaur*, ayant pour objectifs de maximiser à la fois la précision globale et la diversité.

Ceci nous permettra ensuite de comparer ces deux approches : ajout d'un mécanisme de diversification via fenêtre glissante au sein de l'algorithme *LinUCB*, et portfolio d'algorithmes de*MAB* et *CMAB*.

5.3 Approche portfolio d'algorithmes de bandits-manchots : *Gorthaur*

Dans la section précédente, nous avons présenté un mécanisme de diversification via fenêtre glissante sur l'algorithme *LinUCB*. Dans cette section, nous présentons une autre méthode de diversification via une approche portfolio d'algorithmes de *MAB* et *CMAB* ayant pour objectifs de maximiser à la fois la précision globale des recommandations et leur diversité. Cette section fait référence à notre contribution [Gut+19b] (étoile n° 7) présentée à la Figure 2 en introduction de ce mémoire.

5.3.1 Rappel de la méthode portfolio *Compass*

Afin de bâtir notre méthode de sélection sur portfolio d'algorithmes, nous nous basons sur celle du *Compass* [MS08] que nous rappelons dans cette sous-section.

La méthode *Compass* a été créée afin de déterminer les meilleurs opérateurs d'algorithmes génétiques et permettre ainsi leur paramétrage automatique [MS08 ; Mat+09]. Les auteurs de cette méthode ont travaillé sur l'observation de deux critères : ΔQ et ΔD, correspondant respectivement à la variation moyenne de la *qualité* de la population et à la variation moyenne de la *diversité* de la population. À partir de ces deux critères, ils tentent de déterminer un compromis entre Q et D à travers l'usage de divers opérateurs. La compréhension du mode de fonctionnement du *Compass* décrit par la suite peut être facilitée par la lecture de la figure 5.5.

Dans ce travail de recherche utilisant le *Compass*, les auteurs représentent ΔQ sur l'axe des ordonnées et ΔD sur l'axe des abscisses. Ils considèrent un opérateur $i \in [1..k]$ et un numéro de génération t avec d_{it} représenté sur l'axe ΔD, q_{it} représenté sur l'axe ΔQ, et T_{it} le temps d'exécution moyen d'un opérateur i durant ses τ dernières utilisations. Les effets qu'obtient un opérateur sur la population en terme de qualité et de diversité sont ensuite caractérisés par un vecteur $o_{it} = (d_{it}, q_{it})$. Les valeurs obtenues d_{it} et q_{it} sont ensuite normalisées comme suit :

$$d_{it}^n = \frac{d_{it}}{max\{|d_{it}|\}} \text{ et } q_{it}^n = \frac{q_{it}}{max\{|q_{it}|\}}$$

Ceci permet d'obtenir un vecteur normalisé $o_{it}^n = (d_{it}^n, q_{it}^n)$.

Ensuite, les auteurs déterminent un vecteur de référence c, définie selon un angle $\Theta \in [0; \frac{\pi}{2}]$ entre c et l'axe des abscisses (ΔD). Ce vecteur de référence traduit un compromis entre les critères de qualité et de diversité, qui varie selon la valeur de l'angle Θ.

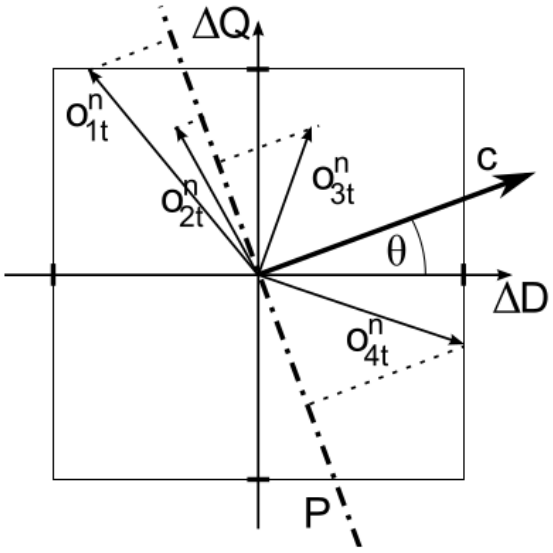


FIGURE 5.5 – Approche du *Compass* selon [MS08]

La sélection de l'opérateur se fait enfin selon une méthode de roulette proportionnelle qui se base sur la capacité (*fitness*) δ_{it} de l'opérateur. La capacité δ_{it} est calculée plus précisément à partir de la projection du vecteur obtenu o^n_{it} sur le vecteur de référence c, c'est-à-dire par exemple $|o_{it}| \cos \alpha_{it}$ où α_{it} correspond à l'angle entre o_{it} et c. Comme dans leur cas d'étude il est possible d'obtenir des valeurs négatives, ils soustraient alors les projections obtenues à la plus petite des projections enregistrées pour un même opérateur. Ils divisent ensuite le résultat de cette soustraction par le temps d'exécution afin de favoriser les opérateurs les plus rapides. Ainsi, ils définissent le calcul de la capacité δ_{it} comme suit :

$$\delta_{it} = \frac{|o^n_{it}| \cos \alpha_{it} - min_i\{|o^n_{it}| \cos \alpha_{it}\}}{T_{it}}$$

où δ_{it} est la capacité (*fitness*) de l'opérateur i à la génération t, et T_{it} le temps moyen d'exécution à l'itération t pour l'opérateur i durant les τ dernières itérations. Ainsi, selon une méthode de sélection de type roulette, ils sélectionnent l'opérateur i avec une probabilité p_{it} définie comme suit :

$$p_{it} = \frac{\delta_{it} + \xi_t}{\sum_{i=1}^{k} \delta_{it} + \xi_t}$$

où ξ_t est une constante permettant à la fois d'éviter les divisions par 0 et d'obtenir un taux minimum utilisable pour la sélection roulette.

Cette méthode possède l'avantage de favoriser les opérateurs dont la distance projetée sur le vecteur de référence c est la plus grande, sachant que c détermine le compromis souhaité entre ces deux critères. On peut donc en déduire que cette méthode favorise la sélection des opérateurs répondant au mieux au compromis fixé par le vecteur de référence c. De ce fait, la

valeur de l'angle Θ est central dans le fonctionnement du *Compass* puisque : 1) quand $\Theta \to 0$ alors on favorisera plutôt l'exploration ; 2) quand $\Theta \to \frac{\pi}{2}$ on favorisera plutôt l'exploitation.

De telles considérations sont très proches des besoins de notre cas d'étude où nous cherchons à déterminer un compromis entre la précision globale et la diversité des recommandations. Dans la sous-section suivante nous décrivons nos motivations.

5.3.2 Motivations

De nos jours, les algorithmes de bandits-manchots sont très largement considérés par de nombreuses applications se heurtant à des problèmes de décision séquentielle comme le sont les systèmes de recommandation (voir Chapitre 2).

Dans le Chapitre 2 nous avons abordé les algorithmes de recommandation selon une approche unitaire, c'est-à-dire en décrivant et en définissant chaque algorithme individuellement. Nous avons observé que l'objectif principal de ces algorithmes était principalement focalisé sur la maximisation de la précision globale. En effet, les algorithmes de bandits-manchots contextuels et non contextuels ont une politique visant à maximiser leurs gains (voir Chapitre 2) c'est-à-dire le cumul des récompenses obtenus à l'horizon T (ou en d'autres termes minimiser le regret).

Or, comme vu au Chapitre 4, dans le cadre des systèmes de recommandation il est intéressant voire incontournable de prendre en compte d'autres critères de performances par exemple la diversité ou encore la précision individuelle.

Indépendamment des preuves théoriques, selon le jeu de données employé nous avons observé au Chapitre 4 que les algorithmes de bandits-manchots contextuels et non-contextuels obtiennent différents résultats sur les critères de précision globale et de diversité qui résultent notamment en une distribution inéquitable de la précision individuelle des recommandations au sein de l'ensemble de la population.

Pour traiter différents critères de performance, il existe des solutions à base de bandits-manchots multi-objectifs [Bus+17] comme *MO-MAB* [DN13 ; Lac17] ou *MOC-MAB* [TT17]. Néanmoins celles-ci agrègent un ensemble de critères de performances en un seul et induisent finalement une résolution mono-objectif.

Par conséquent, les questions qui se posent dans un cadre applicatif sont : comment choisir le bon algorithme correspondant le mieux à mes besoins ? Existe-t-il un algorithme à la fois précis et favorisant la diversité des recommandations ?

De tels constats nous ont amené à considérer une approche portfolio pour la sélection dynamique d'algorithmes de bandits-manchots pour la recommandation. Ainsi, nous avons implémenté une nouvelle approche intitulée *Gorthaur* (**G**eneric-**OR**ien**T**ed **H**euristic **A**lgorithm for **U**ser **R**ecommendation). Celle-ci sélectionne les algorithmes via une roulette proportionnelle visant à maximiser les critères de précision globale et de diversité.

L'approche *Gorthaur*, que nous présentons dans cette section, est une heuristique de gestion d'un portfolio d'algorithmes de bandits-manchots inspirée de la méthode du *Compass* [MS08] que nous avons précédemment décrite. Pour cela, nous explorons le front *pareto* du

couple (précision globale, diversité) obtenu par chacun des algorithmes du porte-feuille pour ensuite sélectionner proportionnellement l'algorithme de *MAB*/*CMAB*.

L'analyse de *Gorthaur* que nous avons effectuée dans cette thèse ouvre des perspectives de meta-sélection d'algorithmes de bandits-manchots jusque là peu expérimentées dans la littérature. En effet, dans le cadre d'une évaluation hors-ligne sur jeu de données ou en ligne via une application réelle, il est souvent difficile voire impossible de savoir à l'avance quel algorithme va le mieux correspondre à la résolution du problème de recommandation. Ainsi, l'utilisation d'un algorithme de type portfolio comme *Gorthaur* peut trouver tout son sens dans le cadre de systèmes de recommandations à base de bandits-manchots sur des applications pour lesquelles on ne connaît pas à l'avance quelle serait la meilleure solution à favoriser/utiliser.

Ainsi, nos trois principales motivations pour implémenter *Gorthaur* sont de pouvoir :

1. Considérer à la fois la précision globale et la diversité ;
2. Utiliser une solution qui reste réellement multi-objectifs c.-à-d., pas de ré-agrégation des objectifs en un seul ;
3. Trouver un compromis et/ou une solution sur des jeux de données multiples qui de nature donnent des résultats différents en fonction des algorithmes de bandits-manchots choisis (c.-à-d., le meilleur algorithme de bandit-manchot à choisir sera différent pour chaque jeu de données).

Nous décrivons précisément notre approche dans la sous-section suivante.

5.3.3 *Gorthaur*

Trois bandits-manchots convergeant sous le ciel,
Sept bandits contextuels dans leurs demeures de pierre,
Neuf heuristiques destinées au trépas,
Un algorithme nommé Gorthaur, Seigneur de l'IA sur son trône,
Au pays de l'apprentissage où s'étend la science des données
Gorthaur :
Un algo pour les gouverner tous,
Un algo pour les trouver,
Un algo pour les amener tous,
Et dans le multicritère les lier.

Outre la ressemblance assumée à l'un des plus fameux roman Fantastique du XX^{ème} siècle [Led72], cette première définition littéraire de *Gorthaur* vulgarise tout à fait son mode de fonctionnement. En effet, le principe de *Gorthaur* est de posséder un porte-feuille d'algorithmes de recommandation et de pouvoir les utiliser pour maximiser le compromis souhaité entre la *précision globale* et la *diversité*. *Gorthaur* permet ainsi de tirer profit du meilleur de chaque algorithme sur ces deux critères en s'inspirant de la méthode du *Compass* décrite précédemment.

Dans cette sous-section, nous rappelons les critères que nous souhaitons maximiser, ensuite nous décrivons notre problème bi-objectifs, la méthode employée pour le résoudre ainsi que les différents paramétrages possibles, et l'algorithme qui en découle : *Gorthaur (**G**eneric-**OR**ien**T**ed **H**euristic **A**lgorithm for **U**ser **R**ecommendation)*.

5.3.3.1 Le problème

Comme rappelé dans la section 5.3.2, dans le cadre des systèmes de recommandation, la précision globale et la diversité peuvent être tout aussi importantes. De ce fait, nous pouvons formaliser le problème sous forme d'optimisation bi-objectifs où il sera question de maximiser à la fois la précision globale (définie au Chapitre 2) et la diversité des recommandations (définie au Chapitre 4).

Notre problème d'optimisation bi-objectifs peut être formulé donc comme suit :

$$\max(acc(t), div(t)) \text{ s.c. } t \in [1, T]$$

5.3.3.2 La méthode

Considérons la variation de précision globale ΔAcc qui sera représentée sur l'axe des ordonnées et la variation de diversité ΔDiv sur l'axe des abscisses. À l'origine, il est possible de paramétrer un vecteur de référence $\vec{c}$ définie selon un angle $\Theta \in [0; \frac{\pi}{2}]$ que fait $\vec{c}$ par rapport à l'axe des abscisses (ΔDiv). Ce vecteur de référence traduit le compromis recherché entre les critères de précision globale et de diversité. Il varie selon la valeur de l'angle Θ (voir Figure 5.6). Ainsi, nous pourrons soit choisir de paramétrer $\vec{c}$ afin de favoriser la précision globale, soit le paramétrer afin qu'il favorise la diversité, soit un équilibre entre les deux selon le front *pareto* obtenu par l'ensemble des algorithmes du porte-feuille (Voir explications à la section 5.3.3.3).

Soient les algorithmes de recommandation $b \in \mathcal{B}$ possédant le même nombre de bras. À chaque itération t, *Gorthaur* choisit un algorithme $b \in \mathcal{B}$ qui sélectionnera un bras a à recommander selon sa propre stratégie. Puis *Gorthaur* mesure en retour la précision globale $acc(b,t)$ de l'algorithme b à l'itération t représentée sur l'axe ΔAcc et sa diversité $div(b,t)$ représentée sur l'axe ΔDiv.

La mesure $acc(b,t)$ correspond à la précision globale moyenne comme rappelé à la section 2.4, c'est-à-dire : $acc(b,t) = \frac{g(b,t)}{t_b}$, où $g(b,t) = \sum_{t=1}^{t} r(b,t)$, c.-à-d., la somme des récompenses obtenues par l'algorithme b à l'itération t, $\forall t \in [1;T]$, et t_b correspond au nombre de fois que l'algorithme b a été sélectionné par *Gorthaur*.

La mesure $div(b,t)$ quant à elle correspond à la diversité de l'algorithme b à l'itération t comme défini section 2.4.

En considérant $N_{b,t} = \{n_{a_1}(b,t), ..., n_{a_k}(b,t)\}$ où $n_{a_j}(b,t)$ correspond au nombre de recommandations effectuées pour le bras a_j à l'itération t par l'algorithme b, nous définissions formellement $c_\nu(N_{b,t}) = \frac{\sigma(N_{b,t})}{\overline{N_{b,t}}}$ où $\overline{N_{b,t}}$ est le nombre moyen de fois qu'un bras a été sélectionné par l'algorithme b à l'itération t, et $\sigma(N_{b,t})$ est son écart-type. Alors, la diversité résultante pour

chaque algorithme b à l'itération t est la suivante [1] :

$$div(b,t) = 1 - \frac{c_\nu(N_{b,t})}{\sqrt{k}}$$

La capacité qu'a un algorithme à réaliser des recommandations à la fois précises et diversifiées est donc ensuite caractérisée par un vecteur $o_{b,t} = (div(b,t), acc(b,t))$. Les valeurs mesurées de $div(b,t)$ et $acc(b,t)$ sont ensuite normalisées comme suit :

$$div^{norm}(b,t) = \frac{div(b,t)}{max\{div(b,t)\}} \tag{5.2}$$

et

$$acc^{norm}(b,t) = \frac{acc(b,t)}{max\{acc(b,t)\}} \tag{5.3}$$

Ceci nous permet d'obtenir un vecteur normalisé :

$$o_{b,t}^{norm} = (div^{norm}(b,t), acc^{norm}(b,t)) \tag{5.4}$$

Ensuite, la sélection des algorithmes se fera également comme pour le *Compass* [MS08], c'est-à-dire selon une méthode de roulette proportionnelle qui s'appuie sur la *fitness* $\delta_{b,t}$ propre à chaque algorithme. La capacité (*fitness*) $\delta_{b,t}$ est calculée plus précisément à partir de la projection du vecteur mesuré et normalisé $o_{b,t}^{norm}$ sur le vecteur de référence $\vec{c}$ (voir Figure 5.6). Il est alors possible de déterminer la capacité $\delta_{b,t}$ pour chaque algorithme $b \in \mathcal{B}$ à l'itération t comme suit :

$$\delta_{b,t} = |o_{b,t}^{norm}| \cos \alpha_{b,t} - min_b\{|o_{b,t}^{norm}| \cos \alpha_{b,t}\} \tag{5.5}$$

Ensuite, à chaque itération t, *Gorthaur* sélectionne un algorithme $b \in \mathcal{B}$ avec une probabilité $p_{b,t}$ définie comme suit :

$$p_{b,t} = \frac{\delta_{b,t} + \xi}{\sum_{i=1}^{|\mathcal{B}|} \delta_{b_i,t} + \xi} \tag{5.6}$$

où $\xi = 2^{-1074}$ est une constante permettant à la fois d'éviter les divisions par 0 et d'obtenir un taux minimum utilisable pour la sélection roulette.

La sélection du bras $a \in A$ se fait alors selon la stratégie de l'algorithme b choisi par *Gorthaur*. Enfin, on observe la récompense r_t et on met à jour le ou les éléments correspondant à la stratégie de l'algorithme sélectionné (par exemple la moyenne des bras pour ε-*Greedy* [SB98], le vecteur de réponse et la matrice de covariance pour *LinUCB* [Li+10]).

La méthode du *Compass* appliquée à notre cas possède l'avantage de favoriser les algorithmes dont la distance projetée sur le vecteur de référence $\vec{c}$ est la plus grande, sachant que $\vec{c}$ détermine le compromis souhaité entre les deux critères (précision et diversité). On peut donc en déduire que cette méthode favorise la sélection des algorithmes répondant au mieux au compromis fixé par le vecteur de référence $\vec{c}$. De ce fait, la valeur de l'angle Θ est cen-

1. Voir définition générique du critère de diversité page 126

trale dans le fonctionnement de *Gorthaur* puisque quand $\Theta \to 0$ alors on favorise plutôt la diversification. A contrario, lorsque $\Theta \to \frac{\pi}{2}$ on favorise la précision.

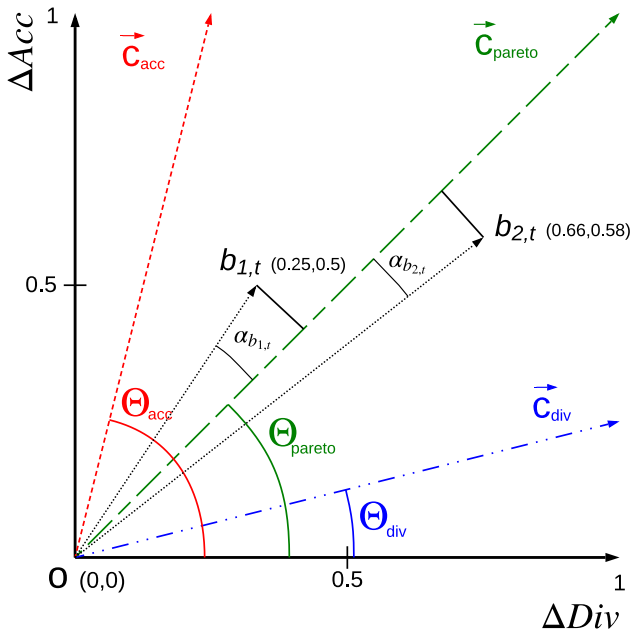


FIGURE 5.6 – Fonctionnement de *Gorthaur* selon le vecteur de référence $\vec{c}$ choisi et deux algorithmes (b_1 et b_2) avec exemple de projections sur le vecteur $\vec{c}_{pareto}$

5.3.3.3 Valeurs de Θ statiques ou dynamiques

Via notre méthode, nous pouvons soit décider d'une valeur statique de Θ, fixée dès le début, soit estimer le front *pareto* à chaque itération t du couple (précision globale, diversité) obtenu par le porte-feuille d'algorithmes et de ce fait re-paramétrer dynamiquement Θ.

Valeur de Θ fixe. Θ pourra être paramétré selon les besoins de l'application : soit avantager la précision globale avec une valeur de Θ proche de $\frac{\pi}{2}$; soit avantager la diversité avec une valeur de Θ proche de 0. Par la suite, nous exprimerons les valeurs de l'angle Θ en pourcentage de $\frac{\pi}{2}$ afin d'exprimer le ratio entre l'équilibre précision globale et diversité. Par exemple, $66,66\%$ de $\frac{\pi}{2}$ correspond à un angle de $\frac{\pi}{3}$ ($60°$) équivalent donc à avantager la précision globale de $+16.66\%$ et désavantager la diversité de -16.66%.

Valeur de Θ recalculée. Si on n'a aucune prédisposition spécifique à avantager soit le critère de précision, soit le critère de diversité, alors on peut laisser l'algorithme calculer Θ à l'*équilibre* de l'ensemble du porte-feuille d'algorithmes. Pour cela, on initialise Θ_{t_0} à $\frac{\pi}{4}$, puis, à chaque itération $t \in T$, *Gorthaur* recalcule l'angle Θ_t du front *pareto* (ligne 15 de l'algorithme 3), comme suit (voir Figure 5.7) :

1. Déterminer $o_{b_i,t} = \arg\max_{b \in \mathcal{B}} acc(b,t)$, c.-à-d., le vecteur obtenu par l'algorithme ayant la précision globale la plus forte ;
2. Déterminer $o_{b_j,t} = \arg\max_{b \in \mathcal{B}} div(b,t)$„ c.-à-d., le vecteur de référence de l'algorithme obtenant la diversité la plus forte ;
3. Calculer $\beta = \arccos \frac{\vec{o}_{b_i,t} \cdot \vec{o}_{b_j,t}}{||\vec{o}_{b_i,t}|| \, ||\vec{o}_{b_j,t}||}$ correspondant à l'angle entre les vecteurs $\vec{o}_{b_i,t}$ et $\vec{o}_{b_j,t}$;
4. Calculer $\gamma = \arccos \frac{\vec{o}_{b_j,t} \cdot \vec{v}_{div}}{||\vec{o}_{b_j,t}||}$ correspondant à l'angle entre l'axe des abscisses (diversité) et le vecteur $\vec{o}_{b_j,t}$, où $\vec{v}_{div}$ est le vecteur unitaire de référence de diversité tel que $\vec{v}_{div} = (1,0)$;
5. Calculer le nouveau :

$$\Theta_t = \frac{\beta}{2} + \gamma \tag{5.7}$$

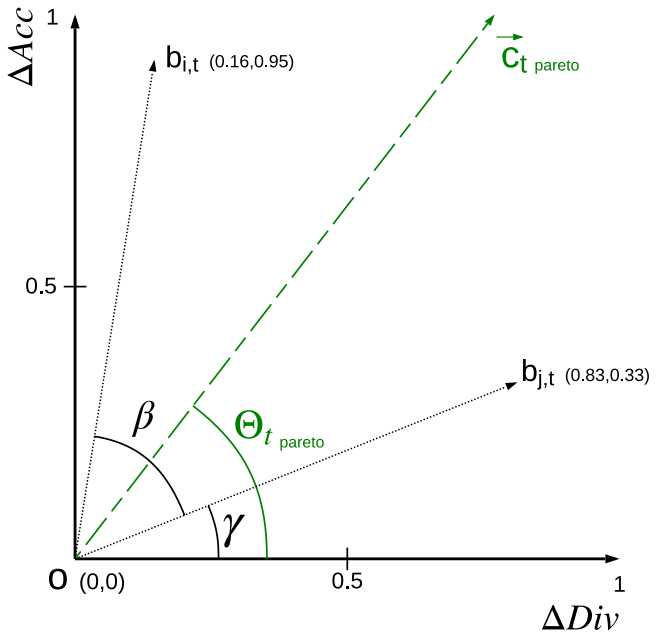


FIGURE 5.7 – Illustration du calcul de Θ_{pareto} et du vecteur $\vec{c}$ associé

5.3.3.4 L'algorithme

L'objectif de *Gorthaur* est de sélectionner les algorithmes dont ils disposent, proportionnellement à leur capacité de correspondre au mieux aux deux objectifs de performance : précision globale et diversité. Un système de recommandation utilisant *Gorthaur* fonctionne selon la description de l'algorithme 3.

Algorithme 3 : *Gorthaur* pour la recommandation

Données : La liste des utilisateurs $u \in U$ et leur contexte $x \in X$, la liste des k éléments à recommander associés aux bras $a \in A$, le porte-feuille d'algorithmes $b \in \mathcal{B}$, l'angle Θ selon la stratégie choisie (voir le paragraphe 5.3.3.3 pour les valeurs de Θ conseillées).

Initialisation de $\delta_{b_{i,1}} = 1$ et $t_{b_i} = 0$ pour $i = 1, .., |\mathcal{B}|$;
pour $t = 1$ *à* T **faire**
  Sélectionner aléatoirement un utilisateur $u_t \in U$ et son contexte $x_t \in X$;
  pour tous les $b_i \in \mathcal{B}$ **faire**
    Calculer $p_{b_{i,t}} = \frac{\delta_{b_{i,t}} + \xi}{\sum_{j=1}^{|\mathcal{B}|} \delta_{b_{j,t}} + \xi}$;
  Choisir l'algorithme b aléatoirement selon les probabilités $p_{b_{1,t}}, ..., p_{b_{|\mathcal{B}|,t}}$ *(Roulette proportionnelle)* comme défini à l'Équation (5.6);
  $t_b = t_b + 1$;
  Choisir l'élément $a \in A$ selon la stratégie de l'algorithme b sélectionné précédemment et recommander cet élément à l'utilisateur u_t;
  Observer la récompense r_t retournée ;
  Mettre à jour les paramètres de l'algorithme b sélectionné selon sa stratégie de traitement de la récompense;
  Mettre à jour $acc(b, t)$ selon l'Équation (5.3);
  Mettre à jour $div(b, t)$ selon l'Équation (5.2) ;
  Mettre à jour $o_{b,t}^{norm}$ selon l'Équation (5.4);
  Mettre à jour $\delta_{b,t}$ selon l'Équation (5.5);
  Re-calculer l'angle Θ *selon l'Équation (5.7) décrite dans la Section 5.3.3.3 (Uniquement dans le cas d'une stratégie de calcul dynamique de* Θ*);*

5.3.4 Expérimentations et Résultats

5.3.4.1 Jeux de données

L'évaluation de notre proposition s'est faite sur cinq jeux de données que nous avons déjà utilisées au Chapitre 4 (voir tableau 4.1) et pour lesquels nous n'avons pas obtenu de résultats démontrant la dominance systématique d'un algorithme par rapport aux autres (voir Tableau 4.3). Ils se constituent donc comme des jeux de données idéaux pour évaluer notre méthode de sélection d'algorithmes. Ces cinq jeux de données sont : *Contrôle*, *RS-ASM* (*vc* et *vt*), *Food*, *Poker Hand*, et *Jester*.

5.3.4.2 Porte-feuille d'algorithmes

Notre cas d'étude porte sur les systèmes de recommandation à base de bandits-manchots. Nous avons pré-sélectionné les algorithmes du porte-feuille de *Gorthaur* (voir Chapitre 2) en fonction de deux critères : 1) le niveau de précision et de personnalisation (c.-à-d., borne du

regret, prise en compte du contexte et/ou capacité d'exploration) ; 2) l'applicabilité et la capacité de l'algorithme à s'inscrire dans des contraintes temps-réels (complexité et besoin en CPU).

Ainsi, *Gorthaur* disposera des algorithmes suivants :

— **Bandits-manchots (*Multi-Armed Bandits - MAB*) :** *UCB2* [ACF02], ε*-Greedy* [SB98 ; Wat89] avec ε fixe, *Thompson Sampling* (*TS*) [AG12] et *EXP3* [Aue+02] ;

— **Bandits-manchots contextuels (*Contextual Multi-Armed Bandits - CMAB*) :** *LinUCB* [Li+10], *Contextual Thompson Sampling* (*CTS*) [AG13] et *EXP4.P* [Aue+95].

5.3.4.3 Rappel de l'étude préliminaire

Dans le chapitre 4, nous avons mené une étude préliminaire pour évaluer manuellement chaque algorithme et chaque jeu de données. Les résultats que nous avons obtenus pour les jeux de données sur lesquels nous nous focalisons sont présentés dans le Tableau 4.3. Il y a un avantage certain à utiliser un sélecteur automatique d'algorithme quand on ne sait pas à l'avance lequel sera le meilleur. Ainsi, afin d'expérimenter *Gorthaur*, nous avons retenu les jeux de données (Tableau 5.2) pour lesquels il n'existe pas un **unique** algorithme dominant tous les autres et ce sur chaque critère. Le Tableau 5.2 est une extraction du Tableau 4.3 (voir Chapitre précédent) et se focalise sur les résultats de précision globale et de diversité. Il permettra ainsi de confronter les résultats obtenus par la sélection automatique de *Gorthaur*.

	Précision globale	**Diversité**
Contrôle	*CMABs*	*CMABs*
RS-ASM (vc)	*LinUCB*	*LinUCB*
Food	*LinUCB*, *CTS*	*EXP4.P*
RS-ASM (vt)	*LinUCB*, *CTS*	*LinUCB*, *CTS*
Poker Hand	*LinUCB*, *CTS*	ε*-Greedy*
Jester	*TS*, *UCB2*	*CMABs*, *EXP3*

TABLE 5.2 – Meilleurs algorithmes pour chaque jeu de données et pour chaque critère

5.3.4.4 Protocole Expérimental

Simulation numérique. Concernant la simulation numérique nous emploierons la même méthode qu'au Chapitre 4, c.-à-d., pour simuler un flux de données d'utilisateurs avec leur contexte se présentant pour recevoir une recommandation, nous sélectionnons séquentiellement et aléatoirement les instances disponibles dans l'ensemble du jeu de données et ce jusqu'à un horizon T défini dès le début de la simulation. Pour chaque algorithme, nous simulons : $100,000$ itérations pour les jeux de données possédant peu d'instances (*Food*), $200\ 000$ itérations pour ceux ayant un nombre d'instances moyen : (*RS-ASM* et *Contrôle*), $400\ 000$ itérations pour ceux ayant un grand nombre d'instances (*Jester*), et enfin $10\ 000\ 000$ itérations pour ceux possédant un très grand nombre d'instances (*Poker Hand*).

Paramétrages. Nous expérimenterons trois cas de paramétrages possibles de *Gorthaur* :

— *Gorthaur* avantageant la précision globale (Cas C1) c.-à-d., $\Theta \approx 97,56\%$ de $\frac{\pi}{2}$ c.-à-d., $\Theta = 87.81°$) ;

— *Gorthaur* avantageant la diversité (Cas C2) c.-à-d., $\Theta \approx 6,36\%$ de $\frac{\pi}{2}$ c.-à-d., $\Theta = 5.73°$) ;

— *Gorthaur* « *équilibré* » (Cas C3) calculant Θ de manière dynamique (voir explications dans la section 5.3.3.3).

Ceci nous permettra d'observer la pertinence des sélections automatiques d'algorithmes de *Gorthaur* selon le niveau d'importance accordé à chacun des deux critères.

5.3.5 Analyse des résultats

Dans cette section nous décrivons les résultats obtenus par *Gorthaur* selon les trois cas : C1) Précision globale avantagée (Tableau 5.3), C2) Diversité avantagée (Tableau 5.4), C3) Calcul dynamique de Θ (Tableau 5.5).

TABLE 5.3 – Résultats de Gorthaur avantageant la précision globale, cas C1 ($\Theta \approx 97,56\%$ de $\frac{\pi}{2}$ c.-à-d., $\Theta = 87,81°$)

	Contrôle	***RS-ASM (vc)***	***RS-ASM (vt)***	***Food***	***Poker Hand***	***Jester***
Gorthaur	0,80 / 0,95	0,59 / 0,61	0,55 / 0,48	0,79 / 0,67	0,49 / 0,08	0,54 / 0,57
LinUCB	**24,9%/0,999/0,998**	**19,4%/0,67/0,87**	17,2% / 0,58 / 0,53	**18,6%/0,92/0,9**	**16,8%/0,52/0,08**	7,9% / 0,32 / 0,99
CTS	24,7% / 0,999 / 0,998	15,7% / 0,59 / 0,82	**17,5%/0,59/0,48**	18,3% / 0,91 / 0,85	15,3% / 0,5 / 0,09	7,9% / 0,32 / 0,99
EXP4.P	24,8% / 0,996 / 0,996	9,8% / 0,4 / 0,8	11,9% / 0,45 / 0,59	10,3% / 0,62 / 0,97	15,1% / 0,5 / 10^{-4}	8,1% / 0,33 / 0,99
UCB2	6,5% / 0,25 / 0,66	13,7% / 0,54 / 0,1	10,9% / 0,42 / 0,13	12,7% / 0,73 / 0,1	12% / 0,42 / 0,001	12,7% / 0,47 / 0,41
EXP3	6,5% / 0,25 / 0,5	13,1% / 0,51 / 0,48	15,5% / 0,55 / 0,25	12,9% / 0,72 / 0,53	15,2% / 0,5 / 0,01	16,1% / 0,48 / 0,58
TS	6,4% / 0,25 / 0,6	15,2% / 0,58 / 0,1	16,5% / 0,58 / 0,05	14,8% / 0,8 / 0,06	15,1% / 0,5 / 10^{-4}	**28,6%/0,77/0,03**
ε-Greedy	6,3% / 0,24 / 0,53	13,1% / 0,52 / 0,2	10,5% / 0,41 / 0,2	12,4% / 0,72 / 0,2	10,5% / 0,36 / 0,19	18,7% / 0,56 / 0,35

TABLE 5.4 – Résultats de *Gorthaur* avantageant la diversité, cas C2 ($\Theta \approx 6,36\%$ de $\frac{\pi}{2}$ c.-à-d., $\Theta = 5,73°$)

	Contrôle	***RS-ASM (vc)***	***RS-ASM (vt)***	**Food**	***Poker Hand***	***Jester***
Gorthaur	0,71 / 0,97	0,56 / 0,70	0,52 / 0,52	0,77 / 0,8	0,47 / 0,24	0,42 / 0,8
LinUCB	20,6% / 0,998 / 0,993	**24,5%/0,69/0,85**	**22,1%/0,59/0,53**	21,2% / 0,93 / 0,91	20% / 0,53 / 0,06	17,2% / 0,32 / 0,99
CTS	20,8% / 0,999 / 0,993	23,6% / 0,61 / 0,8	19,9% / 0,59 / 0,5	19,7% / 0,9 / 0,85	23% / 0,5 / 0,08	17,3% / 0,32 / 0,99
EXP4.P	**20,9%/0,995/0,994**	17,4% / 0,49 / 0,42	18,7% / 0,5 / 0,36	**21,8%/0,62/0,98**	4,6% / 0,5 / 0,001	17,3% / 0,32 / 0,99
UCB2	8,8% / 0,25 / 0,48	9,7% / 0,55 / 0,34	10,1% / 0,52 / 0,2	10,1% / 0,61 / 0,3	4,75% / 0,5 / 0,004	16% / 0,47 / 0,56
EXP3	11,6% / 0,25 / 0,81	13% / 0,53 / 0,39	17,1% / 0,54 / 0,33	13,6% / 0,73 / 0,44	8,5% / 0,49 / 0,01	**19%/0,48/0,66**
TS	11,4% / 0,25 / 0,7	5,9% / 0,58 / 0,12	5,9% / 0,57 / 0,1	7,6% / 0,78 / 0,23	4,65% / 0,5 / 0,004	6,6% / 0,77 / 0,08
ε-Greedy	6% / 0,24 / 0,37	5,9% / 0,51 / 0,16	6,3% / 0,51 / 0,18	6% / 0,63 / 0,29	**34,5%/0,42/0,19**	6,6% / 0,49 / 0,43

TABLE 5.5 – Résultats de *Gorthaur* calculant Θ_{pareto} dynamiquement, cas C3

	Contrôle	***RS-ASM (vc)***	***RS-ASM (vt)***	***Food***	***Poker Hand***	***Jester***
Gorthaur	0,77 / 0,95	0,58 / 0,62	0,55 / 0,51	0,78 / 0,8	0,48 / 0,1	0,49 / 0,74
Θ_T **calculé**	50,7% de $\frac{\pi}{2}$, (45,6°)	48,8% de $\frac{\pi}{2}$, (44°)	56% de $\frac{\pi}{2}$, (50,48°)	43,33% de $\frac{\pi}{2}$, (39,29°)	44,84% de $\frac{\pi}{2}$, (40,36°)	57,28% de $\frac{\pi}{2}$, (51,56°)
LinUCB	23,3% / 0,998 / 0,996	**22,3%/0,68/0,87**	**18,8%/0,58/0,54**	**21%/0,93/0,91**	20,6% / 0,53 / 0,08	12,5% / 0,32 / 0,99
CTS	**23,4%/0,999/0,997**	19,9% / 0,6 / 0,8	18,6% / 0,59 / 0,52	19,8% / 0,91 / 0,85	19,4% / 0,49 / 0,09	12,4% / 0,32 / 0,99
EXP4.P	23,1% / 0,995 / 0,995	16,7% / 0,4 / 0,83	18% / 0,49 / 0,44	17,7% / 0,62 / 0,97	8,3% / 0,5 / 0,001	12,5% / 0,32 / 0,99
UCB2	7,7% / 0,247 / 0,47	9,3% / 0,53 / 0,11	10,4% / 0,51 / 0,11	9,7% / 0,64 / 0,22	8,6% / 0,5 / 0,002	15,4% / 0,49 / 0,52
EXP3	7,4% / 0,249 / 0,4	14,8% / 0,54 / 0,42	13,8% / 0,54 / 0,33	15,4% / 0,72 / 0,60	11% / 0,5 / 0,01	**18,1%/0,48/0,69**
TS	9,5% / 0,255 / 0,62	9,2% / 0,58 / 0,11	11,4% / 0,58 / 0,11	9,2% / 0,8 / 0,12	8,2% / 0,5 / 10^{-4}	14,9% / 0,77 / 0,05
ε-Greedy	5,6% / 0,246 / 0,25	7,8% / 0,5 / 0,29	9% / 0,5 / 0,19	7,2% / 0,64 / 0,23	**23,5%/0,42/0,19**	14,2% / 0,53 / 0,51

Dans les tableaux 5.3, 5.4 et 5.5, on observera la première ligne de résultat intitulée « *Gorthaur* » indiquant : la précision globale / la diversité ($acc(T)/div(T)$) obtenue par *Gorthaur*. Enfin, dans les lignes suivantes on indique respectivement pour chaque algorithme du porte-feuille : leur proportion d'utilisation (par *Gorthaur*) en pourcentage par rapport aux autres algorithmes du porte-feuille, leur précision globale, et leur diversité (utilisation en $\%/acc(T)/div(T)$). Notons que le tableau 5.5 quant à lui indique en plus la valeur de Θ calculée (à l'horizon T) en pourcentage de $\frac{\pi}{2}$ et son équivalent en degré.

En analyse préalable, nous observons que *Gorthaur* adapte son choix d'algorithmes en fonction du cas contextuel ou non.

5.3.5.1 Cas avantageant la précision globale (Cas C1)

Dans le tableau 5.3 nous observons que la part de sélection de *LinUCB* est la plus importante pour les jeux de données *Contrôle* $(24,9\%)$, *RS-ASM (vc)* $(19,4\%)$, *Food* $(18,6\%)$ et *Poker Hand* $(16,8\%)$. Concernant le jeu de données *RS-ASM (vt)* par contre *CTS* a été proportionnellement le plus sélectionné $(17,5\%)$ et dans le cas non contextuel de *Jester*, c'est *TS* qui a été le plus sélectionné $(28,6\%)$.

Ces résultats sont en cohérence avec le tableau 4.3 mais également avec les valeurs de précision globale et de diversité propre à chaque algorithme et selon la stratégie choisie (c.-à-d., favoriser la précision globale).

Si nous comparons les résultats obtenus par *Gorthaur* dans notre cas favorisant la précision globale par rapport aux résultats obtenus avec les autres paramétrages, on observe que la précision globale est bien avantagée (p. ex. sur le jeu de données *RS-ASM (vc)* : $acc(T) = 0,59$ contre $acc(T) = 0,56$ et $acc(T) = 0,58$).

Notons un point important : pour sélectionner les algorithmes du porte-feuille, si nous avions choisi une méthode de type bandit-manchot plutôt que de type roulette proportionnelle alors c'est l'algorithme apparaissant en gras dans le tableau 5.3 qui aurait été sélectionné. Cet algorithme est donc « *optimal* » pour maximiser à la fois la précision globale et la diversité dans le cas où on favorise fortement la précision globale.

5.3.5.2 Cas avantageant la diversité (Cas C2)

Dans le tableau 5.4 nous observons que la part de sélection de *LinUCB* est la plus importante pour les jeux de données *RS-ASM (vc)* $(24,5\%)$ et *RS-ASM (vt)* $(22,1\%)$. Concernant les jeux de données *Contrôle* et *Food* par contre c'est *EXP4.P* qui a été proportionnellement le plus sélectionné (respectivement $20,9\%$ et $21,8\%$). Dans le cas non contextuel de *Jester*, c'est *EXP3* qui a été le plus sélectionné (19%). Enfin, pour le jeu de données *Poker Hand* c'est ε*-Greedy* qui été de loin le plus choisi (34.5%).

Ces résultats sont en cohérence avec le tableau 4.3 mais également avec les valeurs de précision globale et de diversité propre à chaque algorithme et selon la stratégie choisie (c.-à-d., favoriser la diversité).

Si nous comparons les résultats obtenus par *Gorthaur* dans notre cas favorisant la diversité par rapport aux résultats obtenus avec les autres paramétrages, on observe que la diversité est bien avantagée (p. ex. sur le jeu de données *RS-ASM (vc)* : $div(T) = 0,7$ contre $div(T) = 0,61$ et $div(T) = 0,62$).

5.3.5.3 Cas de calcul dynamique de Θ (Cas C3)

Dans le tableau 5.5 nous observons que la part de sélection de *LinUCB* est la plus importante pour les jeux de données *RS-ASM (vc)* $(22,3\%)$, *RS-ASM (vt)* $(18,8\%)$ et *Food* (21%). Concernant le jeux de données *Contrôle* par contre c'est *CTS* qui a été proportionnellement le plus sélectionné $(23,4\%)$. Dans le cas non contextuel de *Jester*, c'est *EXP3* qui a été le plus sélectionné $(18,1\%)$. Enfin, pour le jeu de données *Poker Hand* c'est ε*-Greedy* qui été le plus choisi $(23,5\%)$ suivi de près par *LinUCB* $(20,6\%)$ et *CTS* $(19,4\%)$.

Ces résultats semblent bien favoriser l'équilibre entre les deux critères. Si nous comparons les résultats obtenus par *Gorthaur* dans notre cas par rapport aux résultats obtenus avec les autres paramétrages, on observe que l'équilibre entre précision globale et diversité est bien respecté (p. ex. sur le jeu de données *RS-ASM (vc)* : $(acc(T) = 0,58; div(T) = 0,62)$ contre $(acc(T) = 0,59; div(T) = 0,61)$ et $(acc(T) = 0,56; div(T) = 0,70)$).

Enfin, nous pouvons observer les valeurs finales calculées de Θ pour chaque jeu de données. Ces valeurs indiquent le front *pareto* pour l'ensemble des algorithmes sur les critères de diversité et de précision globale.

5.3.6 Analyse complémentaire sur la précision individuelle

Dans cette sous-section, nous réalisons une analyse complémentaire des résultats obtenus par *Gorthaur* sur le critère de précision individuelle sur les jeux de données issus du monde réel.

Aux Figures 5.8, 5.9, 5.10, et 5.11, nous observons que la précision individuelle est plus équitablement répartie sur l'ensemble de la population que ce que nous avions obtenu avec n'importe quel algorithme observé individuellement (voir le Chapitre 4 et les Figures B.3, B.13, B.11, et B.16 en annexe). De plus, on remarque que la proportion d'utilisateurs pour laquelle la précision individuelle est très basse (c.-à-d., dans l'intervalle $[0,0;0,1]$) est moins importante avec *Gorthaur*.

Enfin, en fonction du jeu de données, on remarque qu'il y a une incidence sur la précision individuelle selon le paramétrage choisi de Θ :

— concernant RSASM qui est un jeu de données spécifique à la recommandation, Figures 5.8 et 5.9 on remarque un avantage à utiliser un paramétrage en faveur de la diversité dans les deux cas (*vc* ou *vt*) puisque celui-ci obtient, de manière globale, une meilleure FDC de la précision individuelle ;

— concernant, le jeu de données *Poker Hand* par contre, les résultats restent à nuancer. Une forte diversification permet concrètement de réduire la proportion de personnes

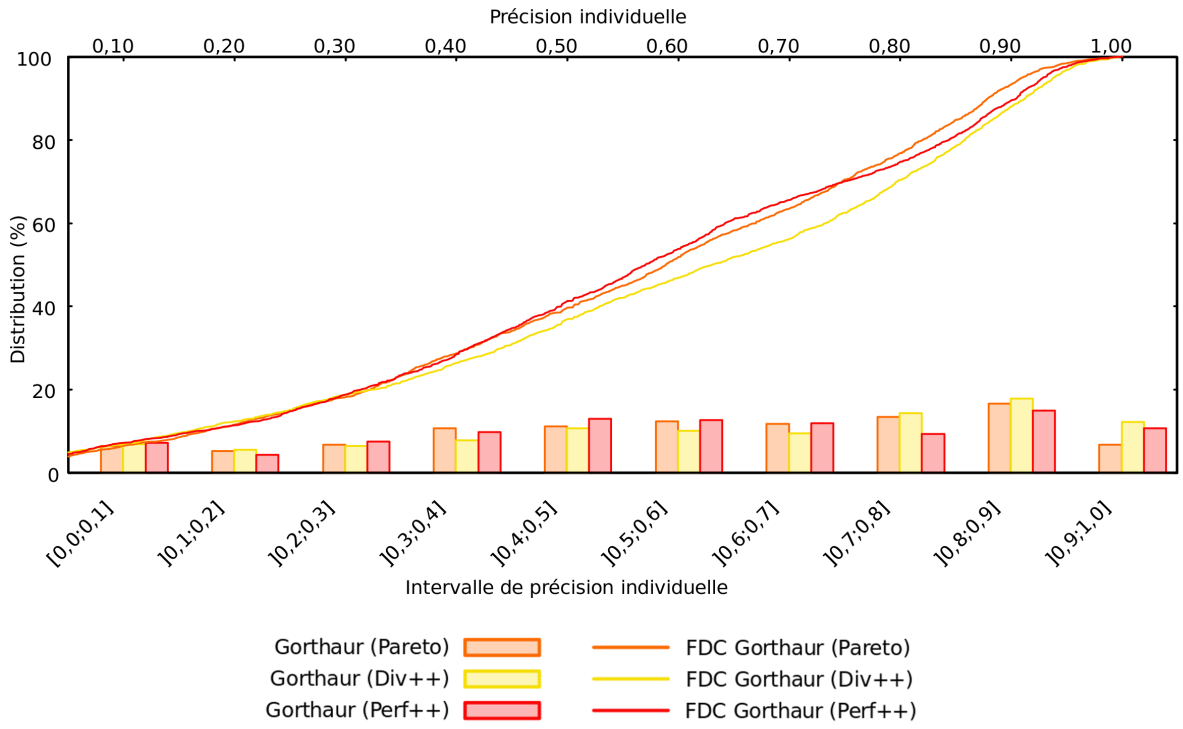


FIGURE 5.8 – Distribution de la précision individuelle pour chaque paramétrage de *Gorthaur* sur le jeu de données *RSASM (vc)*

dont la précision individuelle est très basse (c.-à-d., $[0,0;0,1]$). Néanmoins, ce résultat opère au détriment des personnes dont la précision individuelle était forte. Aussi, on peut en déduire en observant la FDC sous un angle global que la précision individuelle est ré-équilibrée et recentrée sur la moyenne $(0,47)$. Dans le cadre de ce jeu de données, on préférera donc peut être un paramétrage favorisant la précision globale donnant des résultats finalement plus équilibrés que les deux autres formes de paramétrage. Ceci peut s'expliquer par la nature même du jeu de données pour lequel il existe finalement peu de relation entre le contexte et les classes obtenues par ailleurs (exceptées les classes : « *Nothing in hand* » et « *One Pair* » qui semblent différentiables).

— concernant le jeu de données *Jester*, on remarque que le paramétrage favorisant la précision globale est le plus avantageux. En effet, ce paramétrage offre à la fois une garantie de base de diversifier (de part la nature même de la méthode de diversification de *Gorthaur*) tout en cherchant à maximiser la précision globale. Trop favoriser la diversification devient contre-performant du fait de la composition même du porte-feuille de *Gorthaur*. En effet, dans notre cas *Gorthaur* possède trois algorithmes de *CMAB* qui, ne pouvant s'appuyer sur des caractéristiques du contexte, restent piégés en phase d'ex-

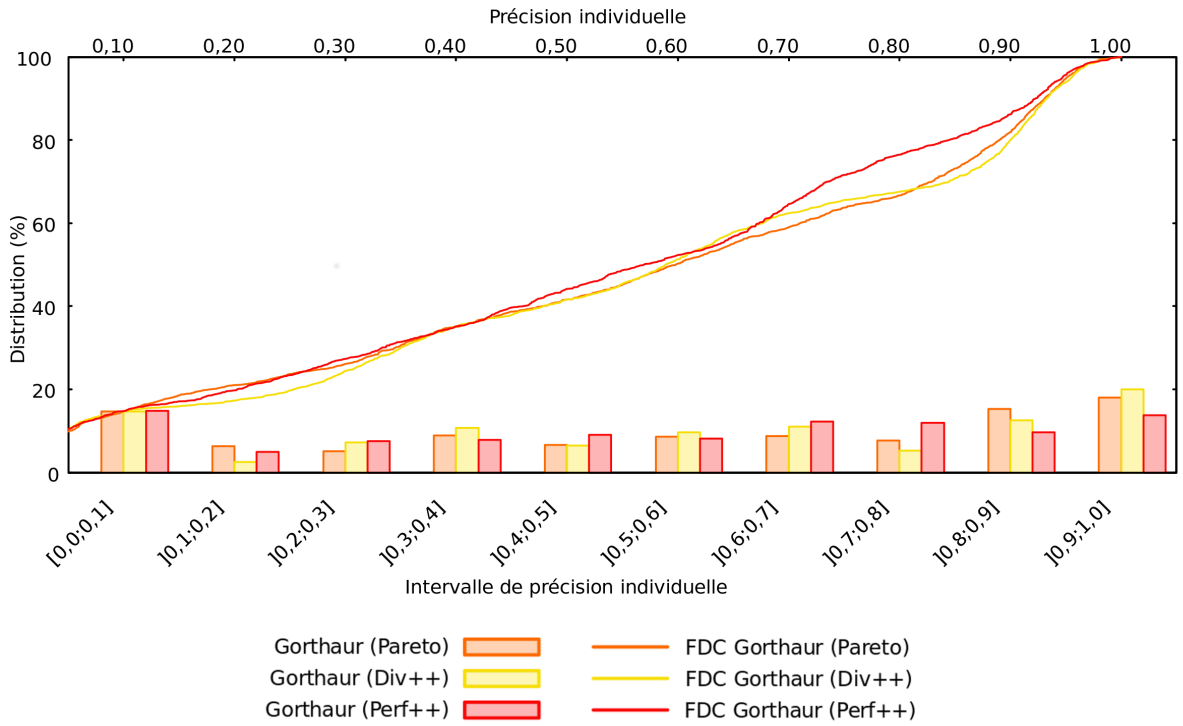


FIGURE 5.9 – Distribution de la précision individuelle pour chaque paramétrage de *Gorthaur* sur le jeu de données *RSASM (vt)*

ploration. De ce fait, sur-favoriser l'exploration via *Gorthaur* devient contre-performant et ce même pour la précision individuelle.

L'analyse complémentaire que nous venons de réaliser ouvre deux nouvelles perspectives :

1. Dans le cas de jeux de données non-contextuels les résultats que nous avons obtenus révèlent qu'il n'est pas pertinent d'intégrer des algorithmes de *CMAB* dans le portefeuille de *Gorthaur* puisqu'ils restent incapables de converger sans données de contexte sur lesquelles s'appuyer. Néanmoins, il pourrait être intéressant de tirer profit des algorithmes de *MAB* pour construire du contexte exploitable par les algorithmes de *CMAB* qui deviendraient ainsi intéressant à utiliser. Cette perspective sera traitée dans le Chapitre 7 à la Section 7.4 ;

2. Nous avons énoncé le problème de *Gorthaur* sous la forme d'une maximisation bi-objectifs (c.-à-d., maximiser à la fois la précision et la diversité). L'analyse complémentaire que nous venons de réaliser sur le critère de précision individuelle, nous amène à envisager, en perspective, de prendre en considération la précision individuelle comme troisième critère à maximiser dans *Gorthaur*. Le problème deviendrait alors multi-objectifs (c.-à-d., maximiser à la fois la précision globale, la diversité et la

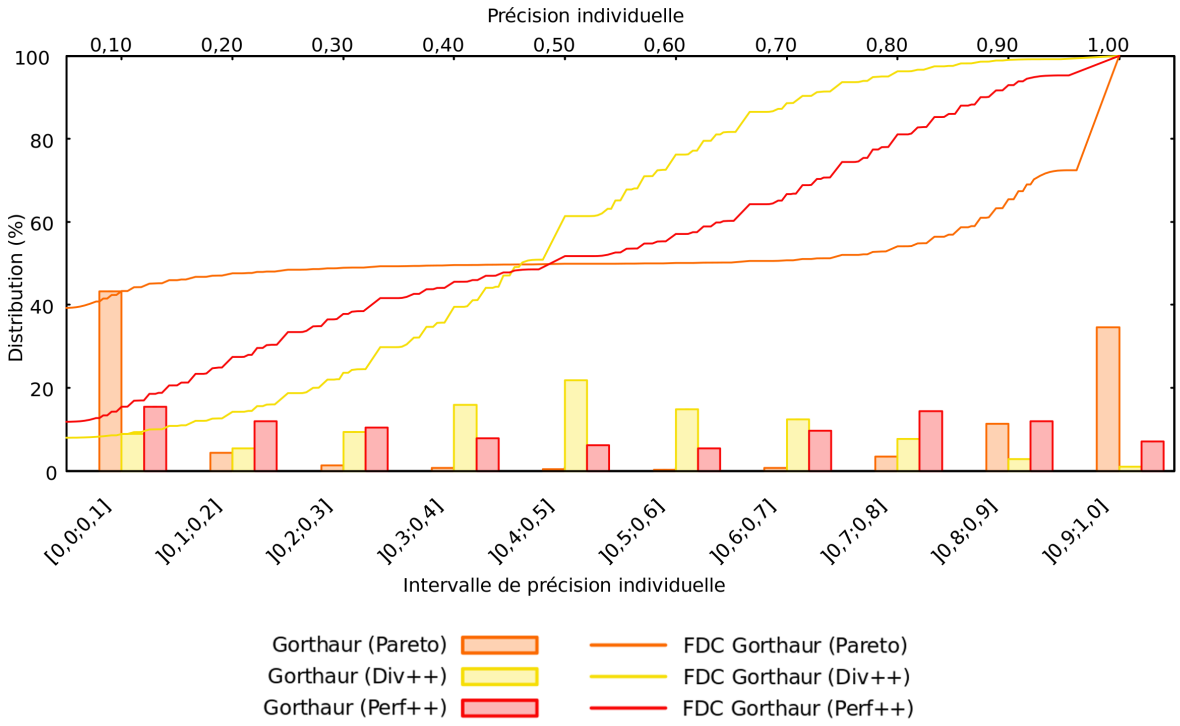


FIGURE 5.10 – Distribution de la précision individuelle pour chaque paramétrage de *Gorthaur* sur le jeu de données *Poker Hand*

précision individuelle). Cette seconde perspective n'a pas été exploitée dans cette thèse et reste donc totalement ouverte.

5.3.7 Conclusion et Perspectives

Dans cette section, nous avons proposé *Gorthaur* : une nouvelle approche de type portfolio d'algorithmes de bandits-manchots pour la recommandation. Notre problème est bi-objectifs : maximiser à la fois la précision globale et la diversité des recommandations. Ainsi, le principe de *Gorthaur* est de sélectionner les algorithmes proportionnellement (c.-à-d., via roulette proportionnelle) à leur capacité à maximiser ces deux critères.

Par la suite, il pourrait être intéressant, d'ajouter d'autres critères à maximiser comme par exemple la précision individuelle. Naturellement, *Gorthaur* différencie les problèmes contextuels des problèmes non contextuels et sélectionne selon le cas les algorithmes de la bonne catégorie (*MAB* ou *CMAB*). D'autres parts, les proportions de sélection de chaque algorithme que nous avons obtenues via *Gorthaur* sont cohérentes vis à vis des évaluations que nous avons effectuées au préalable pour chaque algorithme.

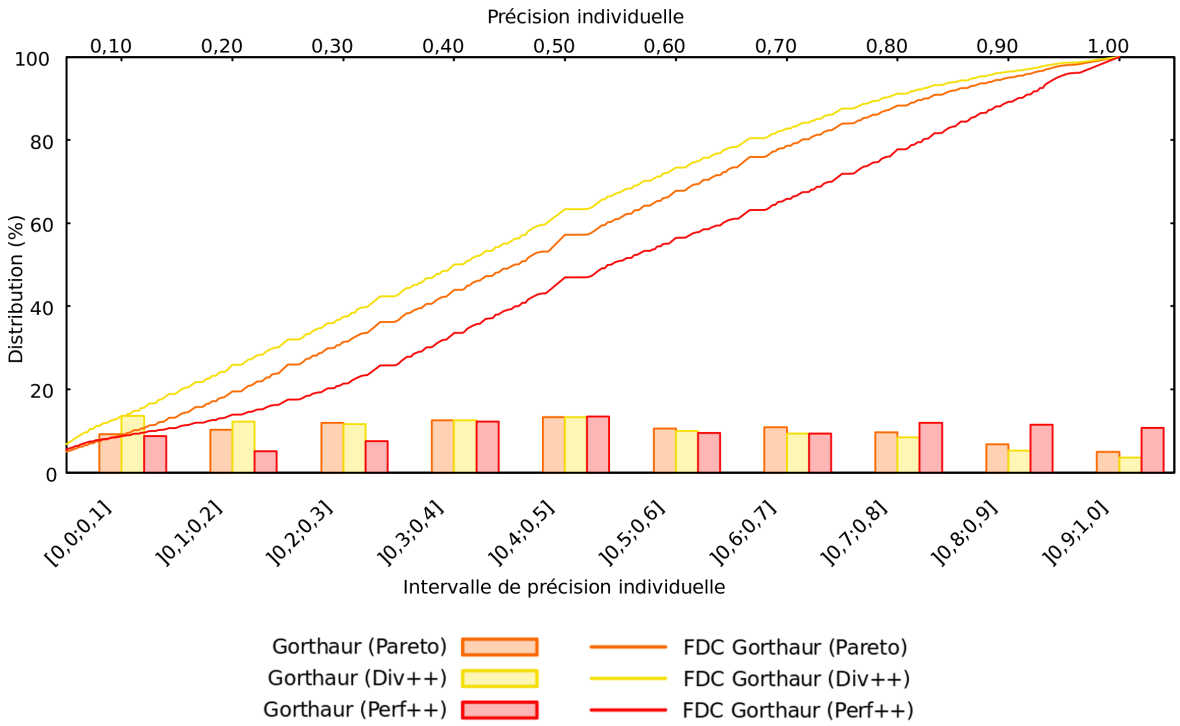


FIGURE 5.11 – Distribution de la précision individuelle pour chaque paramétrage de *Gorthaur* sur le jeu de données *Jester*

De ce fait, en fonction du besoin applicatif du système de recommandation, il sera possible d'utiliser *Gorthaur* selon deux modes de sélection possibles des algorithmes du porte-feuille :

1. Sélection de type roulette proportionnelle comme évalué dans ce chapitre afin de trouver un compromis entre précision et diversité en se reposant sur la capacité intrinsèque de chacun des algorithmes du porte-feuille ;

2. Sélection de type bandit-manchot afin de plutôt choisir l'algorithme optimal permettant de répondre au mieux aux deux critères qu'on cherche à maximiser.

Cette seconde possibilité reste à expérimenter et se constitue naturellement comme une imminente perspective.

Enfin, une autre perspective intéressante à prendre en compte dans un avenir proche serait de faire évoluer *Gorthaur* sous forme multi-objectifs et d'y intégrer le critère (à maximiser) de précision individuelle.

5.4 Diversité et précision individuelle : Conclusion et Perspectives

Dans ce chapitre, nous avons évalué deux méthodes de diversification des recommandations : l'une par le biais d'une fenêtre glissante permettant de pénaliser les bras qui ont été trop sélectionnés, la seconde en utilisant une approche portfolio d'algorithmes visant à maximiser à la fois la précision globale et la diversité. Dans cette section, nous comparons donc ces deux méthodes qui permettent d'obtenir des résultats similaires selon deux différentes approches.

Dans un premier temps, nous nous poserons la question d'utiliser *SW-LinuCB* ou *Gorthaur*, ou bien d'utiliser *SW-LinuCB* dans *Gorthaur*. Ensuite nous réfléchirons à l'applicabilité de ces méthodes. Enfin, nous conclurons.

5.4.1 *Gorthaur* ou *SW-LinUCB* ? *Gorthaur* avec *SW-LinUCB* ?

***Gorthaur* ou *SW-LinUCB* :** nous considérons ces deux approches indépendamment l'une de l'autre ce qui les rend de ce fait « *concurrentes* ».
***Gorthaur* avec *SW-LinUCB* :** *SW-LinUCB* est incorporé dans le porte-feuille d'algorithmes de *Gorthaur* ce qui les rend de ce fait « *collaborantes* ».

5.4.1.1 Philosophie de *Gorthaur*

La question posée en titre de cette sous-section est intéressante puisqu'elle nous ouvre vers le fondement même de la création de *Gorthaur*. En effet, en conclusion de la Section 5.3 traitant de *Gorthaur*, nous avons évoqué le fait de pouvoir utiliser deux types de sélections possibles des algorithmes du porte-feuille (c.-à-d., sélection roulette ou sélection de type *MAB*). Aussi, pour répondre à cette question, tout dépend de l'application qu'on souhaite en faire et des résultats attendus. Ainsi, il sera possible :

1. Soit de décider de rechercher un compromis entre diversité et précision globale sur l'ensemble du porte-feuille d'algorithmes (c.-à-d., sélection roulette) ;
2. Soit de décider de rechercher le meilleur algorithme (optimal) offrant le meilleur compromis entre diversité et précision globale sur l'ensemble du porte-feuille d'algorithmes (c.-à-d., sélection de type *MAB*).

Dans le premier cas (sélection roulette), le résultat de *SW-LinUCB* sera intégré en proportion dans *Gorthaur* selon le paramétrage de Θ choisi, dans le second cas (sélection *MAB*), si *SW-LinUCB* s'avère être le meilleur algorithme alors les résultats de *Gorthaur* seront identiques à ceux de *SW-LinUCB*.

De ce fait, nous pouvons plus particulièrement considérer que *Gorthaur* ne s'oppose pas à *SW-LinUCB* mais doit plutôt l'inclure dans tous les cas. Ainsi, ***Gorthaur* avec *SW-LinUCB*** serait la meilleure solution.

5.4.1.2 Comparaison de *Gorthaur* avec *LinUCB* et *SW-LinUCB* sur le jeu de données *RSASM (vt)*

Précédemment, nous avons argumenté en quoi *SW-LinUCB* ne s'opposait pas à *Gorthaur* mais pouvait plutôt l'enrichir. Néanmoins, il semble intéressant de pouvoir les comparer afin de voir ce que *SW-LinUCB* apporterait à *Gorthaur*.

Quid de la précision globale ? Nous observons que *Gorthaur* a obtenu une précision globale de l'ordre de $0,55$ dans le cas d'un paramétrage favorisant la performance ou l'équilibre, et $0,52$ dans le cas d'un paramétrage favorisant la diversité. *SW-LinUCB* quant à lui obtient une précision globale de $0,56$. Il ne semble pas y avoir d'avantage significatif à utiliser l'une où l'autre des méthodes, néanmoins on observe que si nous intégrions *SW-LinUCB* dans le portefeuille de *Gorthaur*, alors il aurait tendance à légèrement améliorer la précision globale de ce dernier.

Quid de la diversité ? Nous observons que *Gorthaur* a obtenu une valeur de diversité de $0,52$ dans le cas d'un paramétrage favorisant la diversité, $0,51$ dans le cas d'un paramétrage à l'équilibre, et $0,48$ dans le cas d'un paramétrage favorisant la performance. A contrario *SW-LinUCB* obtient une valeur de diversité de $0,69$. De ce fait, intégrer *SW-LinUCB* dans le portefeuille de *Gorthaur* aurait tendance à améliorer la diversité de ce dernier.

Enfin, en termes de précision individuelle, nous observons à la Figure 5.12 que *Gorthaur* et *SW-LinUCB* obtiennent une tendance similaire en offrant ainsi la garantie d'une précision individuelle mieux répartie au sein de la population que tout autre méthode utilisée seule (voir résultats des analyses préliminaires sur la précision individuelle en Annexe B).

Ainsi, nous remarquons selon les trois critères (Acc, Div, et Acc_u) que *SW-LinUCB* serait un algorithme pertinent à intégrer dans *Gorthaur*. En effet, si *SW-LinUCB* s'avérait être l'algorithme le plus pertinent pour résoudre le problème bi-objectifs alors dans le cas d'une sélection roulette *Gorthaur* le sélectionnerait en proportion plus fréquemment et améliorait de ce fait sa propre performance. D'autres parts dans le cas d'une sélection de type *MAB*, *Gorthaur* sélectionnerait systématiquement *SW-LinUCB* puisqu'il serait l'algorithme « optimal » pour résoudre le problème bi-objectifs.

5.4.2 Applicabilité

Notre objectif final est d'intégrer un système de recommandation dans une application mobile réelle : Le *scéno*[2]. Les contraintes de telles applications nécessitent d'employer à la fois des approches plus dynamiques, sensibles au contexte mais également génériques afin de répondre à tout problème. En effet, dans de telles applications, il est difficile voire impossible de savoir à l'avance quels algorithmes fonctionneront le mieux. De plus, les préférences utilisateurs peuvent évoluer selon le contexte et/ou au cours du temps.

2. Projet RFI-Atlanstic2020 *Event-AI* : `https://play.google.com/store/apps/details?id=sceno.android.pfe.com.sceno&hl=fr`

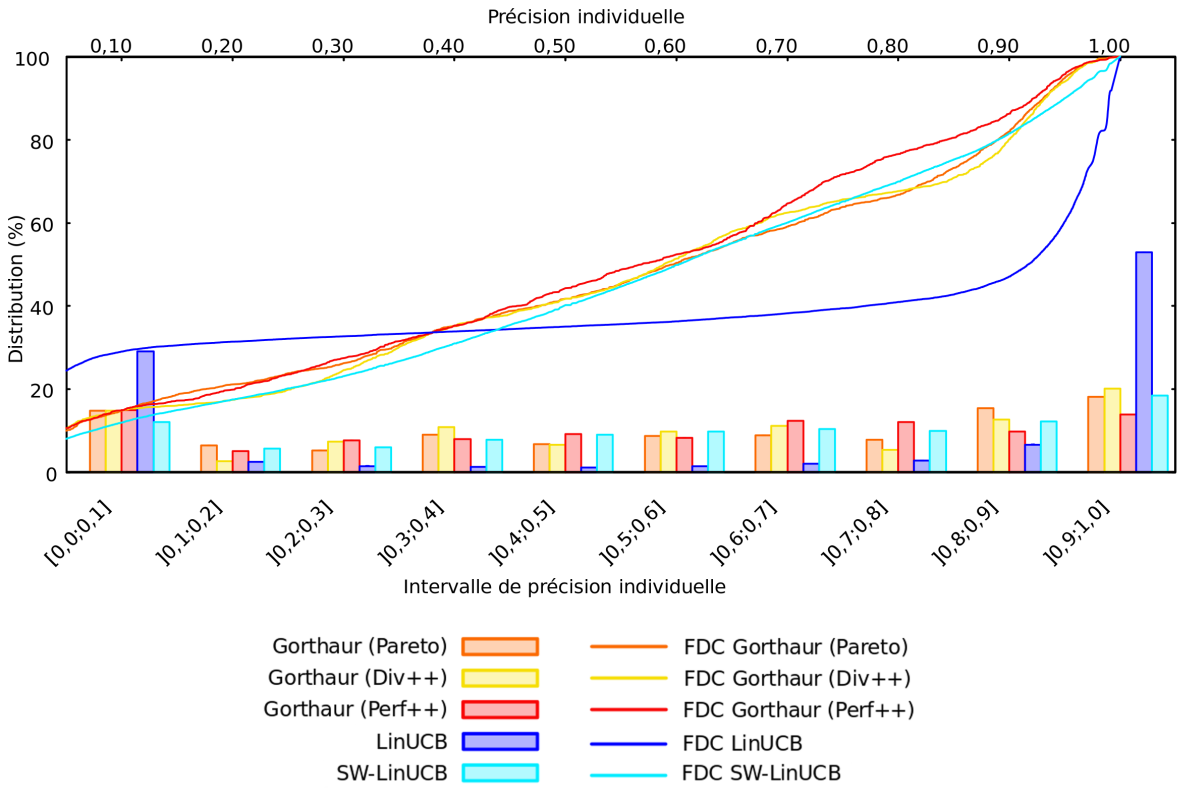


FIGURE 5.12 – Distribution de la précision individuelle pour chaque algorithme sur le jeu de données *RSASM (vt)*

De ce fait, l'utilisation d'une approche dynamique telle que *SW-LinUCB* ou à la fois générique comme *Gorthaur*, est une perspective tout à fait pertinente que nous envisageons dans les mois qui suivront.

5.4.3 Conclusion

Dans ce chapitre nous avons mis en lumière l'intérêt de la diversification et de prendre en considération une mesure de la précision individuelle. Aussi, le prochain chapitre fera entre autres l'objet de l'utilisation de la précision individuelle mesurée afin de créer du contexte exploitable par les algorithmes de bandits-manchots contextuels pour la recommandation (voir Section 7.4 du Chapitre 7).

De même, dans ce chapitre nous avons pu observer l'impact d'une restriction sur les informations de contexte sur la précision des recommandations faites par les algorithmes de *CMAB* c.-à-d., tronquer une partie pertinente du vecteur de contexte résulte à la fois en une diminution de la précision globale, de la diversité et de la précision individuelle. De ce fait, dans la prochaine partie, nous évoquerons comment pallier le manque d'informations contextuelles

complètes en capturant le contexte dans le cas pratique de la ville intelligente (Voir Chapitre 6), et comment rendre ce contexte exploitable sous une forme de représentation structurée pour les algorithmes de *CMAB* pour la recommandation (Voir les Chapitres 6 et 7).

TROISIÈME PARTIE

Contributions à l'élaboration du contexte pour les algorithmes de recommandation

Introduction de la partie III

« Le contexte est ce qui n'intervient pas directement dans la résolution d'un problème mais contraint sa résolution » [Bré02].

Plusieurs études dans le domaine des problèmes de décision contextuels se concentrent sur la modélisation ou l'amélioration du contexte [Bou+17 ; Guo+15 ; Sli14] afin d'améliorer la précision globale ou de gérer des problèmes de complexité en temps. Dans la partie II, nous avons évalué les algorithmes de bandits-manchots à travers deux nouvelles métriques : la précision individuelle et la diversité. Nous avons observé que les résultats de précision globale, diversité et précision individuelle sont très dépendants du niveau de description et de pertinence du contexte qui leur est fourni en entrée.

En effet, dans le problème des bandits-manchots contextuels (*CMAB*) basés sur un modèle linéaire, on suppose une dépendance linéaire entre les récompenses et les caractéristiques du contexte [LZ08]. Les contextes sont fournis séquentiellement aux algorithmes de *CMAB* qui choisissent alors la meilleure action en conséquence [AG13 ; Li+10]. De tels algorithmes ont des bornes supérieures de regret théoriquement basses et atteignent finalement une personnalisation totale ; parfois après un grand nombre d'itérations [ZB16]. Néanmoins, dans diverses applications du monde réel telles que les systèmes de recommandation [Li+10], les essais cliniques [VBW15] ou les applications mobiles en santé [Gre+17], le contexte peut être donné restreint (c.-à-d., incomplet), en raison par exemple d'une mauvaise modélisation du contexte [SDO18], de restrictions liées à la confidentialité des données, ou encore de profils mal renseignés. Ces contextes incomplets (caractéristiques manquantes / non observées) peuvent entraîner des problèmes de dérive conceptuelle (*concept-drift*) ayant un impact sur les algorithmes de *CMAB* [Qui+09] (voir Sous-section 2.3.7).

Par exemple, imaginons un cas réel simple de non stationnarité par partie, où chaque partie est associée à une condition météorologique particulière. Pour simplifier, nous considérerons deux conditions : pluie/soleil. Durant la partie où le soleil brillera, les utilisateurs seront ravis de recevoir des recommandations de glaces. A contrario, lorsqu'il pleuvra, les utilisateurs seront ravis de recevoir des recommandations de boissons chaudes. Si notre système à base de bandits-manchots contextuels, a un accès restreint au contexte météorologique (c.-à-d. il ne possède pas ces informations contextuelles), alors ne pouvant s'appuyer sur ces caractéristiques, il observera des regrets liés à la dérive contextuelle (non stationnarité). En revanche, si le système possède ces caractéristiques de contexte météorologique, alors il sera en mesure de pallier cette dérive en s'appuyant sur ces dimensions pertinentes ce qui rendra ses recommandations plus précises.

Par conséquent, les algorithmes de *CMAB* doivent opérer sur des caractéristiques pertinentes et variées du contexte, afin de bien différencier les diverses situations. Une description trop pauvre du contexte, ne permettrait plus aux algorithmes de différencier des contextes, pourtant différent, mis à leur disposition. Ces contextes considérés alors comme similaires n'obtiendront pourtant pas les mêmes récompenses pour un même bras donné. En d'autres

termes, pour une même observation ceci conduit à obtenir des récompenses différentes et à des regrets inattendus.

De telles considérations, nous ont amenées à travailler, à travers le spectre du paradigme de *Mobile Crowd Sensing and Computing (MCSC)*, sur une meilleure modélisation du contexte, sur sa capture et sur sa transformation, afin que les algorithmes de bandits-manchots contextuels pour la recommandation puissent mieux l'exploiter.

Ainsi dans le Chapitre 6, après avoir proposé une modélisation du contexte pour de la recommandation de services à des utilisateurs mobiles dans la ville, nous présentons deux cas d'études concrets de capture et d'interprétation du contexte dans le cadre de la ville intelligente. Chacun des cas repose soit :

— sur l'existence d'une infrastructure de points d'accès Wi-Fi urbain (*Wifilib*[3]), mise à disposition gratuitement dans les zones les plus fréquentées de plusieurs villes en France ;
— sur l'existence d'une application mobile que nous avons développée : *scéno*[4]. Elle permet de visualiser les événements culturels géolocalisés autour de la position de l'utilisateur et de recevoir des recommandations contextuelles sur ces événements culturels.

Ensuite dans le Chapitre 7 nous présentons quatre méthodes de raisonnement contextuel dans le cadre de la ville intelligente, dont certaines que nous avons pu appliquer aux systèmes de recommandation :

1. **Raisonnement contextuel que nous n'avons pas appliqué à un système de recommandation :** sur la base des données de connexions géolocalisées d'utilisateurs à l'ensemble des points d'accès de l'infrastructure de Wi-Fi urbain *Wifilib* à Angers, nous proposons deux raisonnements contextuels permettant d'étudier les trajectoires et la prédiction de la mobilité des différents utilisateurs mobiles ;

2. **Raisonnement contextuel que nous avons appliqué à un système de recommandation :**
   — sur la base des données de connexions géolocalisées d'utilisateurs à l'ensemble des points d'accès de l'infrastructure de Wi-Fi urbain *Wifilib*, nous proposons un raisonnement contextuel permettant de déduire des quartiers dans la ville en utilisant des techniques de partitionnement spectral. Les résultats obtenus sont ensuite employés en tant qu'informations de contexte spatial par le système de recommandation d'événements culturels de notre application mobile *scéno* ;
   — sur la base des recommandations faites à des utilisateurs, nous proposons une nouvelle approche de raisonnement contextuelle appelée *Individual Context Enrichment (ICE)* qui vise à pallier l'incomplétude du contexte rencontré dans la plupart des applications du monde réel. Plus précisément, notre méthode vise à différencier de manière itérative des contextes considérés comme « similaires » auxquels différents utilisateurs attribuent des récompenses différentes. Nous supposons que cette heuristique permettra aux algorithmes de *CMAB* originaux basés sur un modèle linéaire,

3. https://www.wifilib.com/index.html
4. https://play.google.com/store/apps/details?id=sceno.android.pfe.com.sceno&hl=fr

de fonctionner avec des caractéristiques de contexte supplémentaires plus pertinentes et de gagner ainsi en précision.

CHAPITRE 6

MODÉLISATION, CAPTURE ET ANALYSE PRÉLIMINAIRE DU CONTEXTE POUR LA RECOMMANDATION

Mobile Crowd Sensing and Computing

« *A new sensing paradigm that empowers ordinary citizens to contribute data sensed or generated from their mobile devices, aggregates and fuses the data in the cloud for crowd intelligence extraction and human-centric service delivery.* [Guo+15]

Bin Guo et al. - 2015 »

Sommaire

6.1 Introduction

Ce chapitre fait référence à nos contributions [Gut+17 ; Gut+18d] (étoile n° 1), [Gut+18c] (étoile n° 3), [Gut+18e] (étoile n° 2), ainsi qu'à [Gut+19c] (étoile n° 8) présentées à la Figure 2 en introduction de ce mémoire.

Dans ce chapitre, notre travail porte sur la modélisation, la capture et une analyse préliminaire du contexte dans la ville d'Angers à des fins de recommandations de services dans la ville intelligente.

La problématique de recommandation de services dans une ville peut se rapprocher de celle traitant des systèmes de recommandation géo-sociaux reposant sur les *Location-Based Social Networks (LBSN)* [SNM14] (c.-à-d. des réseaux sociaux basés sur la localisation des utilisateurs). Malheureusement en ce qui concerne la ville d'Angers les *LBSNs* contiennent trop peu d'informations pour être utilisables.

En revanche, l'existence de *Wifilib*[1], un vaste réseau Wi-Fi gratuit diffusé sur une infrastructure de plus de 200 points d'accès localisés principalement dans le centre-ville d'Angers,

1. https://www.wifilib.com/index.html

constitue, à travers les journaux de connexions qui en découlent, une source très riche de données contextuelles. Ces données sont principalement des données de profils (âge, sexe, catégories socioprofessionnelles) mais également des données de contexte environnemental (localisation, date et heure).

D'autre part, dans le cadre du projet de recherche *Event-AI*[2], nous avons développé une application mobile de visualisation et de recommandation d'événements culturels : *scéno*[3]. Dans l'application *scéno*, tous les utilisateurs sont géolocalisés en temps-réel et peuvent visualiser l'ensemble des événements disponibles autour de leur position. De plus, au sein de cette application, nous avons déployé un système de recommandation contextuel d'événements culturels basé sur les bandits-manchots. Ce système s'appuie entre autres sur un composant de capture d'informations de contexte visant à alimenter les algorithmes de bandits-manchots contextuels pour la recommandation (voir Chapitre 7).

Ainsi, les objectifs principaux de ce chapitre et le cadre applicatif qui en découle s'inscrivent pleinement dans l'un des contextes technologiques des plus populaires : le *Mobile Crowd Sensing and Computing* (MCSC) [Had14]. Le *MCSC* est un paradigme qui a émergé durant cette dernière décennie et qui a été identifié comme très prometteur. Le terme *MCSC* englobe toute application scientifique et/ou technologique qui traite de : la génération, la collecte et l'utilisation des données à des fins d'analyse, d'apprentissage automatique, et de décision. Les applications de *MCSC* permettent notamment de collecter une grande variété de données à partir des téléphones mobiles des utilisateurs tout en respectant leur vie privée [Had14]. Le *MCSC* englobe un large panel d'applications possibles [Guo+15] dans des domaines variés tels que :

- la surveillance de l'environnement ;
- la surveillance et la gestion de la circulation, les transports ;
- la mise en évidence de la dynamique urbaine ;
- les systèmes de recommandation mobiles ;
- la santé ;
- la sécurité publique.

Ces applications couvrent des problématiques allant de la collecte de données, jusqu'à leur traitement et leur analyse, en passant par leur stockage. Notre travail s'inscrit dans cette dynamique et se concentre principalement sur la recommandation de services à des utilisateurs mobiles. La recommandation de services dans les *MCSC* est un vaste domaine de recherche dont le principal objectif est de développer des systèmes d'aide à la décision qui proposent aux utilisateurs des suggestions personnalisées de services ou d'informations selon leurs profils, leurs préférences ou leurs habitudes. Pour ce faire ces systèmes reposent sur des informations contextuelles riches et pertinentes.

Ainsi, dans la Section 6.2 de ce chapitre, nous présentons notre analyse préliminaire du contexte brut de la ville d'Angers afin d'en évaluer la pertinence et la richesse. Dans la Sec-

2. Projet labellisé par le pôle de compétitivité Images & Réseaux, effectué dans le cadre d'Atlanstic 2020 – `https://atlanstic2020.fr` – financé par le RFI Pays de La Loire

3. Disponible sur *Google Play* : `https://play.google.com/store/apps/details?id=sceno.android.pfe.com.sceno&hl=fr`

tion 6.3 de ce chapitre, nous présentons comment nous capturons dynamiquement l'information contextuelle brute dans l'application mobile *scéno* et comment nous la structurons.

6.2 Modélisation, capture et analyse préliminaire du contexte à Angers

Cette section fait référence à nos contributions [Gut+17 ; Gut+18d] (étoile n° 1) présentées Figure 2 en introduction de ce mémoire.

Dans cette section, après avoir rappelé la notion de *contexte* et comment nous le modélisons, nous montrons comment l'analyse de données de contexte brut d'un environnement urbain permet de déduire des informations contextuelles riches. Notre principal objectif est de montrer, à travers des données réelles (journaux connexions au réseau Wi-Fi urbain Wifilib à Angers), un exemple concret de capture d'informations contextuelles de profil utilisateur et de données liées à la mobilité, pour un système de recommandation dans la ville intelligente.

6.2.1 Jeu de données

Notre travail repose sur un réseau Wi-Fi urbain continu *Wifilib* déployé par le groupe *Afone*, un opérateur virtuel basé à Angers. L'infrastructure de ce réseau est composée d'un ensemble de points d'accès mis à disposition dans les zones les plus fréquentées de plusieurs villes françaises.

Les fichiers journaux de connexions qui nous ont été fournis, représentent une source de données très intéressante puisqu'ils contiennent des informations contextuelles permettant de caractériser les utilisateurs et l'environnement dans lequel ils évoluent. Ces informations sont principalement des données de profil, la description du matériel de l'utilisateur, les coordonnées GPS du point d'accès sur lequel l'utilisateur est connecté, la durée et l'horodatage de la connexion. Le détail des informations que contiennent les fichiers de journaux sont disponibles en annexe D.1.

6.2.2 Notre modélisation du contexte

La modélisation de notre contexte a été spécialement construite pour répondre au besoin de capture du contexte à des fins de recommandation à des utilisateurs mobiles dans la ville. Ainsi, dans ce cadre nous rappelons ici la notion de contexte et décrivons notre proposition de modélisation.

La notion de contexte est un élément central pour l'adaptabilité et la personnalisation. Notre objectif est de capturer et exploiter les informations contextuelles afin de recommander des éléments (p. ex., services, événements culturels) pertinents à la bonne personne et au bon moment. Ces informations contextuelles (voir Chapitre 3) peuvent être, par exemple, l'emplacement de l'utilisateur, la date et l'heure de la journée ou l'activité de l'utilisateur. La définition

du contexte a fait l'objet de nombreux travaux [Bro+07 ; Che+04] dont la définition la plus populaire est celle donnée par [Dey01] et que nous avons rappelée au Chapitre 3.

Afin de modéliser le contexte dans le cadre de la recommandation de services dans la ville intelligente, nous partons de différentes définitions et approches le concernant et proposons un modèle de contexte générique qui devra répondre aux trois exigences suivantes [HMC15] :

— être suffisamment général pour être utilisé par différentes applications mobiles centrées sur l'utilisateur ;
— être suffisamment spécifique pour couvrir les principales entités contextuelles pour les applications mobiles sensibles au contexte ;
— être suffisamment souple pour permettre une extension et prendre en compte de nouvelles entités spécifiques à un domaine d'application donné.

En plus de ces trois exigences, nous proposons de spécifier certaines informations de contexte plus précisément, en particulier dans le cadre du *MCSC*. C'est le cas des données contextuelles de type *profil* et *environnement* qui doivent être définies précisément en particulier pour les applications mobiles sensibles au contexte comme c'est le cas des systèmes de recommandation à des utilisateurs mobiles évoluant dans la ville.

En ce qui concerne ces applications mobiles sensibles au contexte, notre modèle se concentre sur le *who*, le *where*, le *when* et le *what* (c.-à-d., quelles activités se déroulent) et utilise cette information pour déterminer le *why* [Kru09] (c.-à-d., pourquoi une situation se produit). Parmi ces *5w*, le *why* devrait donc être déduit des quatre autres « *w* » (*who*, *where*, *when*, *what*) pouvant être capturés : l'utilisateur, son emplacement, l'heure et son activité.

Ainsi, la Figure 6.1 illustre notre proposition de représentation du contexte générique. Nous décrivons ci-dessous les différentes entités qui composent cette représentation :

— **Utilisateur :** personne ayant un état et un profil. Un utilisateur évolue dans un environnement et utilise des dispositifs informatiques pour consulter ou recevoir des recommandations d'éléments. L'état d'un utilisateur peut être statique ou mobile ;
— **Profil :** un profil est fortement attaché à l'utilisateur et contient les informations qui le décrivent. Un utilisateur peut avoir un profil statique ou dynamique. Le profil statique rassemble des informations décrivant l'utilisateur et qui ne changent pas (ou très rarement) dans le temps. Cela peut être la date de naissance, le nom, la langue ou le sexe. Au contraire, le profil dynamique de l'utilisateur contient des propriétés qui peuvent évoluer dans le temps et refléter sa situation actuelle p. ex., les préférences, les intentions, les désirs, les contraintes de l'utilisateur. Par exemple, le but d'un touriste à la recherche d'un restaurant est de dîner. Dans ce cas, un profil peut donner des informations sur ses préférences culinaires ou sa localisation ;
— **Activité :** l'activité dans laquelle l'utilisateur est engagé peut être une information clé pour décider quel élément, est pertinent pour lui p. ex., un service dans la ville. Cependant, reconnaître les activités des utilisateurs reste une tâche difficile (voir Chapitre 3) ;
— **Matériel :** c'est le dispositif informatique (dans notre cas : smartphone) de l'utilisateur. Cela nous permet entre autres de capturer des informations contextuelles de l'envi-

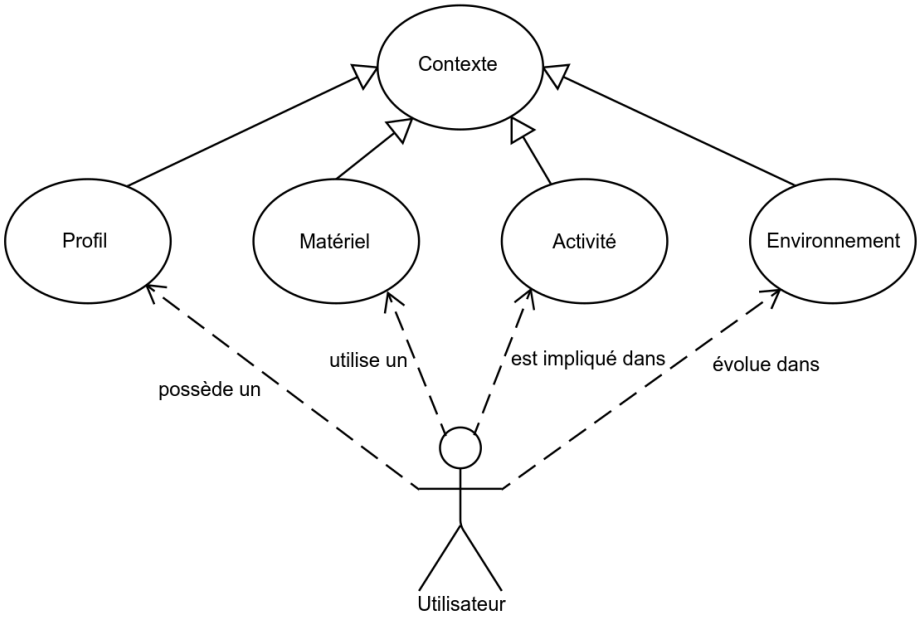


FIGURE 6.1 – Représentation du contexte

ronnement. Le périphérique peut donner des informations concernant son type (p. ex., tablette, ordinateur portable, smartphone), l'application et le réseau ;

— **Environnement :** il contient toutes les informations décrivant l'environnement de l'utilisateur et de son périphérique, qui peuvent être pertinentes pour l'application. Il peut comprendre différents types d'informations telles que :
  — **informations de contexte spatial**, p. ex., Emplacement, ville, destination, vitesse ;
  — **informations de contexte temporel**, p. ex., Temps, date, saison ;
  — **informations de contexte météorologique**, p. ex., Température, type de temps (pluie, soleil, etc.).

6.2.3 Capture du contexte

Notre processus de capture de contexte repose sur l'existence de *Wifilib* (voir la description à la sous-section 6.2.1) et de ces journaux de connexions. Si on se base par exemple sur les données de connexions d'une année en 2017, ceux-ci sont pourvus dans la seule ville d'Angers d'environ $15\ 000\ 000$ connexions d'environ $80\ 000$ utilisateurs à plus de 200 points d'accès.

Pour chaque point d'accès du réseau *Wifilib* à Angers, les journaux contiennent entre autres pour chaque connexion : les coordonnées GPS du point d'accès, le profil anonymisé de l'utilisateur, l'horodatage de déconnexion, la durée de la session (voir Annexe D.1). Ainsi, nos résultats sont calculés en fonction d'un point d'accès spécifique et affiché sur une carte. Lorsqu'ils sont croisés avec les informations d'un service de cartographie, nos résultats peuvent nous aider à découvrir p. ex. des points d'intérêt comme les magasins, les trajets de tramway, les gares, les lieux et rues fréquentés.

Plus précisément, les données capturées sont exploitables sous forme de fichiers journaux au format Json et sont consultables en Annexe D.1.

6.2.4 Visualisation et analyse des informations de contexte brut capturées

Cette partie de notre étude est la phase de *Data Story Telling* que tout *Data Scientist* doit effectuer afin d'expliquer les données et résultats qu'il observe. Après l'étape de capture du contexte *(MCS)*, nos données (profils, géolocalisation) peuvent être analysées avant d'être fournies sous une forme structurée et exploitable à un algorithme de recommandation contextuel.

6.2.4.1 Notre outil : *Urban Mobility Visualizer* (*Ur-MoVe*)

Nous avons examiné et interprété les données de profil et de géolocalisation afin de comprendre ce qu'elles représentent dans le monde réel. Afin de réaliser notre analyse, nous avons développé *Ur-MoVe* (*Urban Mobility Visualizer*), un outil de visualisation de la mobilité et de la dynamique urbaine.

Notre outil est capable de visualiser la fréquentation de lieux des utilisateurs et de prédire leur mobilité par l'analyse des journaux de connexion au réseau Wi-Fi de la ville. Avec ces données, nous pouvons observer et prédire divers indicateurs concernant l'activité du réseau mais aussi travailler sur la dynamique urbaine. L'outil *Ur-MoVe* se constitue à ce titre comme un point d'entrée intéressant permettant de bien comprendre la fréquentation, la mobilité et les comportements des utilisateurs. Il propose : l'affichage d'informations de contexte brut capturé comme par exemple le nombre de personnes connectées à un instant t à une borne Wi-Fi donnée, la durée de connexion, le sexe des utilisateurs, ou encore la langue utilisée etc. *Ur-MoVe* permet aussi d'autres raisonnements contextuels plus complexes comme l'étude des trajectoires, ou encore des flux de déplacement que nous présenterons au Chapitre traitant du raisonnement contextuel (Chapitre 7).

Nous présentons les visualisations de contexte brut de *Ur-MoVe* Figure D.1. Pour cela nous avons utilisé le moteur de cartographie *Leaflet* utilisant la technologie *javascript* que nous avons couplé au fond de carte *OpenStreetMap (OSM)*. Chaque information spécifique est observée sur chaque point d'accès Wi-Fi et affichée sur la carte à son emplacement.

Les différents résultats de la Figure D.1 (en annexe) sont décrits ci-dessous :

1. **Densité des connexions :** la densité des connexions pour chaque point d'accès Wi-Fi est représentée sur une échelle progressive (dégradé) de couleurs allant du jaune au rouge foncé en fonction du nombre d'utilisateurs connectés à l'heure sélectionnée. Ces résultats nous permettent de déterminer les lieux les plus fréquentés et de déduire des périodes spécifiques de la journée, telles que les heures de pointe. De plus, ces données peuvent nous permettre de calculer plus en détail les flux dans la ville. Ils seront notamment nécessaires pour prédire la position d'un utilisateur (voir Chapitre 7) ;

2. **Durée des connexions :** une indication sur la durée des connexions aux différents points d'accès est très importante lors de la catégorisation des lieux et des rues afin d'identifier p. ex., les lieux de passage, ou les lieux où on s'arrête. Par exemple, dans la Figure D.1 représentant le centre-ville d'Angers, on remarque que les utilisateurs restent plus longtemps sur la *place du Ralliement* du centre-ville d'Angers ou encore dans les rues commerçantes où il y a des cafés, des restaurants, des magasins dans lesquels ils peuvent s'arrêter. A contrario, des rues, telles que *rue Saint-Laud*, apparaissent comme des voies de passage avec des durées de connexion courtes aux points d'accès Wi-Fi ;
3. **Principaux langages :** dans notre exemple, nous remarquons les utilisateurs francophones en bleu et les utilisateurs anglophones en rouge. Cette information pourrait être très pertinente pour un système de recommandations contextuel qui, à partir des quartiers fréquentés par des utilisateurs parlant une langue donnée prépondérante, pourrait tirer partie de cette entité géographique particulièrement typée. Cela pourrait être notamment intéressant pour les applications du domaine du tourisme ou encore des villes possédant des regroupements géo-ethniques où la langue d'origine est couramment parlée (par exemple *Chinatown* à *New York*) ;
4. **Échelles d'âge :** les âges sont regroupés par décades de 0 à 100 ans et sont affichés pour chaque point d'accès sur une échelle de couleurs allant du jaune au rouge en fonction de l'âge moyen des utilisateurs connectés. Si elles sont utilisées en combinaison avec l'heure, ces informations peuvent être très pertinentes pour classer les quartiers en fonction de l'âge des habitants de la région et du moment de la journée. Par exemple, la figure montre que davantage de personnes âgées de 40 à 50 ans fréquentent un «Grand Magasin de Culture et Technologies», alors que les utilisateurs âgés de 20 à 30 ans ont tendance à se détendre dans les bars et les cafés disposés aux alentours de la *place du Ralliement* ;
5. **Sexes :** notre application nous permet d'afficher les connexions aux points d'accès par sexe. Les points d'accès auxquels une majorité de femmes sont connectées sont représentés en rose et deviennent bleus lorsque la majorité des hommes sont connectés. Dans la Figure D.1, nous remarquons qu'une catégorie de personnes a tendance à se trouver dans des lieux tels que des pubs, des cafés ou des magasins vendant entre autres de la technologie, tandis qu'une autre catégorie préfère les boutiques de vêtements. Le nombre exact d'utilisateurs appartenant à chaque sexe connecté au point d'accès Wi-Fi peut être dévoilé en cliquant dessus ;
6. **Graphes des routes :** nos données peuvent également être utilisées pour déduire des informations contextuelles plus complexes telles que des graphes de routes. Ceux-ci donnent des indications sur les itinéraires les plus empruntés de la ville. Nous déterminons les itinéraires d'utilisateurs en considérant toutes les connexions entre les temps t et $t + 15min$. Nous déduisons ensuite de ces itinéraires d'utilisateurs un graphe des routes. Ces informations contextuelles brutes sont le point d'entrée incontournable pour calculer par exemple des top-k routes (itinéraires les plus empruntés par les utilisa-

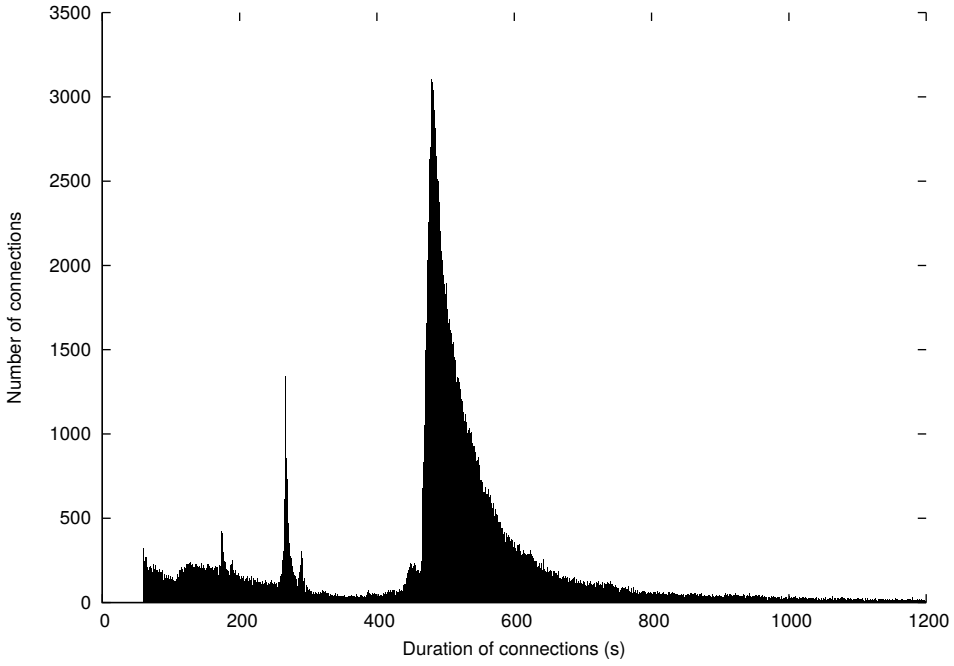


FIGURE 6.2 – Distribution de la durée des connexions (Du 1^{er} au 4 Oct. 2015)

teurs dans la ville) ou encore pour déterminer les matrices de transition utilisées dans le traitement des chaînes de Markov pour la prédiction de la mobilité. Nous traiterons ultérieurement de ces deux raisonnements contextuels au Chapitre 7.

6.2.4.2 Analyses temporelles

Il est également important de réaliser une analyse temporelle afin de déterminer la période de capture moyenne pour chaque utilisateur mobile détecté ainsi que les flux globaux de connexions sur 24 heures. Ainsi, nous avons observé la durée moyenne de connexions qui est de 8 minutes et 2 secondes et un écart type de 33,47 secondes dans la population étudiée (voir Figure 6.2).

De plus, nous avons montré qu'il n'y avait aucune relation entre la durée des connexions et l'heure de la journée (voir Figure 6.3) :

— il n'y a pas de corrélation linéaire entre l'heure du jour et la durée des connexions c.-à-d., corrélation de *Bravais-Pearson* $r(T, D) = 0,0002$ avec

$$r(T, D) = \frac{\mathrm{Cov}(T, D)}{\sigma_T \sigma_D}$$

où T est l'horodatage de la connexion et D la durée de la connexion ;

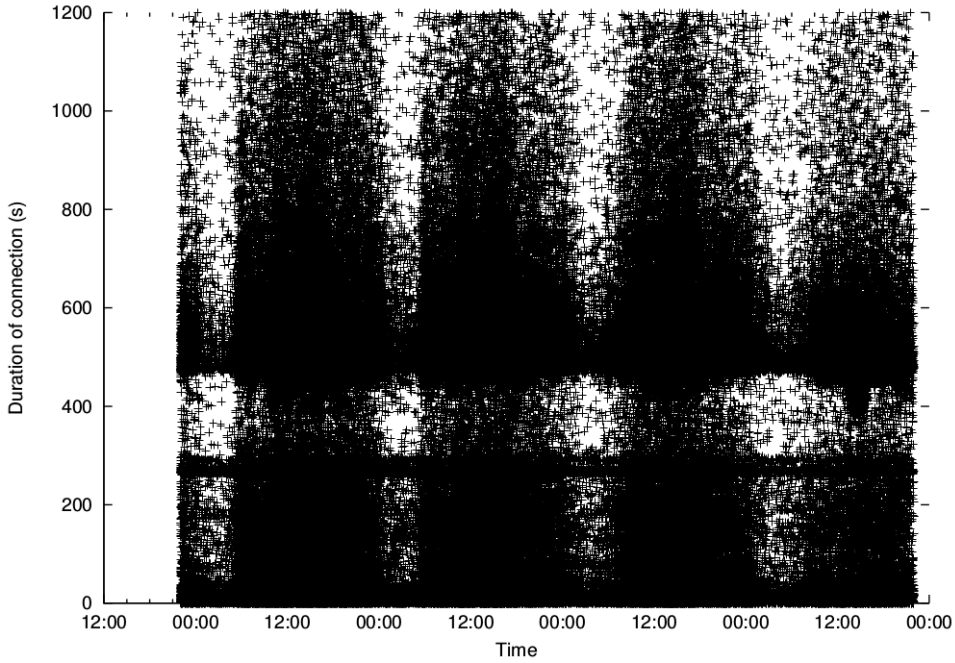


FIGURE 6.3 – Durées de connexions vs horodatages (Du 30 Sept. au 4 Oct. 2015)

— il n'y a pas de corrélation non linéaire entre l'heure du jour et la durée des connexions c.-à-d., corrélation de rang de *Spearman* $r_s(T, D) = 0,0003$ avec

$$r_s(T, D) = 1 - \frac{6. \sum_{i=1}^{N} (rg(T_i) - rg(D_i))^2}{N^3 - N^2}$$

où N est le nombre d'observations, $rg(T_i)$ est le rang de T_i dans la distribution $T_1..T_n$ et $rg(D_i)$ est le rang de D_i dans la distribution $D_1..D_n$;

— nous avons observé graphiquement à la Figure 6.3 qu'il n'y a pas non plus de relations non monotones.

Ces informations sont également intéressantes à prendre en compte pour de futurs raisonnements contextuels. En effet, pour le calcul de prédiction de la mobilité que nous effectuerons pour chaque utilisateur donné (voir Sous-section 7.2.2), il ne semblera pas pertinent d'aller au-delà de 15 minutes de prédiction. De plus, ces analyses nous permettent aussi d'observer que cette règle des 15 minutes s'appliquera à n'importe quel moment de la journée. En revanche, ce ne sera pas le cas la nuit, voir Figure 6.3 où nous observons qu'il y a moins de connexions et que les utilisateurs restent connectés pendant des périodes plus courtes (Voir Figures 6.2 et 6.4).

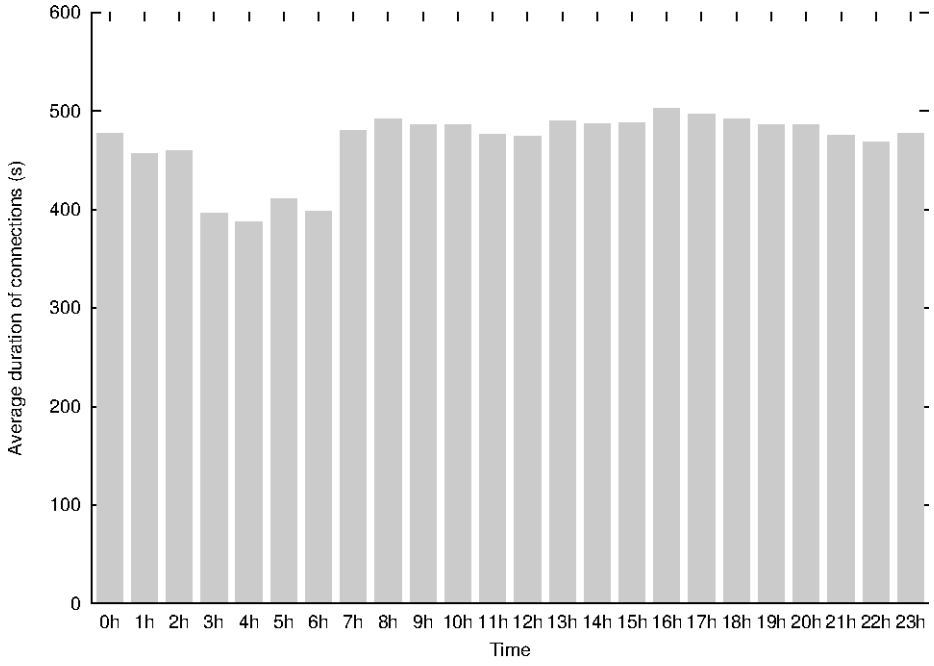


FIGURE 6.4 – Durée moyenne de connexions en fonction du moment de la journée (1^{er} Oct. 2015)

6.2.5 Conclusion et Perspectives

Dans cette section, nous avons donné une description globale de notre modélisation du contexte et des informations contextuelles brutes qui peuvent être tirées d'un composant *MCS* capturant le contexte d'utilisateurs mobiles connectés aux points d'accès Wi-Fi de la ville d'Angers (France).

De plus, nous avons mis en lumière *Ur-MoVe*, un outil que nous avons développé afin de visualiser et analyser ces données de contexte brut. *Ur-MoVe* permet plus particulièrement le calcul et la visualisation de plusieurs informations urbaines de contexte telles que la densité de connexions à des points d'accès donnés, ou des informations concernant les profils des utilisateurs connectés à ces points d'accès dans le temps et dans l'espace.

Nous verrons au chapitre suivant qu'à partir des données contextuelles brutes visualisables et analysables dans *Ur-MoVe*, il sera possible de réaliser des raisonnements plus poussés et plus complexes tels que les calculs des itinéraires les plus populaires dans la ville, de prédiction de la mobilité, ou encore de géo-partitionnement (*geo-clustering*). Ces analyses correspondent à des techniques de raisonnements contextuels que nous décrirons au Chapitre 7. Concernant le géo-partitionnement notons tout de même que nous avons pu utiliser les résultats qui en sont issus en les intégrant dans le composant de recommandation de notre application mobile *scéno*. Nous avons pu ainsi obtenir des résultats concrets à partir d'une évaluation en ligne et observer la pertinence d'inférer du contexte spatial dans le cadre de la recommandation d'évè-

nements culturels à des utilisateurs mobiles. Cette évaluation sera décrite dans la Section 7.3 au Chapitre 7.

6.3 Mise en pratique concrète du *Mobile Crowd Sensing* via le projet *Event-AI*

Cette section présente comment nous capturons le contexte dans le cadre du projet *Event-AI* basé à la fois sur notre application mobile *scéno*[4] (voir Figure D.2 en annexe Section D.4) et sur les principes du *Mobile Crowd Sourcing and Computing (MCSC)*.

Après avoir rappelé ce qu'est le projet de recherche *Event-AI* et l'entreprise *scéno*, nous décrirons l'application mobile que nous avons développée et les données contextuelles brutes que nous capturons via notre architecture *MCSC*. Enfin nous conclurons et proposerons des perspectives.

6.3.1 Projet *Event-AI*

Le projet *Event-AI* est aujourd'hui un projet Régional financé par la région Pays de la Loire dans le cadre du programme Atlanstic 2020[5] porté par RFI. Le projet *EVENT-AI* a pu officiellement débuter en février 2019 pour une durée de un an. Néanmoins, avant son lancement officiel, il a nécessité deux années de travail dans le cadre de cette thèse afin d'implémenter un prototype (version Beta) de l'application *scéno* et son système de recommandation contextuel à base de bandits-manchots. Dans cette section et plus généralement dans ce mémoire, nous nous préoccuperons uniquement des résultats obtenus concernant les analyses issues du prototype (période allant de juillet 2018 à juillet 2019). En effet, les résultats des nouvelles cohortes du projet officiel ne seront exploitables qu'en fin d'année 2020 (soit environ un an après leur lancement).

6.3.2 Le *scéno*

Le *scéno*, basé à Angers, met à disposition depuis 2005 des informations culturelles locales. Plus précisément, le produit *scéno* est un agenda de sorties culturelles qu'il répartit en treize catégories : musique/concert, théâtre, danse, littérature, cinéma, conférences, exposition, loisirs, spectacle, festival, cirque, sport, autre. Le *scéno* existe en support papier et web : `sceno.fr`. Ces supports permettant entre autres de consulter les événements culturels disponibles selon une date et une localisation.

4. Disponible sur *Google Play* : `https://play.google.com/store/apps/details?id=sceno.android.pfe.com.sceno&hl=fr`

5. `https://atlanstic2020.fr`

6.3.3 L'application *scéno*

L'application *scéno* possède aujourd'hui 204 utilisateurs ayant renseigné leur profil avec un potentiel de croissance important tant le nombre de lecteurs pour la version papier est lui estimé à 50 000.

En résumé, elle permet à ses utilisateurs de se géolocaliser et de visualiser l'ensemble des événements culturels disponibles autour de leur position. De plus, cette application possède un système de recommandation contextuel d'événements culturels basé sur les bandits-manchots. Ce système de recommandation correspond au composant intelligent de l'application et nécessite que nous capturions les données contextuelles pertinentes sur lesquelles il puisse s'appuyer afin de personnaliser au mieux ses recommandations. Ainsi, au même titre que nous avions pu l'obtenir indirectement avec *Wifilib*, nous devons capturer les informations de contexte mais cette fois en temps-réel et via notre propre composant de *MCS* intégré à l'architecture de l'application *scéno*. L'étude des données contextuelles et la modélisation du contexte que nous avons présentées à la section précédente ont donc été primordiales pour cette phase de mise en pratique du composant *MCS* du projet *Event-AI*.

6.3.3.1 Visualisation

L'application mobile « prototype » actuelle *scéno* (voir Figure D.2 en annexe Section D.4) permet de visualiser les événements culturels autour de la position de l'utilisateur sur les trois prochains jours (voir Figure D.4 en annexe Section D.4). Il est également possible de se projeter sur plusieurs semaines à l'aide du module de calendrier (voir Figure D.5). L'utilisateur à la possibilité d'afficher les événements soit sur une carte, soit sous forme de liste. Par ailleurs, il est possible de cliquer sur chaque événement afin de consulter son détail (voir Figure D.6 en annexe Section D.4). L'utilisateur peut alors consulter, réserver, partager ou encore tracer son itinéraire vers cet événement. D'autres parts, l'application permet aussi de régler ses préférences en correspondance aux catégories d'événements du *scéno* afin de pré-filtrer et afficher uniquement les types d'événements que l'on souhaite visualiser par défaut (voir Figure D.7 en annexe Section D.4).

6.3.3.2 Recommandation et collecte

Les recommandations sont envoyées à l'utilisateur sous forme de notifications via l'application mobile. Elles sont consultables en cliquant sur la petite *cloche* en bas à gauche de l'écran principal. L'utilisateur peut alors d'une part visualiser la recommandation, et d'autre part l'évaluer simplement (Positif ou Négatif) en cliquant sur un émoticône (☺ ou ☹) (voir Figure D.8 en annexe Section D.4).

Dans l'application *scéno*, le problème de la recommandation d'événements culturels dans la version prototype évaluée est posé sous la forme de bandits-manchots (*Multi-Armed Bandit - MAB*) [Aue02 ; LR85 ; SB98] et de bandits-manchots contextuels (*Contextual Multi-Armed Bandits - CMAB*) [LZ08]. Entre juillet 2018 et juillet 2019, nous avons ainsi confronté deux

types d'algorithmes d'apprentissage par renforcement pour résoudre le problème de *MAB* et de *CMAB* : ε-*Greedy* [SB98] pour le problème de *MAB* ; *LinUCB* [Li+10] pour le problème de *CMAB*. Ces algorithmes sont connus dans la littérature comme étant performant pour résoudre ce type de problème [Gal15].

6.3.3.3 Perspectives de l'application *scéno*

L'application *scéno* intégrera à termes Gorthaur avec son porte-feuille d'algorithmes. De même, afin d'apporter une *Interface Homme-Machine (IHM)* encore plus pertinente pour l'utilisateur, un travail d'ingénierie et de prise en compte de l'expérience utilisateur (*UX Design*) est actuellement en cours de réalisation par un ingénieur de développement dédié. Enfin, le module de *MCS* est également en cours d'amélioration afin de pouvoir capturer et raisonner sur des informations contextuelles encore plus variées.

6.3.4 Architecture *MCSC* du projet *EVENT-AI*

Dans cette sous-section nous décrivons l'architecture *MCSC* que nous avons mise en place afin de répondre aux problématiques de capture du contexte pour la recommandation.

Notre architecture *MCSC* actuelle vise à terme à répondre à deux objectifs : 1) collecter et analyser les informations des utilisateurs mobiles de l'application *scéno* tout en garantissant le respect de leur vie privée ; 2) recommander des événements culturels aux utilisateurs selon leur profil et le contexte dans lequel ils évoluent.

L'architecture que nous avons conçue est synthétisée dans la Figure 6.5. Nous la décrivons comme suit :

— le composant ***Base de Données*** contient les événements présents et à venir ainsi que ceux des 14 dernières années. Ce composant permet également de stocker les résultats obtenus par l'algorithme. Ce composant est situé côté serveur de base de données ;
— le composant ***Mobile Crowd Sensing*** capture les données relatives aux utilisateurs mobiles dans la ville. Il s'agit principalement de données de contexte (profils utilisateurs et localisation) sur autorisation des utilisateurs mobiles conformément à la politique de confidentialité de l'application (voir Section D.3 en annexe). Ce composant est directement situé au niveau de l'application mobile *scéno* côté client ;
— le composant ***Computing*** correspond quant à lui au ***système de recommandation contextuel*** permettant de recommander des événements culturels à des utilisateurs mobiles basé sur les bandits manchots. Ce composant est situé côté serveur web et notifie explicitement les utilisateurs via des *webservices* ;
— enfin, ***la sécurité et la vie privée*** sont inhérentes à l'ensemble du système. La sécurité et la confidentialité de l'information sont mises en œuvre dans le cadre d'une politique de confiance numérique conforme au Règlement Général sur la Protection des Données : RGPD (voir l'intégralité de cette politique en Annexe D.3). Après consultation de la politique de confidentialité, l'utilisateur est invité à l'accepter ou à la décliner (Voir Figure D.9 en annexe Section D.4).

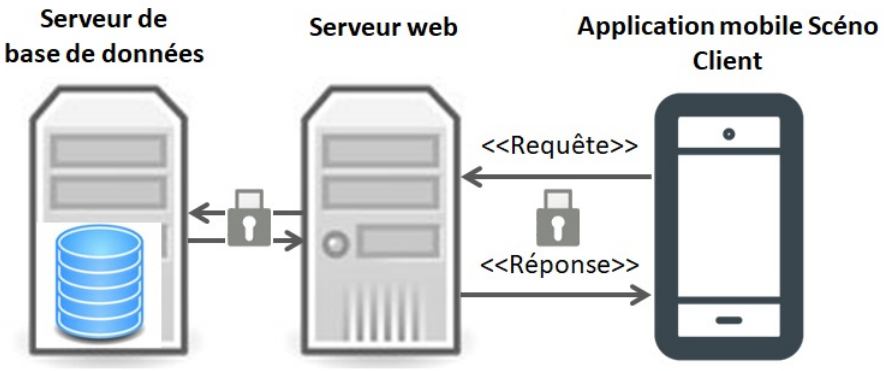


FIGURE 6.5 – Architecture technique *MCSC* du projet *Event-AI*

6.3.5 Informations de contexte capturées

Les utilisateurs mobiles du *scéno* sont encouragés à renseigner un certain nombre d'informations concernant leur profil. Dans tous les cas, nous leur laissons libre choix de renseigner ou non ces informations et de rester uniquement un statut d'utilisateur « invité ».

Dans cette sous-section, nous décrivons les informations de contexte que nous utiliserons ultérieurement dans notre système de recommandation d'événements culturels. Quoi qu'il en soit, les informations de contexte seront transformées en variables catégorielles, puis en variables binaires pour construire un vecteur de contexte de type *"one-hot"* (vecteur binaire). Nous rappelons pour chaque type de données comment nous les avons transformées.

6.3.5.1 Données démographique

Âge. Dans le module de création/mise à jour du profil, l'utilisateur peut renseigner ou non son âge en indiquant sa date de naissance (Voir Figure D.10 en annexe Section D.4). Donner le choix de ne pas renseigner sa date de naissance permet de pallier en partie les fausses informations généralement remplies par défaut. L'information contextuelle de type âge sera ensuite transcrite selon une représentation structurée de l'information afin d'être utilisée par le système de recommandation. Dans le cas de l'âge, nous avons décidé de découper en variables catégorielles par tranches de 5 ans à partir de 18 ans et ce jusqu'à l'âge de 98 ans. Chacune des seize catégories qui en découle est ensuite associée à une seule variable catégorielles à partir de laquelle nous pouvons facilement calculer le vecteur binaire correspondant. Notons donc qu'au total il existe seize vecteurs distincts possibles.

Sexe. Au même titre que pour l'âge, il est possible d'indiquer son sexe (Voir Figure D.10 en annexe Section D.4). Cette variable est découpée en quatre catégories possibles : *{Homme, Femme, Non précisé, Personnalisé}*. Chacune des quatre catégories est associées à une seule

variable catégorielle à partir de laquelle nous pouvons facilement calculer le vecteur binaire correspondant. Notons donc qu'au total il existe quatre vecteurs distincts possibles.

6.3.5.2 Données socio-professionnelles

Dans le module de création/mise à jour du profil de nouveau, l'utilisateur peut renseigner ou non son activité en indiquant sa catégorie socio-professionnelle : *{Agriculteur, Artisan ou commerçant ou chef d'entreprise, cadre et profession intellectuelle, Profession intermédiaire, Employé, Ouvrier, Retraité, Étudiant, Sans activité professionnelle, Non précisé}* (Voir Figure D.11 en annexe Section D.4). Au même tire que l'âge, ou encore le sexe de l'utilisateur, nous pensons qu'il est important de connaître le niveau de catégorie socio-professionnelle pour être plus précis dans les recommandations d'événements culturels. Chaque catégorie socio-professionnelle est associée à une seule variable catégorielle à partir de laquelle nous pouvons facilement calculer le vecteur binaire correspondant. Notons donc qu'au total il existe donc 10 vecteurs distincts possibles.

6.3.5.3 Préférences utilisateurs

Préférences culturelles et filtres. Les utilisateurs peuvent également définir leurs préférences culturelles parmi les 13 catégories existantes : {musique/concert, théâtre, danse, littérature, cinéma, conférences, exposition, loisirs, spectacle, festival, cirque, sport, autre}.

Un mécanisme permet également aux utilisateurs de filtrer les événements qui leur sont affichés (voir Figure D.7 en annexe D.4). Ils peuvent à tout moment choisir de limiter l'affichage aux événements appartenant à des catégories particulières (de une à treize). Notons que cette information pourrait être utilisée comme contexte qui représenterait les préférences du moment.

En ce qui concerne les préférences culturelles renseignées au niveau des écrans de profil (Voir Figure D.12 en annexe Section D.4), nous les limitons au nombre de cinq maximum afin de mieux raffiner les intérêts culturels des utilisateurs. En effet, même si les utilisateurs à un moment donné peuvent choisir de visualiser sur leur carte des événements spécifiques d'une, de plusieurs ou de toutes les catégories, nous conservons malgré tout en mémoire des choix plus pérennes jusqu'à cinq préférences favorites. Le système de recommandation peut donc ensuite s'appuyer sur ces préférences de long terme.

Rayons de recherche. Dans l'application mobile, les utilisateurs ont un certain nombre de paramétrages possibles dont celui du rayon de recherche des événements culturels pouvant varier de 1 à à 50km (Voir Figure D.13 en annexe Section D.4). Outre la fonction de filtrage que nous offre cette fonctionnalité, il semble également intéressant de prendre en compte les recommandations en fonction du rayon de recherche paramétré. Ainsi, nous définissons trois catégories de rayon : {proche : < 5km ; raisonnable : ≥ 5km et < 20km ; éloigné : ≥ 20km}.

6.3.5.4 Données d'environnement

Données météorologiques et temporelles Comme nous pouvons l'observer Figure D.3 en annexe D.4, l'application *scéno* récupère des informations sur les prévisions météorologiques par périodes de 3 heures sur trois jours consécutifs et les affiche dans l'écran principal de visualisation des événements dans la tranche horaire sélectionnée. En revanche, dans la version prototype, l'information météorologique n'est pas intégrée dans l'écran de recommandation en tant qu'information pouvant influencer la décision de l'utilisateur sur la pertinence de la recommandation.

Au vu du caractère incontournable de cette donnée contextuelle, en perspective de la livraison de la version finale de l'application *scéno* début 2020, nous souhaitons pouvoir faire bénéficier l'utilisateur de cette information dans l'écran de recommandation et de pouvoir ainsi la considérer dans le contexte observé par le système de recommandation.

L'une des pistes envisagées est de partir de la hiérarchie du modèle correspondant au domaine des conditions météorologiques [HMP12 ; SPV07] afin de construire les variables catégorielles à partir des données météorologiques capturées. Dans ce modèle, les conditions météorologiques sont des attributs pouvant être hiérarchisées selon leur niveau de détails N_i auquel elles correspondent. Formellement, une hiérarchie d'attributs est un treillis $(N, \prec) : N = (N_1, ..., N_{ALL})$ de n niveaux et $\prec$ est un ordre partiel entre les niveaux de N tel que $\forall 1 < i < n, N_1 \prec N_i \prec N_{ALL}$ [SPV07]. Ceci a pour signification que $\forall i \in [1; n]$ les concepts représentés par N_i sont plus précis que ceux représentés par N_{ALL}. Ainsi les conditions météorologiques pourraient être représentées ainsi : {très froid, froid, doux, chaud, très chaud} concernant les conditions détaillées de niveau N_1, {bonne ou mauvaise} pour la caractérisation de ces conditions de niveau N_2, et ALL qui regroupe toutes les valeurs en une seule de niveau N_{ALL} [HMP12]. Notons ainsi que dans ce treillis $N_1 \prec N_2 \prec N_{ALL}$.

Géolocalisation L'application mobile *scéno*, permet de récupérer la localisation de l'utilisateur afin de lui afficher les événements culturels à proximité (Voir Figure D.4 en annexe Section D.4). C'est la seule condition obligatoire pour que l'application fonctionne. En revanche, même si cette information est facilement utilisable pour afficher les événements culturels alentours selon un rayon donné, son intégration dans un système de recommandation nécessite un raisonnement plus complexe.

Il existe une hiérarchie du modèle concernant la localisation : c.-à-d., Pays (N_3), Région(N_2), Ville (N_1) [SPV07], où il pourrait devenir intéressant dans le cadre de notre système de recommandation d'événements culturels, de descendre au niveau du quartier [Cra+12]. Dans le Chapitre 7, nous présentons une approche de partitionnement spectral pour la détection de quartier afin de pouvoir exploiter l'information de géolocalisation capturée.

6.3.6 Conclusion

Dans cette section, nous avons dans un premier temps décrit le projet *Event-AI* qui repose sur le paradigme *MCSC*. Nous en avons rappelé l'intérêt et les perspectives qu'il offre dans le domaine des systèmes de recommandations à des utilisateurs mobiles.

Nous avons également décrit l'application mobile *scéno* autour de laquelle s'articule le projet *Event-AI*. Cette application permet la visualisation et la recommandation d'événements culturels dans la ville intelligente.

Nous avons ensuite expliqué l'architecture *MCSC* du projet *EVENT-AI* déployé pour l'application *scéno*.

Enfin, nous avons détaillé les informations contextuelles brutes capturées par notre module de *MCS* et comment nous les avons transcrites en représentation structurée pour un algorithme de bandits-manchots contextuels pour la recommandation.

6.4 Bilan et perspectives

Dans ce chapitre nous avons mis en lumière l'intérêt de capturer et analyser l'information de contexte dans la ville pour les systèmes de recommandation à des utilisateurs mobiles.

Pour ce faire, dans la première section de ce chapitre, nous avons proposé un modèle de contexte, puis nous avons visualisé et analysé des données contextuelles de la ville d'Angers à partir de sources de fichiers journaux de connexions aux points d'accès Wi-Fi urbain de la ville.

Dans la seconde section de ce chapitre, nous avons mis en pratique le principe de capture de contexte, mais cette fois de manière dynamique dans le cadre d'un projet de recherche et développement nommé *Event-AI*. Le premier prototype de ce projet, l'application mobile *scéno*, permet la visualisation et la recommandation d'événements culturels à des utilisateurs mobiles dans la ville d'Angers. Cette seconde section a également été l'occasion de décrire les tenants et aboutissants du projet *Event-AI*, de son application concrète, de son architecture *MCSC* dédiée, ainsi que la description précise des informations contextuelles brutes capturées pour le système de recommandation de l'application mobile.

Dans le chapitre suivant (Chapitre 7) nous décrirons comment nous raisonnons sur des informations contextuelles brutes, permettant d'enrichir les systèmes de recommandation de façon à leur fournir des caractéristiques de contextes plus pertinentes et plus exploitables. Ceci peut s'avérer parfois incontournable, comme c'est le cas notamment dans la prise en compte de la localisation des utilisateurs.

Ainsi, dans le prochain chapitre nous présenterons trois méthodes de raisonnement contextuel que nous avons mises en oeuvre à partir des données *Wifilib*. Nous en détaillerons une plus particulièrement : le géo-partitionnement. Les résultats obtenus par cette méthode (quartiers déduits de la ville d'Angers) ont pu être intégrés dans le système de recommandation de l'application mobile *scéno* et ont donc pu être évalués en ligne. Enfin, en dernière section du prochain chapitre, nous décrirons comment il est possible de déduire des préférences uti-

lisateurs afin d'enrichir le contexte fourni aux systèmes de recommandation. Cette dernière étude a fait l'objet pour le moment d'une évaluation hors ligne sur plusieurs jeux de données du monde réel.

CHAPITRE 7

RAISONNEMENTS CONTEXTUELS ET APPLICATION AUX SYSTÈMES DE RECOMMANDATION

Context Reasoning

« *Context-aware applications use context information to evaluate whether there is a change to the user and/or computing environment context ; taking a decision whether any adaptation to that change is necessary often requires reasoning capabilities.* [Bet+10]
Claudio Bettini et al. - 2010 »

Sommaire

7.1 Introduction

Ce chapitre fait référence à nos contributions [Gut+17 ; Gut+18d] (étoile n° 1) pour la Section 7.2, [Gut+18c] (étoile n° 3) ainsi qu'à [Gut+19c] (étoile n° 8) pour la Section 7.3, [Gut+18b ; Gut+19d] (étoiles n° 4 et 5), pour la Section 7.4, présentées à la Figure 2 en introduction de ce mémoire.

Au chapitre 6, nous avons décrit comment il était possible de modéliser, capturer et déduire de l'information de contexte brut dans la ville, sans raisonnement contextuel complexe. Or, dans certains cas pratiques comme c'est le cas pour les systèmes de recommandation, il devient nécessaire de rendre lisible et exploitable toute donnée de contexte brut qui resterait inutilisable en l'état par le système. C'est le cas notamment de l'information de localisation des utilisateurs mobiles, qui sans raisonnement contextuel reste inexploitable [SDO18].

Ainsi, ce chapitre est dédié à la présentation de quatre méthodes de raisonnement contextuel. Deux de ces méthodes ont pu être intégrées et évaluées dans des systèmes de recommandation. Ce chapitre est donc organisé comme suit :

- dans la Section 7.2, après avoir rappelé un état de l'art de l'étude de la dynamique urbaine dans le cadre des *MCSC*, nous présentons deux méthodes de raisonnement contextuel sur la mobilité dans la ville, à partir du jeu de données *Wifilib* (voir sous-section 6.2.1 pour la description du jeu de données). La première méthode, utilise les graphes des routes présentés dans la sous-section 6.2.4, afin de déduire des résultats plus complexes, tels que des itinéraires les plus fréquentés, permettant entre autres de mieux comprendre la mobilité urbaine. La seconde méthode utilise des chaînes de *Markov*, afin d'estimer où un nouvel utilisateur qui se connecte à un point d'accès donné à t_0 sera entre $t_0{+}1\ min$ et $t_0{+}15\ min$;
- dans la Section 7.3 nous nous intéressons à déduire des quartiers de la ville à partir des journaux de connexions au réseau Wi-Fi urbain des utilisateurs mobiles sur plusieurs villes en France. Pour ce faire, nous employons une méthode d'apprentissage non supervisé : le partitionnement spectral, en utilisant la technique des k-moyennes pour définir des zones géographiques regroupant les points d'accès selon leur fréquentation. Nous avons publié les résultats obtenus par cette méthode sur un site web : `http://www.wifilib-clustering.info/`, en y mettant également à disposition de la communauté un échantillon anonymisé de notre jeu de données. De plus, dans cette section nous présentons une évaluation en ligne des résultats de géo-partionnement dans la ville d'Angers (quartiers déduits) que nous avons intégré dans le système de recommandation de l'application mobile *scéno* présentée au Chapitre 6 ;
- dans la Section 7.4, nous proposons une nouvelle méthode de raisonnement et d"amélioration du contexte nommée *Individual Context Enrichment - ICE*, s'associant aux algorithmes de bandits-manchots contextuels (*Contextual Multi-Armed Bandit - CMAB*) pour la recommandation. *ICE* permet aux algorithmes de *CMAB* de s'appuyer sur d'autres caractéristiques de contexte pertinentes. Ces caractéristiques sont déduites selon la mesure de précision individuelle des utilisateurs (mesure définie au Chapitre 4) vis à vis des recommandations qui leur sont faites, dans des contextes donnés que nous définissons comme étant des paires utilisateur-contexte (u, x).

7.2 La mobilité urbaine

Dans les applications *MCSC*, l'étude de la dynamique urbaine fait l'objet de nombreux travaux qui analysent la mobilité et les comportements liés à l'activité humaine dans les zones urbaines. Différentes méthodes sont en cours d'élaboration et utilisées pour réaliser ces études. Par exemple, dans [Nou+12], les auteurs déduisent des modèles de mobilité depuis les historiques de notifications géolocalisées d'utilisateurs de réseaux sociaux de type *LBSN (Location Based Social Network)*. Certaines études utilisent par exemple les *LBSNs* pour la recommandation d'événements socioculturels [Que+10], ou pour découvrir les trajectoires agrégées des

utilisateurs [ZZ11] ou encore pour mettre en évidence des typologies de quartier dans la ville [Cra+12]. Cependant, une grande quantité de données reste nécessaire pour tirer des conclusions suffisamment précises, significatives et représentatives. C'est pourquoi ces méthodes ne sont efficaces et utilisables que dans le cas d'un usage important des réseaux sociaux par les utilisateurs. D'autres méthodes se basant sur l'utilisation du Wi-Fi dans la ville ont été utilisées et peuvent être une alternative permettant de suivre les téléphones mobiles et estimer leurs trajectoires. C'est notamment le cas de [ME12] qui démontre comment, en s'appuyant sur des traces de connexion Wi-Fi, les chaînes de *Markov* peuvent être utilisées pour estimer des trajectoires.

Les deux méthodes de raisonnement contextuelles que nous proposons dans cette section fait référence à nos contributions [Gut+17 ; Gut+18d]. Ces deux méthodes s'appuient sur le jeu de données *Wifilib* que nous avons décrit au Chapitre 6.

7.2.1 Top-k routes.

À partir du jeu de données *Wifilib* (voir sous-section 6.2.1 pour la description du jeu de données), notre objectif est de mettre en évidence une approximation des itinéraires (c.-à-d. association de chemins et de routes empruntés) les plus parcourus à une date et une période données. Ce raisonnement constitue l'un des points d'entrée permettant de déduire des schémas de mobilité dans la ville.

Différentes approches sont décrites dans la littérature afin d'obtenir ces résultats. Par exemple, [Bao+15 ; Han+14] décrit comment des top-k trajectoires peuvent être calculées à partir des notifications issues des réseaux sociaux basées sur la localisation de l'utilisateur *(LBSN)* ou du temps de trajet. La plupart des méthodes proposées utilisent la théorie des graphes pour résoudre efficacement ces problèmes.

Par conséquent, nous calculons au préalable le graphe de routes correspondant à un intervalle de temps donné. Le graphe de routes est construit en identifiant toutes les arêtes. Une arête correspond à la transition entre une déconnexion d'un point d'accès suivie d'une connexion à un autre, sur une période donnée. Le poids de chaque arête est initialisé à 1. Il est ensuite incrémenté chaque fois qu'une transition entre deux mêmes points d'accès est observée. Le graphe de routes résultant est le point de départ de tous les calculs liés aux itinéraires. Ainsi, les itinéraires les plus empruntés dans un intervalle de temps donné sont calculés comme suit :

1. Pour chaque utilisateur $u \in \mathcal{U}$, nous déterminons la liste $\mathcal{L}_u$ de tous les itinéraires $l \in \mathcal{L}_u$ des quinze dernières minutes. Nous considérons un itinéraire utilisateur l comme étant un ensemble d'au plus cinq de ses connexions $c_{u,i}; \forall i \in [1,5]$ aux différents points d'accès a_i ordonnancés selon leur horodatage sur l'intervalle $T = [t, t + 15min]$. Soit $\mathcal{L} = \bigcup_{\forall u \in \mathcal{U}} \mathcal{L}_u$ l'ensemble de tous les itinéraires de 4 segments $f_{u,j}; j \in [1,4]$ pour tous les utilisateurs ;
2. Nous sommons ensuite l'ensemble des flux de chaque segment f_j pour tous les itinéraires $l \in \mathcal{L}$. Ceci correspond aux flux d'utilisateurs entre chaque paire de point d'accès

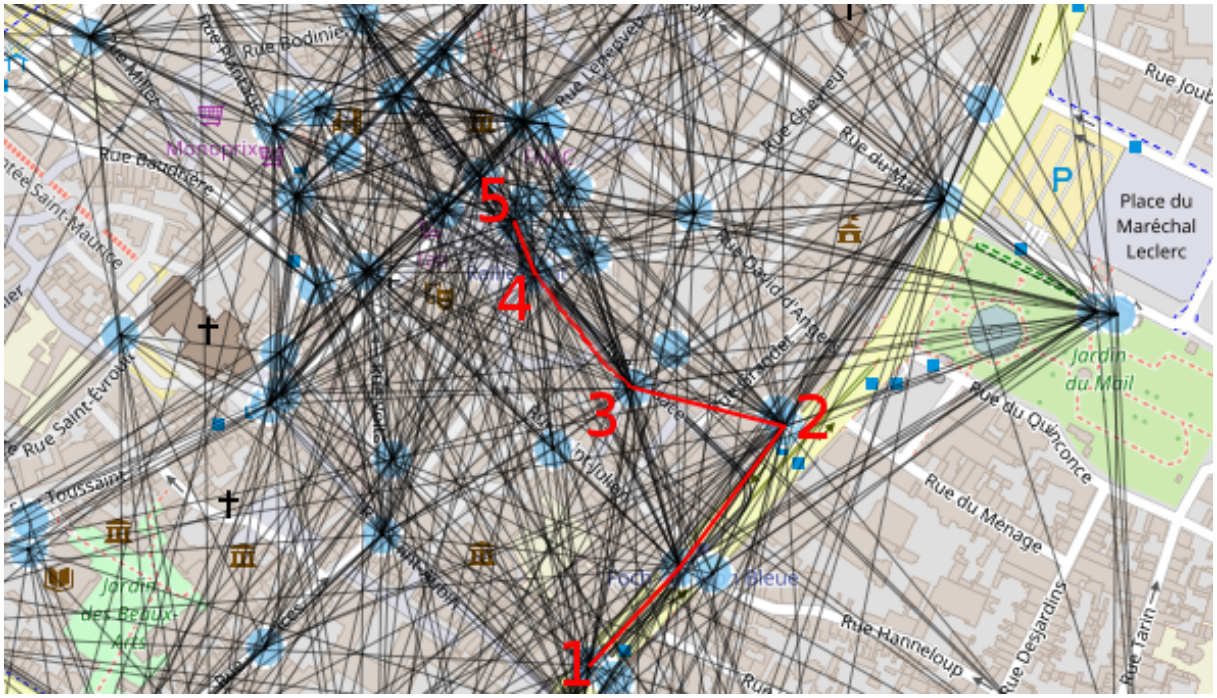


FIGURE 7.1 – Itinéraire le plus emprunté à Angers le 03/10/2015 entre 16h00 et 16h15 selon les flux de connexions Wi-Fi

entre t et $t + 15min$. Cela nous permet de mettre en évidence les segments les plus empruntés pour un itinéraire approximatif l final recalculé ;

3. Nous déterminons ensuite l'itinéraire le plus emprunté en sélectionnant

$$\arg\max_{f_j \in l} \sum_{j=1}^{5} f_j$$

La Figure 7.1 nous permet par exemple d'observer l'itinéraire le plus parcouru à Angers le Dimanche 3 Octobre 2015 entre 16h00 and 16h15.

En croisant nos connaissances sur la ville d'Angers et l'itinéraire calculé le plus emprunté, nous remarquons que celui-ci correspond à la ligne de tramway de la ville. Ceci correspond ainsi aux principaux lieux de fréquentation et aux flux que nous nous attendions à obtenir à l'heure et au jour donnés.

7.2.2 Prédictions de mobilité

Dans le cadre de cette thèse, nous n'avons pas eu l'opportunité d'appliquer la prédiction de mobilité à un système de recommandation en ligne, et par conséquent nous n'avons pas pu l'évaluer. Néanmoins, nous soutenons qu'à terme il sera pertinent d'intégrer au contexte à la

fois la position courante de l'utilisateur mais aussi prédite. En effet, les utilisateurs intéressés par un service géolocalisé (p. ex., événement culturel, passage du prochain tram) ont plus de chance d'y répondre favorablement s'ils vont dans la direction où se déroule ce service recommandé.

Ainsi, notre objectif consiste à prédire l'emplacement futur d'un utilisateur en partant du principe que l'information de localisation prédite pourra être intéressante dans le cadre des systèmes de recommandation de services à des utilisateurs mobiles dans la ville.

Au chapitre 6, nous avons effectué un travail préliminaire qui visait à la fois à étudier différents aspects des informations contextuelles tirées des connexions au Wi-Fi urbain et à comprendre l'échelle de temps dans laquelle les utilisateurs évoluent. Comme nous prévoyons d'utiliser des matrices de transition de *Markov* dynamiques (c.-à-d., qui évoluent dans le temps), nous nous sommes basés sur notre étude préliminaire des durées de connexions (voir nos analyses temporelles au Chapitre précédent 6.2.4.2) afin de déterminer à la fois la période de rafraîchissement de ces matrices et le moment, dans le futur, pour lesquels il est pertinent de calculer la prédiction d'emplacement. À la lumière de ces analyses, il semble qu'une période de prédiction de mobilité pouvant aller de 1 à 15 minutes, et un rafraîchissement des matrices de transitions toutes les minutes soient le plus pertinent.

7.2.2.1 Énoncé du problème

Pour résoudre notre problème de prédiction, nous proposons d'utiliser des chaînes de *Markov*. Nous considérons l'ensemble dénombrable $\mathcal{A} = A \cup \varepsilon$ comme étant l'ensemble des états de la chaîne de *Markov*, où A est l'ensemble des points d'accès Wi-Fi ($\{a_i, a_j, a_k, ...\} \in A$) et ε représente un point d'accès *imaginaire* où seront « connectés » les utilisateurs qui ne se reconnecteront à aucun point d'accès physique réel de la ville au prochain pas de temps $t+1$ (c.-à-d., ceux qui se déconnectent d'un point d'accès et ne se reconnectent jamais à un autre). Ainsi, chaque $a_i \in \mathcal{A}$ est considéré comme un état de $\mathcal{A}$ qui représente notre ensemble dénombrable d'espace d'états.

Soit un espace de probabilité $(\Omega, \mathcal{F}, P)$. Ω est l'ensemble de résultats, $\mathcal{F}$ est une tribu de Ω, et $\forall \phi \in \mathcal{F}$, $P(\phi)$ est la mesure de probabilité de ϕ. Dans notre étude de mobilité nous observons une séquence de variables $X_0, X_1, ...$ prenant leurs valeurs dans $\mathcal{A}$. Ainsi, une variable observée X avec des valeurs dans $\mathcal{A}$ est une fonction $X : \Omega \rightarrow \mathcal{A}$. Ensuite, on observe un vecteur ligne $\lambda = (\lambda a_i : a_i \in \mathcal{A})$ qui est considéré comme une mesure si $\forall a_i, \lambda_{a_i} \geq 0$. Dans notre problème de prédiction de la prochaine connexion, nous commençons par une distribution initiale sur $\mathcal{A}$, spécifiée par $\lambda_{a_i} : a_i \in \mathcal{A}$ telle que $0 \leq \lambda_{a_i} \leq 1$ pour tout a_i et $\sum_{a_i \in \mathcal{A}} \lambda_{a_i} = 1$.

Ensuite nous considérons chaque nouvelle connexion dans $\mathcal{A}$ pour déterminer à quel a_i notre utilisateur est connecté. Cette nouvelle connexion observée d'un utilisateur, nous informera qu'il y a une probabilité de 1 que celui-ci soit dans l'état a_i au moment t_0. Nous notons ainsi le vecteur ligne résultant de cette information comme suit : $\lambda = \delta_{a_i} = (0, ..., 1, ..., 0)$. À partir de cette mesure initiale, pour chaque nouvelle connexion utilisateur à l'instant t_0, nous

pouvons ainsi calculer les probabilités de connexion de l'utilisateur à un autre $a_j \in \mathcal{A}$ à partir des pas de temps de $t+1\ min$ à $t+15\ min$.

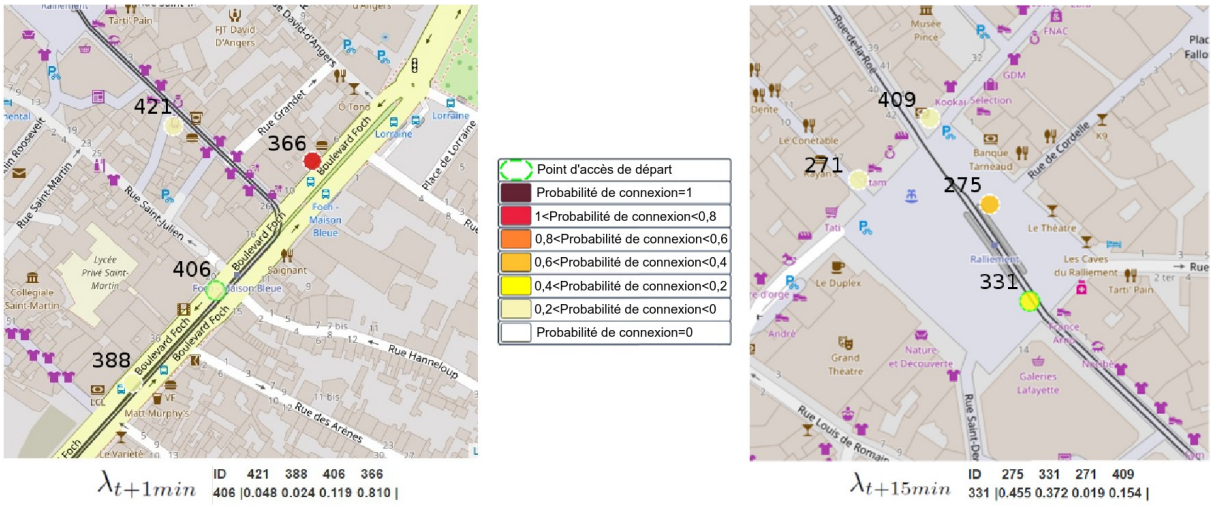
Pour ce faire, nous calculons la matrice de transition de *Markov* P. Ainsi, nous exploitons les données issues des fichiers journaux de connexions pour calculer : 1) les flux entre chaque paire de points d'accès ; 2) le nombre d'utilisateurs restant au même endroit ; 3) le nombre d'utilisateurs ne se reconnectant pas. Cela nous permet de déduire la probabilité de changement d'état dans le temps. Pour chaque nouvelle connexion d'un utilisateur à un point d'accès $a_i \in \mathcal{A}$ au temps t_0, nous calculons les probabilités qu'au moment $t_0+k; k \in [1,15]$, l'utilisateur sera soit : 1) connecté à un autre point d'accès ; 2) toujours connecté à a_i ; 3) déconnecté et aura donc atteint l'état ε. Ainsi, soit P la matrice de transition, $P = (p_{a_i a_j} : a_i, a_j \in \mathcal{A})$ avec $\forall a_i, a_j, p_{a_i a_j} \geq 0$ et $\sum_{a_j \in \mathcal{A}} p_{a_i a_j} = 1$ (chaque ligne de P est une distribution sur $\mathcal{A}$). La matrice P est donc une matrice stochastique. Dans notre cas, $(X_n)_{n\geq 0}$ est une chaîne de *Markov* avec la distribution initiale λ et la matrice de transition P évolue chaque minute de P_{t_0} à $P_{t_0+15\ min}$.

En termes plus pratique, nous calculons les probabilités $p_{a_i a_j}$ de la matrice de transition en considérant chaque flux entre chaque paire de points d'accès dans $\mathcal{A}$. Ces flux sont donnés en pourcentages puis transcrits en probabilités. La matrice de transition est mise à jour chaque minute afin de limiter les erreurs pouvant être induites par la non stationnarité. Ainsi, supposons que u soit un nouvel utilisateur connecté dans n'importe quel état $a_i \in \mathcal{A}$ au temps t_0, alors $\lambda = \delta_{a_i} = (0, ..., 1, ..., 0)$. Nous pouvons calculer ses prochaines probabilités de changement d'état en utilisant la matrice de transition comme suit : $\lambda_{t_0+1\ min} = \delta_{a_i} P_{t_0}$ qui est le vecteur ligne de probabilité au temps $t_0 + 1min$, $\lambda_{t_0+2\ min} = \delta_{a_i} P_{t_0} P_{t_0+1\ min}$ qui est le vecteur ligne de probabilité au temps $t_0 + 2min$, et $\lambda_{t_0+3\ min} = \delta_{a_i} P_{t_0} P_{t_0+1\ min} P_{t_0+2\ min}$ qui est le vecteur ligne de probabilité au temps $t_0 + 3min$. Ainsi, d'ordre général, $\forall k \in [1, n]$ (dans notre cas $n = 15$), le vecteur ligne calculé est $\lambda = \delta_{a_i} \prod_{k=1}^{n} P_{t_0+k}$. Enfin, cela permettra en outre à notre système d'envisager la prochaine connexion $a_i \in \mathcal{A}$ (localisation) prédite au temps $t_0 + k$ tel que $\lambda_{a_i} = \arg\max_{a_i \in \mathcal{A}} \lambda_{a_i}$ qui représente ainsi la plus forte probabilité pour l'utilisateur u d'être connecté au prochain point d'accès a_i au temps $t_0 + k$.

7.2.2.2 Resultats

Nous avons décidé d'observer deux types de lieux à Angers sur lesquels nous comptons mettre en évidence différents types de résultats : les lieux où l'on s'arrête et les lieux de passage. La Figure 7.2 montre plusieurs résultats de prédiction que nous avons obtenu le 1^{er} Oct. 2015 à différent moment de la journée à deux types d'emplacements différents.

— la première expérience (Voir Figure 7.2.1) a pour objectif d'étudier l'une des principales avenue d'Angers — *Avenue Foch* — à 8 heures du matin, un jour de travail habituel. Ainsi, nous nous attendons à ce que les utilisateurs se déplacent plutôt qu'ils restent sur place. Dans ce cas, $\{a_{421}; a_{388}; a_{406}; a_{366}\} \in A$ sont les points d'accès étudiés. Sur cette figure, nous observons qu'après 1 minute, la prochaine connexion prédite d'un utilisateur connecté à a_{406} est estimée à a_{366} avec une probabilité de 0,81. Nous pensons que cela est dû aux caractéristiques du lieu qui est une grande avenue typique où on circule ;



1- Prédiction de mobilité à t+1min
Avenue Foch 8h00
Points d'accès {421; 388; 406; 366}

2- Prédiction de mobilité à t+15min
Place du Ralliement 15h00
Points d'accès {275; 331; 271; 409}

FIGURE 7.2 – Prédiction de la mobilité à partir des journaux de connexions à *Wifilib*

— la seconde expérience (Voir Figure 7.2.2) a pour objectif d'étudier l'une des places principales du centre-ville d'Angers — *la Place du Ralliement* — à 15 heures durant un jour de week-end. On s'attend à ce que les personnes restent plus longtemps que sur une avenue. Notons qu'au centre de cette place il y a une station de Tramway. Dans ce cas, $\{a_{275}; a_{331}; a_{271}; a_{409}\} \in A$ sont les points d'accès étudiés. À la Figure 7.2.2 on observe qu'après 15 minutes, la prochaine connexion prédite d'un utilisateur connecté à a_{331} est estimée en a_{275} avec une probabilité de 0,455 et d'être encore connecté à a_{331} avec une probabilité de 0,372. Avant de prédire la mobilité après 15 minutes, nous avons exploré les 14 premières minutes qui ont montré qu'un utilisateur connecté à a_{331} a une probabilité plus forte de rester connecté à a_{331} que de se déplacer à un autre point d'accès. Nous pensons que cela est dû aux caractéristiques de la place du Ralliement, qui est un lieu typique où les personnes s'arrêtent quelque temps p. ex., dans un café, faire les magasins ou encore attendre le tramway.

Dans cette étude, nous avons travaillé avec quatre jours de données collectées du 1^{er} au 4 Octobre 2015, ce qui nous permet de valider l'exactitude de nos prévisions en les comparant aux données réelles.

7.3 Géo-partitionnement (*geo-clustering*) appliqué aux systèmes de recommandation

Cette section fait référence à nos contributions [Gut+18c] (étoile n° 3) ainsi qu'à [Gut+19c] (étoile n° 8) présentées à la Figure 2 en introduction de ce mémoire.

L'une des informations dynamique les plus importantes à prendre en compte dans les applications de villes intelligentes est l'emplacement de l'utilisateur [SDO18]. Ce type de données est généralement acquis par divers capteurs [AT05], p. ex., comme nous l'avons abordé au chapitre précédent : il peut s'agir du GPS des téléphones mobiles, des journaux de connexion Wi-Fi en milieu urbain. Néanmoins, ce contexte brut doit être analysé et transformé afin de devenir exploitable. C'est pourquoi, la localisation des utilisateurs mobiles a été largement étudiée dans le cas d'applications *MCSC*, car les analyses qui en découlent peuvent améliorer considérablement l'exploitabilité du contexte [Cra+12 ; GS11 ; Guo+15 ; Had14 ; ME12]. Ces analyses peuvent par exemple conduire à des caractérisations de la mobilité urbaine [Nou+12] telles que des trajectoires ou des prédictions de déplacement [ME12]. De plus, dans le domaine de la détection de la dynamique urbaine et de la recommandation mobile, les réseaux sociaux basés sur la localisation *(LBSN)*, qui traitent non seulement les données relatives aux utilisateurs, telles que les amitiés, les activités, les événements, les intérêts communs ou les connaissances partagées, tiennent également compte des relations entre les utilisateurs et leurs emplacements [Bao+13 ; Zhe11].

Dans notre cas applicatif, les *LBSNs* concernant la ville d'Angers contiennent trop peu d'informations pour être utilisables. Ainsi dans notre cas nous devrons uniquement nous appuyer sur les journaux de connexion aux points d'accès *Wifilib* et sur l'application mobile *scéno*.

7.3.1 Notre cas pratique de géo-partitionnement appliqué à la recommandation

Les techniques de raisonnement contextuel deviennent incontournables afin d'extraire de la connaissance *exploitable* [SDO18]. En ce qui concerne le contexte spatial, cette exploitation peut se traduire sous la forme de prises de décisions : soit humaine après traitement et analyse de l'information (via l'informatique décisionnelle) ; soit purement informatique dans le cadre, pour notre cas des systèmes de recommandation (via l'apprentissage automatique).

Les informations géographiques telles que la latitude et la longitude, sont des données typiques que l'ont peut utiliser pour construire un contexte spatial. En revanche, même si ces informations sont importantes, il peut devenir très compliqué d'en tirer profit en les abordant uniquement sous leur forme brute [SDO18]. Comme les systèmes de recommandation ne sauraient uniquement se baser sur ces coordonnées brutes, il est nécessaire d'employer des techniques d'extraction de connaissances afin que ces systèmes puissent les exploiter et tirer parti d'un apprentissage efficace. De plus, la découverte automatisée de lieux peut aider à mieux comprendre les habitudes utilisateurs et leur comportement [Fro+06], et plus généralement les informations de lieux peuvent aider à la caractérisation du contexte [Cam+08].

Notre travail s'inscrit dans cette dynamique et repose sur l'existence d'une infrastructure de points d'accès Wi-Fi urbain (*Wifilib*) (voir sous-section 6.2.1 pour la description du jeu de données), mise à disposition gratuitement dans les zones les plus fréquentées de plusieurs villes en France. Pour effectuer le calcul de géo-partionnement, nous utiliserons cette fois une année complète (2017-2018) de journaux de connexions des utilisateurs mobiles à l'ensemble des points d'accès *Wifilib*[1] dont nous connaissons les coordonnées GPS. Notre objectif est de catégoriser les lieux et d'en déduire des *quartiers* dans la ville, en étudiant les fréquentations des utilisateurs à ces points d'accès. Nous proposons d'aborder le problème sous l'angle de l'apprentissage non supervisé en utilisant, pour le résoudre, une approche de type partitionnement spectral employant la technique des k-moyennes. La principale contribution de cette section est la construction de la matrice d'affinité basée sur la fréquentation (le nombre de connexions) des différents points d'accès par les utilisateurs mobiles.

L'ensemble de nos résultats est disponible sur le site web `http://www.wifilib-clustering.info/` qui permet d'observer, entre autres, les quartiers déduits sur les 15 villes que nous avons traitées.

L'objectif de nos travaux a été de traduire de l'information spatiale brute (coordonnées GPS) en données utilisables par le système de recommandation d'événements culturels de l'application mobile *scéno*. Ainsi, lorsqu'un utilisateur de l'application se connecte à un endroit précis de la ville, nous pouvons prendre en considération sa localisation selon le quartier (inféré) depuis lequel il s'est connecté. L'évaluation en ligne concernant la partie système de recommandation portera sur la ville d'Angers uniquement.

7.3.2 Partitionnement spectral pour l'identification de quartiers

L'une des approches populaire employée pour la déduction de quartier est le partitionnement spectral. Elle a été largement étudiée et obtient de bons résultats empiriques lorsqu'elle est bien paramétrée [NJW02 ; Nou+11 ; Ryu+17] (voir Chapitre 3). Cette méthode a notamment été utilisée pour le projet *Livehoods*[2] [Cra+12] qui, à partir de données de *Foursquare* (LBSN), a pu inférer des dynamiques urbaines et sociales se traduisant visuellement par le découpage de zones dans les grandes villes Nord Américaines. Les résultats obtenus via partitionnement spectral par le projet *Livehoods* sur *Foursquare* ont été évalués par des enquêtes et des sondages, et ont été jugés pertinents.

En revanche, sur les connexions aux réseaux Wi-Fi urbains à grande échelle, à notre connaissance, aucune étude n'a ni proposé de méthode de partitionnement spectral, ni mis en évidence les performances d'une telle approche.

L'algorithme que nous utilisons et présentons dans cette section est une variante de celui proposé par [Cra+12]. Plus précisément, la méthode *spectral_clustering()* de la bibliothèque *Scikit-learn*[3] que nous avons utilisée pour mettre en oeuvre notre algorithme est issue de techniques de partitionnement spectral telles que définies entre autres par [Von07].

1. `https://www.wifilib.com/index.html`
2. `http://livehoods.org/`
3. Bibliothèque de *Python* dédiée à l'apprentissage automatique : `http://scikit-learn.org/`

7.3.2.1 Notre méthode de partitionnement spectral

Le principe de notre approche est d'utiliser une méthode de partitionnement spectral visant à classer les zones de la ville en fonction des visites (connexions) des utilisateurs aux points d'accès Wi-Fi disponibles (c.-à-d., en fonction des utilisateurs qui les fréquentent). À partir des journaux de connexions, l'objectif de notre travail est ainsi d'inférer du contexte spatial via les connexions utilisateurs aux points d'accès Wi-Fi dont nous possédons les coordonnées GPS exactes.

Construction de la matrice d'affinité. Pour chaque ville, nous disposons de n utilisateurs $U = \{u_1, ..., u_n\}$, et de w points d'accès Wi-Fi $P = \{p_1, ..., p_w\}$. Sur l'année 2017, nous disposons également d'un ensemble C de connexions de cet ensemble utilisateurs U aux points d'accès de P. Notre approche étant basée sur la fréquentation des points d'accès par les utilisateurs mobiles, à partir de C nous calculons le nombre de connexions de chaque utilisateur $u \in U$ à chaque point d'accès $p \in P$. À partir de ce calcul nous construisons w vecteurs $c_p \in \mathbb{N}^n$ où le premier composant du vecteur c_p correspond au nombre de connexions de l'utilisateur u_1 au point d'accès p et le $n^{\text{ème}}$ composant du vecteur c_p correspond au nombre de connexions de l'utilisateur u_n au point d'accès p. À partir de ces vecteurs, il est possible de calculer les similarités de fréquentation entre chaque paire de points d'accès $sim(p_i, p_j)$. Dans notre cas nous avons choisi d'utiliser la mesure de similarité de cosinus (CP - Cosine similarity between access Points) telle que :

$$CP(p_i, p_j) = \frac{c_{p_i} \cdot c_{p_j}}{||c_{p_i}||\ ||c_{p_i}||}$$

Enfin, nous construisons la matrice d'affinité $S = (s_{i,j})_{1 \leq i \leq w, 1 \leq j \leq w}$ telle que $s_{i,j} = CP(p_i, p_j)$.

Le calcul et la nature même de cette matrice d'affinité — matrice de similarité de fréquentation — est l'une de nos principales contributions. À noter que dans notre exemple nous considérons que les points d'accès ne sont décrits que par leurs fréquentations.

Construction de la matrice Laplacienne normalisée. Préliminairement au calcul de la matrice normalisée L_{sym} l'algorithme calcule la matrice diagonale D, de diagonale $(d_1, ..., d_w)$, telle que $D_{i,i} = d_i = \sum_{j=1}^{w} s_{i,j}$ et construit ensuite la matrice Laplacienne non normalisée $L = D - S$. Enfin, il est possible de calculer la matrice Laplacienne normalisée L_{sym} par division symétrique telle que

$$L_{sym} = D^{-1/2}\ L\ D^{-1/2}$$

Déduction des vecteurs propres et partitionnement k-moyennes. L'étape suivante nécessite de déterminer le nombre de groupes g (quartiers) à inférer par le partitionnement spectral. Deux cas s'offrent à nous : 1) nous connaissons le nombre g exact de groupes (quartiers) que l'algorithme doit déterminer ; 2) nous ne le connaissons pas et laissons soit ce paramètre par défaut, soit déterminons un intervalle $[g_{min}, g_{max}]$ dans lequel le calcul des valeurs propres peut se faire [Cra+12]. Dans notre cas, notre application doit fournir des recommandations en

temps réel. Ainsi, puisque la complexité de *LinUCB* est $\mathcal{O}(d^3)$ et que sa borne de regret est $\mathcal{O}\left(d\sqrt{T\ln((1+T)/\delta)}\right)$ où d est la taille du vecteur de contexte, nous devons limiter le nombre de dimensions qui enrichissent le contexte original x afin d'éviter de devoir à traiter trop de dimensions supplémentaires (c.-à-d., la malédiction de la dimensionnalité [Bel15]). Nous décidons donc de limiter le nombre de clusters déduits à un maximum de $g_{max} = 10$.

Ensuite, *Scikit-Learn* se base sur une stratégie de décomposition utilisant entre autres le solveur *arpack* conçu pour calculer les g valeurs propres les plus grandes $\lambda_1, .., \lambda_g$ et déterminer ainsi les g premiers vecteurs propres correspondants de L_{sym} c.-à-d., l'algorithme détermine les g premiers vecteurs propres $v_1, ..., v_g$ de L_{sym} tels que : $v_i = (v_{i,1}, ..., v_{i,w}), i \in [1, w]$.

À partir des g premiers vecteurs propres $v_i, i \in [1, g]$, l'algorithme construit une matrice non normalisée $V = [v_1, ..., v_g] \in \mathbb{R}^{w \times g}$, puis construit enfin une matrice normalisée de V que nous nommons $V' \in \mathbb{R}^{w \times g}$, telle que :

$$v'_{i,j} = \frac{v_{i,j}}{\left(\sum_j \ v_{i,j}^2\right)^{1/2}}$$

Chaque ligne de V' est ensuite traitée comme un point de $\mathbb{R}^g$ qu'on peut définir tel que : Pour $i = 1, ..., w$, soit $y_i \in \mathbb{R}^g$ le vecteur correspondant au $i^{ème}$ rang de V'.

La dernière étape de l'algorithme consiste, via la méthode des k-moyennes, à classer les points $(y_i)_{i \in 1,..,w}$ en g clusters $G_1, ..., G_g$.

Affichage sur la carte. Pour chaque cluster on attribue une couleur, et pour chaque point d'accès p appartenant à un cluster on fixe une épingle de localisation sur la carte de la couleur du cluster aux coordonnées GPS de p. Enfin, on applique un algorithme de *Quick Hull* [BDH93] afin de déterminer l'enveloppe convexe formée par l'ensemble des points d'accès d'un même cluster, puis on trace l'enveloppe associée.

7.3.3 Résultats

7.3.3.1 Site web de résultats

Nous avons construit un site web [4] permettant de visualiser nos résultats [5] (clusters c.-à-d., quartiers calculés). De plus, afin de pouvoir mettre à disposition de la communauté un exemple de notre jeu de données (sensible et confidentiel), nous en avons anonymisé un échantillon correspondant à un mois de connexions de l'une des 15 villes traitées. Pour ce faire, nous avons placé les points d'accès sur la carte d'un monde imaginaire c.-à-d., la carte de la *Terre du milieu* issue du *Seigneur des Anneaux* [6] [Led72]. Pour cela, nous avons transformé les coordonnées d'origines des points d'accès vers le système de coordonnées de notre monde imaginaire tout en conservant la cohérence des quartiers détectés. Sur le site web, nous mettons à disposition ce jeu de données, les méta-données, les distances géographiques d'origine

4. `http://www.wifilib-clustering.info/`
5. Sur des cartes *OpenStreetMap* : `https://www.openstreetmap.fr`
6. `https://en.wikipedia.org/wiki/The_Lord_of_the_Rings`

entre chaque point d'accès, ainsi que les résultats que nous obtenons via notre algorithme de partitionnement spectral. Nous ne transmettons pas les détails de la transformation afin de garantir l'anonymisation des coordonnées.

7.3.3.2 Quantité et Qualité des résultats

Le jeu de données *JSON Wifilib* (voir Annexe D.1) qui nous a été fourni nous a permis de traiter 55 000 000 de connexions de près de 1 500 000 utilisateurs sur l'année 2017, sur un ensemble de 2 200 points d'accès Wi-Fi, dans 15 villes en France. L'ensemble des résultats sont disponibles sur le site via un menu interactif représentant la carte de France [7]. Pour chaque ville, à partir de ce menu, il est possible de voir en sur-brillance : le nombre de points d'accès, le nombre d'utilisateurs, et le nombre de connexions traités. De plus, à partir du niveau de couverture, nous avons pu déterminer trois niveaux de qualités de résultats obtenus :

— vert - Villes possédant plus de 100 points d'accès Wi-Fi ;
— orange - Villes possédant entre 50 et 100 points d'accès Wi-Fi ;
— rouge - Villes possédant moins de 50 points d'accès Wi-Fi.

Les deux villes les plus couvertes [8] par *Wifilib* sont Angers et Paris. Les résultats qui en découlent sont de ce fait les plus complets.

7.3.3.3 Focus sur Angers

L'évaluation en ligne que nous avons réalisée sur notre système de recommandation *scéno* a été réalisée sur Angers. Ainsi, nous décrivons plus précisément ci-dessous les résultats que nous obtenons dans cette ville en termes de quartiers déduits (voir la Figure 7.3) :

1. Cluster 1 : le premier cluster correspond à une partie du quartier de « *La Doutre* » situé sur la rive opposée du centre-ville d'Angers ;
2. Cluster 2 : le deuxième cluster correspond au centre-ville historique d'Angers où l'on peut visiter le château, la cathédrale et flâner dans des rues pavées datant du moyen-âge ;
3. Cluster 3 : le troisième cluster correspond au « *Quartier de la gare* », où les gens arrivent de différentes villes et cherchent ensuite p.ex. un restaurant où déjeuner/dîner, leur hôtel ;
4. Cluster 4 : le quatrième cluster 4 correspond à la « *Place du Ralliement* » qui est une place où les gens peuvent prendre un verre ou déjeuner/dîner. Ils peuvent aussi faire les magasins ou attendre le prochain tramway. Ce quartier peut être considéré comme l'un des plus fréquentés d'Angers ;

7. `http://www.wifilib-clustering.info/#FRMAP`
8. Angers : `http://www.wifilib-clustering.info/cities/Angers20172018.html`, Paris : `http://www.wifilib-clustering.info/cities/Paris20172018.html`

5. Cluster 5 : le cinquième cluster correspond à un quartier d'affaires dans lequel nous trouvons des sièges sociaux de différentes assurances, banques, et opérateurs de télécommunications et l'université St Serge ;
6. Cluster 6 : le sixième cluster correspond à la « *Place Imbach* » possédant un grand parking entouré de plusieurs restaurants et une église romane à visiter ;
7. Cluster 7 : le septième cluster correspond au « *Jardin du Mail* » où les familles peuvent se promener et où les enfants peuvent jouer dans une aire de jeux ou un manège ;
8. Cluster 8 : le huitième cluster correspond à la rue « *Bressigny* », où les étudiants se rassemblent le soir et la nuit (pour travailler...) ;
9. Cluster 9 : le neuvième cluster correspond à une voie de passage qui à partir du « *Jardin des plantes* et de la rue » *Boreau* » permet d'accéder au centre-ville d'Angers ;
10. Cluster 10 : le dixième cluster correspond à certain nombre de rues de restaurants et de pubs où les gens peuvent prendre un verre ou déjeuner/dîner.

Ces résultats ont été vérifiés et discutés avec plusieurs citoyens de la ville d'Angers et des personnes impliquées dans le service urbain public de la ville d'Angers qui les ont jugés pertinents.

7.3.4 Utilisation du géo-contexte déduit par notre système de recommandation

Comme décrit dans le Chapitre 2, un algorithme de bandits-manchots contextuel basé sur un modèle linéaire utilise une représentation structurée du contexte sous forme d'un vecteur dénommé x. Afin que ce type d'algorithme puisse utiliser le contexte spatial déduit, nous considérons chaque cluster i à travers un vecteur de type « *one-hot* » x'_i à combiner avec le vecteur original x (c.-à-d., $x \oplus x'_i$). Le contexte spatial de toute localisation située en dehors d'un cluster identifié est noté comme appartenant au "cluster 0" et sera considéré comme tel par l'algorithme de recommandation. Ainsi dans notre cas pour Angers, $\forall x'_i \in \{0,1\}^{g+1}$ et $g = 10$, concernant le cluster 0 le vecteur correspondant est $x'_{i=0}$ c.-à-d., $\{1,0,0,0,0,0,0,0,0,0,0\}$, concernant le cluster 1 le vecteur correspondant est $x'_{i=1} = \{0,1,0,0,0,0,0,0,0,0,0\}$, et pour le cluster 5 $x'_{i=5} = \{0,0,0,0,0,1,0,0,0,0,0\}$.

7.3.5 Évaluation des résultats : application au système de recommandation de *scéno*

Afin d'observer la pertinence des clusters inférés pour un système de recommandation, nous avons effectué une évaluation en ligne via notre système de recommandation d'événements culturels que nous avons intégré au sein de notre application mobile *scéno*.

7.3.5.1 Algorithmes comparés

Les algorithmes que nous avons comparés dans cette évaluation en ligne sont les suivants :

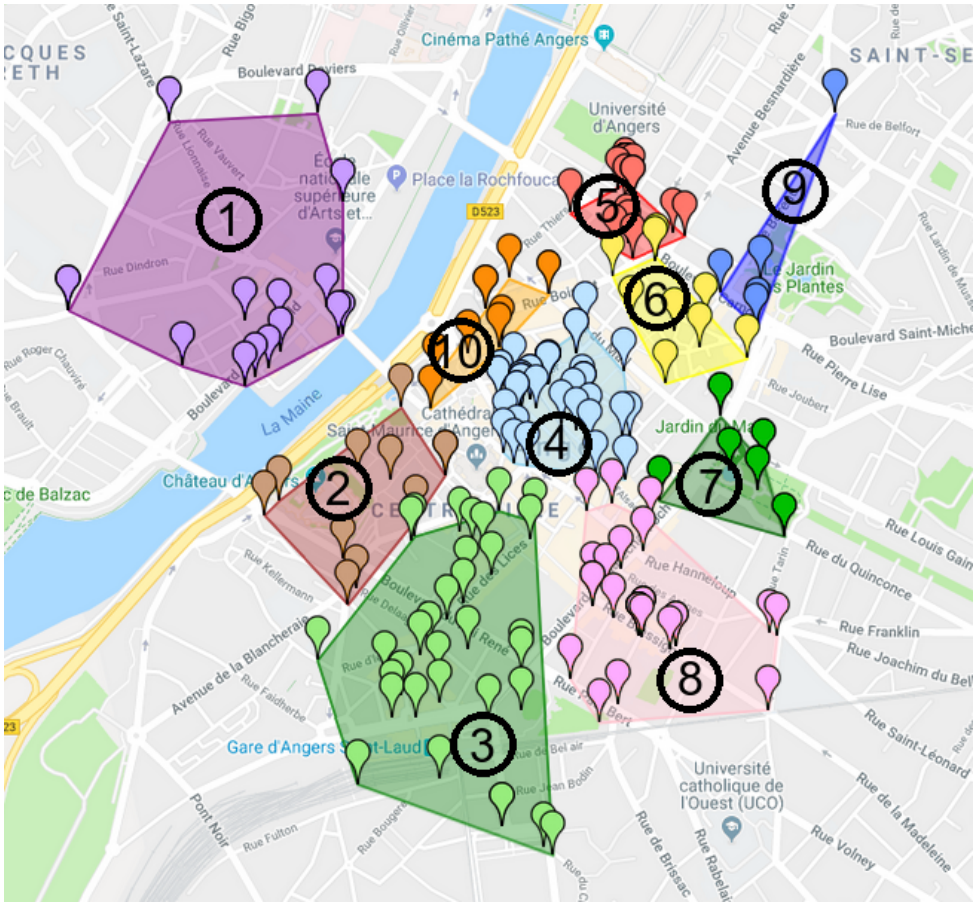


FIGURE 7.3 – Quartiers déduits (clusters) pour la ville d'Angers

— *LinUCB* [Li+10] opérant sur le contexte d'origine x, c'est à dire les informations de profil sans prise en compte du contexte spatial ;
— *LinUCB* opérant sur $x \oplus x'_i$, c'est à dire les informations de profil avec prise en compte du contexte spatial déduit par notre méthode de partitionnement spectral ;
— à des fins de contrôle, nous comparons les deux cas précédent avec deux autres algorithmes : 1) recommandation *aléatoire* ; 2) ε-*Greedy* [SB98] qui est un algorithme de bandits-manchots non contextuel (voir Chapitre 2).

À chaque itération, un utilisateur se présente dans un contexte donné et on sélectionne aléatoirement l'un des quatre algorithmes comparés. Ensuite, l'algorithme sélectionné recommande les événements culturels selon sa propre politique de sélection des bras. Notons que dans notre évaluation en ligne, nous n'avions pas encore pu intégrer *Gorthaur* (voir Section 5.3) qui sélectionne le meilleur algorithme proportionnellement à sa capacité de maximiser la précision globale et la diversité.

7.3.5.2 Données expérimentales et critères d'évaluation

Durant cette phase d'évaluation de 12 mois, l'application *scéno* a effectué 5705 recommandations à 204 utilisateurs. Parmi ces recommandations, 723 ont été évaluées (c.-à-.d. on reçu un retour positif ou négatif de l'utilisateur).

Nous évaluons notre méthode selon deux critères : la précision globale (voir sous-section 4.3.2) et le taux de retour (en pourcentage) qui correspond à la participation (Positive comme Négative) des utilisateurs par rapport aux recommandations qui leur sont faites.

Le taux de retour τ_b pour chaque algorithme $b \in B$ peut être défini comme suit : Soit $n_{U,b}$ le nombre de retours effectués par l'ensemble U des utilisateurs lorsqu'une recommandation a été produite par l'un des quatre algorithmes $b \in B$ comparés, et soit N_b le nombre total de recommandations produites par l'algorithme $b \in B$. Alors le taux de retour de l'algorithme b est $\tau_b = n_{U,b}/N_b$ exprimé en pourcentage.

7.3.5.3 Résultats

Dans le tableau 7.1, nous observons la précision globale obtenue par les quatre algorithmes de recommandation intégrés à notre application mobile. Les pourcentages représentent l'augmentation de la précision globale par rapport à celle d'une stratégie de recommandation *aléatoire*. Dans le tableau 7.1, nous remarquons d'abord que l'algorithme ε-*Greedy* est supérieur à l'algorithme de recommandation aléatoire. De plus, les deux algorithmes *LinUCB* surpassent sans surprise l'algorithme de recommandation aléatoire et ε-Greedy. Enfin, nous observons que *LinUCB* obtient une meilleure précision globale lorsqu'il bénéficie de caractéristiques de contexte spatial que nous avons déterminé avec notre méthode de partitionnement spectral (augmentation de +21,6% par rapport à *LinUCB* avec vecteur de contexte sans contexte spatial).

Algorithmes	Précision globale	Augmentation de la précision
Aléatoire	$0,3744$	0%
ε-*Greedy*	$0,5376$	$43,58\%$
LinUCB	$0,5827$	$55,63\%$
LinUCB avec contexte spatial	$0,7086$	$89,26\%$

TABLE 7.1 – Évaluation en ligne sur 12 mois de 204 utilisateurs réguliers ayant fait des retours sur 723 recommandations de notre application mobile *scéno*.

De plus, dans le tableau 7.2, nous observons les taux de retour τ_b des utilisateurs pour chaque algorithme $b \in B$. Nous observons que les algorithmes de bandits-manchots contextuels (*Contextual Multi-Armed Bandit* - *CMAB*) obtiennent de meilleurs taux de retours des utilisateurs que toute autre méthode évaluée. De plus, nous observons plus précisément que lorsque le contexte spatial est fourni, *LinUCB* surpasse toutes les méthodes. Par conséquent, nous supposons que si les recommandations ne sont pas appréciées par les utilisateurs, ils

perdent leur motivation à répondre et ne prennent pas le temps de donner leur avis. Une perspective pourrait donc être de considérer qu'une « non-évaluation » doit être considérée comme un retour négatif de la part des utilisateurs.

Algorithmes	**Taux de retour**
Aléatoire	$10,05\%$
ε-*Greedy*	$11,12\%$
LinUCB	$21,11\%$
LinUCB avec contexte spatial	$27,82\%$

TABLE 7.2 – Résultats de taux de retours obtenus, en ligne, sur 12 mois par les différents algorithmes de recommandation intégrés dans l'application prototype *scéno*.

7.3.5.4 Discussion

Selon les résultats que nous avons obtenus dans notre évaluation en ligne, la question qui se pose est la suivante : « Qu'est-ce qui fait vraiment la différence entre les deux cas d'utilisation de *LinUCB* ? ». Nous décidons donc de compléter notre évaluation en étudiant la précision globale obtenue dans les deux cas en observant leur résultats obtenus pour les 3 catégories (parmi les 13 catégories existantes) pour lesquelles ils ont obtenu les meilleures résultats de précision globale.

Nous observons dans le tableau 7.3 que dans les deux cas les algorithmes *LinUCB* parviennent à trouver les bras optimaux correspondant aux catégories de « conférences » et de « cinéma ». En revanche, en l'absence de contexte spatial, *LinUCB* ne recommande pas efficacement concernant les événements de la catégorie « loisirs ». Notre constat est donc le suivant : considérant que *LinUCB* suppose une dépendance linéaire entre les récompenses et les caractéristiques du vecteur de contexte, les caractéristiques de contexte spatial sont donc un élément central pour la recommandation d'événements ayant trait aux loisirs.

Algorithmes	**Précision globale**	**Catégorie**
LinUCB	$0,92$	Conférences
	$0,89$	Cinéma
	$0,69$	Loisirs
LinUCB avec contexte spatial	$0,93$	Loisirs
	$0,90$	Conférences
	$0,88$	Cinéma

TABLE 7.3 – Précisions globales obtenues pour les deux cas de *LinUCB* (avec et sans contexte spatial) pour les 3 meilleures catégories d'événements culturels

7.3.6 Conclusion et perspectives

Dans cette section nous avons présenté notre méthode de partitionnement spectral visant la déduction de quartiers à partir des journaux de connexions des utilisateurs mobiles aux réseaux Wi-Fi urbains. Les résultats obtenus ont été employés en tant qu'informations de contexte spatial par le système de recommandation d'événements culturels de notre application mobile *scéno*. Les informations de contexte spatial déduites et intégrées au contexte de systèmes de recommandation d'une application mobile réelle a permis d'obtenir des résultats de précision globale supérieures à tout autre méthode ne s'appuyant pas sur ces données.

Ainsi, de ces résultats il est possible de tirer une triple conclusion :

— l'information de contexte spatial est très importante pour les systèmes de recommandation à des utilisateurs mobiles dans la ville intelligente pour effectuer de la recommandation d'événements culturels ;

— les quartiers que nous avons inférés pour notre système semblent pertinent du fait des meilleurs résultats obtenus par *LinUCB* lorsque celui-ci s'appuie sur ces nouvelles caractéristiques de contexte ;

— le vecteur de contexte original x avait des composants manquants en terme de contexte spatial. Avec notre méthode de raisonnement contextuelle utilisant une technique de partitionnement spectral, nous parvenons à compléter ces composants manquants dans x et ainsi à améliorer la précision des recommandations d'événements ayant trait aux loisirs.

De telles observations nous incitent à réfléchir à la manière dont, lors de l'utilisation d'algorithmes de *CMAB*, nous pouvons dynamiquement et rapidement pointer des composants manquants (plus généraux) dans x, puis utiliser une méthode d'enrichissement pour les compléter.

Ainsi, dans la section suivante, nous traiterons de cette problématique d'enrichissement et proposerons une méthode plus générale d'apprentissage de préférence utilisateurs permettant d'enrichir le contexte d'origine x. Cette méthode *ICE* (*Individual Context Enrichment*), vise l'amélioration du contexte en se basant sur la précision individuelle mesurée pour chaque utilisateur.

7.4 Application aux systèmes de recommandation : cas d'apprentissage des préférences utilisateurs avec la méthode *ICE*

« *Most of the time, user preferences depend on context, that is, they may have different values depending on the situation of the user* » [HMP12]

Cette section fait référence à nos contributions [Gut+18b ; Gut+19d] (étoiles n° 4 et 5) présentées à la Figure 2 en introduction de ce mémoire.

Dans le domaine des systèmes de recommandation, nous avons vu que le contexte était une information clé permettant d'améliorer la performance des algorithmes de recommandation. De nombreux travaux se focalisent sur l'amélioration du contexte en termes de modéli-

sation [BHB18], de capture [Guo+15] ou encore de réduction de dimensions [Bou+17]. Ces travaux ont pour objectif d'améliorer la précision globale [Sli14] ou encore de diminuer la complexité en temps [Bou+17].

Ainsi, dans cette section nous présentons notre méthode *ICE* (*Individual Context Enrichment*) dont le principal objectif est d'enrichir le contexte original fourni en considérant chaque utilisateur u et le retour qu'il produit par rapport à un bras dans un contexte donné x. Nous appelons cette mesure de performance : la précision individuelle de la paire utilisateur-contexte (u, x). La notion de paire utilisateur-contexte a été évoquée au Chapitre 4 et sera précisée à la sous-section 7.4.2.

À notre connaissance, concernant les algorithmes de bandits-manchots contextuels (*Contextual Multi-Armed Bandit - CMAB*), aucune approche ne traite spécifiquement le problème d'enrichissement du contexte en observant des dimensions centrées sur l'utilisateur telles que la précision individuelle des recommandations.

Dans cette section, nous proposons également une expérimentation hors-ligne de notre méthode *ICE* sur plusieurs jeux de données du mondes réels. Celle-ci obtient de meilleurs résultats en termes de précision globale et de regrets cumulés que toute autre méthode de *MAB* ou de *CMAB* originale (c.-à-d., sans *ICE*).

7.4.1 Principe de *ICE*

ICE classifie les paires *utilisateur-contexte* en fonction de leur précision individuelle et utilise les classes (de précision individuelle) obtenues pour enrichir le contexte original. Pour être efficace, notre méthode nécessite d'opérer sur des utilisateurs « abonnés ». Notre méthode est donc particulièrement intéressante dans le cas d'applications comportant des utilisateurs identifiables et régulièrement identifiés, p. ex., systèmes de recommandation en ligne, essais cliniques, applications mobiles.

7.4.2 Le contexte

Ici, nous représentons le contexte [Bou+17 ; Bré99 ; Dey01 ; Li+10] comme défini au Chapitre 2 c.-à-d., en un vecteur de caractéristiques qui peut être composé p.ex. du profil, des préférences, du contexte spatio-temporel, et caractéristiques des éléments à recommander. Afin de donner une notation unique, nous résumons toutes ces informations et nous y référerons sous le terme de vecteur de contexte représenté par x.

De plus, dans les applications du monde réel, les utilisateurs évoluant dans un contexte donné peuvent être des « abonnés » pouvant être identifiés régulièrement. Par conséquent, nous considérons qu'un même utilisateur u, peut rencontrer à plusieurs reprises des situations représentées par le même vecteur de caractéristiques x. Ceci est exprimé par la paire utilisateur-contexte (u, x).

7.4.3 Énoncé du problème

Dans un problème de bandit-manchot contextuel linéaire, des contextes (observations) identiques reçoivent les mêmes récompenses pour un même bras optimal si le modèle linéaire fourni est juste et que les caractéristiques de contexte données en entrée sont complètes et pertinentes. Si le contexte est incomplet (c.-à-d., restreint), on peut observer des regrets dus à la dérive conceptuelle (concept-drift) (voir Chapitre 2) puisque la non-stationnarité affecte des caractéristiques non appréciées du contexte observé [Qui+09].

En considérant un problème de *CMAB* linéaire (voir le Chapitre 2), soit $O \subseteq A$ l'ensemble des bras optimaux pour ce problème. Soit $\forall x_i, x_j \in X$, $CCT(x_i, x_j)$ le calcul de similarité de cosinus (CCT - Cosine similarity between ConTexts) entre chaque paire de vecteur de contexte tel que : $CCT(x_i, x_j)) = \frac{x_i \,.\, x_j}{||x_i|| \,.\, ||x_j||}$.

Soit le modèle linéaire entre les récompenses obtenus et les caractéristiques du contexte $\widehat{\theta}_a^\top x_t$, et soit $x^* \in X^* \subseteq \mathbb{R}^d$ le vecteur optimal de caractéristiques. Ce vecteur hypothétique possède, avec une confiance de 100%, les caractéristiques les plus pertinentes associées au(x) bras optimal(aux) correspondant(s) $a^* \in O$ pour chaque contexte disponible. Ainsi, $\forall x_i^*, x_j^* \in X^*$, si $CCT(x_i^*, x_j^*) = 1$, alors $\forall a^* \in O$, nous avons $r_{a^*,x_i^*} = r_{a^*,x_j^*}$. Par conséquent, comme *LinUCB* et *CTS* supposent une dépendance linéaire entre la récompense attendue d'une action et son contexte tel que $\mathbb{E}[r_{t,a}|x_t] = \theta_a^\top x_t$, alors quand x^* est fourni, ils convergeront vers une précision globale de 100% (Voir Table 7.4 - *Contrôle (vc)* (vc : vecteur complet et optimal x^*)). Ils atteignent ainsi une personnalisation totale des recommandations faites aux utilisateurs.

Néanmoins, quand le contexte x est fourni de manière incomplète (c.-à-d., restriction sur le contexte), il se peut que deux contextes (observations) qui sont normalement différents soient cette fois représentés par la même description structurée via le même vecteur de caractéristiques. La différence qui résiderait entre ces deux contextes concernerait donc seulement des dimensions contextuelles manquantes (non appréciées car non observables).

Ainsi, à chaque itération, même si le bras sélectionné par *LinUCB* sera généralement le meilleur, il pourrait être erroné dans certains cas en raison des dimensions de contexte observé manquantes. C'est à dire, $\exists x_i, x_j \in X^2$, et $\exists a^* \in O$ où $CCT(x_i, x_j) = 1$ et $r_{a^*,x_i} \neq r_{a^*,x_j}$. Naturellement, dans des situations réelles, il peut manquer des caractéristiques au vecteur x pour différentes raisons : les caractéristiques des bras ne sont pas disponibles, il existe des restrictions liées à la politique de confidentialité, le contexte a mal été modélisé, les profils ont été mal renseignés, les informations sur l'environnement de l'utilisateur sont manquantes (p. ex., localisation temporairement indisponible). Dans ces cas où $x \neq x^*$, les algorithmes de *CMAB* doivent faire face à des contraintes de restriction sur le contexte puisque les caractéristiques d'un hypothétique x^* sont manquantes dans x et ne peuvent donc être prises en compte par l'algorithme.

Ainsi, on peut considérer deux possibilités d'amélioration afin de pallier les problématiques de restriction :

1. Supposer que le modèle est non linéaire et de ce fait proposer une autre méthode [Gre+17] ;

2. Travailler à améliorer le contexte en lui-même [BHB18 ; Bou+17 ; Bré99], ce qui est l'axe principal de notre travail.

7.4.4 Une histoire de précision

Les algorithmes de *CMAB* comme *LinUCB* [Li+10] et *Contextual Thompson Sampling* (*CTS*) [AG13] ont été élaborés en supposant une relation linéaire globale entre l'ensemble des vecteurs de contexte de dimension d et les récompenses réelles retournées [Gre+17]. Ainsi, lorsque la dépendance linéaire entre les contextes et les récompenses est invariante dans le temps (stationnaire) et que le vecteur de contexte donné est optimal (x^*), ces algorithmes atteignent une personnalisation complète des recommandations faites aux utilisateurs. De plus, même si des mesures globales telles que les regrets cumulés ou les moyennes de récompenses (précision globale) sont tout à fait pertinentes dans la plupart des travaux, lorsqu'un seul bandit est modélisé pour l'ensemble de la population, il devient impossible de déterminer si on atteint une personnalisation des recommandations pour chaque paire d'utilisateur-contexte [MRK06]. Aucune étude n'effectue d'évaluation en tenant compte et observant la précision individuelle pour chaque contexte. Pourtant, les applications du monde réel disposent rarement d'un vecteur de contexte suffisamment complet et pertinent ($x \neq x^*$). Nous soutenons que pour prendre en compte de tels problèmes applicatifs, nous devons observer les performances d'un autre point de vue en nous concentrant sur l'individu. Au chapitre 4 nous avons étudié la précision individuelle ($Acc_u(T)$) afin de mesurer le niveau de personnalisation pour un utilisateur donné. Ici nous décidons donc d'étendre cette métrique en définissant cette fois la précision individuelle du couple utilisateur-contexte (u, x) ($Acc_{u,x}(T)$) afin de mesurer le niveau de personnalisation pour un utilisateur donné dans un contexte donné. Ce critère supplémentaire nous servira également à indiquer si des caractéristiques semblent manquer dans x.

7.4.5 Utiliser la précision individuelle mesurée

Notre méthode s'appuie sur l'étude préliminaire effectuée au Chapitre 4 et qui s'est concentrée entres autres sur l'observation de la précision individuelle $Acc_u(T)$ obtenue par différents algorithmes de *MAB* et de *CMAB*. Cette étude introductive nous a permis d'observer la distribution de précision individuelle sur U à l'horizon T, en étudiant sa fonction de distribution cumulative (FDC). Dans cette étude préliminaire, nous avons observé 12 jeux de données dont certains selon deux cas : avec un vecteur de contexte complet (*vc*), ou avec un vecteur de contexte tronqué (*vt*). Cela nous a permis de constater qu'avec les algorithmes de *MAB*, ou les algorithmes *CMAB* opérant sur des contextes tronqués, même si la précision globale est élevée, elle n'est pas équitablement répartie parmi les individus (utilisateurs).

Ces observations nous ont amenés à créer une nouvelle méthode inspirée des idées du *Logical Analysis of Data* (*LAD*) [HB05].

Notre méthode vise à enrichir un vecteur de contexte qui se voudrait incomplet avec de nouvelles caractéristiques élaborées à partir des retours utilisateurs vis à vis des recomman-

dations qui leur sont faites dans un contexte spécifique. Ainsi, dans ce cas, nous allons calculer et considérer ces informations comme étant la précision individuelle de la paire utilisateur-contexte $(u, x) \in U \times X$. Cette mesure est une extension du critère de précision individuelle définie au Chapitre 4 qui ne se focalise que sur les utilisateurs. Cette fois nous mesurons la précision individuelle par paire utilisateur-contexte (u, x), $Acc_{u,x}(T)$. Celle-ci peut être définie comme étant :

$$\forall (u, x) \in \mathcal{U} \times X, Acc_{u,x}(T) = \frac{\sum_{t=1}^{T} r_{t,u,x}}{obs(T, u, x)}$$

où $obs(T, u, x)$ représente le nombre de fois qu'un utilisateur avec son contexte a été sélectionné à l'horizon T, et $r_{u,x}(t) \in \{0, 1\}$ correspond à la récompense retournée par u à l'itération t dans le contexte x.

7.4.6 Notre méthode : *Individual Context Enrichment (ICE)*

Sur la base de l'énoncé du problème (Sous-section 7.4.3) et des observations formulées dans notre étude préliminaire au Chapitre 4, nous soutenons que les retours individuels des utilisateurs (récompenses) sont étroitement liées aux actions réalisées précédemment par l'algorithme, qui sont elles-mêmes définies par sa propre politique. Ainsi, nous supposons qu'il est possible de définir la précision individuelle pour chaque couple (u, x) résultant de la stratégie de l'algorithme et utiliser ces informations pour enrichir les vecteurs de contexte d'origine. Nous supposons qu'un tel enrichissement de la description du contexte pourrait améliorer la précision globale de l'algorithme.

Nous décidons donc de conserver une approche linéaire du problème de *CMAB* tant pour *LinUCB* que *CTS* qui présentent de bonnes garanties théoriques. Néanmoins, nous proposons de les combiner avec notre méthode d'enrichissement individuel du contexte : *ICE* (*Individual Context Enrichment*).

7.4.6.1 Description de la méthode *ICE*

Afin de pallier les regrets cumulés obtenus en conséquence d'un manque de description du contexte, nous proposons de différencier, de manière précoce et itérative, des paires utilisateur-contexte identiques qui obtiennent des récompenses différentes, en prenant en compte leurs précisions individuelles c.-à-d., $Acc_{u,x}(T)$ (Voir la sous-section 7.4.5 où nous définissons cette métrique). Notre méthode consiste à enrichir le vecteur de caractéristiques en considérant la précision individuelle $Acc_{u,x}$ comme faisant partie intégrante du contexte. *ICE* vise à regrouper des paires utilisateur-contexte dans d' classes en fonction de leurs précisions individuelles. Ainsi, une paire utilisateur-contexte (u, x) doit être prise en compte au moins $d' - 1$ fois avant que le vecteur x correspondant puisse être enrichi.

Soit $x' \in \mathbb{R}^{d'}$ le vecteur de contexte binaire additionnel qui sera finallement concaténé avec le vecteur de contexte original $x \in \mathbb{R}^d$. Soit $I = \{i_q | q \in [0, d' - 1]\}$ l'ensemble des d' classes de

précision individuelle, tel que :

$$\begin{cases} i_0 = 0 \\ i_q =]\frac{q-1}{d'-1}, \frac{q}{d'-1}] \quad , \forall q \in]0, d'-1] \end{cases}$$

Notons que nous choisissons un groupe spécial i_0 composé d'utilisateurs totalement insatisfaits dans des contextes spécifiques (c.-à-d., $Acc_{u,x} = 0$). Ensuite, nous classifions (Voir ligne 13 de l'Algorithme 4) chaque paire utilisateur-contexte selon leur $Acc_{u,x}(t)$ (où t correspond à l'itération à laquelle la taille de l'échantillon mentionné dans la section 7.4.6.2 est atteinte), comme suit :

1. Déterminer à quelle classe de précision i_q la paire utilisateur-contexte appartient c.-à-d., $Acc_{u,x}(t) \in i_q$;
2. Calculer $x' = (0)_q \oplus (1) \oplus (0)_{d'-(q+1)}$ où $a \oplus b$ représente la concaténation entre les vecteurs a et b, et $(0)_q$ représente un vecteur de 0 de dimension q. Par exemple, pour $d' = 4$, si $Acc_{u,x}(t) = 0$ alors nous avons $Acc_{u,x}(t) = i_0$ donc $x' = (0)_0 \oplus (1) \oplus (0)_3 = 1000$. Si $Acc_{u,x}(t) \in]0; 0,33]$ alors nous avons $Acc_{u,x}(t) \in i_1$ donc $x' = (0)_1 \oplus (1) \oplus (0)_2 = 0100$. Enfin, si $Acc_{u,x}(t) \in]0,33; 0,66]$ donc $x' = 0010$, et si $Acc_{u,x}(t) \in]0,66; 1]$ donc $x' = 0001$;
3. Finalement, nous pouvons déterminer un vecteur de contexte enrichi $x_{ice} \in \mathbb{R}^{d+d'}$ tel que $x_{ice} = x \oplus x'$.

Cependant, avant de déclencher un enrichissement du contexte dans une telle approche d'apprentissage par renforcement et sans connaissances a priori, nous devons fournir ce contexte supplémentaire tout en essayant de garantir le modèle non biaisé. Cela signifie que nous devons considérer un nombre suffisant d'observations sur une population représentative de paires utilisateur-contexte. Par conséquent, nous observerons la précision individuelle sur un échantillon statistiquement représentatif de la population globale avant de pouvoir enrichir x avec x'.

7.4.6.2 Calcul de la taille d'échantillon pour *ICE*

En statistiques, et spécialement dans le cadre de sondages ou d'évaluations d'essais cliniques, on peut déduire des informations sur une population en observant un nombre fini d'individus de cette population, c'est-à-dire que l'échantillonnage de la population est réalisé en supposant que les caractéristiques de l'échantillon sont représentatives du total de la population. Lorsque la taille N de la population totale est inconnue et qu'on la suppose beaucoup plus grande que la taille de l'échantillon ($N \gg n_0$), nous pouvons utiliser le calcul général de la taille de l'échantillon de *Cochran* [Coc63] :

$$n_0 = \frac{z^2 \, \widehat{p} \, (1 - \widehat{p})}{m^2} \tag{7.1}$$

où z correspond au z-score défini selon l'intervalle de confiance choisi, $\widehat{p}$ est la proportion d'individus d'une classe dans l'échantillon qui doit être égale à $0,5$ si nous n'avons aucune

hypothèse a priori, et m est la marge d'erreur. Quand la taille de la population totale N est connu (et $N \ll n_0$), en suivant la formule de *Cochran* on peut ajuster la taille de l'échantillon (n_0 défini à l'Équation (7.1)) comme suit :

$$n = \frac{n_0}{1 + \frac{(n_0 - 1)}{N}} \tag{7.2}$$

Dans notre expérimentation, N sera le nombre d'instances du jeu de données (nombre de $x \in X$ disponibles pour les utilisateurs u considérés). Afin de fournir un x' suffisamment juste et combinable avec le vecteur x original, et ce afin d'éviter de biaiser le modèle d'apprentissage des algorithmes de *CMAB*, nous choisissons un niveau de confiance de 99% correspondant à $z = 2,575$, et une marge d'erreur $m = 0,01$.

Algorithme 4 : *ICE* pour les algorithmes de *CMABs* linéaires

Données : L'horizon T, l'ensemble des k bras $a \in A$ disponibles (c.-à-d., les éléments à recommander), l'ensemble des N contextes X (fixes) avec les utilisateurs u à considérer, l'algorithme de *CMAB* choisi, les paramètres spécifiques de l'algorithme de *CMAB* choisi, le nombre de d' classes de *ICE* voulues, la marge d'erreur m, le z-score z, et la proportion de la population $\widehat{p}$.

$obsCount \leftarrow 0$;
Calculer la taille de l'échantillon n comme défini à l'Équation (7.2);
pour tous les $x \in X$ **faire**
    $x' \leftarrow 0_{d'}$;
    $x \leftarrow x \oplus x'$
pour $t = 1$ *à* T **faire**
    Considérer l'utilisateur u évoluant dans un contexte $x_t \in X \subset \mathcal{R}^{d+d'}$ représenté par la paire utilisateur-contexte (u, x);
    Calculer $obs(t, u, x)$ le nombre de fois que la paire utilisateur-contexte (u, x) a été précédemment observée par l'algorithme;
    si *(*$obs(t, u, x) = 0$*)* **alors**
        $obsCount \leftarrow obsCount + 1$;
    si *(*$obsCount \geq n$ *et* $obs(t, u, x) \geq d' - 1$ *et* $x'_t = 0_{d'}$*)* **alors**
        Calculer $Acc_{u,x}(t)$;
        Classer chaque paire utilisateur-contexte (u, x) (dans une classe de précision) selon sa valeur de $Acc_{u,x}(t)$ et mettre à jour le vecteur x'_t;
        Mettre à jour x_t en remplaçant ses d' dernières dimensions initialisées à 0 par x'_t ;
    Choisir l'élément $a \in A$ selon la stratégie de l'algorithme de *CMAB* et recommander cet élément à l'utilisateur u_t;
    Observer la récompense r_t retournée;
    Mettre à jour les paramètres de l'algorithme de *CMAB* choisi selon sa stratégie de traitement de la récompense;

Dans la prochaine sous-section nous présentons nos expérimentations et les résultats de notre méthode *ICE*.

7.4.7 Expérimentations et Résultats

Afin d'expérimenter notre méthode, nous avons défini trois cas d'utilisation qu'il est possible de rencontrer dans les applications en ligne :

1. Nous utilisons le contexte « complet » (vecteur complet - *vc*) d'origine fourni par les jeux de données et tentons de l'améliorer avec *ICE* afin d'obtenir une meilleure précision globale ;
2. Nous tronquons le contexte d'origine (vecteur tronqué - *vt*) pour qu'il soit restreint. Ceci nous permet d'observer comment *ICE* réduit l'impact de la restriction de contexte sur la précision globale ;
3. Enfin, nous considérons un troisième cas particulier où aucune information contextuelle n'est disponible. Dans ce cas « non-contextuel », les algorithmes de *CMAB* ne peuvent pas être utilisés et un algorithme de *MAB* ne s'appuyant pas du contexte est préféré. Nous montrons ici comment de tels problèmes avec contexte caché peuvent être transformés avec *ICE* en problèmes contextuels équivalents pour lesquelles les algorithmes de *CMAB* sont plus adaptés et plus précis.

Dans tous les cas d'utilisation, indépendamment de l'algorithme de *MAB* ou de *CMAB* choisi initialement, nous capturons la précision individuelle de chaque paire (u, x) et l'utilisons pour créer du contexte en utilisant la méthode *ICE*. Nous considérons que les habitudes de récompense individuelles des utilisateurs dans un contexte donné forment des classes qui peuvent être en elles-mêmes utilisées comme information contextuelle caractérisant l'utilisateur.

7.4.7.1 Résultats pour les cas contextuels

Ces résultats font écho à notre contribution [Gut+18b] (étoile n° 4) présentée Figure 2 en introduction de ce mémoire.

Paramètres. Dans le cadre de nos expériences, afin d'observer différents niveaux d'enrichissement du vecteur de contexte, nous testons deux niveaux d'enrichissement : $d' = 4$ que nous nommons *ICE-4D*, et $d' = 10$ que nous nommons *ICE-10D*. Ceci correspondra à enrichir le contexte d'origine respectivement de quatre dimensions et de dix dimensions supplémentaires.

Jeux de données. L'évaluation de notre proposition se base sur cinq jeux de données que nous avons déjà décrits et utilisés au Chapitre 4 (voir tableau 4.1). Ces cinq jeux de données sont : *Contrôle* (*vc* et *vt*), *RS-ASM* (*vc* et *vt*) [9], *Food*, *Covertype* et *Poker Hand*.

9. Ce jeu de données a cette fois été utilisé dans son intégralité (parties *été* et *hiver*), soit 2152 instances.

Algorithmes comparés. Nous comparons des algorithmes de *CMAB LinUCB* et *CTS* qui ont été enrichis avec notre méthode *ICE* vs les algorithmes suivants : 1) Aléatoire ; 2) *MABs* : ε-*Greedy* ($\varepsilon = 0.20$) [SB98], *UCB2* [ACF02], et *Thompson Sampling*(*TS*) [Tho33] ; 3) *CMABs* : *LinUCB* [Li+10] et *CTS*[AG13] sans enrichissement (c.-à-d., sans utiliser *ICE*).

Protocole. Pour chaque jeu de données, nous exécutons 10 simulations pour chaque expérience : 1) *Aléatoire* ; 2) Algorithmes de *MAB* ; 3) Algorithmes de *CMAB* sans *ICE* ; 4) Algorithmes de *CMAB* avec *ICE-4D* et *ICE-10D*.

De plus, pour simuler un flux de données d'utilisateurs évoluant dans un contexte donné et se présentant pour recevoir une recommandation (voir ligne 7 de l'algorithme 4), nous sélectionnons les paires utilisateur-contexte une par une aléatoirement dans le jeu de données. Enfin, comme le nombre d'instances est différent pour chacun des jeux de données, nous définissons un horizon T différent afin d'obtenir des valeurs de précisions individuelles suffisamment pertinentes. Par conséquent, en fonction du nombre d'instances du jeu de données, nous paramétrons $T = 10\ 000\ 000$ itérations pour *Poker Hand* et *Covertype*, et $T = 200\ 000$ pour les autres jeux de données.

Sur les 10 simulations, pour chaque jeu de données et expérience, nous calculons : 1) La précision globale, $Acc(T)$, et son écart-type. Ces résultats sont observables dans le Tableau 7.4 ; 2) les regrets cumulés $\sum_{t=1}^{T} \rho(t)$ qui sont observables dans les Figures 7.4, 7.5, 7.6, 7.7, et 7.8.

De plus, nous réalisons un test de *Kruskal-Wallis* afin de mettre en lumière les inégalités entre les différents résultats des algorithmes. Pour chaque jeu de données (Table 7.4) le test de *Kruskal-Wallis* indique qu'il y a des différences significatives entre les résultats de précisions globales obtenus par chacun des algorithmes ($p < 0,01$). Ainsi, nous réalisons des tests de rangs signés de *Wilcoxon* sur la précision globale afin de vérifier (deux à deux) si les résultats des deux méthodes sont significativement différents. Cette significativité sera donnée via la p valeur en considérant si celle-ci est au minimum inférieure à $0,01$.

Impact de la restriction sur le contexte. Afin d'observer l'impact de la restriction sur le contexte c.-à-d., le manque de caractéristiques présentes observables dans le vecteur de contexte, nous avons au préalable expérimenté le jeu de données *Contrôle (vt)* (Voir Figure 7.4), puis nous avons expérimenté sur un jeu de données réel *RSAM (vt)* pour lequel nous avons tronqué une partie du vecteur de contexte (Voir Figure 7.5).

Par rapport aux versions *vc*, lorsque nous expérimentons sur des versions *vt* des jeux de données, nous observons que la précision globale diminue significativement pour chaque algorithme de *CMAB* ($p < 0,01$) (Voir Table 7.4 jeux de données en version *vc* vs jeux de données en version *vt*).

Comme les algorithmes de *MAB* ne s'appuient pas sur le contexte, nous n'observons aucun impact de la restriction sur le contexte concernant leurs résultats. Néanmoins, ceux-ci donnent malgré tout des résultats inférieurs ($p < 0,01$) que les *CMABs* et ce dans les deux cas (*vc* et *vt*).

D'autres parts, les résultats de précision globale dans le Tableau 7.4 soulignent que notre méthode de *CMAB+ICE* reste la meilleure pour chaque jeu de données (*Contrôle (vt)*, *RS-ASM (vt)* et *RS-ASM (vc)*) par rapport aux méthodes de *MAB* et de *CMAB* ($p < 0,01$). De plus, les regrets cumulés sont inférieurs à ceux obtenus avec toute autre méthode : Voir Figure 7.4 pour le contrôle et Figure 7.5 pour *RS-ASM*. Également, nous pouvons noter que lorsque x^* est fourni (*Contrôle (vc)*), il n'y a pas de différence entre les méthodes *CMAB* d'origine et *CMAB+ICE* (Voir Table 7.4) puisque le contexte donné en entrée est déjà optimal et ne nécessite pas d'enrichissement complémentaire. Néanmoins, nous remarquons que le vecteur de contexte original fourni dans *RS-ASM (vc)* semble malgré tout incomplet et nécessite d'être complété par notre méthode *ICE*. En effet, nous observons pour *RS-ASM (vc)* que les résultats de précision globale et de regrets cumulés obtenus avec *ICE* sont meilleurs que ceux obtenus par les algorithmes de *CMAB* d'origines ($p < 0,01$).

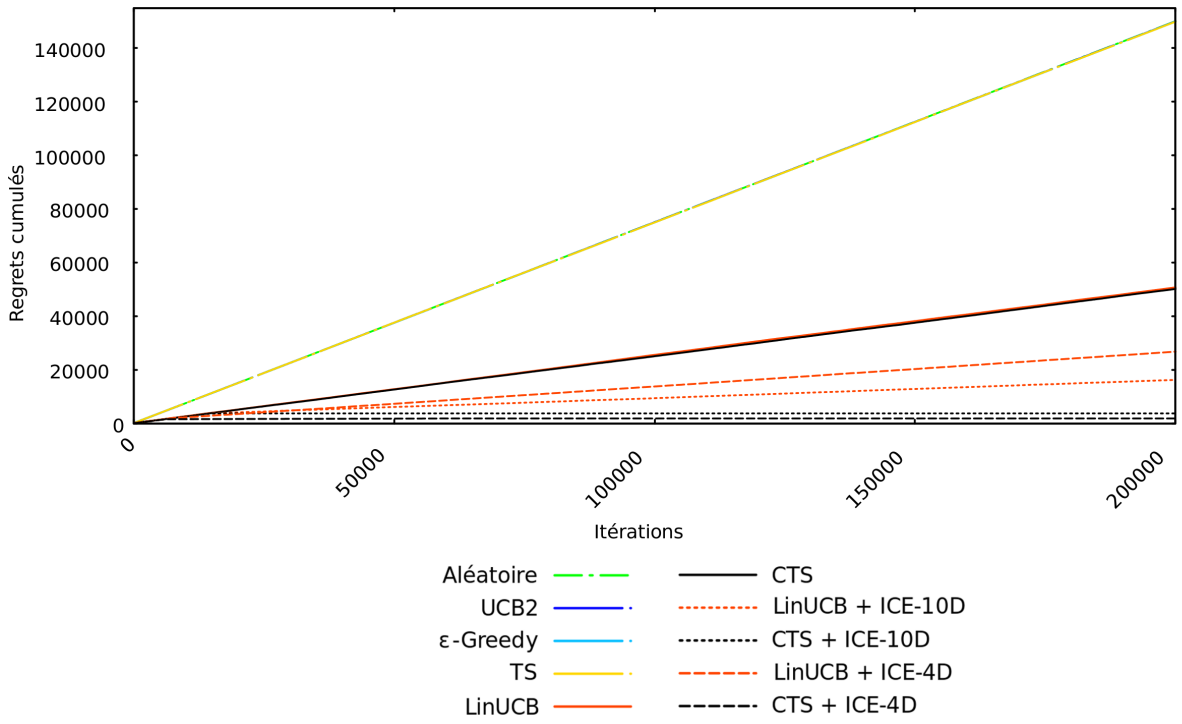


FIGURE 7.4 – Regrets cumulés pour *Contrôle (vt)*

Dans les Figures 7.6, 7.7,et 7.8 ainsi que dans le Tableau 7.4 nous observons que *ICE* donne de meilleurs résultats que les méthodes traditionnelles de *MAB* et *CMAB* ($p < 0,01$) sauf pour le jeu de données *Poker Hand* avec le paramétrage *ICE-4D* pour lequel il n'y a aucune différence significative ($p > 0,05$). Comme pour ce même jeu de données le paramétrage *ICE-10D* obtient finalement de meilleurs résultats que les *MABs* et *CMABs* ($p < 0,01$). Nous

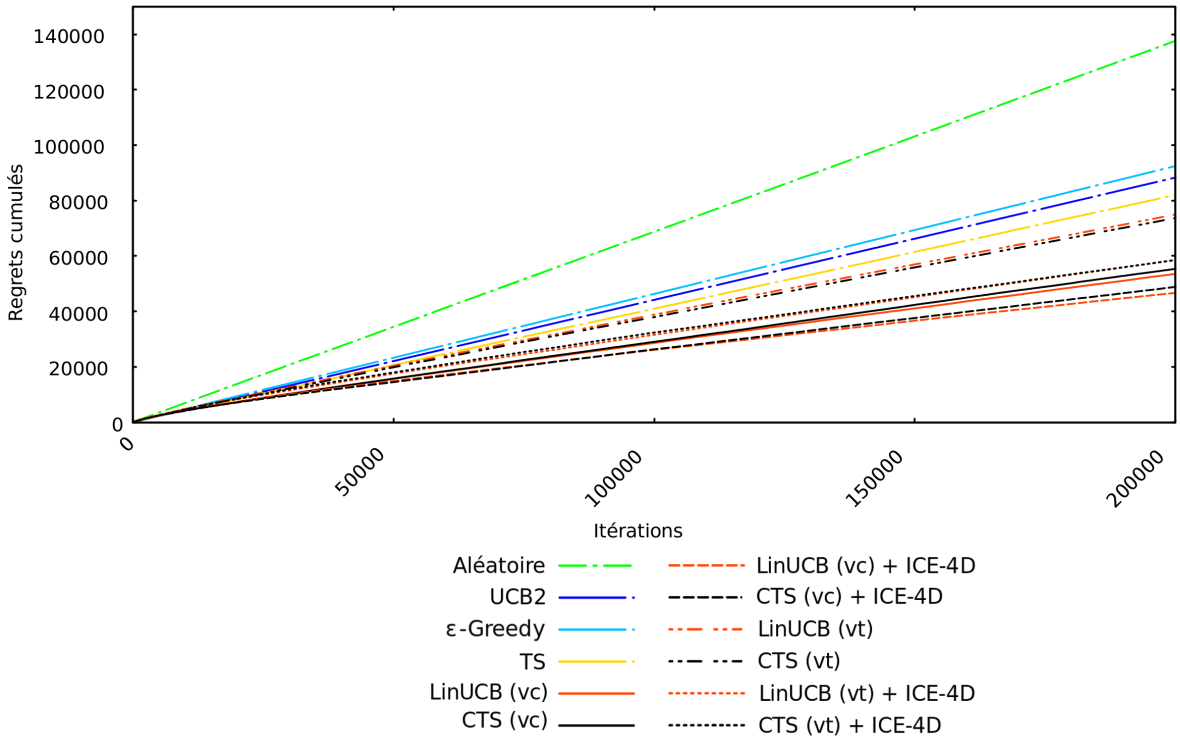


FIGURE 7.5 – Regrets cumulés pour *RS-ASM (vc)* et *(vt)*

pouvons donc en déduire que pour le jeu de données *Poker Hand* un enrichissement de seulement quatre dimensions ne suffit pas à bien différencier les contextes et que celui-ci nécessite un découpage plus important en termes de niveau de précision.

Enfin, à partir des Figures 7.6, 7.7,et 7.8, nous pouvons déduire que plus le niveau de précision ($d' = 4$ ou $d' = 10$) est élevé, plus les algorithmes mettent du temps à converger. Par conséquent, il est nécessaire de déterminer un compromis entre la limite asymptotique de la précision globale et le temps nécessaire pour l'atteindre. Nous observons que *ICE* paramétré avec $d' = 4$ diminue plus rapidement le nombre de regrets cumulés (en comparaison à un algorithme de *CMAB* sans *ICE*) que s'il est paramétré avec $d' = 10$. Par exemple, dans le cas des systèmes de recommandation où l'on doit atteindre rapidement l'efficacité, $d' = 4$ serait un paramètrage plus intéressant.

Conclusion sur les cas contextuels. Nous proposons une nouvelle méthode d'enrichissement du contexte (*ICE*) qui peut être appliquée aux algorithmes de *CMAB* linéaires afin d'améliorer leurs performances. Les principaux avantages de notre méthode sont que : 1) *ICE* fournit expérimentalement une précision globale plus élevée et des regrets cumulatifs moins élevés ; 2) *ICE* est également suffisamment générique pour être combiné à différents algorithmes de

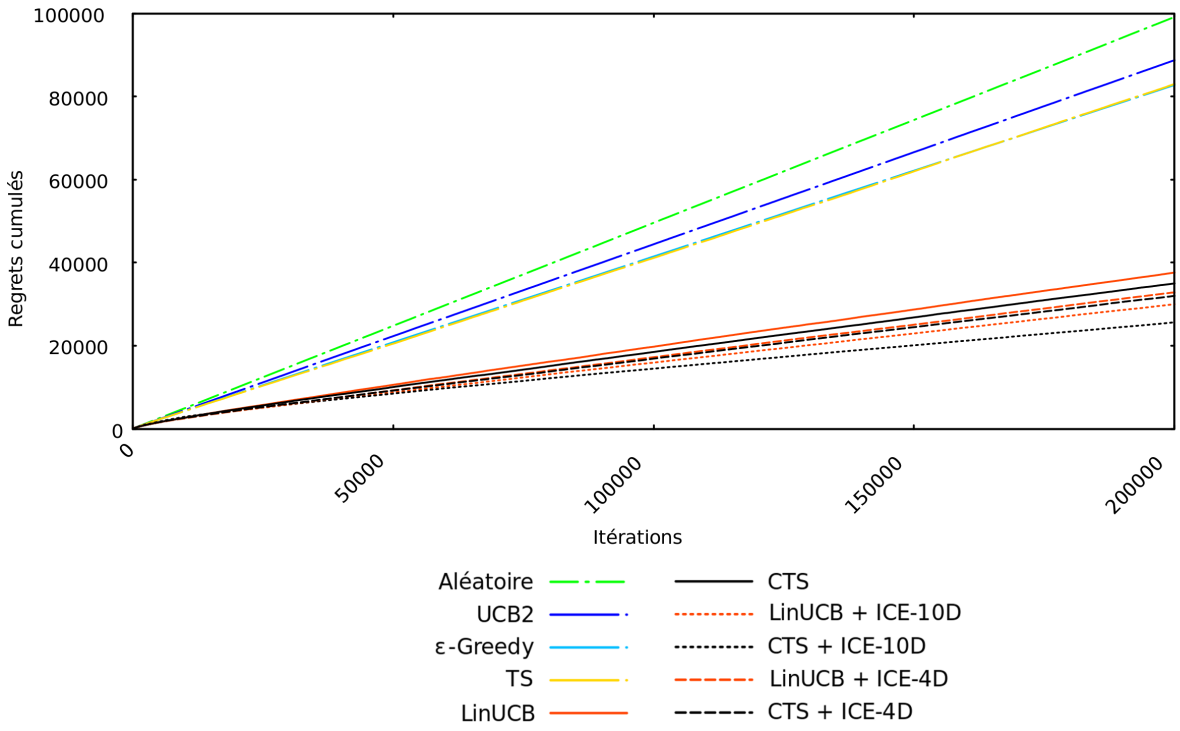


FIGURE 7.6 – Regrets cumulés pour *Food*

CMAB linéaires. Cependant en contrepartie, utiliser notre méthode augmente la taille de la dimension de contexte d dont dépend le regret attendu théorique. Ainsi, en fonction du niveau d'enrichissement que nous souhaitons obtenir, des itérations supplémentaires peuvent être nécessaires avant que des améliorations en termes de précision ne soient constatées.

Il est désormais intéressant de nous pencher sur un cas spécifique où aucune information contextuelle n'est disponible. Nous allons donc vérifier si notre méthode permet à un problème non contextuel de devenir contextuel, en augmentant la précision globale et réduisant les regrets cumulés.

7.4.7.2 Résultats pour les cas non-contextuels

Ces résultats font échos en partie à notre contribution [Gut+18b] (étoile n°4) et plus particulièrement [Gut+19d] (étoile n°5) présentées Figure 2 en introduction de ce mémoire.

Comme les jeux de données dépourvus de contexte ne fournissent aucune caractéristique sur laquelle un algorithme de *CMAB* puisse s'appuyer, nous ne pouvons pas commencer par utiliser *LinUCB* ou *CTS* au départ. Il est donc nécessaire d'amorcer le système de recommandation avec un algorithme de *MAB*. Au Tableau 7.4, on observe parmi les algorithmes de *MAB* que *Thompson Sampling* est meilleur que ε-*Greedy* ou encore *UCB2*. En conséquence de quoi

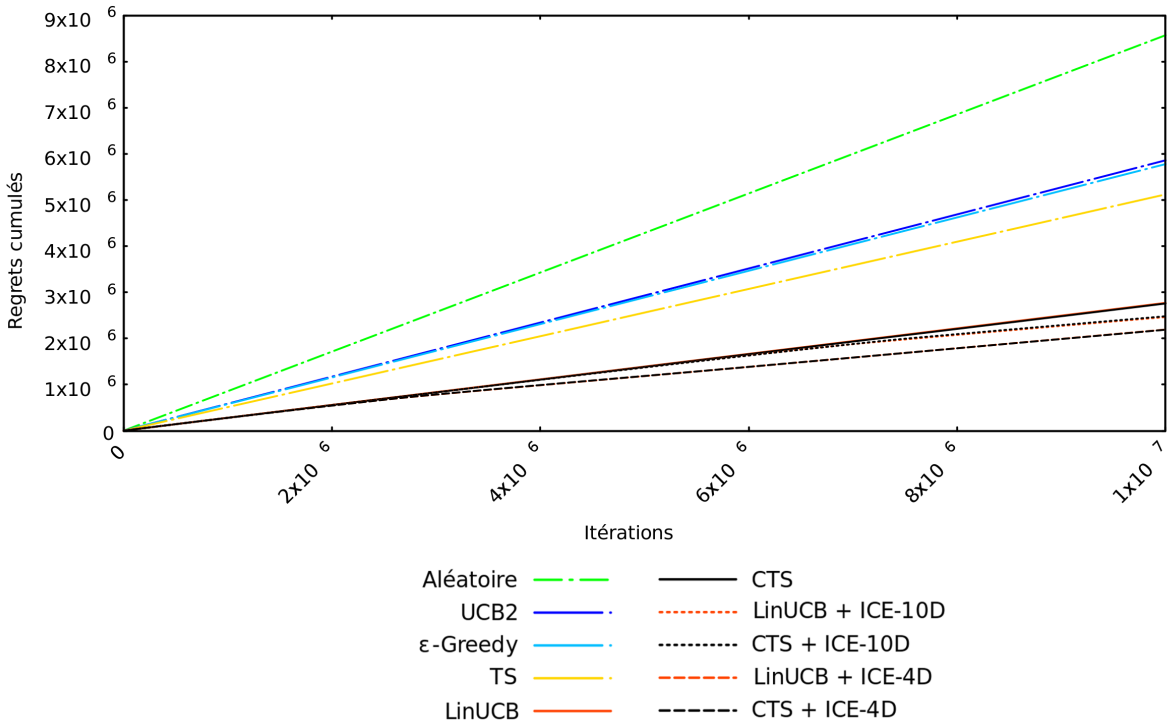


FIGURE 7.7 – Regrets cumulés pour *Covertype*

pour les cas non contextuels nous décidons d'amorcer la méthode en utilisant *TS* et continuons avec *LinUCB+ICE* une fois que *ICE* peut être déclenchée (voir ligne 11 de l'algorithme 4).

Étude préliminaire avec le jeu de données *Jester*. L'évaluation de notre étude préliminaire est basée sur *Jester* qui est un jeu de données dépourvu de contexte (voir sa description au Chapitre 4). En partant d'un jeu de données naturellement non contextuel, cette étude est nécessaire afin de confirmer : d'une part le bon algorithme de *MAB* à utiliser, d'autre part la configuration *ICE-4D* ou *ICE-10D* à employer.

En nous basant sur les résultats obtenus aux Tableaux 7.4 et 7.5 ainsi qu'à la Figure 7.9, nous confirmons que *TS* surpasse les autres algorithmes de *MAB* et que c'est ainsi l'algorithme le plus pertinent à utiliser pour amorcer la méthode. De plus, nous remarquons aussi (Tableaux 7.4 et 7.5) qu'utiliser quatre classes d'enrichissement ($d = 4$) semble être un bon compromis entre la limite asymptotique de la précision globale et le temps nécessaire pour l'atteindre. De ce fait, pour l'étude des cas non contextuels nous continuerons avec une configuration de type *ICE-4D* et amorcerons la méthode avec *TS*.

	Aléatoire	*UCB2*	*ε-Greedy*	*TS*	*LinUCB*	*LinUCB + ICE-4D*	*LinUCB + ICE-10D*	*CTS*	*CTS + ICE-4D*	*CTS + ICE-10D*
Contrôle (vc)	$0,249_{\pm\xi}$	$0,250_{\pm\xi}$	$0,250_{\pm 0,001}$	$0,250_{\pm 0,001}$	$1,000_{\pm\xi}$	$1,000_{\pm\xi}$	$1,000_{\pm\xi}$	$1,000_{\pm\xi}$	$1,000_{\pm\xi}$	$1,000_{\pm\xi}$
Contrôle (vt)	N/A[6]	N/A[6]	N/A[6]	N/A[6]	$0,749_{\pm 0,001}$	$0,866_{\pm 0,05}$	$0,901_{\pm 0,003}$	$0,750_{\pm 0,003}$	$0,991_{\pm 0,003}$	$0,982_{\pm 0,001}$
RS-ASM (vc)	$0,311_{\pm\xi}$	$0,558_{\pm 0,06}$	$0,538_{\pm 0,001}$	$0,591_{\pm 0,001}$	$0,740_{\pm 0,001}$	$0,766_{\pm 0,005}$	$0,778_{\pm 0,005}$	$0,735_{\pm 0,001}$	$0,756_{\pm 0,005}$	$0,777_{\pm 0,005}$
RS-ASM (vt)	N/A[6]	N/A[6]	N/A[6]	N/A[6]	$0,625_{\pm 0,001}$	$0,707_{\pm 0,005}$	$0,738_{\pm 0,005}$	$0,628_{\pm 0,001}$	$0,707_{\pm 0,005}$	$0,733_{\pm 0,005}$
Food	$0,504_{\pm\xi}$	$0,556_{\pm 0,03}$	$0,586_{\pm 0,007}$	$0,584_{\pm 0,001}$	$0,812_{\pm 0,002}$	$0,835_{\pm 0,004}$	$0,850_{\pm 0,004}$	$0,824_{\pm 0,001}$	$0,840_{\pm 0,002}$	$0,872_{\pm 0,002}$
Covertype	$0,143_{\pm\xi}$	$0,413_{\pm 0,06}$	$0,422_{\pm 0,008}$	$0,487_{\pm 0,002}$	$0,722_{\pm 0,001}$	$0,780_{\pm 0,001}$	$0,752_{\pm 0,001}$	$0,724_{\pm 0,001}$	$0,781_{\pm 0,001}$	$0,751_{\pm 0,001}$
Poker Hand	$0,100_{\pm\xi}$	$0,471_{\pm 0,04}$	$0,425_{\pm 0,005}$	$0,500_{\pm 0,001}$	$0,533_{\pm 0,001}$	$0,534_{\pm 0,001}$	$0,581_{\pm 0,002}$	$0,533_{\pm 0,001}$	$0,534_{\pm 0,001}$	$0,584_{\pm 0,002}$

TABLE 7.4 – Cas contextuels : Précision globale $Acc(T)$ obtenus par les différents algorithmes comparés pour chaque jeu de données expérimenté
($\xi = 0,0009$)

	Aléatoire	*UCB2*	*ε-Greedy*	*TS*	*LinUCB*	*LinUCB + ICE-4D*	*LinUCB + ICE-10D*	*CTS*	*CTS + ICE-4D*	*CTS + ICE-10D*
Jester	$0,335_{\pm\xi}$	$0,504_{\pm 0,04}$	$0,696_{\pm 0,001}$	$0,745_{\pm 0,001}$	N/A[7]	$0,834_{\pm 0,005}$	$0,788_{\pm 0,005}$	N/A[7]	$0,833_{\pm 0,004}$	$0,796_{\pm 0,004}$

TABLE 7.5 – Étude préliminaire : Précision globale $Acc(T)$ obtenus par les différents algorithmes comparés sur le jeu de données *Jester*
($\xi = 0,0009$)

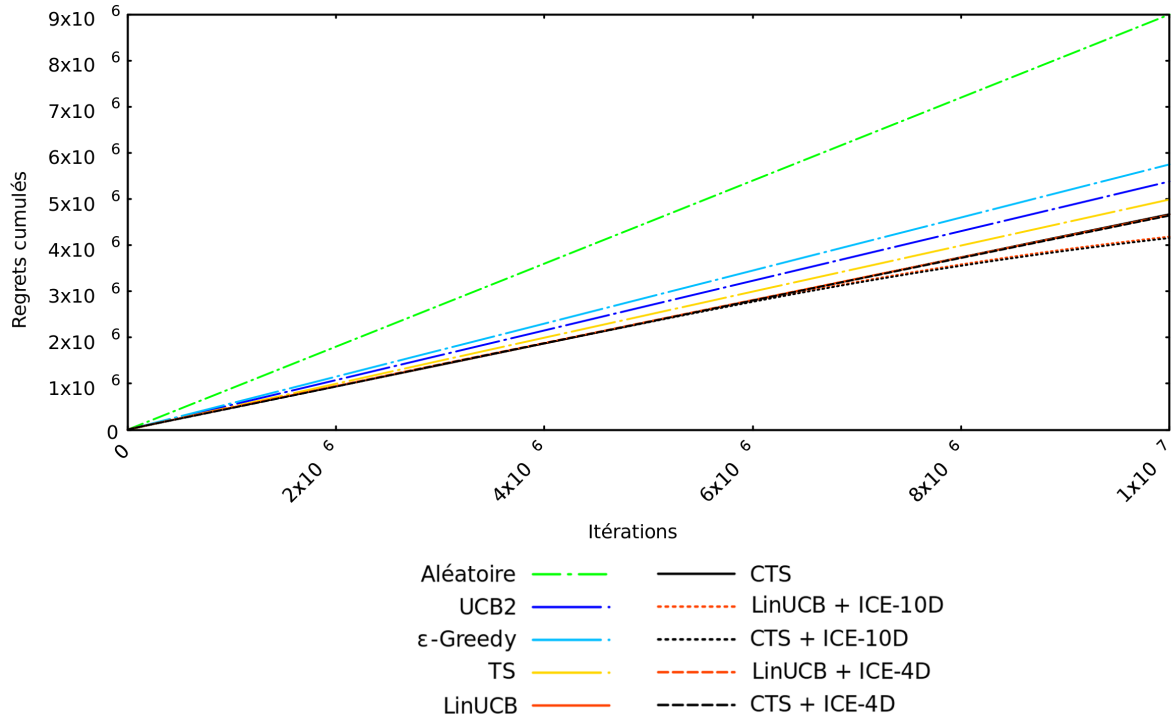


FIGURE 7.8 – Regrets cumulés pour *Poker Hand*

Autres cas réels non-contextuels. L'évaluation de notre proposition est basée sur quatre jeux de données parmi lesquels un jeu de données artificiel sert de contrôle et trois sont des jeux de données du monde réel dans lesquels nous avons caché le contexte (contexte caché - *cc*) :

— ***Contrôle*, *RS-ASM*, et *Food*** qui sont décrits au Chapitre 4 ;

— ***CNAE-9*** qui est un jeu de données issu du dépôt d'apprentissage automatique de l'*UCI*[10] (*UCI Machine Learning Repository*). Il contient des descriptions commerciales d'entreprises brésiliennes classées selon neuf catégories d'activité économique (c.-à-d., selon le tableau *National Classification of Economic Activities*).

Le contexte est caché (*cc*) dans tous les jeux de données listés ci-dessus : nous avons supprimé toutes les informations contextuelles disponibles.

L'objectif est donc d'étudier la capacité qu'à *ICE* à transformer un problème non contextuel en un problème contextuel équivalent pour lequel nous devons minimiser les regrets cumulés.

Les algorithmes comparés seront *TS* et *LinUCB+ICE-4D*. Ainsi pour chaque jeu de données, nous exécutons 10 simulations pour chaque expérience : 1) concernant l'algorithme *TS*

10. `http://archive.ics.uci.edu/ml/`

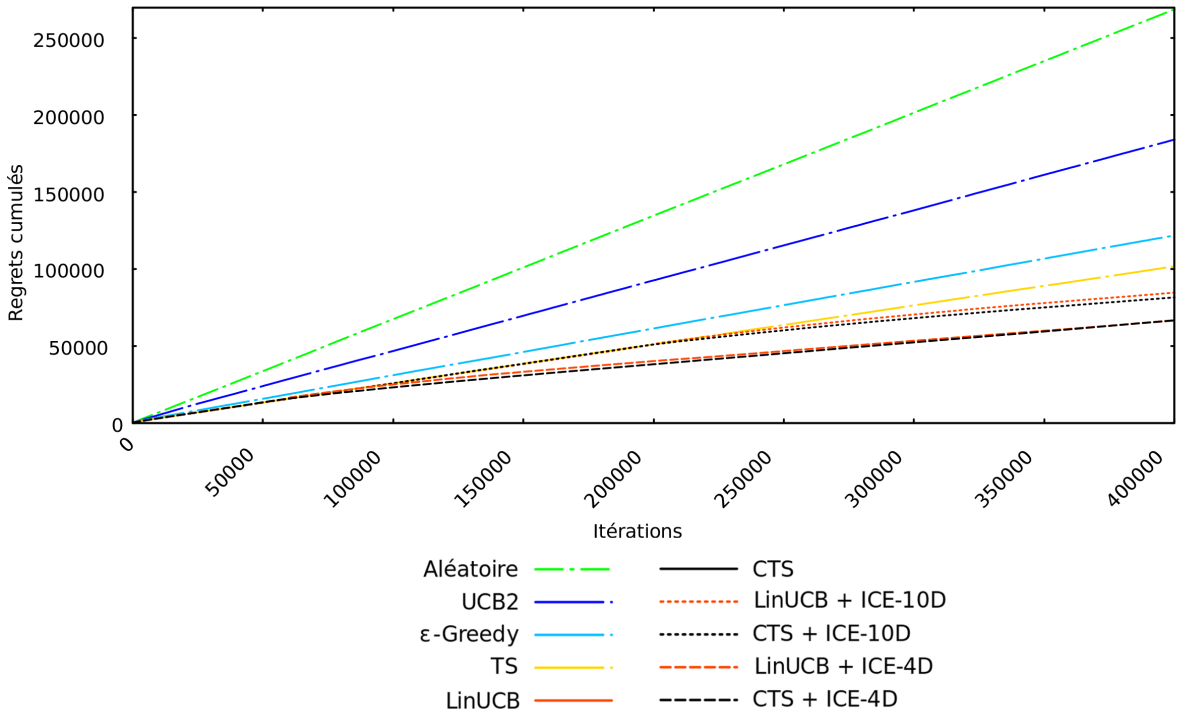


FIGURE 7.9 – Regrets cumulés obtenus par les différents algorithmes comparés sur le jeu de données *Jester*

et 2) concernant l'algorithme *LinUCB* avec *ICE-4D*. De nouveau, pour simuler un flux de données d'utilisateurs évoluant dans un contexte donné et se présentant pour recevoir une recommandation (voir ligne 7 de l'algorithme 4), nous sélectionnons les paires utilisateur-contexte une par une aléatoirement dans le jeu de données. De plus, nous fixons l'horizon à $T = 100\ 000$ afin d'obtenir des mesures suffisamment précises de la précision individuelle. Sur les 10 simulations, pour chaque jeu de données et expérience, nous calculons les regrets cumulés $\sum_{t=1}^{T} \rho(t)$ montrés à la Figure 7.10.

Nous observons les résultats au Tableau 7.6 et à la Figure 7.10.

Nous observons dans la Figure 7.10 et dans le Tableau 7.6 que *ICE* obtient de meilleurs résultats dans tous les cas. Sur les jeux de données choisis, la création du contexte avec la méthode *ICE* permet de réduire les regrets cumulés à l'horizon jusqu'à -33%.

Conclusion sur ces cas non contextuels. La méthode *ICE* a été initialement conçue pour améliorer les performances des algorithmes de *CMAB* en enrichissant leur contexte. Nous l'avons étendue sur des cas avec contexte caché (*cc*) et l'avons évaluée sur plusieurs jeux

11. Les deux méthodes (avec ou sans *ICE*) ont un nombre de regrets cumulés égal jusqu'au point de rentabilité.

	Regrets cumulés		***Comparaison***	
Jeu de données	***TS***	***LinUCB +ICE-4D***	**Gain**	**Point de rentabilité** [11]
Contrôle	74 848	51 666	31%	$t \approx 7\ 500$
RS-ASM	41 034	27 535	33%	$t \approx 16\ 000$
Food	40 884	36 743	10%	$t \approx 5\ 000$
CNAE-9	88 851	82 637	7%	$t \approx 30\ 000$

TABLE 7.6 – Cas non-contextuel : Regrets cumulés obtenus par *TS* et *LinUCB+ICE4D* pour chaque jeux de données en version *cc*

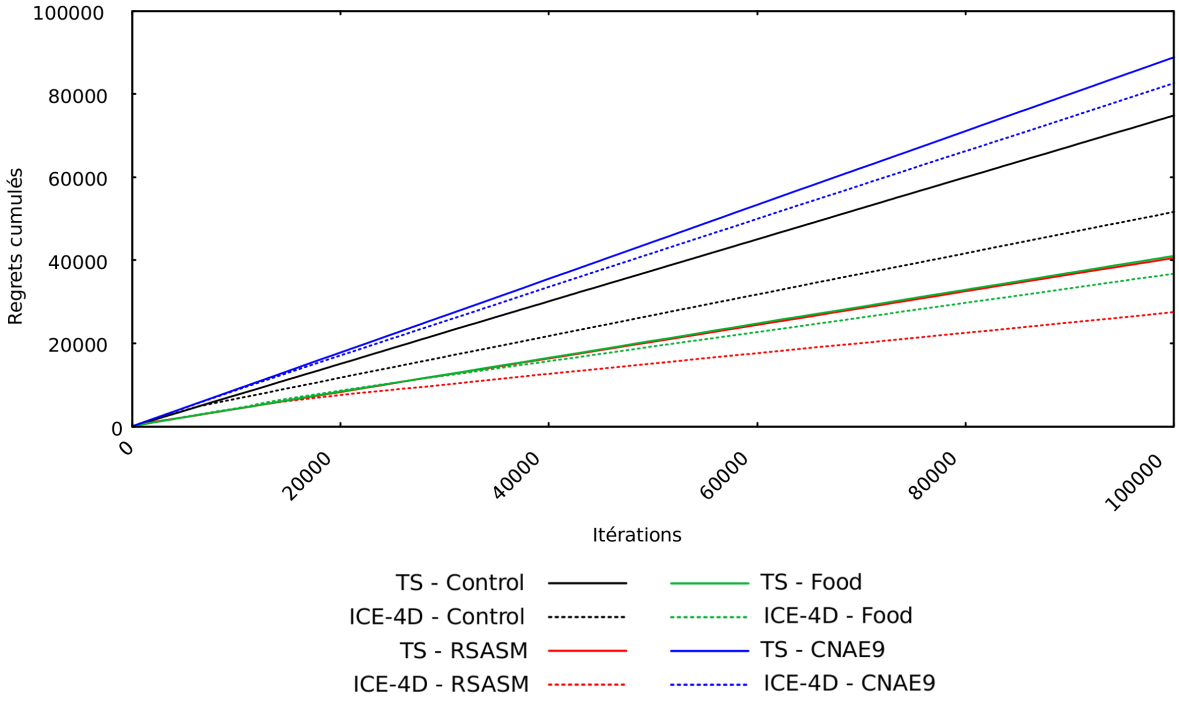


FIGURE 7.10 – Regrets cumulés obtenus par chaque algorithme comparé pour chaque de jeux de données expérimenté en version (*cc*)

de données du monde réel. Nos expériences ont montré qu'en recréant un contexte pour des problèmes sur contexte caché qui devraient être résolus normalement par un algorithme *MAB* uniquement, notre méthode *ICE* obtient des regrets cumulés plus bas que *TS*, un algorithme de *MAB* non contextuel.

Cette nouvelle méthode montre ainsi de bons résultats empiriques, mais ses bornes de regrets doivent encore être prouvées théoriquement. De plus, notre méthode devrait aussi être observé lors d'une utilisation en ligne afin de vérifier que ses performances suivent celles

que nous avons obtenus dans nos expérimentations hors-ligne sur des jeux de données pré-existants.

7.5 Conclusion et Perspectives

Dans ce chapitre nous avons décrit nos contributions portant sur les techniques de raisonnement contextuel en vue de déduire des caractéristiques de contexte exploitables et pertinentes pour les systèmes de recommandation.

Les deux premières contributions : Top-k routes et prédiction de mobilité offrent des perspectives tout à fait intéressantes pour les systèmes de recommandation mais n'ont pas encore pu être évaluées en ligne.

En revanche, les deux techniques suivantes ont pu être évaluées dans un système de recommandation, soit :

1. En ligne, où nous avons raisonné sur des informations de contexte spatial brut via une méthode d'apprentissage non supervisée : le partitionnement spectral, afin de déduire des informations structurées et exploitables par le système de recommandation de *scéno* ;
2. Hors ligne, à l'aide de jeux de données du monde réel où nous avons mis en place une méthode, *ICE*, permettant d'enrichir du contexte en déduisant des préférences utilisateurs de la précision individuelle mesurée par rapport aux recommandations précédemment effectuées.

L'ensemble des résultats présentés dans ce chapitre montre l'importance de prendre en considération de nouvelles caractéristiques pertinentes du contexte pour que les systèmes de recommandation gagnent en précision.

En termes de perspectives pour des applications réelles concernant les système de recommandation, dans lesquelles la dépendance entre les contextes et les récompenses peuvent varier dans le temps, il semblerait pertinent d'opérer un raisonnement contextuel plus dynamique.

De ce fait, en ce qui concerne le géo-partionnement, il serait intéressant de considérer une méthode dynamique de calcul des clusters afin de tenir à jour les quartiers inférés fournis au système de recommandation. En effet, si on prend l'exemple du tourisme dans une ville qui posséderait à la fois la mer et la montagne, un cluster de fréquentation touristique qui aurait été détecté pour la mer l'été, pourrait très bien évoluer l'hiver et se retrouver en montagne. Cette saisonnalité dans la prise en compte de l'information géographique est importante à considérer tant elle influe sur la performance des systèmes de recommandation.

De même, concernant *ICE*, un enrichissement dynamique du contexte permettrait de contrer une possible non-stationnarité. D'autre part, puisque la taille de dimension résultante $d + d'$ du vecteur de contexte peut devenir trop importante, il serait intéressant d'alterner de manière itérative l'enrichissement de contexte et la réduction de dimensions de contexte afin de ne conserver que les éléments les plus importants (c.-à-d., les caractéristiques pertinentes).

CONCLUSION ET PERSPECTIVES

Dans cette thèse, nous nous sommes intéressés aux algorithmes de bandits-manchots contextuels et non contextuels pour la recommandation. Ce sujet possède de multiples facettes et a nécessité que nous l'abordions selon trois axes : les systèmes de recommandation ; les algorithmes de bandits-manchots contextuels et non contextuels ; le contexte.

Un état de l'art triple...

Ainsi, dans la Partie I, nous avons dressé un état de l'art de ces trois axes sur trois chapitres : le Chapitre 1 rappelant les différentes approches employées dans les systèmes de recommandation ; le Chapitre 2 formalisant le problème de bandits-manchots pour la recommandation et rappelant les principaux algorithmes permettant de le résoudre ; le Chapitre 3 explicitant la notion de contexte à travers sa modélisation, son acquisition, et son raisonnement.

Chapitre 1 - Les systèmes de recommandation

Dans le Chapitre 1, nous avons rappelé différentes approches existantes utilisées par les systèmes de recommandation. Nous les avons confrontées afin de choisir celles qui correspondraient le mieux à nos besoins applicatifs finaux. Nous avons opté pour une approche contextuelle de la recommandation basée sur les modèles : les bandits-manchots contextuels. L'avantage d'une telle approche est qu'elle permet d'améliorer l'explicabilité des algorithmes et la résolution du problème de recommandation. De plus, elle reste efficace en environnement dynamique (p. ex., où les goûts et préférences utilisateurs évoluent au cours du temps). D'autre part, dans notre application finale, nous ne possédons aucune connaissance a priori sur la communauté utilisateur. C'est pourquoi, l'apprentissage par renforcement et notamment les bandits-manchots qui se prêtent bien à cette situation, où les informations se dévoilent de manière séquentielle, restent très bien adaptés.

Chapitre 2 - Les bandits-manchots pour la recommandation

Dans le Chapitre 2, nous avons défini le problème du bandit-manchot contextuel (*Contextual Multi-Armed Bandit - CMAB*) et non contextuel (*Multi-Armed Bandit - MAB*) pour la recommandation et avons présenté différents algorithmes permettant de le résoudre. Nous avons réalisé un tableau comparatif de l'ensemble de ces algorithmes sur les critères suivants : précisions et personnalisation, applicabilité et temps-réel, résistance à la non stationnarité. Sur la base de ces critères, nous avons sélectionné sept algorithmes afin de pouvoir les évaluer hors ligne

avant de pouvoir en employer certains pour nos applications futures. Plus particulièrement, parmi les algorithmes de *MAB* nous avons retenu : *UCB2*, ε*-Greedy* avec ε fixe, *Thompson Sampling*, et *EXP3*. Parmi les algorithmes de *CMAB* nous avons retenu : *LinUCB*, *Contextual Thompson Sampling* et *EXP4.P*. Les informations contextuelles pouvant être parfois restreintes voire non pertinentes ou temporairement non capturables, nous évaluons systématiquement nos algorithmes de bandits-manchots contextuels avec des équivalents non contextuels (en ligne comme en hors ligne).

Chapitre 3 - Le contexte

Dans le Chapitre 3, nous avons décrit les principes de modélisation, d'acquisition, et de raisonnement contextuel ainsi que les principales définitions données au contexte. Le contexte est une composante importante du problème de recommandation puisqu'il en contraint la résolution. D'autre part dans tout problème de décision, la question de la qualité et de l'exploitabilité des données fournies en entrée reste primordiale. Ainsi dans ce chapitre nous avons abordé les questions de l'acquisition du contexte et de son raisonnement afin de pouvoir le rendre exploitable par des systèmes de recommandation.

Contributions aux algorithmes de bandits-manchots

Dans la Partie II, nous avons décrit aux Chapitres 4 et 5, nos contributions concernant les algorithmes de *MAB* et de *CMAB*.

Chapitre 4 - Algorithmes de bandits-manchots : une histoire de précision

Le Chapitre 4 porte sur une étude préliminaire où nous avons expérimenté les sept algorithmes de *MAB* et de *CMAB* retenus dans le Chapitre 2. Pour ce faire, nous nous sommes appuyés sur douze jeux de données dont dix sont issus du monde réel et deux ont été générés de manière artificielle. Nous avons observé et analysé les résultats de ces expériences sur trois critères : la précision globale des recommandations ; la diversité des recommandations ; la précision individuelle des recommandations.

Dans un premier temps, nous avons observé que les méthodes de *MAB* comme les méthodes de *CMAB* même si elles obtenaient en règle générale de bons résultats de précisions globales, n'étaient pas équitables en termes de précision individuelle des recommandations au sein de la population. Nous avons également remarqué, que la diversité des recommandations pour les *CMAB* et encore plus pour les *MAB* pouvait être faible voire inexistante dans le sens où ces algorithmes sont prévus pour maximiser leurs gains (en termes de précision des recommandations). À noter par contre que, si le contexte donné en entrée des algorithmes de *CMAB*, est suffisamment complet et pertinent, nous observons un net avantage à utiliser un algorithme de *CMAB* par rapport à un algorithme de *MAB* en termes de précision globale, de diversité et de précision individuelle. C'est pourquoi, afin de pallier le manque de diversité observé pour

les algorithmes de *CMAB* dans cette phase d'expérimentation préliminaire, nous avons proposé une modification de l'algorithme *LinUCB* en y ajoutant un mécanisme de diversification s'appuyant sur une fenêtre glissante.

D'autre part, malgré des preuves théoriques solides en ce qui concerne les algorithmes de *MAB* et de *CMAB*, l'évaluation empirique hors ligne a montré que chaque algorithme obtenait des résultats de précision globale, de diversité et de précision individuelle différents pour chaque jeu de données, si bien qu'il est totalement impossible de prédire à l'avance quel algorithme sera le plus performant pour chaque jeu de données. De telles observations nous ont amené à considérer que dans le cas d'applications (en ligne) de recommandation à des utilisateurs mobiles, il est impossible de décider à l'avance quel algorithme utiliser pour obtenir les meilleurs résultats sur ces 3 critères. C'est pourquoi, afin de pallier ce problème nous avons mis en place *Gorthaur* : une nouvelle méthode, de type portfolio, de sélection dynamique d'algorithmes de *MAB* et de *CMAB*.

L'ensemble de ces observations préliminaires est le point d'entrée des principales contributions qui ont suivi : nous avons émis l'hypothèse que la diversité pourrait en partie pallier le manque d'équité en termes de précision individuelle (voir Chapitre 5) ; nous avons considéré que la précision individuelle qu'obtient un algorithme de *MAB* ou de *CMAB* pour chaque individu (c.-à-d., utilisateur) est un critère intéressant à prendre en compte dans l'enrichissement du contexte des systèmes de recommandation (voir Chapitre 7).

Chapitre 5 - Améliorer la précision individuelle : diversifier les recommandations

Dans le Chapitre 5, nous avons abordé la question de la diversification des recommandations : soit afin de tenir compte du fournisseur d'éléments à recommander (p. ex., enseignes publicitaires, pourvoyeur d'événements) et qui souhaite que ses éléments soit suffisamment pris en compte ; soit afin de pallier le manque d'équité en termes de précision individuelle au sein des individus. Afin de gagner en diversité, nous avons créé deux nouvelles méthodes : *SW-LinUCB* et *Gorthaur*.

La première méthode a consisté à faire évoluer un algorithme de *CMAB* : *LinUCB*, en y intégrant un mécanisme de diversification basé sur une fenêtre glissante. Ce nouvel algorithme : *SW-LinUCB*, est une heuristique qui a permis d'obtenir une meilleur diversification des recommandations mais également de diminuer le nombre d'utilisateurs pour lesquels la précision individuelle était faible. Ces améliorations ont cependant un coût en précision global pouvant aller jusqu'à 10% de moins.

La seconde méthode a consisté à employer une approche porte-folio d'algorithmes de *MAB* et *CMAB*. Cette nouvelle méthode : *Gorthaur*, est une heuristique qui a pour objectif de maximiser à la fois la précision globale des recommandations et leur diversité. Pour ce faire, *Gorthaur* sélectionne proportionnellement les algorithmes de son porte-feuille selon leurs capacités à maximiser ces deux critères. Cette méthode a l'avantage de pallier le fait qu'il est difficile voire impossible de choisir au préalable l'algorithme qui correspond le mieux à notre problème (c.-à-d., pour n'importe quel jeu de données ou applications en ligne) selon les critères de précisions

globales et de diversité. De plus, *Gorthaur* permet de limiter le nombre d'individus pour lesquels la précision individuelle est faible (inférieure à $0,25$).

Ces deux méthodes (*SW-LinUCB* et *Gorthaur*), bien qu'elles ont montré un certain nombre d'avantages à être employées en évaluation hors ligne, restent néanmoins à évaluer dans une application en ligne.

Contributions portant sur le contexte

Dans la Partie III, nous avons décrit aux Chapitres 6 et 7, nos contributions concernant l'acquisition de contexte et le raisonnement contextuel pour les systèmes de recommandation.

Chapitre 6 - Modélisation, capture et analyse préliminaire du contexte pour la recommandation

Dans le Chapitre 6, nous avons en premier lieu décrit les résultats de notre étude préliminaire sur le contexte dans la ville d'Angers. L'objectif a été de mettre en lumière comment, à partir de journaux de connexions aux points d'accès d'un réseau Wi-Fi urbain, il est possible d'extraire des informations contextuelles exploitables par un système de recommandation.

En seconde section de ce chapitre nous avons présenté notre application mobile *scéno*, permettant la visualisation et la recommandation d'événements culturels dans la ville d'Angers. Le système de recommandation de notre application mobile *scéno* utilise des algorithmes de *MAB* et de *CMAB*. Nous y avons détaillé l'architecture et les composants permettant d'acquérir du contexte brut en partie exploitable par le système de recommandation.

Certaines des informations contextuelles de ces sources de données (Wi-Fi urbain et application mobile), ne peuvent cependant pas être exploitées sous leur forme brute par les systèmes de recommandation et nécessitent que nous raisonnions sur le contexte.

Chapitre 7 - Raisonnements contextuels et application aux systèmes de recommandation

Dans le Chapitre 7, nous avons présenté plusieurs méthodes de raisonnement contextuel pour les systèmes de recommandation, tirant profit de deux différentes sources de données : les journaux de connexions au Wi-Fi urbain ; la précision individuelle des utilisateurs vis à vis des recommandations qui leur sont faites.

Ainsi dans ce chapitre nous avons proposé trois méthodes de raisonnement contextuel à partir des journaux de connexions au Wi-Fi urbain : l'étude des trajectoires des utilisateurs mobiles dans la ville, la prédiction de la mobilité, et la déduction de quartiers dans la ville via l'emploi d'un algorithme de partitionnement spectral. Les résultats de ce dernier raisonnement ont pu être intégrés et évalués directement au sein de notre composant de recommandation de l'application mobile *scéno*. Nous avons pu ainsi observer que fournir du géo-contexte raisonné à notre système de recommandation d'événements culturels de l'application mobile *scéno*, permet d'améliorer la précision globale de ses recommandations.

Enfin, nous avons proposé une ultime méthode de raisonnement contextuel. Elle consiste en l'utilisation de la mesure de précision individuelle des utilisateurs évoluant dans un contexte donné afin d'inférer du contexte et d'enrichir les informations contextuelles fournies aux algorithmes de *CMAB* utilisés pour la recommandation. Cette dernière méthode : *ICE*, a pu être évaluée au préalable en hors ligne sur des jeux de données du monde réel. Nous avons observé que les algorithmes de *CMAB* combinés à *ICE* obtiennent une meilleure précision globale que s'ils ne l'utilisent pas. Cette méthode devra néanmoins être évaluée en ligne notamment dans l'application mobile *scéno*.

Perspectives

Au regard des différentes contributions de cette thèse, plusieurs perspectives s'offrent à nous.

Concernant les algorithmes *SW-LinUCB* et *Gorthaur*, il serait intéressant de pouvoir les évaluer en ligne dans l'application mobile *scéno*. En effet, les évaluations hors ligne sont limitées du fait que nous tournons en boucle sur les jeux de données et que les utilisateurs ne se lassent jamais d'une même recommandation, ce qui n'est pas le cas dans le monde réel. D'autre part, il serait judicieux d'établir une preuve formelle sur la borne asymptotique des regrets de ces heuristiques. Enfin, concernant *Gorthaur*, nous envisageons de pouvoir considérer la mesure de précision individuelle comme l'un des critères à maximiser dans le cadre de la sélection des algorithmes du porte-feuille. À la suite de ces travaux d'expérimentations préliminaires, il sera intéressant d'étudier et d'analyser ces méthodes afin de garantir des preuves de convergence.

Concernant notre méthode de raisonnement contextuel via l'usage de partitionnement spectral, il serait intéressant pour l'application mobile *scéno* de pouvoir s'émanciper de la source externe de données *Wifilib* (c.-à-d., journaux de connexions au Wi-Fi urbain). Nous projetons d'étendre notre méthode de partitionnement en utilisant comme source de données les historiques de géolocalisation de l'application mobile *scéno* directement. Ceci aurait pour avantage de gagner en réactivité et en dynamisme concernant les clusters inférés.

D'autre part, nous envisageons de faire évoluer notre méthode d'enrichissement de contexte *ICE* afin que celle-ci puisse évoluer dynamiquement. En effet, la méthode actuelle permet de recréer du contexte par rapport à la précision individuelle mesurée selon une période donnée mais n'est jamais plus reconsidérée tout au long des itérations et ce jusqu'à l'horizon. Dans le cadre d'applications en ligne comme c'est le cas de notre application mobile *scéno*, il semble incontournable de pouvoir reconsidérer dynamiquement le contexte calculé (la précision individuelle) par *ICE*.

Enfin, à l'instar de l'utilisation des bandits-manchots dans les systèmes de recommandation il serait intéressant de se pencher sur d'autres méthodes du paradigme de l'apprentissage par renforcement. Nous envisageons notamment d'explorer des techniques de *Q-Learning*. Cela nous permettrait ainsi d'explorer une matrice d'états plutôt qu'une distribution de probabilités qui s'avérerait peut être plus pertinente dans le cas contextuel.

BIBLIOGRAPHIE

[Abo+97] Gregory D. ABOWD et al., « Cyberguide : A mobile context-aware tour guide », in : *Wireless networks* 3.*5* (1997), p. 421–433.

[Adi15] Ladji ADIAVIAKOYE, « Observatoire de trajectoire de piétons à l'aide d'un réseau de télémètre laser à balayage : application à l'intérieur des bâtiments », thèse de doct., Université d'Angers, 2015.

[AT05] Gediminas ADOMAVICIUS et Alexander TUZHILIN, « Towards the next generation of recommender systems : A survey of the state-of-the-art and possible extensions », in : *IEEE Transactions on Knowledge & Data Engineering 6* (2005), p. 734–749.

[AT11] Gediminas ADOMAVICIUS et Alexander TUZHILIN, « Context-aware recommender systems », in : *Recommender systems handbook*, Springer, 2011, p. 217–253.

[Ado+05] Gediminas ADOMAVICIUS et al., « Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach », in : *ACM Trans. Inf. Syst.* 23.*1* (jan. 2005), p. 103–145, ISSN : 1046-8188, DOI : 10.1145/1055709.1055714, URL : http://doi.acm.org/10.1145/1055709.1055714.

[AC17] Ajay AGARWAL et Minakshi CHAUHAN, « Similarity Measures used in Recommender Systems : A Study », in : (2017).

[Aga+14] Alekh AGARWAL et al., « Taming the monster : A fast and simple algorithm for contextual bandits », in : *International Conference on Machine Learning*, 2014, p. 1638–1646.

[Agi+06] Eugene AGICHTEIN et al., « Learning user interaction models for predicting web search result preferences », in : *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, 2006, p. 3–10.

[ABR09] Alessandra AGOSTINI, Claudio BETTINI, et Daniele RIBONI, « Hybrid reasoning in the CARE middleware for context awareness », in : *International journal of Web engineering and technology* 5.*1* (2009), p. 3–23.

[Agr95] Rajeev AGRAWAL, « Sample mean based index policies by O (log n) regret for the multi-armed bandit problem », in : *Advances in Applied Probability* 27.*4* (1995), p. 1054–1078.

[AG12] Shipra AGRAWAL et Navin GOYAL, « Analysis of Thompson sampling for the multi-armed bandit problem », in : *Conference on Learning Theory*, 2012, p. 39–1.

[AG13] Shipra AGRAWAL et Navin GOYAL, « Thompson sampling for contextual bandits with linear payoffs », in : *International Conference on Machine Learning*, 2013, p. 127–135.

[Ait11] Soraya AIT CHELLOUCHE, « Délivrance de services média suivant le contexte au sein d'environnements hétérogènes pour les réseaux médias du futur », thèse de doct., Bordeaux 1, 2011.

[All16] Robin ALLESIARDO, « Bandits Manchots sur Flux de Données Non Stationnaires », thèse de doct., Paris Saclay, 2016.

[AFB14] Robin ALLESIARDO, Raphaël FÉRAUD, et Djallel BOUNEFFOUF, « A neural networks committee for the contextual bandit problem », in : *International Conference on Neural Information Processing*, Springer, 2014, p. 374–381.

[AP15] Xavier AMATRIAIN et Josep M. PUJOL, « Data mining methods for recommender systems », in : *Recommender Systems Handbook*, Springer, 2015, p. 227–262.

[Ank+99] Mihael ANKERST et al., « OPTICS : ordering points to identify the clustering structure », in : *ACM Sigmod record*, t. 28, 2, ACM, 1999, p. 49–60.

[AEK00] Asim ANSARI, Skander ESSEGAIER, et Rajeev KOHLI, *Internet recommendation systems*, 2000.

[Ard+01] Liliana ARDISSONO et al., « Tailoring the recommendation of tourist information to heterogeneous user groups », in : *Workshop on adaptive hypermedia*, Springer, 2001, p. 280–295.

[Ash+15] Azin ASHKAN et al., « Optimal Greedy Diversity for Recommendation. », in : *In International Joint Conferences on Artificial Intelligence (IJCAI)* (2015), p. 1742–1748.

[Aue02] Peter AUER, « Using confidence bounds for exploitation-exploration trade-offs », in : *Journal of Machine Learning Research* 3.*Nov* (2002), p. 397–422.

[AC98] Peter AUER et Nicolo CESA-BIANCHI, « On-line learning with malicious noise and the closure algorithm », in : *Annals of mathematics and artificial intelligence* 23.*1-2* (1998), p. 83–99.

[ACF02] Peter AUER, Nicolo CESA-BIANCHI, et Paul FISCHER, « Finite-time analysis of the multiarmed bandit problem », in : *Machine learning* 47.*2-3* (2002), p. 235–256.

[Aue+95] Peter AUER et al., « Gambling in a rigged casino : The adversarial multi-armed bandit problem », in : *focs*, IEEE, 1995, p. 322.

[Aue+02] Peter AUER et al., « The nonstochastic multiarmed bandit problem », in : *SIAM journal on computing* 32.*1* (2002), p. 48–77.

[Bal+18] Avinash BALAKRISHNAN et al., « Using Contextual Bandits with Behavioral Constraints for Constrained Online Movie Recommendation. », in : *IJCAI*, 2018, p. 5802–5804.

[Bao+13] Jie BAO et al., « A survey on recommendations in location-based social networks », in : *ACM TIST* (2013).

[Bao+15] Jie BAO et al., « Recommendations in location-based social networks : a survey », in : *Geoinformatica* 19.*3* (2015), p. 525–565.

[BDH93] C. Bradford BARBER, David P. DOBKIN, et Hannu HUHDANPAA, *The quickhull algorithm for convex hull*, rapp. tech., Technical Report GCG53, The Geometry Center, MN, 1993.

[BDG11] Cristina BARBERO, Paola DAL ZOVO, et Barbara GOBBI, « A flexible context aware reasoning approach for iot applications », in : *Mobile Data Management (MDM), 2011 12th IEEE International Conference on*, t. 1, IEEE, 2011, p. 266–275.

[Bel+12] Paolo BELLAVISTA et al., « A survey of context data distribution for mobile ubiquitous systems », in : *ACM Computing Surveys (CSUR)* 44.*4* (2012), p. 24.

[Bel15] Richard E. BELLMAN, *Adaptive control processes : a guided tour*, t. 2045, Princeton university press, 2015.

[Bel+13] Alejandro BELLOGIN et al., « An empirical comparison of social, collaborative filtering, and hybrid recommenders », in : *ACM Transactions on Intelligent Systems and Technology (TIST)* 4.*1* (2013), p. 14.

[Ben15] Idir BENOUARET, « Un système de recommandation sensible au contexte pour la visite de musée. », in : *CORIA*, 2015, p. 515–524.

[Ben17] Idir BENOUARET, « Un système de recommandation contextuel et composite pour la visite personnalisée de sites culturels », thèse de doct., Université de Technologie de Compiègne, 2017.

[Bet+10] Claudio BETTINI et al., « A survey of context modelling and reasoning techniques », in : *Pervasive and Mobile Computing* 6.*2* (2010), p. 161–180.

[Bey+11] Alina BEYGELZIMER et al., « Contextual bandit algorithms with supervised learning guarantees », in : *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, p. 19–26.

[Bik+07] Antonis BIKAKIS et al., « A survey of semantics-based approaches for context reasoning in ambient intelligence », in : *European Conference on Ambient Intelligence*, Springer, 2007, p. 14–23.

[BP98] Daniel BILLSUS et Michael J. PAZZANI, « Learning Collaborative Information Filters. », in : *Icml*, t. 98, 1998, p. 46–54.

[Bob+11] Jesus BOBADILLA et al., « Improving collaborative filtering recommender system results and performance using genetic algorithms », in : *Knowledge-based systems* 24.*8* (2011), p. 1310–1316.

[BHB18] Boudjemaa BOUDAA, Slimane HAMMOUDI, et Sidi Mohammed BENSLIMANE, « Towards an Extensible Context Model for Mobile User in Smart Cities », in : *Computational Intelligence and Its Applications* (2018).

[BTB09] Ourdia BOUIDGHAGHEN, Lynda TAMINE-LECHANI, et Mohand BOUGHANEM, « Vers la définition du contexte d'un utilisateur mobile de système de recherche d'information », in : *Proceedings of the 5th French-Speaking Conference on Mobility and Ubiquity Computing*, ACM, 2009, p. 25–31.

[Bou14] Djallel BOUNEFFOUF, « Recommandation mobile, sensible au contexte de contenus évolutifs : Contextuel-E-Greedy », in : *arXiv preprint arXiv :1402.1986* (2014).

[BF16] Djallel BOUNEFFOUF et Raphael FERAUD, « Multi-armed bandit problem with known trend », in : *Neurocomputing* 205 (2016), p. 16–21.

[Bou+17] Djallel BOUNEFFOUF et al., « Context Attentive Bandits : Contextual Bandit with Restricted Context », in : *International Joint Conferences on Artificial Intelligence (IJCAI)* (2017).

[BCR09] Oliver BRDICZKA, James L. CROWLEY, et Patrick REIGNIER, « Learning situation models in a smart home », in : *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39.*1* (2009), p. 56–63.

[BHK98] John S. BREESE, David HECKERMAN, et Carl KADIE, « Empirical analysis of predictive algorithms for collaborative filtering », in : *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, Morgan Kaufmann Publishers Inc., 1998, p. 43–52.

[Bré99] Patrick BRÉZILLON, « Context in Artificial Intelligence : A survey of the literature », in : *Computers and artificial intelligence* 18 (1999), p. 321–340.

[Bré02] Patrick BRÉZILLON, « Hors du contexte, point de salut », in : *Séminaire" Objets Communicants* 48 (2002), p. 74–90.

[Bri+05] Derek BRIDGE et al., « Case-based recommender systems », in : *The Knowledge Engineering Review* 20.*3* (2005), p. 315–320.

[Bro+07] Gregor BROLL et al., « Modeling context information for realizing simple mobile services », in : *Proceedings of the 16th IST Mobile & Wireless Communications Summit*, Budapest, Hungary, juil. 2007.

[BBM81] Daniel T. BROOKS, Brandon BECKER, et Jerry R. MARLATT, « Computer applications in particular industries : securities », in : *Computers and the law, 3rd edn. American Bar Association, Section of Science and Technology* (1981).

[Bru96] Peter BRUSILOVSKY, « Methods and techniques of adaptive hypermedia », in : *User modeling and user-adapted interaction* 6.*2-3* (1996), p. 87–129.

[Bur02] Robin BURKE, « Hybrid recommender systems : Survey and experiments », in : *User modeling and user-adapted interaction* 12.*4* (2002), p. 331–370.

[Bur07] Robin BURKE, « Hybrid web recommender systems », in : *The adaptive web*, Springer, 2007, p. 377–408.

[BLL15] Giuseppe BURTINI, Jason LOEPPKY, et Ramon LAWRENCE, « A survey of online experiment design with the stochastic multi-armed bandit », in : *arXiv preprint arXiv :1510.00757* (2015).

[Bus+17] Róbert BUSA-FEKETE et al., « Multi-objective bandits : optimizing the Generalized Gini Index », in : *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, JMLR. org, 2017, p. 625–634.

[Cam+08] Andrew T. CAMPBELL et al., « The rise of people-centric sensing », in : *IEEE Internet Computing* 12.*4* (2008).

[CBC08] Iván CANTADOR, Alejandro BELLOGIN, et Pablo CASTELLS, « A multilayer ontology-based hybrid recommendation model », in : *Ai Communications* 21.*2-3* (2008), p. 203–210.

[CBB13] Sylvain CASTAGNOS, Armelle BRUN, et Anne BOYER, « When Diversity Is Needed... But Not Expected ! », in : *International Conference on Advances in Information Mining and Management*, IARIA XPS Press, 2013, p. 44–50.

[CBB14] Sylvain CASTAGNOS, Armelle BRUN, et Anne BOYER, « La diversité : entre besoin et méfiance dans les systèmes de recommandation », in : *Revue I3-Information Interaction Intelligence* (2014).

[Cen+06] Federica CENA et al., « Integrating heterogeneous adaptation techniques to build a flexible and usable mobile tourist guide », in : *AI Communications* 19.*4* (2006), p. 369–384.

[CF98] Nicolo CESA-BIANCHI et Paul FISCHER, « Finite-Time Regret Bounds for the Multiarmed Bandit Problem. », in : *ICML*, Citeseer, 1998, p. 100–108.

[CL11] Olivier CHAPELLE et Lihong LI, « An empirical evaluation of thompson sampling », in : *Advances in neural information processing systems*, 2011, p. 2249–2257.

[CPF17] Akshay Kumar CHATURVEDI, Filipa PELEJA, et Ana FREIRE, « Recommender System for News Articles using Supervised Learning », in : *arXiv preprint arXiv :1707.00506* (2017).

[Che+04] Harry CHEN et al., « Intelligent agents meet the semantic web in smart spaces », in : *IEEE Internet computing* 8.*6* (2004), p. 69–79.

[Che+52] Herman CHERNOFF et al., « A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations », in : *The Annals of Mathematical Statistics* 23.*4* (1952), p. 493–507.

[CMD99] Keith CHEVERST, Keith MITCHELL, et Nigel DAVIES, « Design of an object model for a context sensitive tourist GUIDE », in : *Computers & Graphics* 23.*6* (1999), p. 883–891.

[CHM97] David Maxwell CHICKERING, David HECKERMAN, et Christopher MEEK, « A Bayesian approach to learning Bayesian networks with local structure », in : *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, Morgan Kaufmann Publishers Inc., 1997, p. 80–89.

[Chi13] Bachir CHIHANI, « Enterprise context-awareness : empowering service users and developers », thèse de doct., Institut National des Télécommunications, 2013.

[CVS07] Christina CHRISTAKOU, Spyros VRETTOS, et Andreas STAFYLOPATIS, « A hybrid movie recommender system based on neural networks », in : *International Journal on Artificial Intelligence Tools* 16.*05* (2007), p. 771–792.

[CD04] Stahl CHRISTOPH et Heckmann DOMINIK, « Using semantic web technology for ubiquitous location and situation modeling », in : *Geographic Information Sciences* 10.*2* (2004), p. 157–165.

[Coc63] WG COCHRAN, « Sampling Techniques, New York, 1953 », in : *Statistical Surveys E. Grebenik and CA Moser* (1963).

[Cor+07] Chris CORNELIS et al., « One-and-only item recommendation with fuzzy logic techniques », in : *Information Sciences* 177.*22* (2007), p. 4906–4921.

[CMB18] Antoine CORNUÉJOLS, Laurent MICLET, et Vincent BARRA, *Apprentissage artificiel : Deep learning, concepts et algorithmes*, Eyrolles, 2018.

[CC05] Joëlle COUTAZ et James L. CROWLEY, « CONTEXTis KEY », in : *Communications of the ACM* 48.*3* (2005), p. 49.

[CAS16] Paul COVINGTON, Jay ADAMS, et Emre SARGIN, « Deep neural networks for youtube recommendations », in : *Proceedings of the 10th ACM conference on recommender systems*, ACM, 2016, p. 191–198.

[Cra+12] Justin CRANSHAW et al., « The livehoods project : Utilizing social media to understand the dynamics of a city », in : *International AAAI Conference on Weblogs and Social Media*, 2012, p. 58.

[CHM15] Susan CRAW, Ben HORSBURGH, et Stewart MASSIE, « Music recommenders : user evaluation without real users ? », in : *In International Joint Conferences on Artificial Intelligence (IJCAI)* (2015).

[DLT04] Joaquin DELGADO, Renaud LAPLANCHE, et Mathias TURCK, *System and method for obtaining user preferences and providing user recommendations for unseen physical and information goods and services*, US Patent 6,801,909, oct. 2004.

[DK04] Mukund DESHPANDE et George KARYPIS, « Item-based top-n recommendation algorithms », in : *ACM Transactions on Information Systems (TOIS)* 22.*1* (2004), p. 143–177.

[Dey01] Anind K. DEY, « Understanding and Using Context », in : *Personal Ubiquitous Computing* 5.*1* (jan. 2001), p. 4–7, ISSN : 1617-4909, DOI : 10.1007/s007790170019.

[Dou04] Paul DOURISH, *Where the action is : the foundations of embodied interaction*, MIT press, 2004.

[DN13] Madalina M. DRUGAN et Ann NOWE, « Designing multi-objective multi-armed bandits algorithms : A study », in : *IJCNN*, IEEE, 2013, p. 1–8.

[Dud+11] Miroslav DUDIK et al., « Efficient optimal learning for contextual bandits », in : *arXiv preprint arXiv :1106.2369* (2011).

[EPC08] Miikka ERMES, Juha PARKKA, et Luc CLUITMANS, « Advancing from offline to online activity recognition with wearable sensors », in : *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, IEEE, 2008, p. 4451–4454.

[EMM02] Eyal EVEN-DAR, Shie MANNOR, et Yishay MANSOUR, « PAC bounds for multi-armed bandit and Markov decision processes », in : *International Conference on Computational Learning Theory*, Springer, 2002, p. 255–270.

[Fab+18] Anne FABER et al., « Modeling and Visualizing Smart City Mobility Business Ecosystems : Insights from a Case Study », in : *Information* 9.*11* (2018), p. 270.

[FB08] Alexander FELFERNIG et Robin BURKE, « Constraint-based recommender systems : technologies and research issues », in : *Proceedings of the 10th international conference on Electronic commerce*, ACM, 2008, p. 3.

[Fel+06] Alexander FELFERNIG et al., « An integrated environment for the development of knowledge-based recommender applications », in : *International Journal of Electronic Commerce* 11.*2* (2006), p. 11–34.

[Fra01] Andrew U. FRANK, « Tiers of ontology and consistency constraints in geographical information systems », in : *International Journal of Geographical Information Science* 15.*7* (2001), p. 667–678.

[Fre75] David A. FREEDMAN, « On tail probabilities for martingales », in : *the Annals of Probability* (1975), p. 100–118.

[Fro+06] Jon FROEHLICH et al., « Voting with your feet : An investigative study of the relationship between place visit behavior and preference », in : *International Conference on Ubiquitous Computing*, Springer, 2006, p. 333–350.

[Gal15] Nicolas GALICHET, « Contributions to Multi-Armed Bandits : Risk-Awareness and Sub-Sampling for Linear Contextual Bandits », thèse de doct., Université Paris-Sud, 2015.

[GM11] Aurélien GARIVIER et Eric MOULINES, « On upper-confidence bound policies for switching bandit problems », in : *International Conference on Algorithmic Learning Theory*, Springer, 2011, p. 174–188.

[Gav+14] Damianos GAVALAS et al., « Mobile recommender systems in tourism », in : *Journal of Network and Computer Applications* 39 (2014), p. 319–333.

[GDJ10] Mouzhi GE, Carla DELGADO-BATTENFELD, et Dietmar JANNACH, « Beyond accuracy : evaluating recommender systems by coverage and serendipity », in : *Proceedings of the fourth ACM conference on Recommender systems*, ACM, 2010, p. 257–260.

[GP14] Mustansar Ali GHAZANFAR et Adam PRÜGEL-BENNETT, « Leveraging clustering approaches to solve the gray-sheep users problem in recommender systems », in : *Expert Systems with Applications* 41.*7* (2014), p. 3261–3275.

[GLS16] Adrien GOËFFON, Frédéric LARDEUX, et Frédéric SAUBION, « Simulating non-stationary operators in search algorithms », in : *Applied Soft Computing* 38 (2016), p. 257–268.

[Gol+92] David GOLDBERG et al., « Using collaborative filtering to weave an information tapestry », in : *Communications of the ACM* 35.*12* (1992), p. 61–70.

[Gol+01] Ken GOLDBERG et al., « Eigentaste : A constant time collaborative filtering algorithm », in : *information retrieval* 4.*2* (2001), p. 133–151.

[GS11] Eric GORDON et Adriana de Souza e SILVA, *Net locality : Why location matters in a networked world*, John Wiley & Sons, 2011.

[GS01a] Philip GRAY et Daniel SALBER, « Modelling and using sensed context information in the design of interactive applications », in : *Engineering for Human-Computer Interaction*, Springer, 2001, p. 317–335.

[Gre+17] Kristjan GREENEWALD et al., « Action Centered Contextual Bandits », in : *Advances in neural information processing systems*, 2017, p. 5979–5987.

[GS01b] Tom GROSS et Marcus SPECHT, « Awareness in context-aware information systems », in : *Mensch & Computer 2001*, Springer, 2001, p. 173–182.

[Gru93] Thomas R. GRUBER, « Towards principles for the design of ontologies used for knowledge sharing in formal ontology in conceptual analysis and knowledge representation », in : *Int. J. of Human-Computer Studies. Kluwer Academic Publishers* 43 (1993), p. 907–928.

[Gua+07] Donghai GUAN et al., « Devising a context selection-based reasoning engine for context-aware ubiquitous computing middleware », in : *International Conference on Ubiquitous Intelligence and Computing*, Springer, 2007, p. 849–857.

[GRS98] Sudipto GUHA, Rajeev RASTOGI, et Kyuseok SHIM, « CURE : an efficient clustering algorithm for large databases », in : *ACM Sigmod Record*, t. 27, 2, ACM, 1998, p. 73–84.

[Guo+15] Bin GUO et al., « Mobile crowd sensing and computing : The review of an emerging human-powered sensing paradigm », in : *ACM Computing Surveys (CSUR)* 48.*1* (2015), p. 7.

[GL12] Lin GUO et Xiongfei LI, « Using Apriori to mine IoT frequent structures on compute cloud », in : *JOURNAL OF INFORMATION &COMPUTATIONAL SCIENCE* 9.*3* (2012), p. 657–665.

[GCC17] Nicolas GUTOWSKI, Olivier CAMP, et Eric CHAUVEAU, « Measuring the Energy Consumption of Massive Data Insertions : an energy consumption assessment of the PL/SQL FOR LOOP and FORALL methods », in : *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, IEEE, 2017, p. 450–457.

[Gut+17] Nicolas GUTOWSKI et al., « A Framework for Context-Aware Service Recommendation for Mobile Users : A Focus on Mobility in Smart Cities », in : *From Data To Decision* (2017).

[Gut+18a] Nicolas GUTOWSKI et al., « Bandits-Manchots Contextuels : Précision Globale Versus Individuelle », in : *4ème conférence sur les Applications Pratiques de l'Intelligence Artificielle APIA 2018*, 2018.

[Gut+18b] Nicolas GUTOWSKI et al., « Context Enhancement for Linear Contextual Multi-Armed Bandits », in : *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, IEEE, 2018, p. 1048–1055.

[Gut+18c] Nicolas GUTOWSKI et al., « Déduction de quartiers à partir des connexions au Wifi Urbain », in : *SAGEO'2018 EXCES*, 2018.

[Gut+18d] Nicolas GUTOWSKI et al., « Mobility and prediction : an asset for crisis management », in : *How Information Systems Can Help in Alarm/Alert Detection*, Elsevier, 2018, p. 33–53.

[Gut+18e] Nicolas GUTOWSKI et al., « Recommandation et Visualisation dans les Villes Intelligentes : Projet EVENT-AI », in : *Atelier Data Intelligence INFORSID 2018*, 2018.

[Gut+19a] Nicolas GUTOWSKI et al., « Global Versus Individual Accuracy in Contextual Multi-Armed Bandit », in : *The 34th ACM/SIGAPP Symposium on Applied Computing*, ACM, 2019.

[Gut+19b] Nicolas GUTOWSKI et al., « Gorthaur : A Portfolio Approach for Dynamic Selection of Multi-Armed Bandit Algorithms for Recommendation », in : *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, IEEE, 2019.

[Gut+19c] Nicolas GUTOWSKI et al., « Improving Bandit-Based Recommendations with Spatial Context Reasoning : An Online Evaluation », in : *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, IEEE, 2019.

[Gut+19d] Nicolas GUTOWSKI et al., « Using Individual Accuracy to Create Context for Non-Contextual Multi-Armed Bandit Problems », in : *RIVF*, IEEE, 2019.

[Gyr15] Amelie GYRARD, « Designing cross-domain semantic Web of things applications », thèse de doct., Télécom ParisTech, 2015.

[Had14] Nicolas HADERER, « APISENSE® : une plate-forme répartie pour la conception, le déploiement et l'exécution de campagnes de collecte de données sur des terminaux intelligents », thèse de doct., Université des Sciences et Technologie de Lille-Lille I, 2014.

[HMP12] Allel HADJALI, Amine MOKHTARI, et Olivier PIVERT, « A fuzzy-rule-based model for handling contextual preference queries », in : *International Journal of Computational Intelligence Systems* 5.*4* (2012), p. 775–788.

[Hag+04] Hani HAGRAS et al., « Creating an ambient-intelligence environment using embedded agents », in : *IEEE Intelligent Systems* 19.*6* (2004), p. 12–20.

[HB05] PL. HAMMER et T. BONATES, « Logical Analysis of Data : From Combinatorial Optimization to Medical Applications. RUTCOR Research Report 10-2005 », in : (2005).

[HMC15] Slimane HAMMOUDI, Valérie MONFORT, et Olivier CAMP, « Model driven development of user-centred context aware services », in : *IJSSC* 5.*2* (2015), p. 100–114, DOI : 10.1504/IJSSC.2015.069227, URL : http://dx.doi.org/10.1504/IJSSC.2015.069227.

[Han+14] Yuxing HAN et al., « Efficiently Retrieving Top-k Trajectories by Locations via Traveling Time. », in : *ADC*, Springer, 2014, p. 122–134.

[HIR03] Karen HENRICKSEN, Jadwiga INDULSKA, et Andry RAKOTONIRAINY, « Generating context management infrastructure from high-level context models », in : *In 4th International Conference on Mobile Data Management (MDM)-Industrial Track*, Citeseer, 2003.

[HLI04] Karen HENRICKSEN, Steven LIVINGSTONE, et Jadwiga INDULSKA, « Towards a hybrid approach to context modelling, reasoning and interoperation », in : *Proceedings of the first international workshop on advanced context modelling, reasoning and management, in conjunction with ubicomp*, t. 2004, 2004.

[HK01] Jonathan L. HERLOCKER et Joseph A. KONSTAN, « Content-independent task-focused recommendation », in : *IEEE Internet Computing* 5.*6* (2001), p. 40–47.

[Her+04] Jonathan L. HERLOCKER et al., « Evaluating collaborative filtering recommender systems », in : *ACM Transactions on Information Systems (TOIS)* 22.*1* (2004), p. 5–53.

[Hil+95] Will HILL et al., « Recommending and evaluating choices in a virtual community of use », in : *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press/Addison-Wesley Publishing Co., 1995, p. 194–201.

[HGN00] Jochen HIPP, Ulrich GÜNTZER, et Gholamreza NAKHAEIZADEH, « Algorithms for association rule mining—a general survey and comparison », in : *ACM sigkdd explorations newsletter* 2.*1* (2000), p. 58–64.

[Hoe63] Wassily HOEFFDING, « Probability inequalities for sums of bounded random variables », in : *Journal of the American statistical association* 58.*301* (1963), p. 13–30.

[HBC10] M. HOFFMAN, D. BLEI, et P. COOK, « Bayesian nonparametric matrix factorization for recorded music », in : *Proceedings of the 27th Annual International Conference on Machine Learning*, 2010, p. 439–446.

[Hsu+07] Shang H. HSU et al., « AIMED-A personalized TV recommendation system », in : *European Conference on Interactive Television*, Springer, 2007, p. 166–174.

[Hu17] Chufeng HU, « Application of Neural Network Based Recommendation System », thèse de doct., UCLA, 2017.

[Hu+17] Liang HU et al., « Diversifying Personalized Recommendation with User-session Context », in : *In International Joint Conferences on Artificial Intelligence (IJCAI)* (2017).

[IS03] Jadwiga INDULSKA et Peter SUTTON, « Location management in pervasive systems », in : *Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003-Volume 21*, Australian Computer Society, Inc., 2003, p. 143–151.

[Jam04] Anthony JAMESON, « More than the sum of its members : challenges for group recommender systems », in : *Proceedings of the working conference on Advanced visual interfaces*, ACM, 2004, p. 48–54.

[JF13] Dietmar JANNACH et Gerhard FRIEDRICH, « Tutorial : recommender systems », in : *International Joint Conference on Artificial Intelligence Beijing*, 2013.

[Jat+08] Luciana C. JATOBA et al., « Context-aware mobile health monitoring : Evaluation of different pattern recognition methods for classification of physical activity », in : *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, IEEE, 2008, p. 5250–5253.

[Jay65] Edwin T. JAYNES, « Gibbs vs Boltzmann entropies », in : *American Journal of Physics* 33.*5* (1965), p. 391–398.

[KST08] Sham M. KAKADE, Shai SHALEV-SHWARTZ, et Ambuj TEWARI, « Efficient bandit algorithms for online multiclass prediction », in : *Proceedings of the 25th international conference on Machine learning*, ACM, 2008, p. 440–447.

[Kan+02] Tapas KANUNGO et al., « An efficient k-means clustering algorithm : Analysis and implementation », in : *IEEE Transactions on Pattern Analysis & Machine Intelligence 7* (2002), p. 881–892.

[KHK99] George KARYPIS, Eui-Hong HAN, et Vipin KUMAR, « Chameleon : Hierarchical clustering using dynamic modeling », in : *Computer* 32.*8* (1999), p. 68–75.

[KK57] J. KATSNELSON et S. KOTZ, « On the upper limits of some measures of variability », in : *Theoretical and Applied Climatology* 8.*1* (1957), p. 103–107.

[KRW09] Carsten KESSLER, Martin RAUBAL, et Christoph WOSNIOK, « Semantic rules for context-aware geographical information retrieval », in : *European Conference on Smart Sensing and Context*, Springer, 2009, p. 77–92.

[Kha+14] Muhammad Aamir KHAN et al., « A novel learning method to classify data streams in the internet of things », in : *Software Engineering Conference (NSEC), 2014 National*, IEEE, 2014, p. 61–66.

[Kim+09] Donnie H. KIM et al., « Discovering semantically meaningful places from pervasive RF-beacons », in : *Proceedings of the 11th international conference on Ubiquitous computing*, ACM, 2009, p. 21–30.

[KA08] Kyoung-jae KIM et Hyunchul AHN, « A recommender system using GA K-means clustering in an online shopping market », in : *Expert systems with applications* 34.*2* (2008), p. 1200–1209.

[Kle02] Roland KLEMKE, « Modelling context in information brokering processes », thèse de doct., Bibliothek der RWTH Aachen, 2002.

[KS06] Levente KOCSIS et Csaba SZEPESVÁRI, « Discounted ucb », in : *2nd PASCAL Challenges Workshop*, 2006, p. 784–791.

[Kon06] Amit KONAR, *Computational intelligence : principles, techniques and applications*, Springer Science & Business Media, 2006.

[KK10] Barbara T. KOREL et Simon GM. KOO, « A survey on context-aware sensing for body sensor networks », in : *Wireless Sensor Network 2.08* (2010), p. 571.

[Kor08] Yehuda KOREN, « Factorization meets the neighborhood : a multifaceted collaborative filtering model », in : *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2008, p. 426–434.

[KBV09] Yehuda KOREN, Robert BELL, et Chris VOLINSKY, « Matrix factorization techniques for recommender systems », in : *Computer 8* (2009), p. 30–37.

[Kru09] John KRUMM, « A survey of computational location privacy », in : *Personal and Ubiquitous Computing* 13.*6* (2009), p. 391–399, DOI : 10.1007/s00779-008-0212-5, URL : http://dx.doi.org/10.1007/s00779-008-0212-5.

[KP17] Matevž KUNAVER et Tomaž POŽRL, « Diversity in recommender systems–A survey », in : *Knowledge-Based Systems* 123 (2017), p. 154–162.

[Lac15] Anisio LACERDA, « Contextual Bandits for Multi-objective Recommender Systems », in : *Intelligent Systems (BRACIS), 2015 Brazilian Conference on*, IEEE, 2015, p. 68–73.

[Lac17] Anisio LACERDA, « Multi-Objective Ranked Bandits for Recommender Systems », in : *Neurocomputing* 246 (2017), p. 12–24.

[LR85] Tze Leung LAI et Herbert ROBBINS, « Asymptotically efficient adaptive allocation rules », in : *Advances in applied mathematics* 6.*1* (1985), p. 4–22.

[LZ08] John LANGFORD et Tong ZHANG, « The epoch-greedy algorithm for multi-armed bandits with side information », in : *Advances in neural information processing systems*, 2008, p. 817–824.

[Led72] John Ronald Reuel Tolkien (trad. Francis LEDOUX), *Le Seigneur des Anneaux*, 1972, p. 67.

[Lev+12] Justin J. LEVANDOSKI et al., « Lars : A location-aware recommender system », in : *2012 IEEE 28th International Conference on Data Engineering*, IEEE, 2012, p. 450–461.

[Li+10] Lihong LI et al., « A contextual-bandit approach to personalized news article recommendation », in : *Proceedings of the 19th international conference on World wide web*, ACM, 2010, p. 661–670.

[Li+11] Lihong LI et al., « Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms », in : *Proceedings of the fourth ACM international conference on Web search and data mining*, ACM, 2011, p. 297–306.

[LK03] Qing LI et Byeong Man KIM, « Clustering approach for hybrid recommender system », in : *Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003)*, IEEE, 2003, p. 33–38.

[LKG16] Shuai LI, Alexandros KARATZOGLOU, et Claudio GENTILE, « Collaborative Filtering Bandits », in : *The 39th International ACM SIGIR Conference on Information Retrieval (SIGIR)*, 2016.

[Li+15] Xin LI et al., « Context aware middleware architectures : survey and challenges », in : *Sensors* 15.*8* (2015), p. 20570–20607.

[LKH14] Blerina LIKA, Kostas KOLOMVATSOS, et Stathes HADJIEFTHYMIADES, « Facing the cold start problem in recommender systems », in : *Expert Systems with Applications* 41.*4* (2014), p. 2065–2073.

[LL05] Tsung-Nan LIN et Po-Chiang LIN, « Performance comparison of indoor positioning techniques based on location fingerprinting in wireless networks », in : *Wireless Networks, Communications and Mobile Computing, 2005 International Conference on*, t. 2, IEEE, 2005, p. 1569–1574.

[Lou+15] Jonathan LOUËDEC et al., « Systemes de recommandations : algorithmes de bandits et évaluation expérimentale », in : *47emes Journees de Statistique de la SFdS (JDS 2015)*, 2015, pp–1.

[Lu+15] Jie LU et al., « Recommender system application developments : a survey », in : *Decision Support Systems* 74 (2015), p. 12–32.

[Luc05] R. Duncan LUCE, *Individual choice behavior : A theoretical analysis*, Courier Corporation, 2005.

[Luc12] R. Duncan LUCE, *Individual choice behavior : A theoretical analysis*, Courier Corporation, 2012.

[Luc] RD. LUCE, *Individual choice behavior. 1959*.

[Lun+17] José Maria LUNA et al., « Mining Context-Aware Association Rules Using Grammar-Based Genetic Programming », in : *IEEE transactions on cybernetics 99* (2017), p. 1–15.

[Maj+13] Abdul MAJID et al., « A context-aware personalized travel recommendation system based on geotagged social media data mining », in : *International Journal of Geographical Information Science* 27.*4* (2013), p. 662–684.

[MT04] Shie MANNOR et John N. TSITSIKLIS, « The sample complexity of exploration in the multi-armed bandit problem », in : *Journal of Machine Learning Research* 5.*Jun* (2004), p. 623–648.

[MS08] Jorge MATURANA et Frédéric SAUBION, « A compass to guide genetic algorithms », in : *International Conference on Parallel Problem Solving from Nature*, Springer, 2008, p. 256–265.

[Mat+09] Jorge MATURANA et al., « Extreme compass and dynamic multi-armed bandits for adaptive operator selection », in : *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, IEEE, 2009, p. 365–372.

[MRF03] Rene MAYRHOFER, Harald RADI, et Alois FERSCHA, *Recognizing and predicting context by learning from user behavior*, na, 2003.

[MB97] John MCCARTHY et Sasa BUVAC, « Formalizing context (expanded notes) », in : (1997).

[MA98] Joseph F. MCCARTHY et Theodore D. ANAGNOST, « MusicFX : an arbiter of group preferences for computer supported collaborative workouts », in : *Proceedings of the 1998 ACM conference on Computer supported cooperative work*, ACM, 1998, p. 363–372.

[McC+06] Kevin MCCARTHY et al., « Cats : A synchronous approach to collaborative group recommendation », in : *Florida Artificial Intelligence Research Society Conference (FLAIRS)*, 2006, p. 86–91.

[MRK06] Sean M. MCNEE, John RIEDL, et Joseph A. KONSTAN, « Being accurate is not enough : how accuracy metrics have hurt recommender systems », in : *CHI'06 extended abstracts on Human factors in computing systems*, ACM, 2006, p. 1097–1101.

[Mer99] Arnd Kohrs-Bernard MERIALDO, « Clustering for collaborative filtering applications », in : *Intelligent Image Processing, Data Analysis & Information Retrieval* 3 (1999), p. 199.

[Mey+11] Frank MEYER et al., « Apport des données thématiques dans les systèmes de recommandation : hybridation et démarrage à froid. », in : *EGC*, 2011, p. 215–220.

[MDS09] Stuart E. MIDDLETON, David DE ROURE, et Nigel R. SHADBOLT, « Ontology-based recommender systems », in : *Handbook on ontologies*, Springer, 2009, p. 779–796.

[Mit97] Tom M. MITCHELL, « Machine learning, ser », in : *Computer Science Series. Singapure : McGraw-Hill Companies, Inc* (1997).

[ME12] ABM MUSA et Jakob ERIKSSON, « Tracking unmodified smartphones using wi-fi monitors », in : *Proceedings of the 10th ACM conference on embedded network sensor systems*, ACM, 2012, p. 281–294.

[Neg15] Elsa NEGRE, *Information and Recommender Systems*, John Wiley & Sons, 2015.

[NJW02] Andrew Y. NG, Michael I. JORDAN, et Yair WEISS, « On spectral clustering : Analysis and an algorithm », in : *Advances in neural information processing systems*, 2002, p. 849–856.

[Nou+11] Anastasios NOULAS et al., « Exploiting Semantic Annotations for Clustering Geographic Areas and Users in Location-based Social Networks. », in : *The social mobile web* 11.*2* (2011).

[Nou+12] Anastasios NOULAS et al., « A tale of many cities : universal patterns in human urban mobility », in : *PloS one* 7.*5* (2012), e37027.

[NF04] Petteri NURMI et Patrik FLORÉEN, « Reasoning in context-aware systems », in : *Helsinki Institute for Information Technology, Position paper* (2004).

[Oco+01] Mark O'CONNOR et al., « PolyLens : a recommender system for groups of users », in : *ECSCW 2001*, Springer, 2001, p. 199–218.

[Oku+06] Kenta OKU et al., « Context-aware SVM for context-dependent information recommendation », in : *Proceedings of the 7th international Conference on Mobile Data Management*, IEEE Computer Society, 2006, p. 109.

[OHS05] Michael P. O'MAHONY, Neil J. HURLEY, et Guénolé CM. SILVESTRE, « Recommender systems : Attack types and strategies », in : *AAAI*, 2005, p. 334–339.

[Ono+09] Chihiro ONO et al., « Context-aware preference model based on a study of difference between real and supposed situation data », in : *User Modeling, Adaptation, and Personalization* (2009), p. 102–113.

[ON08] Houda OUFAIDA et Omar NOUALI, « Le filtrage collaboratif et le web 2.0 », in : *Document numérique* 11.*1* (2008), p. 13–35.

[PLZ08] Amir PADOVITZ, Seng W. LOKE, et Arkady ZASLAVSKY, « The ECORA framework : A hybrid architecture for context-oriented pervasive computing », in : *Pervasive and mobile computing* 4.*2* (2008), p. 182–215.

[Pan+09] Umberto PANNIELLO et al., « Experimental comparison of pre-vs. post-filtering approaches in context-aware recommender systems », in : *Proceedings of the third ACM conference on Recommender systems*, ACM, 2009, p. 265–268.

[POC11] Han-Saem PARK, Keunhyun OH, et Sung-Bae CHO, « Bayesian network-based high-level context recognition for mobile context sharing in cyber-physical system », in : *International Journal of Distributed Sensor Networks* 7.*1* (2011), p. 650387.

[Pat07] Arkadiusz PATEREK, « Improving regularized singular value decomposition for collaborative filtering », in : *Proceedings of KDD cup and workshop*, t. 2007, 2007, p. 5–8.

[PB07] Michael J. PAZZANI et Daniel BILLSUS, « Content-based recommendation systems », in : *The adaptive web*, Springer, 2007, p. 325–341.

[Pen+00] David M. PENNOCK et al., « Collaborative filtering by personality diagnosis : A hybrid memory-and model-based approach », in : *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, Morgan Kaufmann Publishers Inc., 2000, p. 473–480.

[Per+13] Charith PERERA et al., « Semantic-driven configuration of internet of things middleware », in : *Semantics, Knowledge and Grids (SKG), 2013 Ninth International Conference on*, IEEE, 2013, p. 66–73.

[Per+14] Charith PERERA et al., « Context aware computing for the internet of things : A survey », in : *IEEE communications surveys & tutorials* 16.*1* (2014), p. 414–454.

[Pha01] Dzung L. PHAM, « Spatial models for fuzzy clustering », in : *Computer vision and image understanding* 84.*2* (2001), p. 285–297.

[Pie+08] Stefan PIETSCHMANN et al., « Croco : Ontology-based, cross-application context management », in : *2008 Third International Workshop on Semantic Media Adaptation and Personalization*, IEEE, 2008, p. 88–93.

[Pop13] George POPESCU, « Group recommender systems as a voting problem », in : *International Conference on Online Communities and Social Computing*, Springer, 2013, p. 412–421.

[PB14] Davy PREUVENEERS et Yolande BERBERS, « Samurai : A streaming multi-tenant context-management architecture for intelligent and scalable internet of things applications », in : *Proceedings of 10th International Conference on Intelligent Environments (IE'14)*, IEEE Computer Society, 2014, p. 226–233.

[Que+10] Daniele QUERCIA et al., « Recommending social events from mobile phone location data », in : *2010 IEEE International Conference on Data Mining*, IEEE, 2010, p. 971–976.

[Qui+09] Joaquin QUIONERO-CANDELA et al., *Dataset shift in machine learning*, The MIT Press, 2009.

[RC03] Anand RANGANATHAN et Roy H. CAMPBELL, « A middleware for context-aware agents in ubiquitous computing environments », in : *ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing*, Springer, 2003, p. 143–161.

[RB11] Sherif RASHAD et Joshua BRADLEY, « SmartMobiMine : Smart mobile data mining techniques to support 4G mobile networks », in : *Consumer Communications and Networking Conference (CCNC), 2011 IEEE*, IEEE, 2011, p. 703–704.

[Res+94] Paul RESNICK et al., « GroupLens : an open architecture for collaborative filtering of netnews », in : *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, ACM, 1994, p. 175–186.

[RRS15] Francesco RICCI, Lior ROKACH, et Bracha SHAPIRA, « Recommender systems : introduction and challenges », in : *Recommender systems handbook*, Springer, 2015, p. 1–34.

[Ric+06] Francesco RICCI et al., « Case-based travel recommendations », in : *Destination recommendation systems : behavioural foundations and applications* (2006), p. 67–93.

[Ric79] Elaine RICH, « User modeling via stereotypes », in : *Cognitive science* 3.*4* (1979), p. 329–354.

[Rob52] H. ROBBINS, « Some aspects of the sequential design of experiments », in : *Bulletin of the American Mathematical Society* (1952), p. 527–535.

[Roc71] Joseph John ROCCHIO, « Relevance feedback in information retrieval », in : *The SMART retrieval system : experiments in automatic document processing* (1971), p. 313–323.

[Ryu+17] Jegwang RYU et al., « Community-based diffusion scheme using Markov chain and spectral clustering for mobile social networks », in : *Wireless Networks* (2017).

[SME14] Sara SAEEDI, Adel MOUSSA, et Naser EL-SHEIMY, « Context-aware personal navigation using embedded sensor fusion in smartphones », in : *Sensors* 14.*4* (2014), p. 5742–5767.

[Sal10] Ali SALEHI, *Design and implementation of an efficient data stream processing system*, rapp. tech., EPFL, 2010.

[Sar01] Badrul Munir SARWAR, *Sparsity, scalability, and distribution in recommender systems*, University of Minnesota, 2001.

[Sar+00] Badrul SARWAR et al., *Application of dimensionality reduction in recommender system-a case study*, rapp. tech., Minnesota Univ Minneapolis Dept of Computer Science, 2000.

[Sar+01] Badrul SARWAR et al., « Item-based collaborative filtering recommendation algorithms », in : *Proceedings of the 10th international conference on World Wide Web*, ACM, 2001, p. 285–295.

[Sch+07] J. Ben SCHAFER et al., « Collaborative filtering recommender systems », in : *The adaptive web*, Springer, 2007, p. 291–324.

[SAW94] Bill SCHILIT, Norman ADAMS, et Roy WANT, « Context-aware computing applications », in : *Mobile Computing Systems and Applications, 1994. Proceedings., Workshop on*, IEEE, 1994, p. 85–90.

[SV01] Albrecht SCHMIDT et Kristof VAN LAERHOVEN, « How to build smart appliances ? », in : *IEEE Personal Communications* 8.*4* (2001), p. 66–71.

[SKS16] Eric SCHULZ, Emmanouil KONSTANTINIDIS, et Maarten SPEEKENBRINK, « Putting bandits into context : How function learning supports decision making », in : *bioRxiv* (2016), p. 081091.

[SCD15] Omer Berat SEZER, Serdar Zafer CAN, et Erdogan DOGDU, « Development of a smart home ontology and the implementation of a semantic sensor network simulator : An internet of things approach », in : *Collaboration Technologies and Systems (CTS), 2015 International Conference on*, IEEE, 2015, p. 12–18.

[SDO18] Omer Berat SEZER, Erdogan DOGDU, et Ahmet Murat OZBAYOGLU, « Context-aware computing, learning, and big data in Internet of Things : a survey », in : *IEEE Internet of Things Journal* 5.*1* (2018), p. 1–27.

[AB08] Mohammad Yahya H. AL-SHAMRI et Kamal K. BHARADWAJ, « Fuzzy-genetic approach to recommender systems based on a novel hybrid user model », in : *Expert systems with applications* 35.*3* (2008), p. 1386–1399.

[SM95] Upendra SHARDANAND et Pattie MAES, « Social information filtering : algorithms for automating “word of mouth” », in : *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press/Addison-Wesley Publishing Co., 1995, p. 210–217.

[SK12] Subhash K. SHINDE et Uday KULKARNI, « Hybrid personalized recommender system using centering-bunching based clustering algorithm », in : *Expert Systems with Applications* 39.*1* (2012), p. 1381–1387.

[SBZ10] Saman SHISHEHCHI, Seyed Yashar BANIHASHEM, et Nor Azan Mat ZIN, « A proposed semantic recommendation system for e-learning : A rule and ontology based e-learning recommendation system », in : *information technology (ITSim), 2010 international symposium in*, t. 1, IEEE, 2010, p. 1–5.

[SJ08] Roman Y. SHTYKH et Qun JIN, « Capturing user contexts : Dynamic profiling for information seeking tasks », in : *Systems and Networks Communications, 2008. ICSNC'08. 3rd International Conference on*, IEEE, 2008, p. 365–370.

[Sim51] Edward H. SIMPSON, « The interpretation of interaction in contingency tables », in : *Journal of the Royal Statistical Society : Series B (Methodological)* 13.*2* (1951), p. 238–241.

[Sli14] Aleksandrs SLIVKINS, « Contextual bandits with similarity information. », in : *Journal of Machine Learning Research* 15.*1* (2014), p. 2533–2568.

[SPV07] Kostas STEFANIDIS, Evaggelia PITOURA, et Panos VASSILIADIS, « On relaxing contextual preference queries », in : *Mobile Data Management, 2007 International Conference on*, IEEE, 2007, p. 289–293.

[SL04] Thomas STRANG et Claudia LINNHOFF-POPIEN, « A context modeling survey », in : *Workshop on advanced context modelling, reasoning and management, UbiComp*, t. 4, 2004, p. 34–41.

[SB98] Richard S. SUTTON et Andrew G. BARTO, *Reinforcement learning : An introduction*, t. 1, 1, MIT press Cambridge, 1998.

[SNM14] Panagiotis SYMEONIDIS, Dimitrios NTEMPOS, et Yannis MANOLOPOULOS, *Recommender systems for location-based social networks*, Springer, 2014.

[SZ16] Panagiotis SYMEONIDIS et Andreas ZIOUPOS, *Matrix and Tensor Factorization Techniques for Recommender Systems*, t. 1, Springer, 2016.

[TT17] Cem TEKIN et Eralp TURGAY, « Multi-objective contextual multi-armed bandit problem with a dominant objective », in : *arXiv preprint arXiv :1708.05655* (2017).

[TV03] Vagan TERZIYAN et Oleksandra VITKO, « Bayesian metanetworks for modelling user preferences in mobile environment », in : *Annual Conference on Artificial Intelligence*, Springer, 2003, p. 370–384.

[TM17] Ambuj TEWARI et Susan A. MURPHY, « From ads to interventions : Contextual bandits in mobile health », in : *Mobile Health*, Springer, 2017, p. 495–517.

[Tho33] William R. THOMPSON, « On the likelihood that one unknown probability exceeds another in view of the evidence of two samples », in : *Biometrika* 25.*3/4* (1933), p. 285–294.

[TP11] Michel TOKIC et Günther PALM, « Value-difference based exploration : adaptive control between epsilon-greedy and softmax », in : *KI 2011 : Advances in Artificial Intelligence* (2011), p. 335–346.

[Van01] Kristof VAN LAERHOVEN, « Combining the self-organizing map and k-means clustering for on-line classification of sensor data », in : *International Conference on Artificial Neural Networks*, Springer, 2001, p. 464–469.

[VAL01] Kristof VAN LAERHOVEN, Kofi A AIDOO, et Steven LOWETTE, « Real-time analysis of data from many sensors with neural networks », in : *ISWC*, IEEE, 2001, p. 115.

[VPK04] Mark VAN SETTEN, Stanislav POKRAEV, et Johan KOOLWAAIJ, « Context-aware recommendations in the mobile tourist application COMPASS », in : *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, Springer, 2004, p. 235–244.

[VC11] Saúl VARGAS et Pablo CASTELLS, « Rank and relevance in novelty and diversity metrics for recommender systems », in : *Proceedings of the fifth ACM conference on Recommender systems*, ACM, 2011, p. 109–116.

[Vas96] Julita VASSILEVA, « A task-centered approach for user modeling in a hypermedia office documentation system », in : *User modeling and user-adapted interaction* 6.*2-3* (1996), p. 185–223.

[VM05] Joannes VERMOREL et Mehryar MOHRI, « Multi-armed bandit algorithms and empirical evaluation », in : *European conference on machine learning*, Springer, 2005, p. 437–448.

[VBW15] Sofia S. VILLAR, Jack BOWDEN, et James WASON, « Multi-armed bandit models for the optimal design of clinical trials : benefits and challenges », in : *Statistical science : a review journal of the Institute of Mathematical Statistics* 30.*2* (2015), p. 199.

[Von07] Ulrike VON LUXBURG, « A tutorial on spectral clustering », in : *Statistics and computing* 17.*4* (2007), p. 395–416.

[Wal+09] Thomas J. WALSH et al., « Exploring compact reinforcement-learning representations with linear regression », in : *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, AUAI Press, 2009, p. 591–598.

[WGX15] Xin WANG, Yunhui GUO, et Congfu XU, « Recommendation Algorithms for Optimizing Hit Rate, User Satisfaction and Website Revenue. », in : *In International Joint Conferences on Artificial Intelligence (IJCAI)*, 2015, p. 1820–1826.

[Wan+14] Zan WANG et al., « An improved collaborative movie recommendation system using computational intelligence », in : *Journal of Visual Languages & Computing* 25.*6* (2014), p. 667–675.

[Wan+16] Nirandika WANIGASEKARA et al., « A Bandit Approach for Intelligent IoT Service Composition across Heterogeneous Smart Spaces », in : *Proceedings of the 6th International Conference on the Internet of Things*, ACM, 2016, p. 121–129.

[Wat89] Christopher John Cornish Hellaby WATKINS, « Learning from delayed rewards », thèse de doct., King's College, Cambridge, 1989.

[Whi88] Peter WHITTLE, « Restless bandits : Activity allocation in a changing world », in : *Journal of applied probability* 25.*A* (1988), p. 287–298.

[WBE09] Wolfgang WOERNDL, Michele BROCCO, et Robert EIGNER, « Context-aware recommender systems in mobile scenarios », in : *International Journal of Information Technology and Web Engineering (IJITWE)* 4.*1* (2009), p. 67–85.

[WSW07] Wolfgang WOERNDL, Christian SCHUELLER, et Rolf WOJTECH, « A hybrid recommender system for context-aware recommendations of mobile applications », in : *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*, IEEE, 2007, p. 871–878.

[Wu+02] Huadong WU et al., « Sensor fusion using Dempster-Shafer theory [for context-aware HCI] », in : *Instrumentation and Measurement Technology Conference, 2002. IMTC/2002. Proceedings of the 19th IEEE*, t. 1, IEEE, 2002, p. 7–12.

[WIW18] Qingyun WU, Naveen IYER, et Hongning WANG, « Learning Contextual Bandits in a Non-stationary Environment », in : *arXiv preprint arXiv :1805.09365* (2018).

[XB07] Bo XIAO et Izak BENBASAT, « E-commerce product recommendation agents : use, characteristics, and impact », in : *MIS quarterly* 31.*1* (2007), p. 137–209.

[Yag03] Ronald R. YAGER, « Fuzzy logic methods in recommender systems », in : *Fuzzy Sets and Systems* 136.*2* (2003), p. 133–149.

[YDM14] Saba Q. YAHYAA, Madalina M. DRUGAN, et Bernard MANDERICK, « Annealing-pareto multi-objective multi-armed bandit algorithm », in : *Adaptive Dynamic Programming and Reinforcement Learning (ADPRL), 2014 IEEE Symposium on*, IEEE, 2014, p. 1–8.

[Yu+06a] Zhiwen YU et al., « Supporting context-aware media recommendations for smart phones », in : *IEEE Pervasive Computing 3* (2006), p. 68–75.

[Yu+06b] Zhiwen YU et al., « TV program recommendation for multiple viewers based on user profile merging », in : *User modeling and user-adapted interaction* 16.*1* (2006), p. 63–82.

[Yu+07] Zhiwen YU et al., « Ontology-based semantic recommendation for context-aware e-learning », in : *International Conference on Ubiquitous Intelligence and Computing*, Springer, 2007, p. 898–907.

[Yul03] G. Udny YULE, « Notes on the theory of association of attributes in statistics », in : *Biometrika* 2.*2* (1903), p. 121–134.

[ZN09] Azene ZENEBE et Anthony F. NORCIO, « Representation, similarity measures and aggregation methods using fuzzy sets for content-based recommender systems », in : *Fuzzy sets and systems* 160.*1* (2009), p. 76–94.

[Zen+16] Chunqiu ZENG et al., « Online context-aware recommendation with time varying multi-armed bandit », in : *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2016, p. 2025–2034.

[Zha+09] Daqiang ZHANG et al., « Extended dempster-shafer theory in context reasoning for ubiquitous computing environments », in : *Computational Science and Engineering, 2009. CSE'09. International Conference on*, t. 2, IEEE, 2009, p. 205–212.

[Zha+19] Ruiyi ZHANG et al., « Scalable Thompson Sampling via Optimal Transport », in : *arXiv preprint arXiv :1902.07239* (2019).

[ZRL96] Tian ZHANG, Raghu RAMAKRISHNAN, et Miron LIVNY, « BIRCH : an efficient data clustering method for very large databases », in : *ACM Sigmod Record*, t. 25, 2, ACM, 1996, p. 103–114.

[Zhe15] Yong ZHENG, « Context suggestion : Solutions and challenges », in : *Data Mining Workshop (ICDMW), 2015 IEEE International Conference on*, IEEE, 2015, p. 1602–1603.

[ZMB15] Yong ZHENG, Bamshad MOBASHER, et Robin BURKE, « Similarity-based context-aware recommendation », in : *International Conference on Web Information Systems Engineering*, Springer, 2015, p. 431–447.

[Zhe11] Yu ZHENG, « Location-based social networks : Users », in : *Computing with spatial trajectories*, Springer, 2011, p. 243–276.

[ZZ11] Yu ZHENG et Xiaofang ZHOU, *Computing with spatial trajectories*, Springer Science & Business Media, 2011.

[ZB16] Li ZHOU et Emma BRUNSKILL, « Latent contextual bandits and their application to personalized recommendations for new users », in : *In International Joint Conferences on Artificial Intelligence (IJCAI)* (2016).

[Zho+10] Tao ZHOU et al., « Solving the apparent diversity-accuracy dilemma of recommender systems », in : *Proceedings of the National Academy of Sciences* 107.*10* (2010), p. 4511–4515.

[ZLO07] Andreas ZIMMERMANN, Andreas LORENZ, et Reinhard OPPERMANN, « An operational definition of context », in : *International and Interdisciplinary Conference on Modeling and Using Context*, Springer, 2007, p. 558–571.

Annexes

ANNEXE A

PSEUDO-CODES DES ALGORITHMES DE BANDITS-MANCHOTS CONTEXTUELS ET NON CONTEXTUELS

A.1 Algorithme ε-*Greedy*

Algorithme 5 : ε-*Greedy* pour la recommandation

Données : La liste des utilisateurs $u \in U$, la liste des éléments à recommander (c.-à-d., les bras) $a \in A$, le paramètre de sélection aléatoire $\varepsilon \in [0, 1]$.

pour $t = 1$ *à* T **faire**
  Sélectionner aléatoirement un utilisateur $u \in U$;
  si *un élément* a *n'a pas encore été sélectionné* **alors**
    Recommander cet élément a à l'utilisateur u ;
  sinon
    Générer un nombre réel n entre 0 et 1;
    si $\varepsilon < n$ **alors**
      Calculer $\widehat{\mu}_{a,t}$;
      Recommander l'élément $a_t = \arg\max(\widehat{\mu}_{a,t})$ à l'utilisateur u ;
    sinon
      Recommander aléatoirement un élément $a \in A$ à l'utilisateur u ;
  Observer la récompense obtenue r_t et mettre à jour la moyenne $\mu_{a,t}$;

A.2 Algorithme ε-*First*

Algorithme 6 : *ε-first* pour la recommandation

Données : La liste des utilisateurs $u \in U$, la liste des éléments à recommander (c.-à-d., les bras) $a \in A$, le paramètre de sélection aléatoire $\varepsilon \in [0, 1]$.

pour $t = 1$ *à* εT **faire**
  Sélectionner aléatoirement un utilisateur $u \in U$;
  si *un élément* $a \in A$ *n'a pas encore été sélectionné* **alors**
    Recommander cet élément a à l'utilisateur u ;
  sinon
    Sélectionner aléatoirement un élément $a \in A$ et le recommander à l'utilisateur u;
  Observer la récompense obtenue r_t et mettre à jour la moyenne $\mu_{a,t}$;
pour $t = \varepsilon T + 1$ *à* T **faire**
  Sélectionner aléatoirement un utilisateur $u \in U$;
  si a *n'a pas encore été sélectionné* **alors**
    Sélectionner a ;
  sinon
    Sélectionner l'élément $a_t = \arg\max(\widehat{\mu}_{a,t})$ et le recommander à l'utilisateur u ;
  Observer la récompense obtenue r_t et mettre à jour la moyenne $\mu_{a,t}$;

A.3 Algorithme *UCB*

Algorithme 7 : *UCB1* pour la recommandation

Données : La liste des utilisateurs $u \in U$, la liste des éléments à recommander (c.-à-d., les bras) $a \in A$, le paramètre α (α=2).

pour $t = 1$ *à* T **faire**
  Sélectionner aléatoirement un utilisateur $u \in U$;
  si *un élément* $a \in A$ *n'a pas encore été sélectionné* **alors**
    Recommander cet élément a à l'utilisateur u ;
  sinon
    Calculer pour chaque bras $\widehat{\mu}_{a,t}$ et la borne de confiance supérieure $\sqrt{\frac{\alpha \ln t}{t_a}}$;
    Sélectionner l'élément $a_t = \arg\max(\widehat{\mu}_{a,t} + \sqrt{\frac{\alpha \ln t}{t_a}})$ et le recommander à l'utilisateur u;
  Observer la récompense obtenue r_t et mettre à jour la moyenne $\mu_{a,t}$;

A.4 Algorithme *Thompson Sampling*

Algorithme 8 : *Thompson Sampling (TS)* (Bernoulli) pour la recommandation

Données : La liste des utilisateurs $u \in U$, la liste des éléments à recommander (c.-à-d., les bras) $a \in A$, l'initialisation des succès et échecs $\forall a \in A$, $F_a = 0$ et $S_a = 0$.

pour $t = 1$ *à* T **faire**
- Sélectionner aléatoirement un utilisateur $u \in U$;
- **pour tous les** $a \in A$ **faire**
  - Générer l'échantillon $\gamma_{t,a}$ à partir de la distribution :

$$Beta(S_{t,a} + 1, F_{t,a} + 1)$$

- Sélectionner l'élément $a_t = \arg\max(\gamma_{a,t})$ et le recommander à l'utilisateur u ;
- Observer la récompense obtenue r_t;
- **si** r_t*=1* **alors**
  - $S_a = S_a + 1$
- **sinon**
  - $F_a = F_a + 1$

A.5 Algorithme *Softmax Annealing*

Algorithme 9 : *Annealing Softmax* pour la recommandation

Données : La liste des utilisateurs $u \in U$, la liste des éléments (c.-à-d., les bras) à recommander $a \in A$.

pour $t = 1$ *à* T **faire**
- Sélectionner aléatoirement un utilisateur $u \in U$;
- **pour tous les** $a \in A$ **faire**
  - Calculer $e^{\widehat{\mu_a} \log(t)}$;
- Calculer $\sum_{i=1}^{k} e^{\widehat{\mu}_{a_i} \log(t)}$;
- Recommander l'élément a_t à l'utilisateur u selon la probabilité :

$$P_{a,t} \leftarrow \frac{e^{\widehat{\mu_a} \log(t)}}{\sum_{i=1}^{k} e^{\widehat{\mu_a} \log(t)}}$$

- Observer la récompense r_t et mettre à jour la moyenne $\mu_{a,t}$ correspondante ;

A.6 Algorithme *SW-UCB*

Algorithme 10 : *SW-UCB* pour la recommandation

Données : La liste des utilisateurs $u \in U$, la liste des éléments à recommander (c.-à-d., les bras) $a \in A$, le paramètre α, la taille de la fenêtre H.

pour $t = 1$ *à* T **faire**
- Sélectionner aléatoirement un utilisateur $u \in U$;
- **pour tous les** $a \in A$ **faire**
  - **si** *l'élément* a *n'a pas encore été sélectionné* **alors**
    - Recommander cet élément a à l'utilisateur u ;
  - **sinon**
    - Calculer $\sum r_a(H) = \sum_{s=t-H+1}^{t} r_{a,s} \, \mathbb{1}_{\{a_s=a\}}$ et $t_a(H) = \sum_{s=t-H+1}^{t} \mathbb{1}_{\{a_s=a\}}$;
    - Calculer $\overline{\mu}_{a,t}(H) = \frac{1}{t_a(H)} \sum r_{a,t}(H)$;
    - Recommander à l'utilisateur u, l'élément $a_t = \arg\max_{a \in A} \left(\overline{\mu}_{a,t}(H) + \sqrt{\frac{\alpha \ln (t \wedge H)}{t_a(H)}} \right)$
- Observer la récompense obtenue r_t et mettre à jour $\sum r_{a_t,t+1}(H)$ tel que $\sum r_{a_t,t+1}(H) = \sum_{s=t-H+1}^{t} r_{a_t,s} \, \mathbb{1}_{\{a_s=a_t\}} + r_t$

A.7 Algorithme *EXP3*

Algorithme 11 : *EXP3* pour la recommandation

Données : La liste des utilisateurs $u \in U$, la liste des k éléments à recommander $a \in A$ (c.-à-d., les bras), $\eta \in [0, 1]$, $\forall a \in A, w_a = 1$.

pour $t = 1$ *à* T **faire**
- Sélectionner aléatoirement un utilisateur $u \in U$;
- Recommander l'élément a_t à l'utilisateur u selon la probabilité :

$$P_{a,t} = (1 - \eta) \frac{w_{a,t}}{\sum_{i=1}^{k} w_{a_i,t}} + \frac{\eta}{k}$$

- Observer la récompense $r_{a,t}$ retournée par l'utilisateur;
- **pour tous les** $a \in A$ **faire**
  - $\widehat{r}_{a,t} = \begin{cases} r_{a,t}/p_{a,t} & \text{si} \quad a = a_t \\ 0 & \text{sinon} \end{cases}$;
  - Mettre à jour $w_{a,t+1} = w_{a,t} \exp\left(\eta \, \frac{\widehat{r}_{a,t}}{k}\right)$;

A.8 Algorithme *Epoch-Greedy*

Algorithme 12 : *Epoch-Greedy*

Données : L'ensemble des k éléments à recommander (c.-à-d., les bras) $a \in A$ disponibles, l'horizon T, le nombre d'époques L, l'ensemble des n contextes fixes disponibles X, $t_1 \leftarrow 1$, et l'ensemble d'apprentissage W_l avec $W_0 = \{\emptyset\}$.

pour $l = 1, 2..., L$ **faire**
- // Phase d'exploration
- $t \leftarrow t_l$;
- Considérer $x_t \in X$: un utilisateur u et son contexte;
- Sélectionner un élément $a \in A$ depuis une distribution uniforme et le recommander à l'utilisateur u;
- Observer la récompense $r_{a,t} \in [0, 1]$ retournée par l'utilisateur;
- Ajouter $(x_t, a_t, r_{a,t})$ à l'ensemble d'apprentissage W_l c.-à-d., $W_l \leftarrow W_{l-1} \cup \{(x_t, a_t, r_{a,t})\}$;
- Optimiser π sur W_l;
- **pour** $t = t_l + 1, ..., t_{l+1} - 1$ **faire**
  - // Phase d'exploitation
  - Considérer $x_t \in X$: un utilisateur u et son contexte;
  - Sélectionner l'élément $a \in A$ estimé comme étant optimal selon la politique $\pi_t(x_t)$ et le recommander à l'utilisateur u;
  - Observer la récompense $r_{a,t} \in [0, 1]$ retournée par l'utilisateur;

A.9 Algorithme *LinUCB*

Algorithme 13 : *LinUCB (Linear Disjoint)*

Données : L'ensemble des k éléments à recommander (c.-à-d., les bras) $a \in A$ disponibles, $\alpha \in \mathbb{R}^+$, l'horizon T, et l'ensemble des n contextes fixes disponibles X. $\forall a \in A, M_a \leftarrow I_d, b_a \leftarrow 0_d$.

pour $t = 1$ *à* T **faire**
- Considérer $x_t \in X$: un utilisateur et son contexte;
- **pour tous les** $a \in A$ **faire**
  - $\widehat{\theta}_a \leftarrow M_a^{-1} b_a$;
  - $p_{t,a} \leftarrow \widehat{\theta_a}^\top x_t + \alpha\sqrt{x_t^\top M_a^{-1} x_t}$;
- Sélectionner l'élément $a_t = \arg\max_{a_t \in A} (p_{t,a})$ et le recommander à l'utilisateur u;
- Observer la récompense r_t retournée par l'utilisateur u;
- $M_{a_t} \leftarrow M_{a_t} + x_t x_t^\top$;
- $b_{a_t} \leftarrow b_{a_t} + r_t x_t$;

A.10 Algorithme *Contextual Thompson Sampling*

Algorithme 14 : *Contextual Thompson Sampling (CTS)*

Données : L'ensemble des k éléments à recommander (c.-à-d., les bras) $a \in A$ disponibles, l'horizon T, et l'ensemble des n contextes fixes disponibles X.$\forall a \in A, B_a \leftarrow I_d, f_a \leftarrow 0_d, \widehat{\theta_a} \leftarrow 0_d$.

pour $t = 1$ *à* T **faire**
- Considérer $x_t \in X$: un utilisateur et son contexte;
- **pour tous les** $a \in A$ **faire**
  - Générer l'échantillon $\tilde{\theta}_{t,a}$ à partir de la distribution :

$$\mathcal{N}(\widehat{\theta}_{t,a}, v^2 B_{t,a}^{-1})$$

- Sélectionner l'élément $a_t = \arg\max x_{t,a}^{\top} \tilde{\theta}_{t,a}$ et le recommander à l'utilisateur u;
- Observer la récompense r_t retournée par l'utilisateur u;
- $B_{t,a} \leftarrow B_{t,a} + x_{a,t} x_{a,t}^{\top}$;
- $f_{t,a} \leftarrow f_{t,a} + r_t x_{a,t}$;
- $\widehat{\theta}_{t,a} \leftarrow B_{t,a}^{-1}\ f_{t,a}$;

A.11 Algorithme *EXP4*

Algorithme 15 : *EXP4* pour la recommandation

Données : La liste des utilisateurs $u \in U$, la liste des k éléments à recommander (c.-à-d., les bras) $a \in A$, $\eta \in [0, 1]$. $\forall i \in \{1, ..., k\}, w_i = 1$.

pour $t = 1$ *à* T **faire**
- Sélectionner aléatoirement un utilisateur $u \in U$;
- Recommander l'élément a_t à l'utilisateur u selon la probabilité :

$$P_{a,t} = (1 - \eta) \sum_{i=1}^{N} \frac{w_{i,t}\ \xi_{a,t}^{i}}{\sum_{j=1}^{N}\ w_{j,t}} + \frac{\eta}{k}$$

- Observer la récompense r_t retournée par l'utilisateur;
- **pour tous les** $a \in A$ **faire**
  - $\widehat{r}_{a,t} = \begin{cases} r_{a,t}/p_{a,t} & \text{si} \quad a = a_t \\ 0 & \text{sinon} \end{cases}$;
  - **pour** $i = 1, ..., N$ **faire**
    - Calculer $\widehat{y}_{i,t} = \xi^i\ .\ \widehat{r}_t$;
    - Mettre à jour $w_{i,t+1} = w_{i,t}\ \exp\left(\eta\ \frac{\widehat{y}_{i,t}}{k}\right)$;

ANNEXE B

ANALYSES DÉTAILLÉES DE L'ÉTUDE PRÉLIMINAIRE SUR LES ALGORITHMES DE BANDITS-MANCHOTS APPLIQUÉS À LA RECOMMANDATION

B.1 Analyses détaillées de la précision globale : *MABs* versus *CMABs* sur jeux de données avec vecteur complet

***MABs* versus *CMABs*.** Pour chaque résultat de précision globale obtenu (Tableau C.2 et Tableau C.1) sur chacun des jeux de données, le test de *Kruskal-Wallis* (*KW*) indique qu'il existe des différences significatives entre les résultats de précision globale des différents algorithmes et ce pour chacun des jeux de données ($p < 0,01$, H_0 est rejeté).

Au vu des résultats des Tableaux C.2 et C.1, et du test de *KW*, les algorithmes de *CMAB* présentent des résultats significativement supérieurs aux algorithmes de *MAB*. Mais étions-nous en mesure de connaître ce résultat à l'avance ? Si les données de contexte fournies en entrée n'avaient pas été pertinentes ou en l'absence d'une réelle dépendance linéaire entre les récompenses et les dimensions du contexte pour les algorithmes de *CMAB* basés sur les modèles linéaires, alors les algorithmes de *MAB* auraient très bien pu supplanter, et de loin, les algorithmes de *CMAB*. C'est pourquoi, nous explorerons également des cas *vt* et un cas sans contexte disponible (cas non contextuel). D'autre part, nous verrons que ces résultats sont à nuancer en fonction du type d'algorithmes de *CMAB* que nous employons. En effet, même si les algorithmes basés sur les modèles linéaires semblent offrir l'assurance d'une performance supérieure vis à vis des algorithmes de *MAB*, ce n'est pas toujours le cas pour celui basé sur la sélection de politiques (*EXP4.P*).

Ci-dessous, nous détaillons et analysons plus précisément les résultats obtenus pour chaque jeu de données et pour chaque algorithme.

Jeux de données artificiels (*Contrôle (vc)* et *YSE (vc)*). Sur ces deux jeux de données, les tests de *Wilcoxon* indiquent qu'il existe une différence significative entre les résultats obtenus pour chacun des algorithmes de *MAB* et *CMAB* comparé deux à deux ($p < 0,001$, H_0 est rejetée).

En revanche, le test de *KW* appliqué pour chaque catégorie d'algorithmes indépendamment l'une de l'autre indique qu'il n'existe aucune différence significative entre les résultats obtenus par chacun des algorithmes d'une même catégorie ($p > 0,496$; H_0 est acceptée).

Ainsi, lorsqu'un contexte optimal est fourni en entrée d'un algorithme de *CMAB*, celui-ci converge vers une valeur de précision globale proche de $1,0 \pm \varepsilon$ (avec $\varepsilon < 0,0009$). De plus, il n'y a pas de différences statistiquement significatives des résultats entre chacun des trois algorithmes de *CMAB* étudiés (Pour *Contrôle (vc)* et *YSE (vc)* : $Acc(\textit{LinUCB}) = 1,0$, $Acc(\textit{CTS}) = 1,0$ et $Acc(\textit{EXP4.P}) = 0,999$, voir tableau C.1).

Notons donc, qu'entre une stratégie basée sur un modèle linéaire et celle basée sur une sélection de politiques via des experts, ici il n'y a pas d'avantages ou d'inconvénients à utiliser l'une ou l'autre des solutions.

Néanmoins cette observation reste à nuancer du fait que les algorithmes du même type que *EXP4* sont performants lorsque les experts sont d'accords entre eux [Aue+95] (c.-à-d., $\forall i, i'$, $\xi_t^i = \xi_t^{i\prime}$) et d'autant plus si le nombre d'actions à choisir est faible [TM17], ce qui est notre cas ici. Il sera donc important d'observer les résultats de telles approches sur des jeux de données du monde réel, plus dimensionnés ou variés en terme d'actions possibles, mais aussi moins stables en ce qui concerne les experts intervenants dans le processus de décision.

Enfin, il apparaît également avantageux d'utiliser un algorithme de *CMAB* quand le contexte fourni est optimal plutôt qu'une stratégie non contextuelle telle qu'un algorithme de *MAB* qui régresse à une précision globale de $0,25$ pour le jeu de données *Contrôle* et $0,5$ pour le jeu de données *YSE*. Rappelons que le jeu de données *Contrôle* possède quatre bras de probabilités de récompenses égales (25%) et que *YSE (vc)* possède deux bras dont les récompenses sont également réparties de manière équitable (50%). Aussi, sans contexte permettant de différencier l'attribution des classes (c.-à-d., des bras/recommandations), il est évident que les *MAB* dans ce cas sont significativement moins performant. À noter qu'à l'instar des *CMAB*, sur ces jeux de données artificiels, il n'y a pas de différences statistiquement significatives entre les résultats obtenus pour chacun des quatre algorithmes de *MAB* étudiés.

Jeux de données spécifiques à la recommandation (*RS-ASM (vc)* et *Food*). Sur les jeux de données *RS-ASM (vc)* et *Food*, les tests de *Wilcoxon* indiquent qu'il existe une différence significative entre les résultats obtenus pour chacun des algorithmes comparé deux à deux de *MAB* et de *CMAB* basé sur un modèle linéaire uniquement ($p < 0,01$, H_0 est rejetée). Ceux-ci obtiennent ainsi une précision globale supérieure aux algorithmes de *MAB* sur *RS-ASM (vc)* ($Acc(LinUCB) = 0,78; Acc(CTS) = 0,70; Acc(UCB2) = 0,59; Acc(EXP3) = 0,56; Acc(TS) = 0,59; Acc(\varepsilon - Greedy) = 0,59$).

C'est d'autant plus le cas sur *Food* ($Acc(LinUCB) = 0,98; Acc(CTS) = 0,98; Acc(UCB2) = 0,738; Acc(EXP3) = 0,74; Acc(TS) = 0,81; Acc(\varepsilon - Greedy) = 0,77$).

EXP4.P quant à lui n'offre pas de différence statistiquement significative avec les algorithmes de *MAB* sur le jeu de données *RS-ASM (vc)* ($Acc(EXP4.P) = 0,57; p > 0,23$; H_0 est acceptée) et pis encore sur le jeu de données *Food* où il est significativement inférieur à tout algorithme de *MAB* comparé deux à deux en termes de précision globale ($Acc(EXP4.P) = 0,63;$

$p < 0,01$; H_0 est rejetée). Ceci peut s'expliquer par le fait que dans le jeu de données *RS-ASM* et *Food*, il existe un plus grand nombre d'actions (18 dans *RS-ASM* et 20 dans *Food*) que dans les jeux de données artificiels [TM17] (4 pour *Contrôle* et 2 pour *YSE*) et également plus d'experts (8 variables catégorielles dans *RS-ASM* et 80 dans *Food* contre 4 pour les jeux de données artificiels). Dans ce cas, les experts ne sont pas tous d'accord entre eux ce qui rend le choix probabiliste de l'action moins précis [Aue+95].

Le test de *KW* appliqué pour chaque catégorie *MAB* et *CMAB* indépendamment l'une de l'autre indique qu'il existe également une différence significative entre les résultats obtenus par chacun des algorithmes d'une même catégorie ($p < 0,01$; H_0 est rejetée) tant sur le jeu de données *RS-ASM (vc)* que *Food*.

Ainsi, parmi les algorithmes de *CMAB* sur le jeu de données *RS-ASM*, on observera que *LinUCB* offre, en termes de précision globale, une garantie de performance supérieure à *CTS* et *EXP4.P* ($Acc(LinUCB) = 0,78$, $Acc(CTS) = 0,70$; $Acc(EXP4.P) = 0,57$; $p < 0,01$; H_0 est rejetée, test de *Wilcoxon*).

D'autre part, parmi les algorithmes de *CMAB* sur le jeu de données *Food*, bien que nous n'observerons pas de différence statistiquement significative entre *LinUCB* et *CTS* ($Acc(LinUCB) = 0,98$, $Acc(CTS) = 0,98$; $p > 0,54$; H_0 est acceptée, test de *Wilcoxon*). En revanche ceux-ci offrent une garantie de précision globale signigicativement supérieure à *EXP4.P* ($Acc(EXP4.P) = 0,63$; $p < 0,01$; H_0 est rejetée, test de *Wilcoxon*).

Jeux de données de recommandation de type conseil à un tiers humain (*Yeast*, *Adult*, *Mushroom*, *Statlog* et *Students Academics Performance*. Sur l'ensemble de ces jeux de données, les tests de *Wilcoxon* indiquent qu'il existe une différence significative entre les résultats obtenus pour chacun des algorithmes comparé deux à deux de *MAB* et de *CMAB* basé sur un modèle linéaire ($p < 0,01$; H_0 est rejetée). Ceux-ci obtiennent ainsi une précision globale supérieure aux algorithmes de *MAB* sur chacun de ces jeux de données (voir les tableaux C.2 et C.1).

EXP4.P quant à lui n'offre pas de différence statistiquement significative avec les algorithmes de *MAB* sur l'ensemble de ces jeux de données ($p > 0,49$, H_0 est acceptée). La majorité de ces jeux de données possède beaucoup d'experts (*Adult, Mushroom, Statlog, Students Academics Perf.*) qui peuvent se contredire, et pour *Yeast* une combinaison nombre d'experts / nombres d'actions possibles qui n'est également pas le plus optimal en termes de complexité. Ainsi, utiliser *EXP4.P* par rapport à un *MAB* dans ces cas n'offre pas de garanties significativement supérieures de précision globale.

Le test de *KW* appliqué pour chaque catégorie *MAB* et *CMAB* indépendamment l'une de l'autre indique qu'il existe également une différence significative entre les résultats obtenus par chacun des algorithmes d'une même catégorie ($p < 0,01$; H_0 est rejetée) et ce pour chacun des jeux de données (*Yeast, Adult, Mushroom, Statlog et Students Academics Performance*).

Parmi les algorithmes de *CMAB* sur l'ensemble des jeux de données, on n'observera pas de différence statistiquement significative entre *LinUCB* et *CTS* (Voir tableaux C.2 et C.1, au minimum $p > 0,21$ sur l'ensemble des jeux de données, H_0 est donc acceptée, test de *Wilcoxon*).

Notons par ailleurs que ces deux algorithmes obtiennent une précision globale significativement supérieure à *EXP4.P* ($p < 0,01$ sur l'ensemble des jeux de données, H_0 est donc rejetée, test de *Wilcoxon*).

Jeux de données de montée à l'échelle (*Covertype* et *Poker Hand*). Il est important d'observer quels résultats nous obtenons quand les algorithmes passent à l'échelle.

Ainsi, *Coverype* et *Poker Hand*, même s'ils ne constituent pas des jeux de données spécifiquement utilisées pour la recommandation, ont déjà été employés pour vérifier la montée à l'échelle des algorithmes de *MAB* et *CMAB* [All16 ; Bou+17 ; Zha+19].

Sur le jeu de données *Poker Hand*, les tests de *Wilcoxon* réalisés entre chaque algorithme comparé deux à deux de *MAB* et de *CMAB* indiquent qu'il n'existe qu'une seule différence significative entre les résultats obtenus et ce avec l'algorithme ε-*Greedy* qui obtient une précision globale inférieure aux autres méthodes ($Acc(\varepsilon - Greedy) = 0,42$ $p < 0,01$; H_0 est rejetée). Dans les autres cas, *LinUCB* et *CTS* obtiennent une précision globale supérieure aux algorithmes de *MAB* et également par rapport à *EXP4.P* (voir les tableaux C.2 et C.1), mais cette différence est non significative ($p > 0,12$; H_0 est acceptée, test de *Wilcoxon*).

Sur le jeu de données *Covertype*, les tests de *Wilcoxon* indiquent qu'il existe une différence significative entre les résultats obtenus pour chacun des algorithmes comparé deux à deux de *MAB* et de *CMAB* basé sur un modèle linéaire (*LinUCB*, *CTS*) ($p < 0,01$; H_0 est rejetée). Ceux-ci obtiennent ainsi une précision globale supérieure aux algorithmes de *MAB* (voir les tableaux C.2 et C.1).

Pour *EXP4.P*, bien que l'on observe une précision globale supérieure ($Acc(EXP4.P) = 0,5$), à *UCB2* ($Acc(UCB2) = 0,49$) et à *TS* ($Acc(TS) = 0,49$), il n'offre pas de différence statistiquement significative avec ces algorithmes sur le jeu de données *Covertype* ($p > 0,13$; H_0 est acceptée, test de *Wilcoxon*). En revanche, il obtient un résultat de précision globale significativement supérieur que ceux obtenus par les algorithmes *EXP3* et ε-*Greedy* ($p < 0,01$; H_0 est rejetée, test de *Wilcoxon*). *EXP4.P* reste néanmoins inférieur aux algorithmes de *CMAB* basés sur un modèle linéaire sur le jeu de données *Covertype*. Ce jeu de données à la particularité de posséder un grand nombre d'experts (54 variables catégorielles) résultant en des contradictions sur les sept actions possibles. Enfin, on remarquera sur le jeu de données *Covertype* qu'*UCB2* et *TS* obtiennent une meilleure précision globale qu'*EXP3* et ε-*Greedy* ($p < 0,01$, H_0 est rejetée, test de *Wilcoxon*).

B.2 Analyses détaillées de la précision globale : *MABs* versus *CMABs* sur jeux de données avec vecteur de contexte tronqué

Contrôle (vt). Sur le jeu de données *Contrôle*, les tests de *Wilcoxon* indiquent qu'il existe une différence significative entre les résultats obtenus pour chacun des algorithmes de *MAB* et *CMAB* comparé deux à deux ($p < 0,001$; H_0 est rejetée). Il apparaît que les algorithmes

de *CMAB* obtiennent ainsi une meilleure précision globale que les algorithmes de *MAB* et ce malgré la perte d'information. En revanche, les algorithmes de *CMAB* résolvant notre problème de recommandation contraint par un vecteur de contexte tronqué (c.-à-d., restreint) sont moins précis que lorsqu'ils ont à disposition le vecteur de contexte original (c.-à-d., complet). Ainsi, on observe une diminution statistiquement significative de la précision globale entre les deux cas *vc* et *vt* ($p < 0,001$; H_0 est rejetée, tests de *Wilcoxon* voir Tableaux C.2 et C.3).

Le test de *KW* appliqué pour chaque catégorie indépendamment l'une de l'autre indique qu'il n'existe aucune différence significative entre les résultats obtenus par chacun des algorithmes d'une même catégorie ($p > 0,32$; H_0 est acceptée).

Ainsi, sur le jeu de données *Contrôle (vt)*, lorsque le contexte optimal de dimension $d = 4$ (défini pour quatre actions possibles, c.-à-d., quatre bras) est tronqué à hauteur de 25% (on retire la dernière dimension), les algorithmes de *CMAB* convergent vers une valeur de précision globale proche de $0,75$. De plus, il n'y a pas de différences statistiquement significatives des résultats entre chacun des trois algorithmes de *CMAB* étudiés (Pour *Contrôle (vt)* : $Acc($*LinUCB*$) = 0,75$; $Acc($*CTS*$) = 0,751$ et $Acc($*EXP4.P*$) = 0,744$; voir tableau C.3).

Notons ainsi qu'entre une stratégie basée sur un modèle linéaire et celle basée sur une sélection de politiques via des experts, ici il n'y a pas d'avantages ou d"inconvénients statistiquement significatifs à utiliser l'une ou l'autre des méthodes. Néanmoins, même si cette différence n'est pas significative, on peut observer un léger avantage de *CTS* par rapport à *LinUCB*, et de *LinUCB* par rapport à *EXP4.P*.

RS-ASM (vt). Sur le jeu de données *RS-ASM (vt)*, les tests de *Wilcoxon* indiquent qu'il existe une différence significative entre les résultats obtenus pour chacun des algorithmes de *MAB* et *CMAB* basés sur un modèle linéaire comparés deux à deux ($p < 0,01$; H_0 est rejetée). Il apparaît que les algorithmes de *CMAB* basés sur un modèle linéaire obtiennent ainsi une meilleure précision globale que les algorithmes de *MAB* et ce malgré la perte d'information. En revanche, les algorithmes de *CMAB* résolvant notre problème de recommandation contraint par un vecteur de contexte tronqué (c.-à-d., restreint) sont moins précis que lorsqu'ils ont à disposition le vecteur de contexte complet. On observe une diminution statistiquement significative de la précision globale ($p < 0,01$; H_0 est rejetée, tests de *Wilcoxon*), voir tableau C.2 et C.3.

EXP4.P quant à lui n'offre pas de différence statistiquement significative avec les algorithmes de *MAB* sur le jeu de données *RS-ASM (vt)* ($Acc(EXP4.P) = 0,58; p > 0,22$; H_0 est acceptée). En revanche, même si la différence n'est pas significative, *EXP4.P* semble obtenir une précision globale sensiblement meilleure avec vecteur tronqué qu'avec vecteur complet. Ainsi, nous supposons que les choix entre experts s'accordent mieux avec un nombre moins important d'entre eux.

YSE (vt). Sur le jeu de données *YSE(vt)*, les tests de *Wilcoxon* indiquent qu'il existe une différence significative entre les résultats obtenus pour chacun des algorithmes de *MAB* et *CMAB* basés sur un modèle linéaire comparés deux à deux ($p < 0,01$; H_0 est rejetée). Il apparaît

que les algorithmes de *CMAB* basés sur un modèle linéaire obtiennent ainsi une meilleure précision globale que les algorithmes de *MAB* et ce malgré la perte d'information. En revanche, les algorithmes de *CMAB* résolvant notre problème de recommandation contraint par un vecteur de contexte tronqué (c.-à-d., restreint) sont moins précis que lorsqu'ils ont à disposition le vecteur de contexte complet. On observe une diminution statistiquement significative de la précision globale ($p < 0,01$; H_0 est rejetée, tests de *Wilcoxon*), voir Tableaux C.2 et C.3. Il est important de souligner que la baisse de performance de *LinUCB* au même titre que celle de *CTS* est le résultat de l'effet *Yule-Simpson* que nous avons volontairement provoquée en retirant la dimension « pays ».

EXP4.P quant à lui n'offre pas de différence statistiquement significative avec les algorithmes de *MAB* sur le jeu de données *RS-ASM (vt)* ($Acc(EXP4.P) = 0,58; p > 0,22$; H_0 est acceptée). En revanche, *EXP4.P* souffre cette fois de la perte d'experts pertinents dans la prise de décision et est fortement affecté par l'effet *Yule-Simpson*. Ainsi *EXP4.P* avec vecteur tronqué est significativement moins performant qu'avec vecteur complet ($p < 0,01$, H_0 est rejetée, tests de *Wilcoxon*).

B.3 Analyses détaillées de la précision globale : *MABs* versus *CMABs* dans le cas non-stationnaire

Sur le jeu de données *RS-ASM (ns)*, les tests de *Wilcoxon* indiquent qu'il existe une différence significative entre les résultats obtenus pour chacun des algorithmes comparé deux à deux de *MAB* et de *CMAB* basés sur un modèle linéaire uniquement ($p < 0,01$, H_0 est rejetée). Ceux-ci obtiennent ainsi une précision globale supérieure aux algorithmes de *MAB* sur *RS-ASM (ns)* ($Acc(LinUCB) = 0,73; Acc(CTS) = 0,685; Acc(UCB2) = 0,586; Acc(EXP3) = 0,557; Acc(TS) = 0,592; Acc(\varepsilon - Greedy) = 0,516$; voir Tableau C.4). Enfin, nous remarquons qu'*EXP4.P* n'est pas plus performant que les algorithmes de *MAB* pour les mêmes raisons que nous avons évoquées concernant la version (*vc*). Les résultats qu'obtient *EXP4.P* sont même significativement inférieurs à ceux obtenus par les autres méthodes ($Acc(EXP4.P) = 0,52; p < 0,01$; H_0 est rejetée).

B.4 Analyses détaillées de la précision globale : *MABs* versus *CMABs* sur jeux de données dépourvu de contexte

Sur le jeu de données non contextuel *Jester*, les tests de *Wilcoxon* indiquent sans surprise qu'il existe une différence significative entre les résultats obtenus pour chacun des algorithmes de *MAB* et *CMAB* comparé deux à deux ($p < 0,001$, H_0 est rejetée). Les algorithmes de *MAB* obtiennent ainsi une précision globale significativement supérieure aux algorithmes de *CMAB* dans le cas d'une restriction totale de contexte (Voir Tableau C.5, ce qui corrobore les observations faites par [Bou+17]).

Parmi les algorithmes de *MAB*, *UCB2* et *TS* obtiennent des résultats de précision globale significativement supérieurs que ceux obtenus par *EXP3* et ε-*Greedy* ($p < 0,01$; H_0 est rejetée). *UCB2* et *TS* offre également une garantie de précision globale similaire aux deux précédents algorithmes ($Acc(UCB2) = 0,75; Acc(TS) = 0,74 p = 0,34$; H_0 est acceptée, test de *Wilcoxon*). Notons par ailleurs que, même si ce n'est pas significatif, *TS* offre un résultat sensiblement supérieur aux autres méthodes.

B.5 Analyses détaillées de la diversité : *MABs* versus *CMABs* sur jeux de données avec vecteur complet

Jeux de données artificiels (*Contrôle (vc)* et *YSE (vc)*). Sur ces deux jeux de données, les algorithmes de *CMAB* obtiennent des résultats de diversité significativement supérieur que les algorithmes de *MAB* (Tests de *Wilcoxon* pour chacun des algorithmes de *MAB* et *CMAB* comparé deux à deux : $p < 0,001$; H_0 est rejetée).

Le test de *KW* effectué sur la catégorie *CMAB* uniquement, indique qu'il n'existe aucune différence significative entre les résultats obtenus par chacun des algorithmes de *CMAB* ($p = 0,41$; H_0 est acceptée). Ainsi, lorsque le contexte fourni en entrée est optimal, les algorithmes de *CMAB* parviennent a suffisamment diversifier en s'appuyant sur chacune des Caractéristiques pertinentes du contexte. Notons également, qu'entre une stratégie basée sur un modèle linéaire et celle basée sur une sélection de politiques via des experts, ici il n'y a pas d'avantages ou d'inconvénients significatifs à utiliser l'une ou l'autre des méthodes. En effet, *EXP4.P* reste performant sur ces deux jeux de données dont les actions sont peu nombreuses et les experts d'accord entre eux.

En ce qui concerne la catégorie des *MABs*, en revanche, le test de *KW* indique qu'il existe une différence significative entre les résultats obtenus par chacun des algorithmes ($p < 0,01$, H_0 est rejetée). Ainsi, pour un même résultat de précision globale, la diversité des recommandations obtenue par chacun des algorithmes n'est pas la même. On préférera donc employer *UCB2* dans le cadre du jeu de données *Contrôle (vc)* qui obtient significativement la meilleure diversité parmi l'ensemble des algorithmes de *MAB* comparés deux à deux ($Div(UCB2) = 0,59; p < 0,01$; H_0 est rejetée, test de *Wilcoxon*). On préférera néanmoins employer ε-*Greedy* dans le cadre du jeu de données *YSE (vc)* ($Div(\varepsilon - Greedy) = 0,25; p < 0,01; H_0$ est rejetée, test de *Wilcoxon*).

Jeux de données spécifiques à la recommandation (*RS-ASM* et *Food*). Sur le jeu de données *RS-ASM (vc)*[1], les tests de *Wilcoxon* indiquent qu'il existe une différence significative entre les résultats obtenus pour chacun des algorithmes comparé deux à deux de *MAB* et de *CMAB* basé sur un modèle linéaire uniquement ($p < 0,01$; H_0 est rejetée). Ceux-ci

1. Notons que toutes les expérimentations (sur environnement stationnaire) de ce chapitre ont été effectuées sur la partie *été* du jeu de données *RS-ASM*, soit 1076 instances.

obtiennent ainsi une diversité significativement supérieure aux algorithmes de *MAB* sur *RS-ASM (vc)* ($Div(LinUCB) = 0,86; Div(CTS) = 0,79; Div(UCB2) = 0,04; Div(EXP3) = 0,16; Div(TS) = 0,02; Div(\varepsilon - Greedy) = 0,2$). *EXP4.P* quant à lui offre une diversification significativement inférieure aux algorithmes de *CMAB* basés sur un modèle linéaire ($Div(EXP4.P) = 0,04; p > 0,47$; H_0 est acceptée).

Sur le jeu de données *Food*, les tests de *Wilcoxon* indiquent qu'il existe une différence significative entre les résultats obtenus pour chacun des algorithmes comparé deux à deux de *MAB* et de *CMAB* ($p < 0,01$; H_0 est rejetée). Ceux-ci obtiennent ainsi une diversité supérieure aux algorithmes de *MAB* sur *Food* ($Div(EXP4.P) = 0,93; Div(LinUCB) = 0,88; Div(CTS) = 0,84; Div(UCB2) = 0,04; Div(EXP3) = 0,25; Div(TS) = 0,01; Div(\varepsilon - Greedy) = 0,19$). Cette fois *EXP4.P* obtient une diversification supérieure (mais non significative) que celles obtenues par les algorithmes de *CMAB* basés sur un modèle linéaire ($p < 0,05$; H_0 est rejetée).

Le test de *KW* appliqué aux algorithmes de *MAB* indique qu'il existe également une différence significative entre les résultats obtenus par chacun des algorithmes d'une même catégorie ($p < 0,01$; H_0 est rejetée) tant sur le jeu de données *RS-ASM (vc)* que *Food*. *ε-Greedy* obtient un résultat de diversité significativement supérieur à tout autre méthode pour *RS-ASM (vc)* tandis que c'est *EXP3* qui diversifie le mieux pour *Food*.

Jeux de données de recommandation de type conseil à un tiers humain (*Yeast*, *Adult*, *Mushroom*, *Statlog* et *Students Academics Performance*. Sur l'ensemble de ces jeux de données, les tests de *Wilcoxon* indiquent qu'il existe une différence significative entre les résultats obtenus pour chacun des algorithmes comparé deux à deux de *MAB* et de *CMAB* basés sur un modèle linéaire ($p < 0,01$, H_0 est rejetée). Ceux-ci obtiennent ainsi une diversité supérieure aux algorithmes de *MAB* sur chacun de ces jeux de données (voir les tableaux C.2 et C.1). *EXP4.P* quant à lui n'offre pas de différence statistiquement significative avec les algorithmes de *MAB* sur l'ensemble de ces jeux de données ($p > 0,62$, H_0 est acceptée, test de *Wilcoxon*) sauf avec ε-Greedy qui diversifie mieux qu'*EXP4.P* sur l'ensemble des jeux de données ($p < 0,01$, H_0 est rejetée, test de *Wilcoxon*). Il n'existe cependant pas de différence significative entre les résultats de diversité obtenues par *LinUCB* et *CTS* sur l'ensemble des jeux de données ($p > 0,51$, H_0 est acceptée, test de *Wilcoxon*).

Le test de *KW* appliqué pour la catégorie *MAB* indique qu'il existe également une différence significative entre les résultats obtenus par chacun des algorithmes d'une même catégorie ($p < 0,01$, H_0 est rejetée) et ce pour chacun des jeux de données (*Yeast, Adult, Mushroom, Statlog et Students Academics Performance*). Ainsi dans tous les cas, on préférera utiliser *ε-Greedy* qui obtient des résultats de diversité significativement supérieur aux autres algorithmes de *MAB*.

Jeux de données de montée à l'échelle (*Covertype* et *Poker Hand*). Il est important d'observer quels résultats nous obtenons concernant la diversité quand les algorithmes passent à l'échelle.

Sur le jeu de données *Poker Hand*, les tests de *Wilcoxon* réalisés entre chacun des algorithmes comparé deux à deux de *MAB* et de *CMAB* indiquent qu'il n'existe qu'une seule différence significative entre les résultats obtenus en termes de diversité avec l'algorithme ε-*Greedy* qui obtient une diversité supérieure aux autres méthodes ($Div(\varepsilon - Greedy) = 0,2$ $p < 0,01$, H_0 est rejetée). Dans les autres cas, *LinUCB* et *CTS* obtiennent une diversité supérieure aux algorithmes de *MAB* et également par rapport à *EXP4.P* ($p < 0,01$, H_0 est rejetée). Néanmoins, dans un cadre applicatif cette significativité ne sera humainement pas perceptible pour un utilisateur puisque celle-ci s'explique du fait de l'ordre de grandeur de la diversité obtenue allant de 6.10^{-2} à 10^{-5} (voir les tableaux C.2 et C.1). Ceci s'explique par le fait que le modèle linéaire s'applique difficilement sur ce jeu de données c.-à-d., les variables de contexte sur lesquelles on s'appuie n'ont pas forcément de dépendance linéaire avec les récompenses et de ce fait ne sont pas suffisamment pertinentes. On pourrait même supposer que dans ce type de jeu de données (mains de Poker), il n'existe pas véritablement de contexte sur lequel s'appuyer permettant de prédire efficacement une valeur de la « *main* » de Poker qu'on obtiendra dans le jeu, outre celle, représentée par l'action : « *Nothing in hand* » (perdante), ou encore celle qui est certes gagnante, mais assez fréquente : « *One pair* ». Nous pouvons l'observer avec les algorithmes *LinUCB* et *CTS* qui ont obtenus des résultats de diversité les moins faibles de l'ensemble des algorithmes (outre $\varepsilon - Greedy$) avec $Div(LinUCB) = 0,06$ et $Div(CTS) = 0,06$, et qui ont le plus souvent choisi les deux actions suivantes : $94,76\%$ pour « *Nothing in hand* » et $4,73\%$ pour « *One pair* ». Les autres actions (bras) sont effectuées en proportion infime par rapport à ces deux actions principales c.-à-d., une proportion inférieure à $0,0007361\%$. Ainsi, bien que cela ne soit pas significatif, les algorithmes de *CMAB* basés sur un modèle linéaire restent avantageux uniquement du fait qu'ils sont capables, en s'appuyant du contexte, de diversifier un peu plus en prédisant l'action « *One pair* ». Les stratégies non contextuels quant à elles ne peuvent se contenter que de sélectionner le bras optimal « *Nothing in hand* » sans pouvoir plus diversifier leur choix. Notons que ε-*Greedy* même s'il offre une meilleure diversification reste malgré tout risqué dans le cas de *Poker Hand*. En effet, il semble que lorsque la différence de probabilité de récompense entre chaque action est très importante, la diversification aléatoire n'est pas la meilleure stratégie d'exploration à adopter.

Sur le jeu de données *Covertype*, les tests de *Wilcoxon* indiquent qu'il existe une différence significative entre les résultats obtenus pour chacun des algorithmes comparé deux à deux de *MAB* et de *CMAB* basés sur un modèle linéaire uniquement ($p < 0,01$, H_0 est rejetée). Ceux-ci obtiennent une diversité globale supérieure aux algorithmes de *MAB* ($Div(LinUCB) = 0,44$ et $Div(CTS) = 0,44$ sachant que le meilleur score obtenu par les algorithmes de MAB est $Div(\varepsilon$-Greedy$) = 0,19$, voir les tableaux C.2 et C.1). Pour *EXP4.P*, bien qu'on observe une diversité de $Div(EXP4.P) = 0,01$, significativement supérieure à *UCB2* et à *TS*, ($Div(UCB2) = 10^{-4}$, et $Div(TS) = 10^{-5}$,$p < 0,01$, H_0 est rejetée, test de *Wilcoxon*), elle n'est pas statistiquement significative comparée à celle obtenue par *EXP3* ($Div(EXP3) = 0,01$, $p > 0,65$, H_0 est acceptée, test de *Wilcoxon*) et est même significativement surpassée par ε-*Greedy* ($Div(\varepsilon$-Greedy$) = 0,19, p < 0,01$, H_0 est rejetée, test de *Wilcoxon*). *EXP4.P* reste néanmoins nettement contre-performant par rapport aux algorithmes de *CMAB* basés sur un modèle linéaire

($p < 0,01$, H_0 est rejetée). Enfin, on remarquera que sur le jeu de données *Covertype*, ε-*Greedy* obtient une diversité significativement supérieure à celles obtenues par l'ensemble des algorithmes de *MAB* ($p < 0,01$, H_0 est rejetée, test de *Wilcoxon*).

B.6 Analyses détaillées de la diversité : *MABs* versus *CMABs* sur jeux de données avec vecteur tronqué

Contrôle (vt). Sur le jeu de données *Contrôle (vt)*, les tests de *Wilcoxon* indiquent qu'il existe une différence significative entre les résultats obtenus pour chacun des algorithmes de *MAB* et *LinUCB* comparé deux à deux ($Div(LinUCB) = 0,88$, $p < 0,001$, H_0 est rejetée). Ainsi, même si *LinUCB* perd significativement en diversité due à la version tronquée du vecteur ($Div(LinUCB) = 0,88$ en (*vt*) contre $0,99$ en (*vc*), $p < 0,001$, H_0 est rejetée, test de *Wilcoxon*), il n'en reste pas des moins compétitifs en termes de diversité face aux algorithmes de *MAB*. En revanche, il apparaît que les autres algorithmes de *CMAB* régressent au niveau d'*UCB2* en termes de diversité ($Div(CTS) = 0,59$ et $Div(EXP4.P) = 0,59$, $p > 0,55$, H_0 est acceptée, test de *Wilcoxon*).

RS-ASM (vt). Sur le jeu de données *RS-ASM (vt)*, les tests de *Wilcoxon* indiquent qu'il existe une différence significative entre les résultats obtenus pour chacun des algorithmes de *MAB* et *CMAB* basés sur un modèle linéaire comparés deux à deux ($p < 0,01$, H_0 est rejetée).Il apparaît que les algorithmes de *CMAB* basés sur un modèle linéaire obtiennent ainsi une diversité significativement supérieure à celles obtenues par les algorithmes de *MAB* et ce malgré la perte d'information. En revanche, les algorithmes de *CMAB* résolvant notre problème de recommandation contraint par un vecteur de contexte tronqué (c.-à-d., restreint) diversifient moins que lorsqu'ils ont à disposition le vecteur complet. On observe ainsi une diminution statistiquement significative de la diversité ($p < 0,01$, H_0 est rejetée, tests de *Wilcoxon*), voir tableau C.2 et C.3. Notons que *CTS* obtient un résultat de diversité significativement supérieur à *LinUCB* ($Div(CTS) = 0,46$, $Div(LinUCB) = 0,33$, $p < 0,01$, H_0 est rejetée, tests de *Wilcoxon*). *EXP4.P* quant à lui n'offre pas de différence statistiquement significative avec *UCB2* et *TS* sur le jeu de données *RS-ASM (vt)* ($Div(EXP4.P) = 0,04, p > 0,22$, H_0 est acceptée). En revanche, *EXP3* et ε-*Greedy* le surpassent ($p < 0,01$, H_0 est rejetée, tests de *Wilcoxon*).

YSE (vt). Sur le jeu de données *YSE(vt)*, les tests de *Wilcoxon* indiquent qu'il existe une différence significative entre les résultats obtenus pour chacun des algorithmes de *MAB* et *CMAB* basés sur un modèle linéaire comparés deux à deux ($p < 0,01$, H_0 est rejetée). Il apparaît que les algorithmes de *CMAB* basés sur un modèle linéaire obtiennent ainsi une diversité significativement supérieure à celles obtenues par les algorithmes de *MAB* et ce malgré la perte d'information ($p > 0,65$, H_0 est acceptée, test de *Wilcoxon*). De même, les algorithmes de *CMAB* résolvant notre problème de recommandation contraint par un vecteur de contexte tronqué (c.-à-d., restreint) diversifient autant que lorsqu'ils ont à disposition le vecteur complet.

Ceci peut s'expliquer par le faible nombre d'actions disponibles (c.-à-d., deux) pour le jeu de données *YSE (vt)* et qui aurait pu observer un plus fort impact si le nombre d'actions avait été plus élevé (comme observé pour *Contrôle (vt)* ou *RS-ASM (vt)*). *EXP4.P* quant à lui n'offre pas de différence statistiquement significative avec *EXP3* et *TS* ($Div(EXP4.P) = 0,04, p > 0,22$, H_0 est acceptée) mais se fait significativement surpasser par *UCB2* et ε-*Greedy* ($p < 0,01$, H_0 est rejetée).

B.7 Analyses détaillées de la diversité : *MABs* versus *CMABs* dans le cas non-stationnaire

Sur le jeu de données non-stationnaire *RS-ASM (ns)* les tests de *Wilcoxon* indiquent qu'il existe une différence significative entre les résultats obtenus pour chacun des algorithmes de *MAB* et *CMAB* comparé deux à deux ($p < 0,001$, H_0 est rejetée). *LinUCB* obtient la meilleure performance en termes de diversité et représente celui qui résiste le mieux à la non stationnarité. Au niveau des algorithmes de *MAB*, ε-Greedy est celui qui diversifie le mieux en environnement non-stationnaire ($p < 0,001$, H_0 est rejetée).

De plus, on observe que pour l'ensemble des algorithmes de *CMAB*, la diversité reste similaire entre la version stationnaire et non stationnaire ($p > 0,6$, H_0 est acceptée). En revanche, les algorithmes de *MAB* comme *UCB2*, *EXP3* et ε-*Greedy* diversifient plus ($p < 0,01$, H_0 est rejetée). Ceci peut s'expliquer par le fait que le bras optimal change entre chaque phase de stationnarité provoquant ainsi une modification de sélection du bras optimal par les algorithmes de *MAB*.

B.8 Analyses détaillées de la diversité : *MABs* versus *CMABs* sur jeux de données dépourvu de contexte

Sur le jeu de données non contextuel *Jester*, les tests de *Wilcoxon* indiquent sans surprise qu'il existe une différence significative entre les résultats obtenus pour chacun des algorithmes de *MAB* et *CMAB* comparé deux à deux ($p < 0,001$, H_0 est rejetée). Les algorithmes de *MAB* obtiennent ainsi une diversité globale significativement inférieure aux algorithmes de *CMAB* dans le cas d'une restriction totale de contexte. Mais est-ce un résultat positif ? Les algorithmes de *CMAB* restant « *aveugles* », face à un contexte absent, régressent à une stratégie ne leur permettant pas de déterminer le ou les bras optimaux du problème. Ils restent ainsi piégés dans une stratégie de pure exploration sans jamais pouvoir exploiter.

Il est donc plus pertinent de s'intéresser uniquement aux algorithmes de *MAB*, où dans notre cas, *EXP3* offre une diversité supérieure aux autres algorithmes de *MAB* ($Div(EXP3) = 0,23, p < 0,01$, H_0 est rejetée, test de *Wilcoxon*).

B.9 Mémento d'aide à la lecture des valeurs et des figures concernant la précision individuelle.

Prenons en exemple l'algorithme *TS* sur le jeu de données *Contrôle* dans le tableau C.2. Pour interpréter les valeurs calculées, il faut procéder comme suit :

- si $Acc_u(T) = 0,09$ au décile 10% cela exprime qu'au moins 10% de la population obtient une précision individuelle inférieure ou égale à $0,09$;
- si $Acc_u(T) = 0,11$ à Q_1 cela exprime qu'au moins 25% de la population obtient une précision individuelle inférieure ou égale à $0,11$;
- si $Acc_u(T) = 0,14$ à la médiane (Med) cela exprime qu'au moins 50% de la population obtient une précision individuelle inférieure ou égale à $0,14$;
- si $Acc_u(T) = 0,29$ à Q_3 cela exprime qu'au moins 75% de la population obtient une précision individuelle inférieure ou égale à $0,29$;
- si $Acc_u(T) = 0,61$ au décile 90% cela exprime qu'au moins 90% de la population obtient une précision individuelle inférieure ou égale à $0,61$.

Connaissant les différents déciles/quartiles, il est donc possible de déduire les intervalles de précision individuelle et de connaître la proportion de la population qui y est inclus.

Si nous prenons un second exemple dans le Tableau C.1. Concernant l'algorithme *EXP4.P* sur le jeu de données de *Contrôle (vc)* on observera dès le premier décile (10%) que $Acc_u(T) = 0,98$ ce qui correspond au fait qu'au moins 10% de la population obtient une précision individuelle inférieure ou égale à $0,98$. Ce qui est bien plus élevé que ce que nous avions obtenu avec l'algorithme *TS* cité précédemment en exemple sur ce même jeu de données.

Ces observations doivent être complétées graphiquement avec la fonction de distribution cumulative (FDC) facilitant d'autant plus leur interprétation. Afin de compléter les analyses de l'exemple précédent où dans le Tableau C.1 nous avons observé que *TS* obtenait plus d'insatisfaits qu'*EXP4.P* au décile 10%, nous analysons donc la Figure B.1. La FDC concernant *EXP4.P* nous permet ainsi d'observer que celui-ci obtient des valeurs de précisions individuelles pour 10% de la population comprises entre $0,9$ et $0,98$ alors que pour *TS* la FDC augmentent rapidement jusqu'à 75% de la population pour des valeurs de précision individuelle comprises entre 0 et $0,29$.

Notons ainsi que les FDC représentent en continue, ce que les valeurs calculées dans les Tableaux C.2 et C.1 nous permettent d'observer par déciles et quartiles. De ce fait, les FDC sont plus précises. C'est pourquoi nous baserons nos prochaines analyses plus particulièrement sur l'observation des FDC et compléterons (si besoin) par les valeurs de déciles et quartiles calculées.

B.10 Analyses détaillées de la précision individuelle : *MABs* versus *CMABs* sur jeux de données avec vecteur complet

Jeux de données artificiels (*Contrôle (vc)* et *YSE (vc)*). À la Figure B.1 représentant la distribution de la précision individuelle pour le jeu de données *Contrôle (vc)*, on observe pour chaque algorithme de *CMAB* que l'ensemble de la population obtient, sans surprise, une précision individuelle très proche ou égale à $1,00$ (c.-à-d., $Acc_u(T) \approx 1,00$, $\forall u \in \mathcal{U}$). En effet, le contexte fourni en entrée étant optimal, les algorithmes de *CMAB* parviennent à différencier chaque action à sélectionner pour chaque utilisateur.

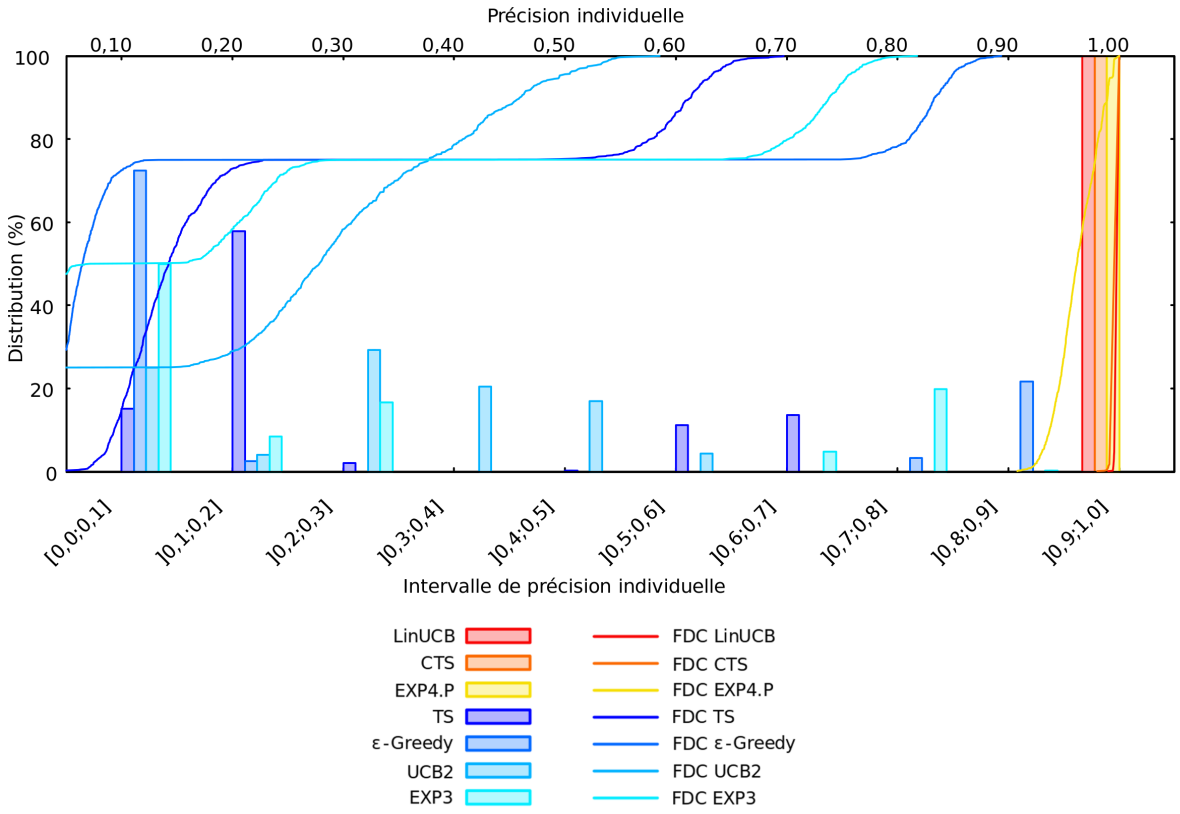


FIGURE B.1 – FDC de différents algorithmes de *MAB* et *CMAB* sur le jeu de données de *Contrôle* (Vecteur complet (*vc*) et optimal)

En revanche, on observe que les algorithmes de *MAB*, ne parviennent pas à satisfaire à 100% à l'ensemble de la population et au contraire atteignent jusqu'à 75% de la population dont la précision des recommandations est comprise entre 0 et $0,1$ (Voir Figure B.1). En effet, les précisions globales obtenues par ces algorithmes ne dépassent pas $0,251$ puisqu'ils ne sont capables que de déterminer le bras optimal à exploiter sans prise en considération du contexte. D'autre part, pour une même précision globale, les quatre algorithmes de *MAB* obtiennent

pourtant une distribution de la précision individuelle complètement différente. On remarquera ainsi Figure B.1 que *TS* provoque le moins d'insatisfaits c.-à-d., dont la précision des recommandations est comprise entre 0 et $0,1$ ($Acc_u(T) = 0,09$ au décile 10%, $Acc_u(T) = 0,11$ à Q_1), suivi d'*UCB2* ($Acc_u(T) = 0,00$ au décile 10%, $Acc_u(T) = 0,11$ à Q_1), *EXP3* ($Acc_u(T) = 0,02$ au décile 10%, $Acc_u(T) = 0,02$ à Q_1) et *ε-Greedy* ($Acc_u(T) = 0,03$ au décile 10%, $Acc_u(T) = 0,05$ à Q_1). Par la suite, on observera que *TS* obtient pour la majorité de la population, une précision individuelle comprise entre $0,1$ et $0,2$, puis entre $0,5$ et $0,7$. *UCB2* quant à lui obtiendra ensuite une précision individuelle dispersée entre $0,1$ et $0,6$. Ceci est dû à son mécanisme alternant exploration et exploitation du bras optimal. Enfin *EXP3* et *ε-Greedy* obtiendront une répartition de la précision individuelle allant jusqu'à respectivement $0,8$ et $0,9$. Ceci est dû à leur mécanisme d'exploration probabiliste (pour EXP3) et aléatoire (pour *ε-Greedy*).

À la Figure B.2 représentant la distribution de la précision individuelle pour le jeu de données *YSE (vc)*, on observe pour chaque algorithme de *CMAB* que l'ensemble de la population obtient une précision individuelle très proche ou égale à $1,00$ (c.-à-d., $Acc_u(T) = 1,00$ pour tout $u \in \mathcal{U}$). De même, le contexte fourni en entrée est optimal, les algorithmes de *CMAB* parviennent donc à différencier chaque action à sélectionner pour chaque utilisateur.

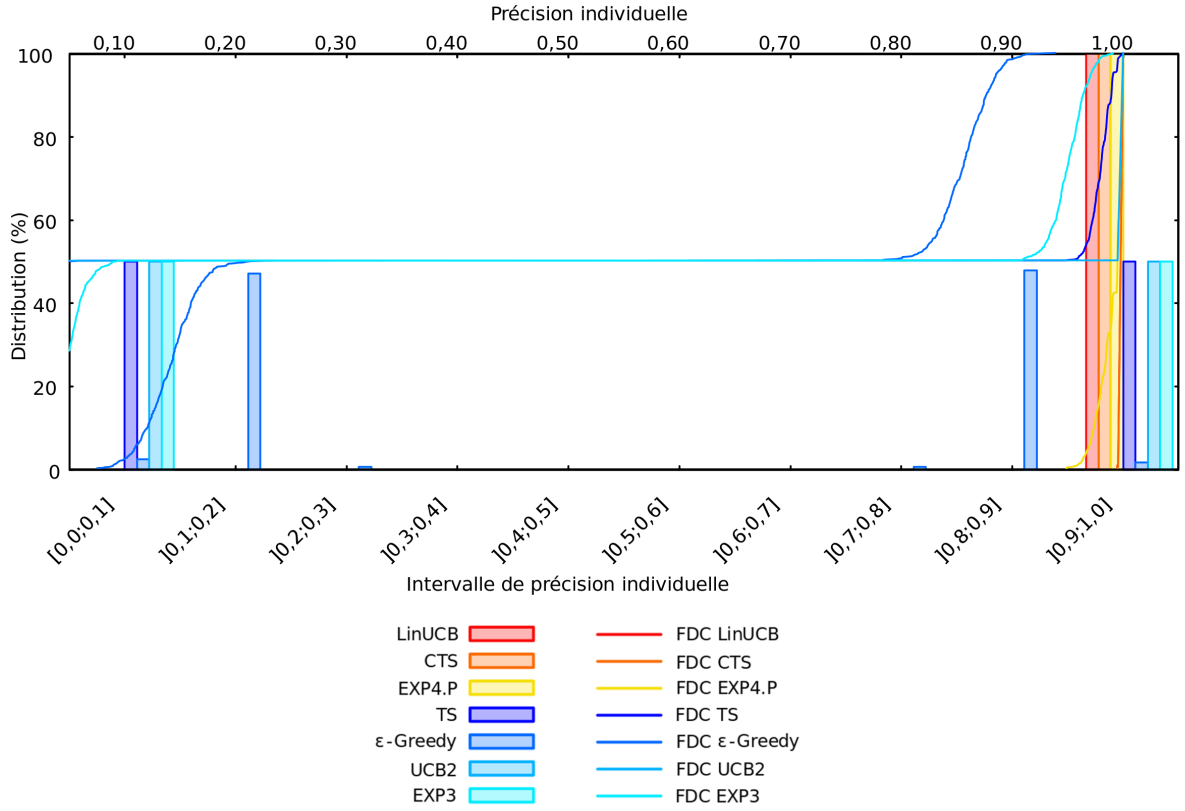


FIGURE B.2 – FDC de différents algorithmes de *MAB* et *CMAB* sur le jeu de données de *YSE* (Vecteur complet - *vc*)

Sur le jeu de données *YSE (vc)*, deux actions, de distribution de récompense équiprobable, sont possibles (c.-à-d., deux bras). Ainsi, les algorithmes de *MAB* ne sélectionnant que le bras optimal sans prise en compte du contexte, provoquent de surcroît une forte insatisfaction pour la moitié de la population (voir Figure B.2). La différence de répartition de la précision individuelle entre chaque algorithme de *MAB* sera donc sensiblement similaire et variera en fonction des mécanismes d'exploration propre à chacun. Ainsi nous observons que l'algorithme qui explore le plus ε-*Greedy* peut être considéré comme le meilleur des algorithmes de *MAB* d'un point de vue de précision individuelle. En effet, il provoque le moins d'insatisfaits c.-à-d., dont la précision des recommandations est comprise entre 0 et $0,1$. ε-*Greedy* obtiendra plutôt des résultats de précision compris entre $0,1$ et $0,2$. En contrepartie ε-*Greedy* n'obtiendra pas de très haut niveau de précision individuelle et plafonnera dans l'intervalle $]0,8; 0,9]$. D'un point de vue de la précision individuelle et pour un résultat de précision globale identique aux autres algorithmes de *MAB*, ε-*Greedy* peut être considéré comme plus équitable.

Jeux de données spécifiques à la recommandation (*RS-ASM (vc)* et *Food*). Concernant le jeu de données *RS-ASM (vc)*, on observe à la Figure B.3 représentant la distribution de la précision individuelle que les algorithmes de *CMAB* basés sur un modèle linéaire (*LinUCB* et *CTS*) obtiennent moins de forte insatisfaction qu'*EXP4.P* et que l'ensemble des algorithmes de *MAB*. Si on compare la FDC de *LinUCB* et *CTS* avec les FDC des autres algorithmes, on remarque que la répartition de la précision individuelle est plus équilibrée au sein des individus.

On observe également que *LinUCB* obtient une précision individuelle nettement meilleure que l'ensemble des algorithmes étudiés notamment *EXP4.P* et les algorithmes de *MAB* : $Acc_u(T) = 0,77$ à Q_1, $Acc_u(T) = 0,95$ pour la médiane (voir Tableau C.1). Sur la Figure B.3, on observe également que *LinUCB* est légèrement supérieure à son compétiteur le plus proche, *CTS* : $Acc_u(T) = 0,47$ à Q_1, $Acc_u(T) = 0,88$ pour la médiane (voir Tableau C.1).

Ainsi, pour une précision individuelle similaire sur l'intervalle $]0,9; 1,0]$ entre *UCB2*, *TS* et *LinUCB*, ce dernier obtient au final une meilleure précision globale car il répartit mieux ensuite la précision individuelle au sein de la population en observant notamment une moins grande proportion d'insatisfaits (intervalle $[0,0; 0,1]$). *CTS* suit cette même répartition équilibrée mais ne parvient pas au même niveau que *LinUCB* sur l'intervalle $]0,9; 1,0]$ ce qui le rend moins performant au regard du critère de précision individuelle (Acc_u).

En ce qui concerne les algorithmes de *MAB*, nous remarquerons qu'ils provoquent un clivage important entre individus insatisfaits (intervalle $[0,0; 0,1]$) et satisfaits (Intervalle $]0,9; 1,0]$) des recommandations qui leur sont faites. Due à l'absence de prise en compte du contexte, ceux-ci restent incapables de diversifier suffisamment sur l'ensemble des 18 actions disponibles dans ce jeu de données et ne sélectionnent ainsi qu'un seul bras optimal. En revanche, le mécanisme d'exploration aléatoire d'ε-*Greedy* lui permet d'un peut mieux répartir la précision individuelle mais ne suffit pas à atténuer la proportion d'insatisfaits sur l'intervalle de précision $[0,0; 0,1]$ (même constat pour *EXP3* et *EXP4.P*).

Concernant le jeu de données *Food*, on observe à la figure B.4 représentant la distribution de la précision individuelle que *LinUCB* et *CTS* obtiennent une précision individuelle très

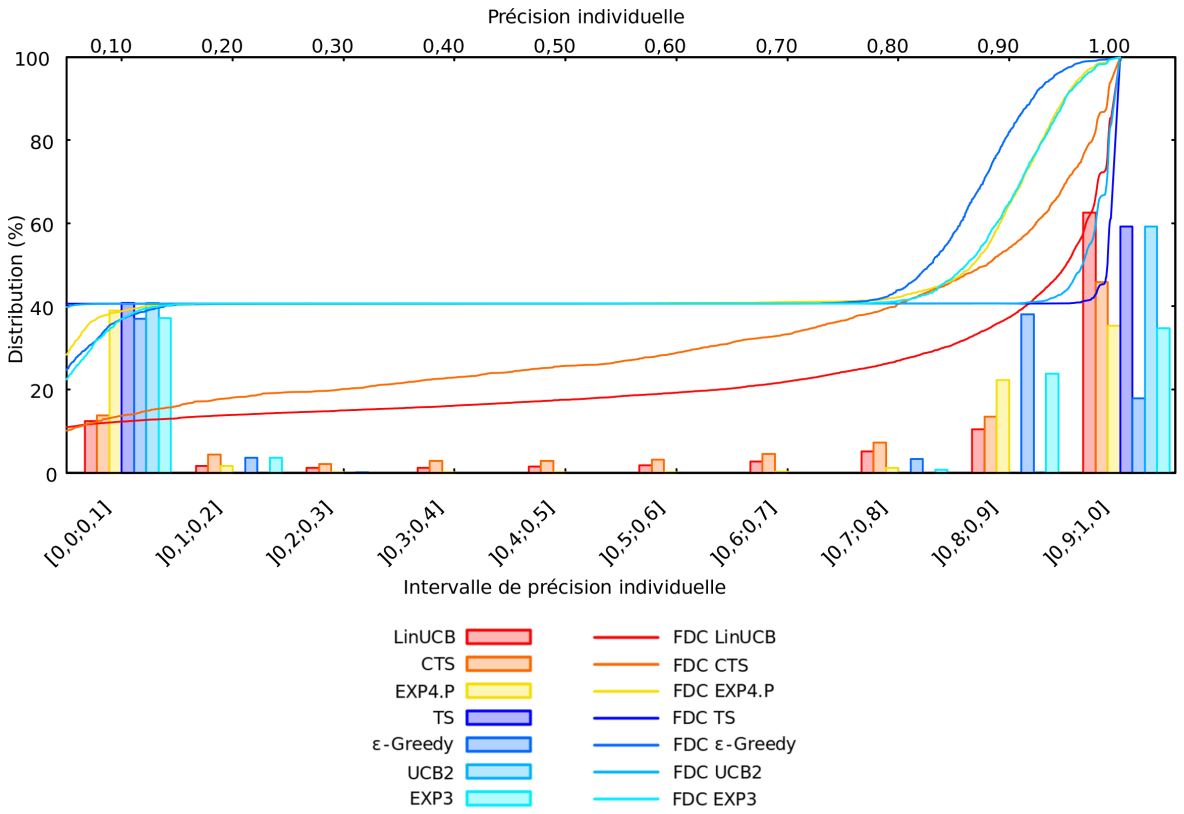


FIGURE B.3 – FDC de différents algorithmes de *MAB* et *CMAB* sur le jeu de données *RS-ASM* (Vecteur complet - *vc*)

proche ou égale à $1,00$ c.-à-d., $Acc_u(T) = 0,97$ pour le décile 10% (correspondant à 10% de la population), $0,98$ pour le quartile Q_1 (correspondant à 25% de la population), $0,99$ pour la médiane (correspondant à 50% de la population), et enfin $Acc_u = 1,00$ pour l'autre moitié de la population. Rappelons que nous avons complété le jeu de données *Food* par filtrage collaboratif afin d'obtenir un jeu de données complet en termes d'évaluations. Ceci revient indirectement à créer une dépendance entre contexte et évaluation propice aux bons résultats des algorithmes de *CMAB* basés sur un modèle qui parviennent ainsi à bien différencier chaque action à sélectionner pour chaque utilisateur.

En revanche, le résultat obtenu par *EXP4.P* reste à nuancer. Rappelons que celui-ci obtient la précision globale la plus faible de l'ensemble des méthodes évaluées. Or, nous observons qu'il répartit équitablement la précision individuelle au sein de la population selon une vision statistique. En effet, après un test de normalité, il apparaît que *EXP4.P* répartit la précision individuelle selon une distribution normale centrée sur $0,625$. Ainsi, certes *EXP4.P* n'obtient que très peu de population très satisfaite (Intervalle $]0,9;1,0]$) mais en contrepartie il n'obtient aucun insatisfaits (Intervalle $[0,0;0,1]$). D'un point de vue de la précision individuelle, ce résultat

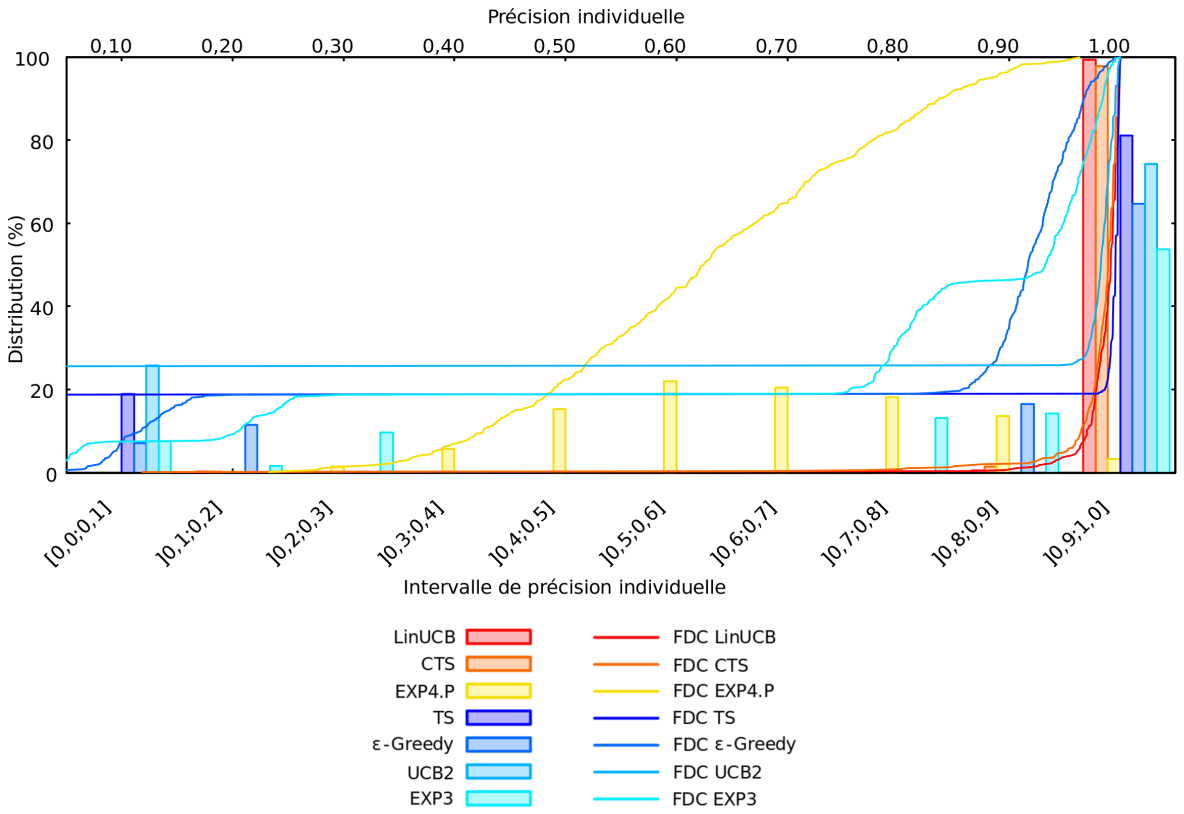


FIGURE B.4 – FDC de différents algorithmes de *MAB* et *CMAB* sur le jeu de données *Food*

est intéressant voire meilleure que les algorithmes de *MAB* qui obtiennent un clivage important entre fortement insatisfaits et fortement satisfaits. Notons que d'un point de vue utilitariste, cette distribution est la moins bonne.

Parmi les algorithmes de *MAB*, nous observons, à la différence du jeu de données *RS-ASM* qu'il y a un avantage à utiliser *EXP3* et ε*-Greedy* puisqu'ils obtiennent moins d'insatisfaits que *TS* et *UCB2*. En revanche, la contrepartie est d'accepter d'obtenir moins de fortement satisfaits.

Jeux de données de recommandation de type conseil à un tiers humain (*Yeast*, *Adult*, *Mushroom*, *Statlog* et *Students Academics Performance*. Concernant le jeu de données *Yeast*, on observe à la figure B.5 représentant la distribution de la précision individuelle que les algorithmes de *CMAB* basés sur un modèle linéaire (*LinUCB* et *CTS*) obtiennent à la fois plus d'individus satisfaits (Intervalle de précision individuelle $]0,9;1,0]$) et moins de forte insatisfaction (Intervalle de précision individuelle $[0,0;0,1]$) qu'*EXP4.P* et que l'ensemble des algorithmes de *MAB*. Concernant les algorithmes de *MAB*, ε*-Greedy* et *EXP3* obtiennent une moins forte insatisfaction qu'*UCB2* et *TS* ce qui coûte au niveau de la proportion de fortement

satisfaits et en termes de précision globale. On peut donc considérer qu'ε-*Greedy* et *EXP3* sont plus équitables sur la répartition de la précision au sein de la population.

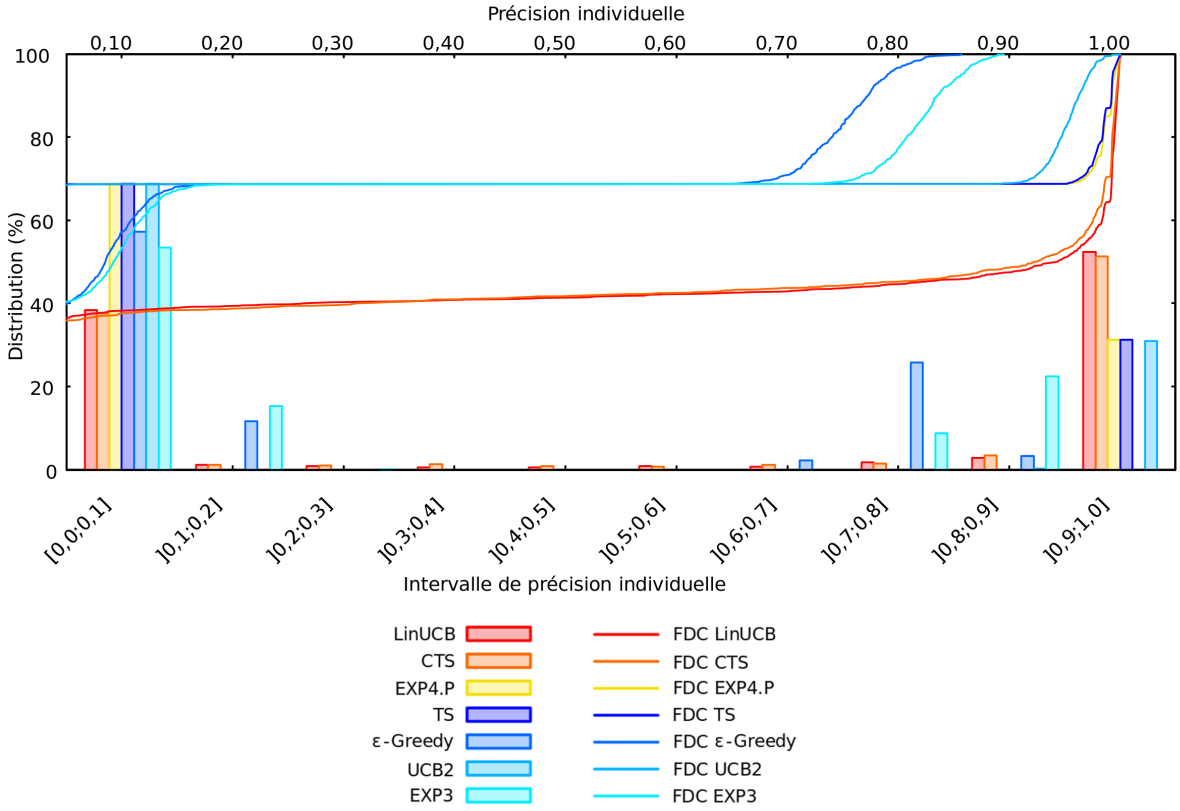


FIGURE B.5 – FDC de différents algorithmes de *MAB* et *CMAB* sur le jeu de données *Yeast*

Concernant le jeu de données *Adult*, on observe à la figure B.6 représentant la distribution de la précision individuelle que les algorithmes de *CMAB* basés sur un modèle linéaire (*LinUCB* et *CTS*) obtiennent légèrement plus de très satisfaits (Intervalle de précision individuelle $]0,9;1,0]$) qu'*EXP4.P* et que l'ensemble des algorithmes de *MAB*, excepté ε-*Greedy* qui est contre-performant. De même, ils obtiennent moins d'insatisfaction (Intervalle de précision individuelle $[0,0;0,1]$) qu'*EXP4.P* et que l'ensemble des algorithmes de *MAB* excepté ε-*Greedy* qui est au même niveau et qui restera meilleur que les autres algorithmes de (*MAB*) en termes de précision individuelle jusqu'à l'intervalle $]0,4;0,5]$.

Concernant le jeu de données *Mushrooms*, on observe à la Figure B.7 représentant la distribution de la précision individuelle que pour *LinUCB* et *CTS* l'ensemble de la population obtient une précision individuelle très proche ou égale à $1,00$ (c.-à-d., $Acc_u(T) \approx 1,00$ pour tout $u \in U$). Le contexte fourni en entrée de ce jeu de données semble de ce fait optimal pour des algorithmes basés sur une dépendance linéaire entre caractéristiques du contexte et

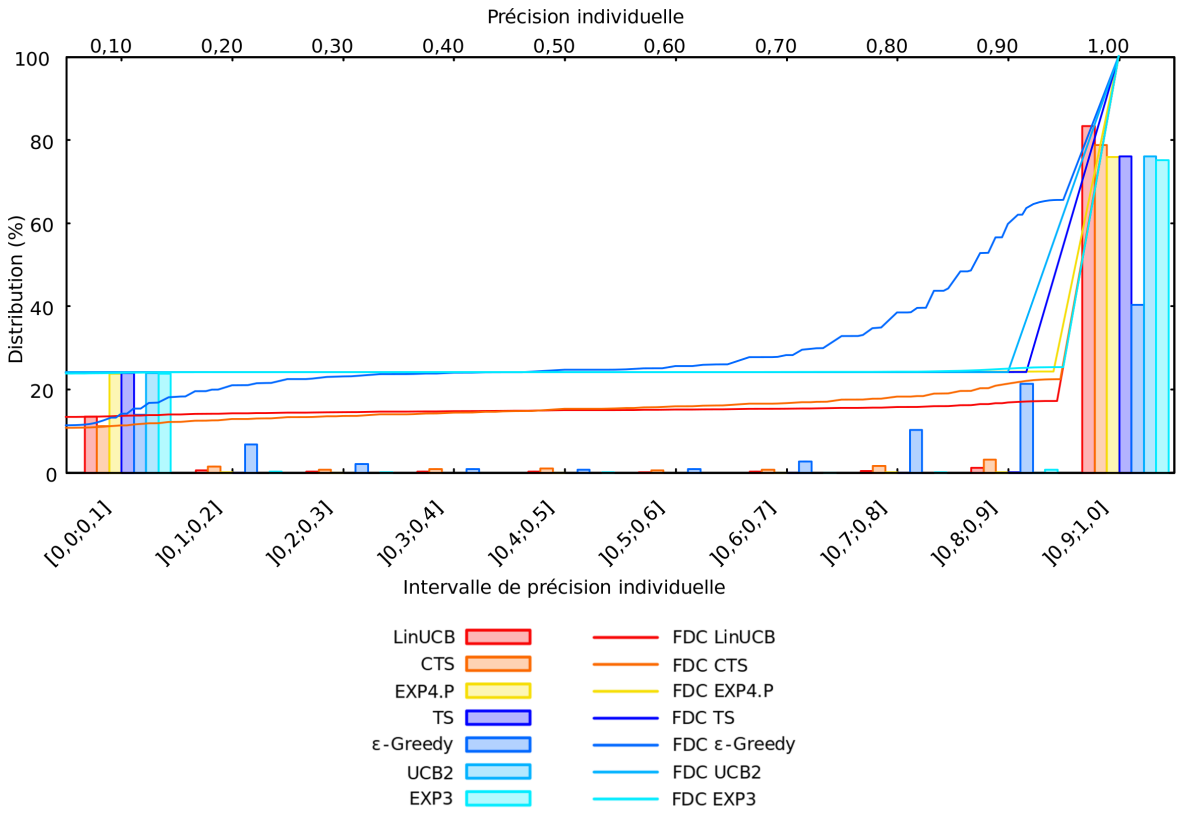


FIGURE B.6 – FDC de différents algorithmes de *MAB* et *CMAB* sur le jeu de données *Adult*

récompenses obtenues pour chaque action. Ainsi *LinUCB* et *CTS* parviennent à différencier chaque action à sélectionner pour chaque utilisateur.

En revanche, on observe que les algorithmes de *MAB* ainsi que *EXP4.P*, ne parviennent pas à satisfaire à 100% de la population et au contraire atteignent jusqu'à 48% de la population dont la précision est comprise entre 0 et $0,1$ (Voir Figure B.7). Parmi l'ensemble de ces algorithmes on observera de nouveau la capacité d'*ε-Greedy* à obtenir moins d'insatisfaits et de mieux répartir la précision individuelle et ce pour un coût moindre sur la précision globale ($-0,01$ de précision globale par rapport aux autres algorithmes de sa catégorie).

Concernant le jeu de données *Statlog*, on observe à la figure B.8 représentant la distribution de la précision individuelle que les algorithmes de *CMAB* basés sur un modèle linéaire (*LinUCB* et *CTS*) obtiennent à la fois plus de très satisfaits (Intervalle de précision individuelle $]0,9;1,0]$) et nettement moins de forte insatisfaction (Intervalle de précision individuelle $[0,0;0,1]$) qu'*EXP4.P* et que l'ensemble des algorithmes de *MAB*. Parmi l'ensemble de ces algorithmes on observera qu'*ε-Greedy* obtient un peu moins d'insatisfaits et répartit la précision individuelle de manière plus équitable et ce pour un coût de précision globale de $-0,013$ par rapport au meilleur algorithme de *MAB* sur ce jeu de données (*TS*).

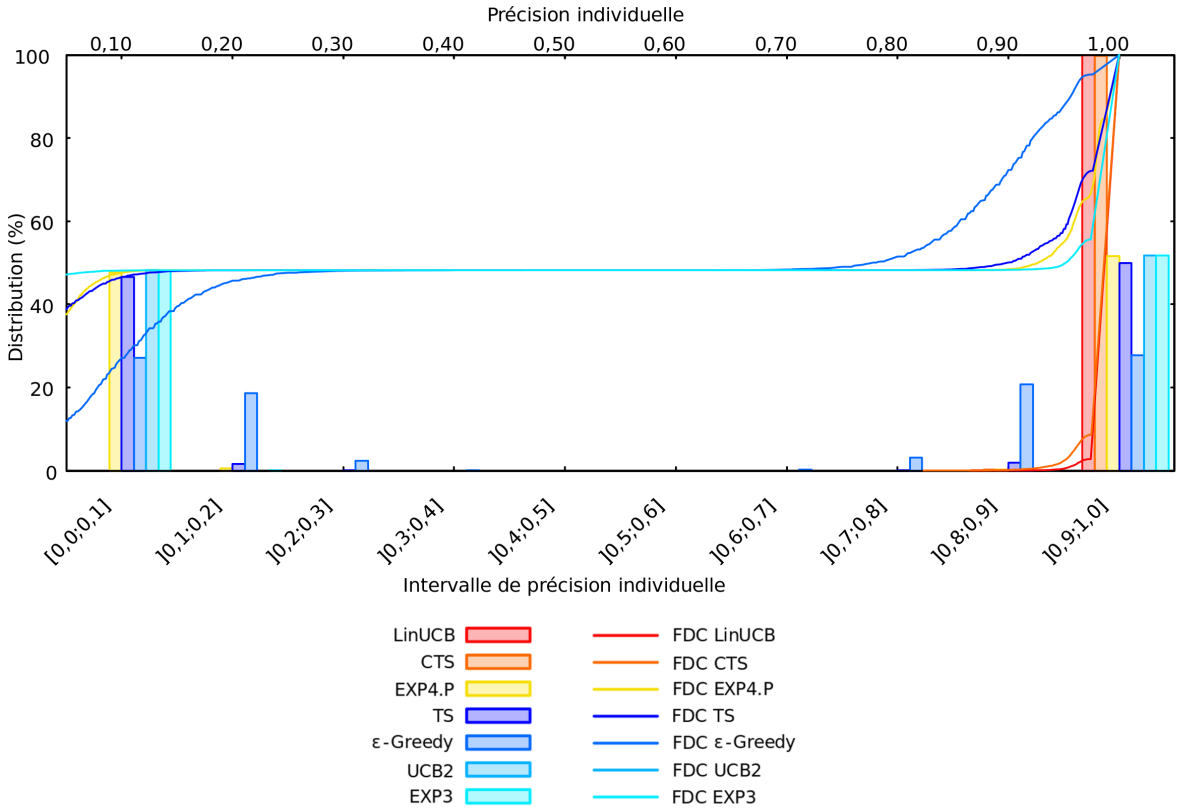


FIGURE B.7 – FDC de différents algorithmes de *MAB* et *CMAB* sur le jeu de données *Mushroom*

Concernant le jeu de données *Student Academics Performance*, on observe à la Figure B.9 représentant la distribution de la précision individuelle que les algorithmes de *CMAB* basés sur un modèle linéaire (*LinUCB* et *CTS*) obtiennent à la fois plus de très satisfaits (Intervalle de précision individuelle $]0,9;1,0]$) et nettement moins de forte insatisfaction (Intervalle de précision individuelle $[0,0;0,1]$) qu'*EXP4.P* et que l'ensemble des algorithmes de *MAB*. Parmi l'ensemble de ces algorithmes on observera pas d'avantage à utiliser l'un ou l'autre pour favoriser la précision individuelle. À noter de plus que cette fois, ε-*Greedy* sera même contre-performant par rapport aux autres algorithmes.

Jeux de données de montée à l'échelle (Covertype et Poker Hand). Concernant le jeu de données *Covertype*, on observe à la figure B.10 représentant la distribution de la précision individuelle que les algorithmes de *CMAB* basés sur un modèle linéaire (*LinUCB* et *CTS*) obtiennent à la fois plus de très satisfaits (Intervalle de précision individuelle $]0,9;1,0]$) et moins de forte insatisfaction (Intervalle de précision individuelle $[0,0;0,1]$) qu'*EXP4.P* et que l'ensemble des algorithmes de *MAB*. Parmi l'ensemble de ces algorithmes on observera cette fois un avan-

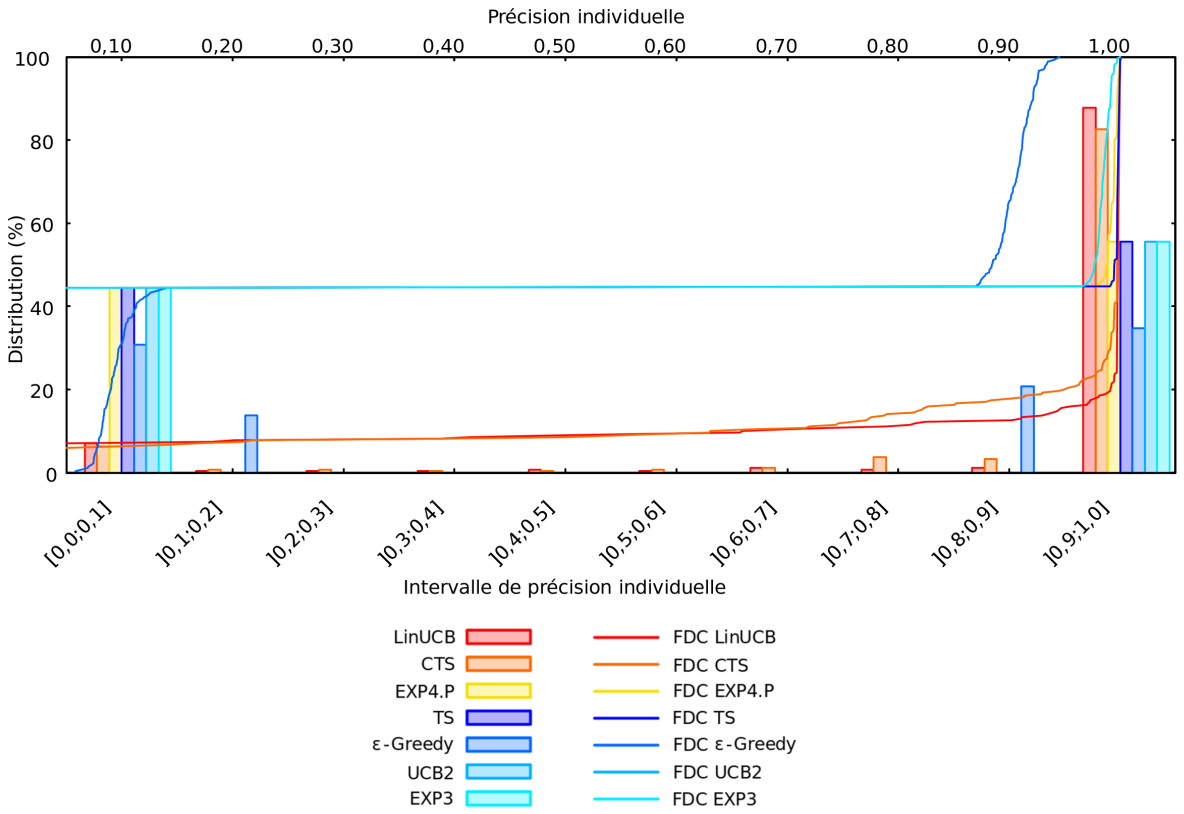


FIGURE B.8 – FDC de différents algorithmes de *MAB* et *CMAB* sur le jeu de données *Statlog*

tage à utiliser *UCB2* ou *TS*. Remarquons la contre-performance d'*EXP3* bien en dessous des autres algorithmes de *MAB*. Au même titre que pour le jeu de données Student Academics Performance, ε-*Greedy* sera également contre-performant par rapport aux autres algorithmes mais meilleur qu'*EXP3*.

Enfin, le jeu de données *Poker Hand* est intéressant à observer car dans ce cas il ne semble pas y avoir de nets avantages à utiliser un algorithme de *CMAB* en ce qui concerne la précision individuelle, a contrario des autres jeux de données vus jusqu'à présent. On observe à la Figure B.11 représentant la distribution de la précision individuelle que les proportions de très satisfaits et de très insatisfaits sont quasi équivalentes entre algorithmes de *CMAB* et de *MAB* (excepté pour ε-*Greedy* qui est moins performant). Néanmoins, on peut observer un léger avantage à utiliser *LinUCB* qui obtient à la fois une proportion de très satisfaits à même hauteur que les algorithmes de *MAB* (ce qui n'est pas le cas pour *CTS*) et un peu moins de très insatisfaits. Notons néanmoins que *CTS* obtient légèrement moins d'insatisfaits que *LinUCB*.

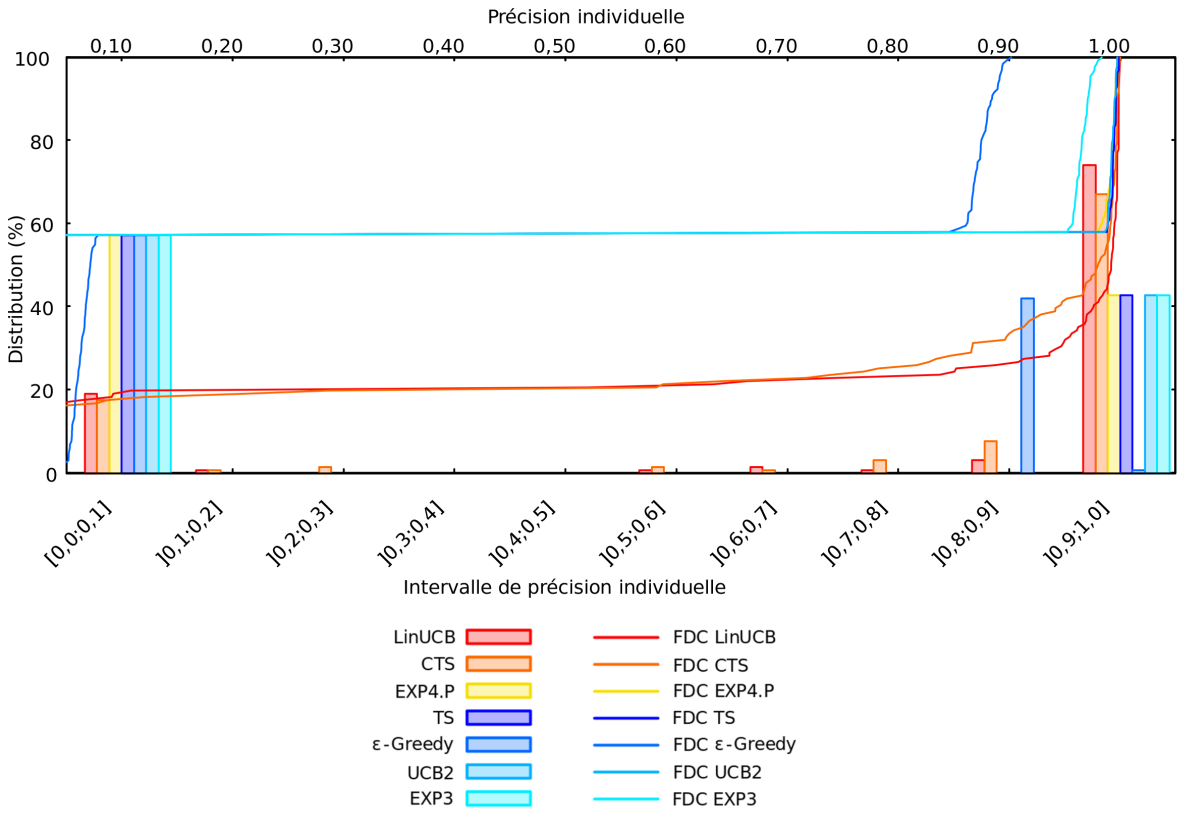


FIGURE B.9 – FDC de différents algorithmes de *MAB* et *CMAB* sur le jeu de données *Students Academics Performance*

B.11 Analyses détaillées de la précision individuelle : *MABs* versus *CMABs* sur jeux de données avec vecteur de contexte tronqué

Il est important d'observer l'effet de la restriction sur le contexte concernant les résultats de précision individuelle. Nous nous attendons à créer de l'insatisfaction pour certains groupes d'utilisateurs pour lesquels les caractéristiques du vecteur qui ont été tronquées étaient pertinentes pour le choix de certaines actions (bras). Ainsi, les algorithmes de *CMAB* devraient perdre certaines relations existantes entre récompenses et variables du contexte tronquées, devenant ainsi incapables de satisfaire aux groupes d'utilisateurs impactés. Notons que nous ne re-observerons pas les algorithmes de *MAB* car, étant non contextuels, la restriction de contexte n'a aucun effet sur eux. Les résultats obtenus par les algorithmes de *MAB* restent donc les mêmes que ceux décrit précédemment.

Concernant le jeu de données *Contrôle (vt)*, on observe sur la Figure B.12 que la perte d'information de contexte a un effet délétère sur la précision individuelle obtenue par les *CMAB*

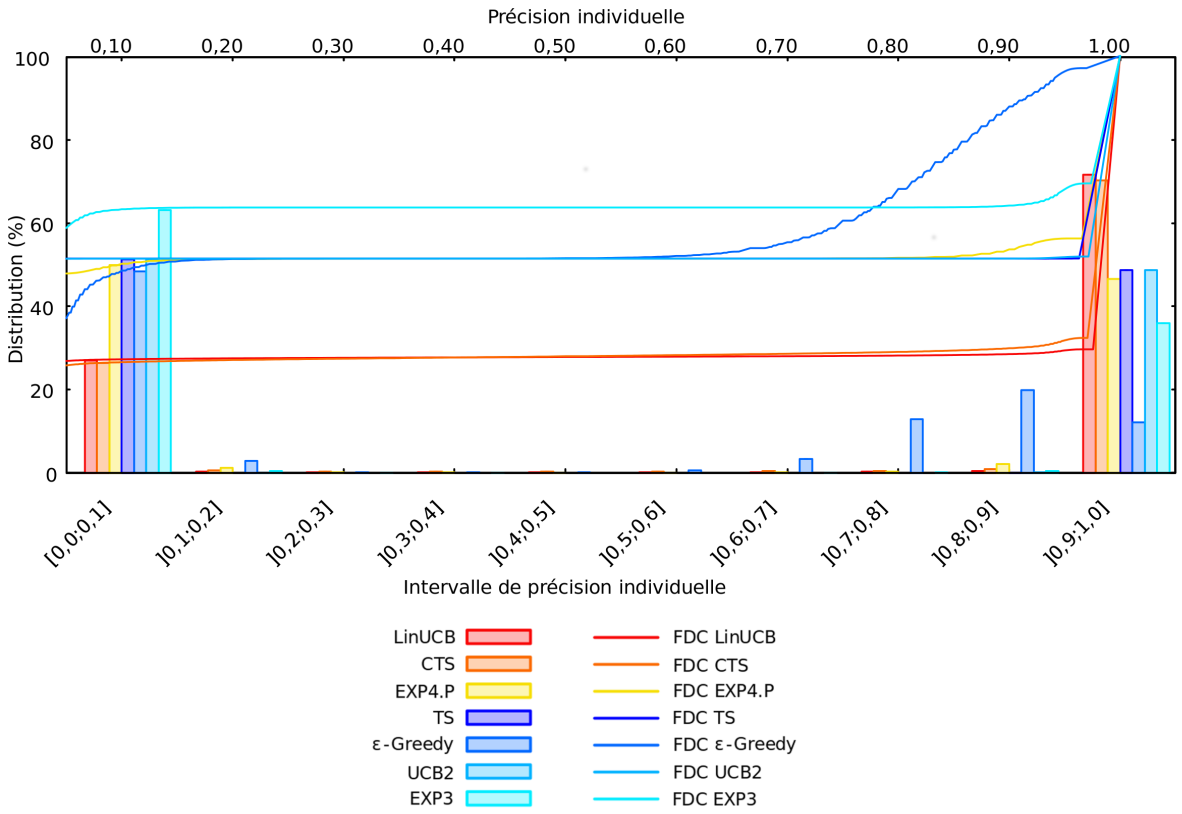


FIGURE B.10 – FDC de différents algorithmes de *MAB* et *CMAB* sur le jeu de données *Covertype*

versus celle obtenue quand le vecteur de contexte est complet. Néanmoins, les algorithmes de *CMAB* restent plus performant en termes de précision individuelle que les algorithmes de *MAB* car le taux de restriction (troncature) apportée au contexte n'est pas suffisant pour totalement le dégrader. Nous remarquons que *CTS* et *EXP4.P* obtiennent une proportion plus importante de très satisfaits (Intervalle de précision $]0,9;1,0]$) que *LinUCB*. En revanche *LinUCB* obtient une très faible proportion de très insatisfaits (Intervalle de précision $[0,0;0,1]$) proposant également une distribution de la précision individuelle plus équitable. Au regard du critère de précision individuelle, même si *LinUCB* obtient moins d'individus très satisfaits, pour une même précision globale que ses homologues, on remarque que la distribution de la précision individuelle est plus équilibrée. À ce titre, nous considérons que *LinUCB* est meilleur que *CTS* et *EXP4.P* sur ce critère de précision individuelle.

Concernant le jeu de données *RS-ASM (vt)*, sur la Figure B.13 on observe de nouveau que la perte d'information de contexte a un effet délétère sur la précision individuelle obtenue par les *CMAB* versus celle obtenue quand le vecteur de contexte est complet. De même, les algorithmes de *CMAB*, même s'ils conservent leur suprématie, ne semblent malgré tout plus

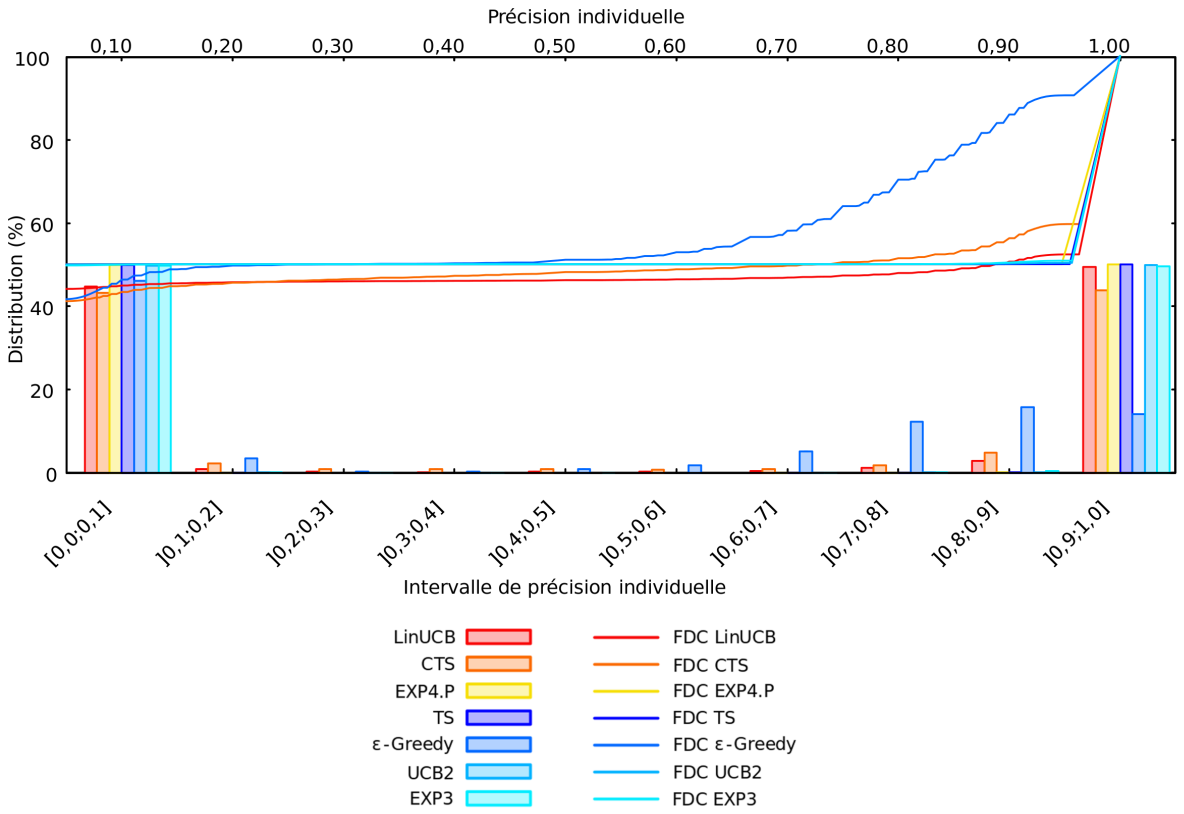


FIGURE B.11 – FDC de différents algorithmes de *MAB* et *CMAB* sur le jeu de données *Poker Hand*

avoir le même avantage flagrant en termes de précision individuelle par rapport aux algorithmes de *MAB*. On remarque ainsi que le taux de restriction sur le contexte (troncature) apportée a fortement dégradé la performance des algorithmes de *CMAB* si bien que ceux-ci sont proches du seuil où les algorithmes de *MAB* les surpassent.

Notons que nous ne remarquons aucune différence entre *CTS* et *LinUCB* sur ce critère de précision individuelle alors que, pour la forme (*vc*) du jeu de données, *LinUCB* était le plus performant. De ce fait, la restriction sur le contexte semble avoir un effet plus atténué sur *CTS* que sur *LinUCB*. Ceci peut s'expliquer par le mécanisme intrinsèque d'exploration de *CTS* qui, à chaque itération, bénéficie de bruit gaussien résultant de son processus de génération $\tilde{\theta}$ à partir de l'opérateur linéaire (vecteur de coefficients) estimé $\widehat{\theta}$.

Concernant le jeu de données *YSE (vt)*, on observe dans la Figure B.14 l'effet *Yule-Simpson* sur les algorithmes de *CMAB* provoqué par la perte d'information de contexte ciblée. L'effet *Yule-Simpson* a un effet délétère sur la précision individuelle obtenue par les *CMAB* versus celle obtenue quand le vecteur de contexte était complet puisque 40% de la population est désormais totalement insatisfaite (intervalle de précision $[0,0;0,5]$). Les algorithmes de *CMAB*

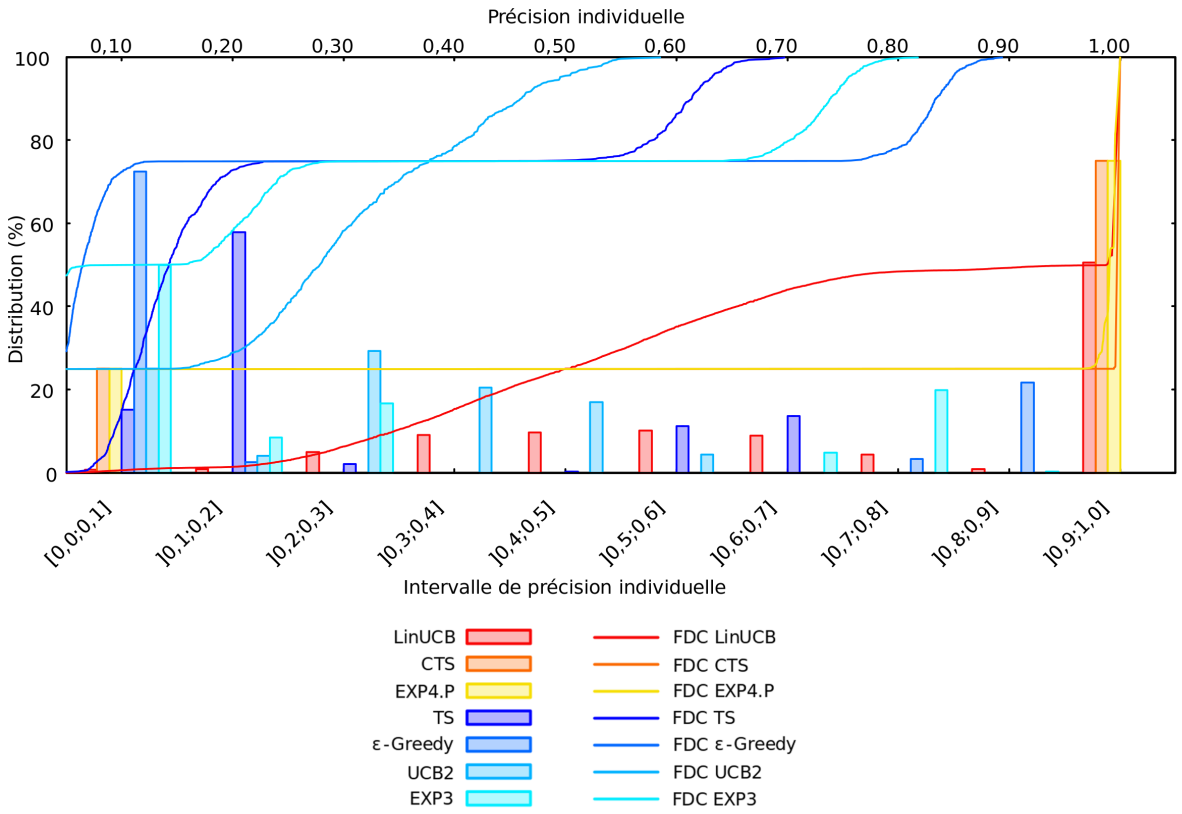


FIGURE B.12 – FDC de différents algorithmes de *MAB* et *CMAB* sur le jeu de données de Contrôle (Vecteur tronqué - *vt*)

sont incapables de rattacher ce groupe d'individus au bon salaire car il leur manque pour cela, la dimension « pays » primordiale pour réussir à choisir le bras pertinent. Néanmoins, les algorithmes de *CMAB* restent légèrement plus performants en termes de précision individuelle que les algorithmes de *MAB*. Cela reste à nuancer du fait de la nature même du jeu de données c.-à-d., la proportion d'hommes gagnants $50\ 000$ euros (300 sur 500), et la proportion de femmes gagnants $15\ 000$ euros (300 sur 500) est de 60%. Si la proportion avait été de 50% nous n'aurions observé aucune différence entre algorithmes de *MAB* et *CMAB*. Notons enfin que *LinUCB* permet d'obtenir une proportion de très satisfaits très légèrement supérieure à *CTS*.

B.12 Analyses détaillées de la précision individuelle : *MABs* Versus *CMABs* dans le cas non-stationnaire

Sur le jeu de données non-stationnaire *RS-ASM (ns)* on observe à la figure B.15 que les algorithmes de *CMAB* basés sur un modèle obtiennent une meilleure répartition de la précision

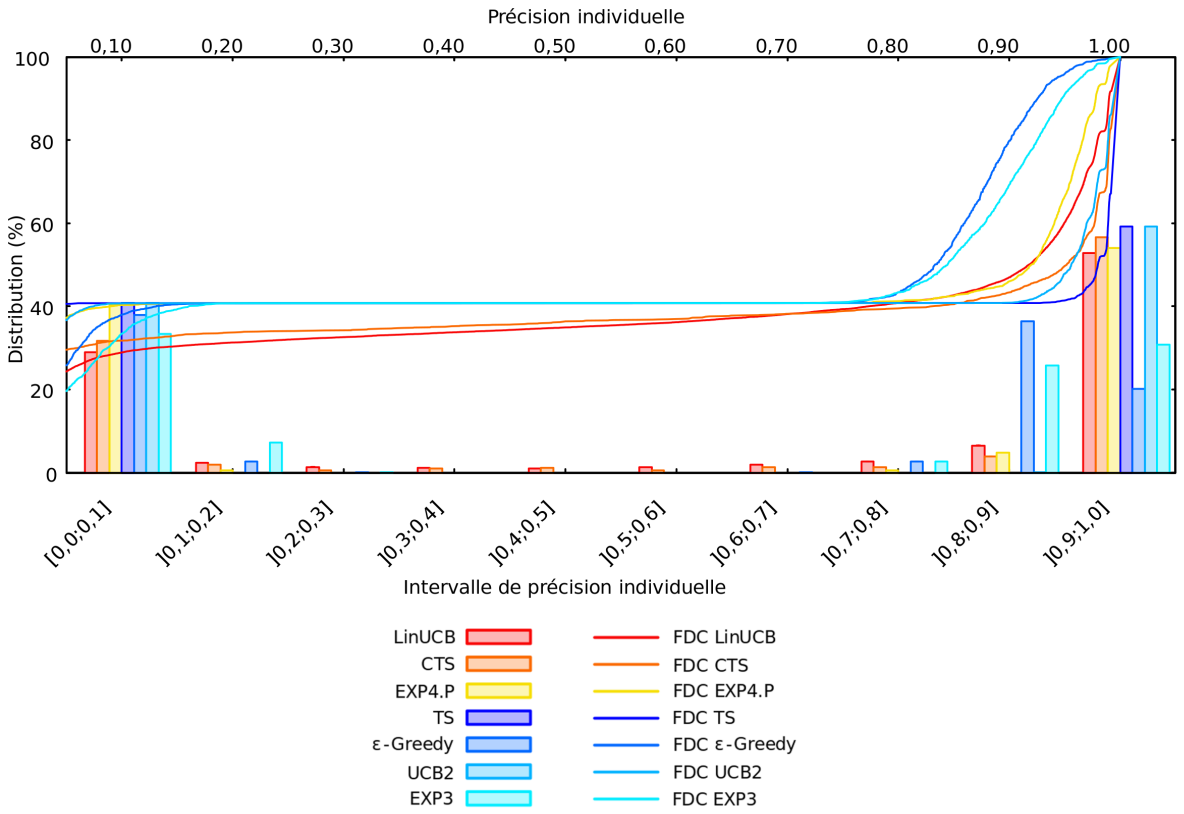


FIGURE B.13 – FDC de différents algorithmes de *MAB* et *CMAB* sur le jeu de données RSASM (Vecteur tronqué - *vt*)

individuelle que les algorithmes de *MAB*. *EXP4.P* quant à lui n'est toujours pas approprié sur ce jeu de données proposant un grand nombre d'actions avec des experts en désaccords. Notons qu'entre les deux meilleurs algorithmes (*LinUCB* et *CTS*), *LinUCB* obtient une meilleure précision individuelle que *CTS*. Parmi les algorithmes de *MAB*, c'est *UCB2* et *EXP3* qui semblent obtenir le meilleur compromis.

B.13 Analyses détaillées de la précision individuelle : *MABs* Versus *CMABs* sur jeux de données dépourvu de contexte

Sur le jeu de données non contextuel *Jester*, on observe sans surprise à la figure B.16 que les algorithmes de *MAB* sont plus performants en termes de précision individuelle que les algorithmes de *CMAB* dans le cas d'une restriction totale de contexte. Les algorithmes de *CMAB* restant « *aveugles* », face à un contexte absent, régressent à une stratégie ne leur permettant

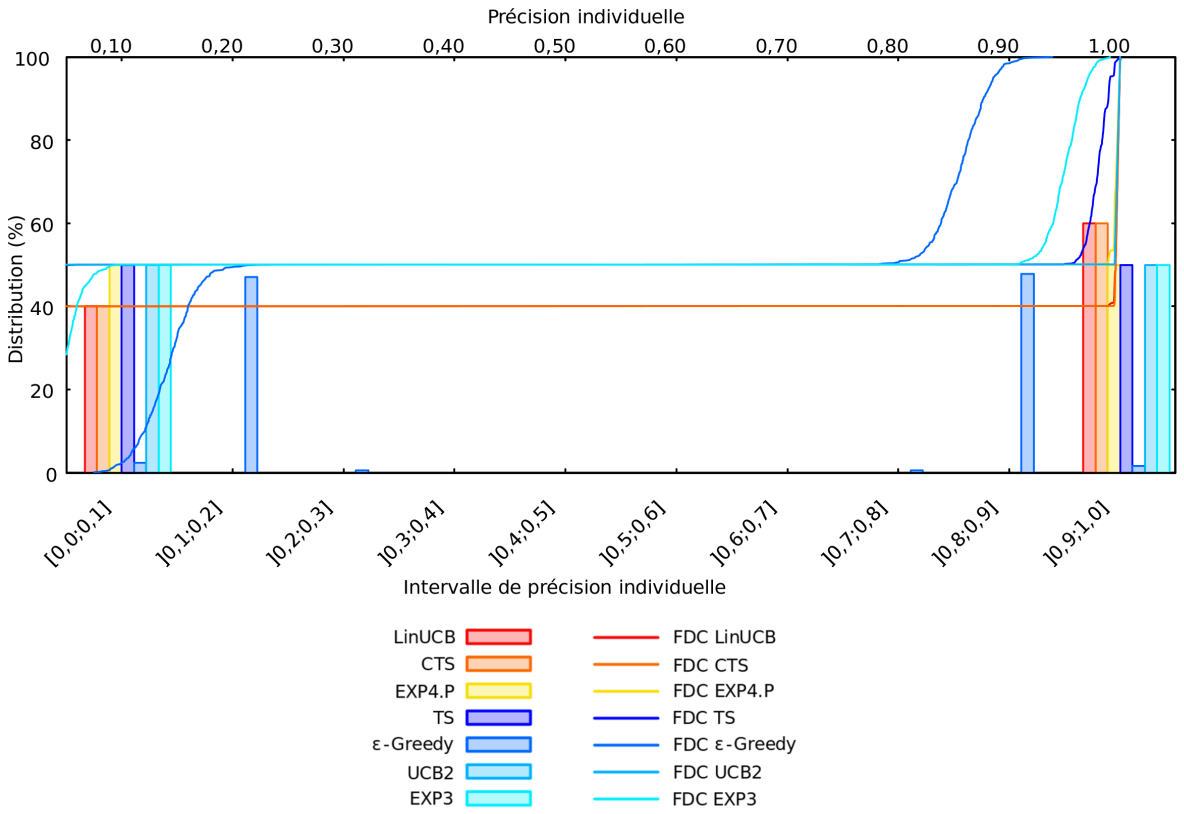


FIGURE B.14 – FDC de différents algorithmes de *MAB* et *CMAB* sur le jeu de données YSE (Vecteur tronqué - *vt*)

plus de déterminer le ou les bras optimaux du problème pour chaque groupe d'individus. Ils ne peuvent donc effectuer de recommandation précise pour chaque individu.

Ainsi, il est plus pertinent de s'intéresser uniquement aux algorithmes de *MAB* suivants : *TS* et *UCB2* qui offrent une précision individuelle supérieure aux autres algorithmes de *MAB* (Concernant les valeurs à la médiane $Acc_u(T) = 1,00$ pour *UCB2* et *TS* contre $Acc_u(T) = 0,89$ pour ε-*Greedy* et $Acc_u(T) = 0,75$ pour *EXP3*, voir Tableau C.5).

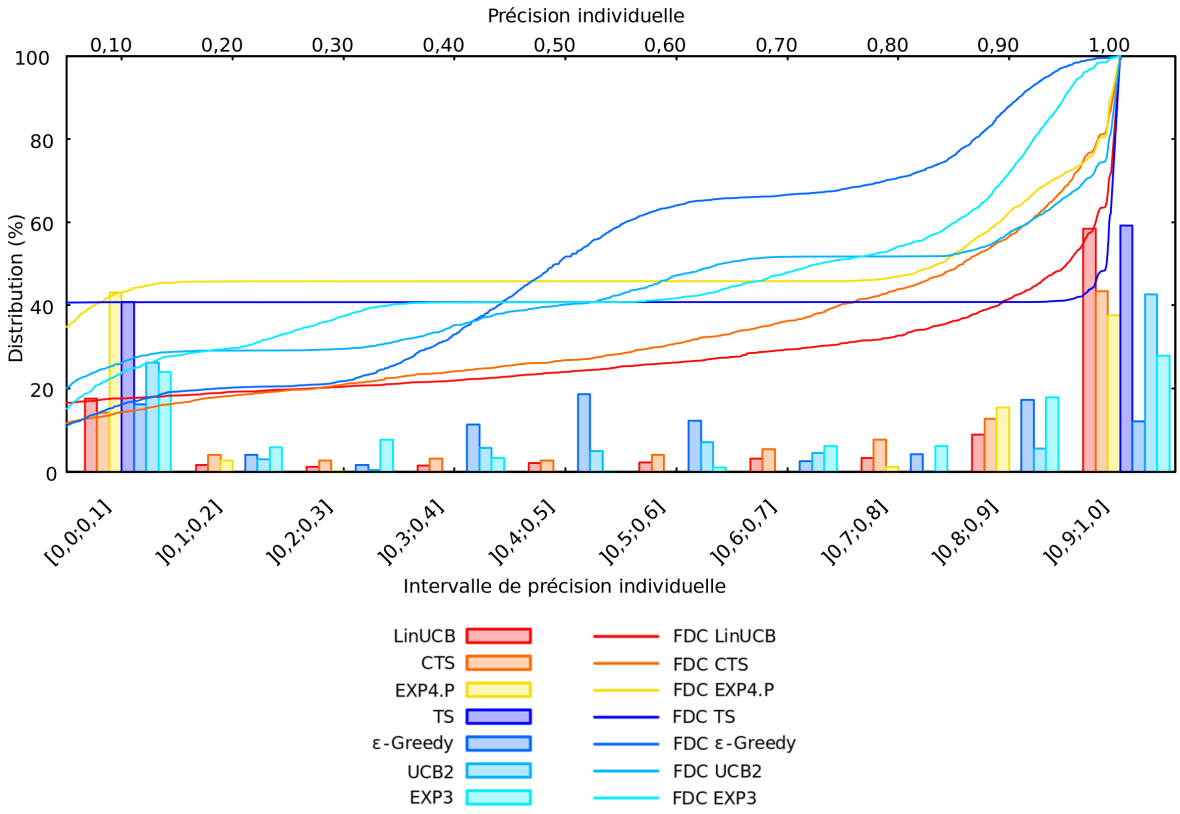


FIGURE B.15 – FDC de différents algorithmes de *MAB* et *CMAB* sur le jeu de données non contextuel *RS-ASM (ns)*

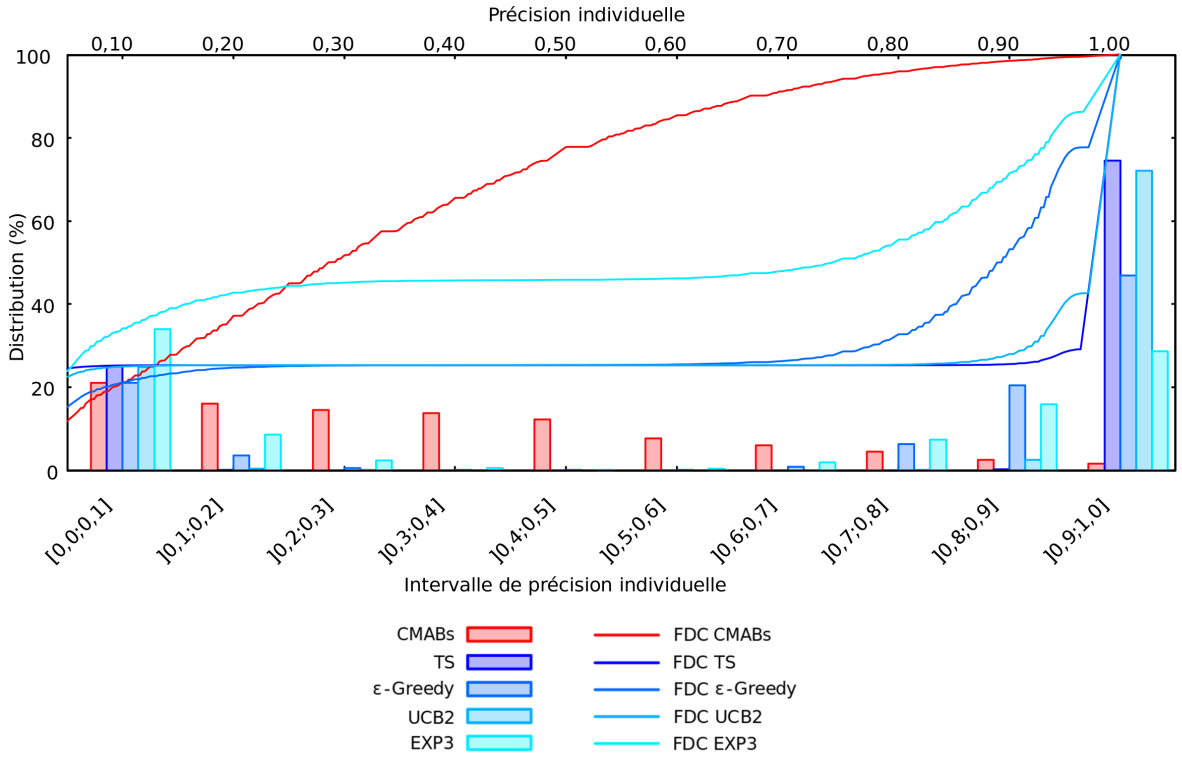


FIGURE B.16 – FDC de différents algorithmes de *MAB* et *CMAB* sur le jeu de données non contextuel *Jester*

ANNEXE C

ÉTUDE PRÉLIMINAIRE : TABLEAUX DE RÉSULTATS DE PRÉCISION GLOBALE, DIVERSITÉ ET DE PRÉCISION INDIVIDUELLE

		Mesures globales		Distribution de la Précision Individuelle				
		Acc	Div	10%	Q_1	Med	Q_3	90%
LinUCB	*Contrôle (vc)*	$1,0_{\pm\varepsilon}$	$0,997_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	YSE (vc)	$1,0_{\pm\varepsilon}$	$0,997_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	RS-ASM (vc)	$0,78_{\pm\varepsilon}$	$0,86_{\pm\varepsilon}$	$0,04_{\pm 0,02}$	$0,77_{\pm 0,02}$	$0,95_{\pm\varepsilon}$	$0,99_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	Poker Hand	$0,53_{\pm\varepsilon}$	$0,06_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$0,90_{\pm 0,01}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	Covertype	$0,72_{\pm\varepsilon}$	$0,44_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	Mushrooms	$0,998_{\pm\varepsilon}$	$0,96_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	Students	$0,78_{\pm 0,01}$	$0,78_{\pm 0,04}$	$0,00_{\pm\varepsilon}$	$0,86_{\pm\varepsilon}$	$0,99_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	Statlog	$0,90_{\pm\varepsilon}$	$0,85_{\pm\varepsilon}$	$0,72_{\pm 0,03}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	Yeast	$0,575_{\pm 0,001}$	$0,63_{\pm 0,001}$	$0,00_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$0,94_{\pm 0,01}$	$0,99_{\pm 0,01}$	$1,00_{\pm\varepsilon}$
	Adult	$0,85_{\pm\varepsilon}$	$0,36_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	Food	$0,98_{\pm\varepsilon}$	$0,88_{\pm\varepsilon}$	$0,97_{\pm\varepsilon}$	$0,98_{\pm\varepsilon}$	$0,99_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
CTS	*Contrôle (vc)*	$1,0_{\pm\varepsilon}$	$0,998_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	YSE (vc)	$1,00_{\pm\varepsilon}$	$0,99_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	RS-ASM (vc)	$0,70_{\pm\varepsilon}$	$0,79_{\pm\varepsilon}$	$0,05_{\pm\varepsilon}$	$0,47_{\pm\varepsilon}$	$0,88_{\pm\varepsilon}$	$0,96_{\pm\varepsilon}$	$0,99_{\pm\varepsilon}$
	Poker Hand	$0,52_{\pm\varepsilon}$	$0,06_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$0,75_{\pm 0,02}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	Covertype	$0,72_{\pm\varepsilon}$	$0,44_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	Mushrooms	$0,996_{\pm\varepsilon}$	$0,96_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	Students	$0,77_{\pm\varepsilon}$	$0,76_{\pm\varepsilon}$	$0,01_{\pm\varepsilon}$	$0,80_{\pm\varepsilon}$	$0,98_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	Statlog	$0,89_{\pm\varepsilon}$	$0,85_{\pm\varepsilon}$	$0,68_{\pm 0,02}$	$0,98_{\pm 0,01}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	Yeast	$0,575_{\pm 0,004}$	$0,62_{\pm 0,002}$	$0,00_{\pm\varepsilon}$	$0,01_{\pm 0,01}$	$0,93_{\pm 0,01}$	$0,99_{\pm 0,01}$	$1,00_{\pm\varepsilon}$
	Adult	$0,85_{\pm\varepsilon}$	$0,34_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	Food	$0,98_{\pm\varepsilon}$	$0,84_{\pm\varepsilon}$	$0,96_{\pm\varepsilon}$	$0,98_{\pm\varepsilon}$	$0,99_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
EXP4.P	*Contrôle (vc)*	$0,99_{\pm 0,001}$	$0,992_{\pm 0,002}$	$0,98_{\pm 0,01}$	$0,98_{\pm 0,01}$	$0,99_{\pm\varepsilon}$	$0,99_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	YSE (vc)	$0,99_{\pm\varepsilon}$	$0,998_{\pm\varepsilon}$	$0,97_{\pm\varepsilon}$	$0,98_{\pm\varepsilon}$	$0,99_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	RS-ASM (vc)	$0,57_{\pm\varepsilon}$	$0,1_{\pm\varepsilon}$	$0,01_{\pm 0,01}$	$0,02_{\pm 0,04}$	$0,92_{\pm 0,07}$	$0,95_{\pm 0,03}$	$0,98_{\pm 0,04}$
	Poker Hand	$0,5_{\pm\varepsilon}$	$10^{-5}{}_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	Covertype	$0,5_{\pm 0,01}$	$0,01_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$0,10_{\pm 0,05}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	Mushrooms	$0,52_{\pm 0,02}$	$0,01_{\pm\varepsilon}$	$0,01_{\pm 0,01}$	$0,03_{\pm 0,03}$	$0,93_{\pm 0,03}$	$0,98_{\pm 0,02}$	$1,00_{\pm\varepsilon}$
	Students	$0,43_{\pm 0,002}$	$0,01_{\pm 0,002}$	$0,00_{\pm 0,00}$	$0,00_{\pm 0,00}$	$0,01_{\pm 0,00}$	$0,99_{\pm 0,00}$	$0,99_{\pm 0,00}$
	Statlog	$0,55_{\pm\varepsilon}$	$0,01_{\pm\varepsilon}$	$0,01_{\pm 0,01}$	$0,01_{\pm 0,01}$	$0,99_{\pm 0,01}$	$0,99_{\pm 0,01}$	$1,00_{\pm\varepsilon}$
	Yeast	$0,30_{\pm 0,02}$	$0,01_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$0,98_{\pm\varepsilon}$	$0,99_{\pm\varepsilon}$
	Adult	$0,76_{\pm 0,001}$	$10^{-4}{}_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	Food	$0,63_{\pm 0,004}$	$0,93_{\pm\varepsilon}$	$0,43_{\pm 0,01}$	$0,52_{\pm 0,01}$	$0,62_{\pm\varepsilon}$	$0,75_{\pm\varepsilon}$	$0,84_{\pm\varepsilon}$

TABLE C.1 – Résultats des algorithmes de *CMAB* sur plusieurs jeux de données avec vecteur complet (*vc*)($\varepsilon = 0,0009$)

		Mesures globales		Distribution de la Précision Individuelle				
		Acc	Div	10%	Q_1	Med	Q_3	90%
UCB2	*Contrôle*	$0,25_{\pm 0,001}$	$0,59_{\pm 0,03}$	$0,00_{\pm \varepsilon}$	$0,11_{\pm 0,02}$	$0,28_{\pm 0,03}$	$0,38_{\pm 0,02}$	$0,46_{\pm 0,03}$
	YSE	$0,5_{\pm 0,01}$	$10^{-5}{}_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,50_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	RS-ASM	$0,59_{\pm 0,04}$	$0,04_{\pm 0,004}$	$0,00_{\pm \varepsilon}$	$0,01_{\pm \varepsilon}$	$0,97_{\pm 0,01}$	$0,99_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	Poker Hand	$0,5_{\pm \varepsilon}$	$10^{-4}{}_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,89_{\pm 0,02}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	Covertype	$0,49_{\pm 0,002}$	$10^{-4}{}_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	Mushrooms	$0,52_{\pm 0,003}$	$10^{-4}{}_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	Students	$0,43_{\pm 0,001}$	$0,01_{\pm 0,004}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,01_{\pm \varepsilon}$	$0,99_{\pm \varepsilon}$	$1,00_{\pm 0,01}$
	Statlog	$0,55_{\pm \varepsilon}$	$10^{-5}{}_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	Yeast	$0,30_{\pm \varepsilon}$	$0,05_{\pm 0,02}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,02_{\pm \varepsilon}$	$0,94_{\pm \varepsilon}$	$0,96_{\pm \varepsilon}$
	Adult	$0,76_{\pm \varepsilon}$	$10^{-6}{}_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	Food	$0,738_{\pm \varepsilon}$	$0,04_{\pm 0,002}$	$0,02_{\pm \varepsilon}$	$0,05_{\pm \varepsilon}$	$0,98_{\pm \varepsilon}$	$0,99_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
EXP3	*Contrôle*	$0,25_{\pm 0,001}$	$0,31_{\pm 0,02}$	$0,02_{\pm \varepsilon}$	$0,02_{\pm \varepsilon}$	$0,09_{\pm 0,02}$	$0,38_{\pm 0,02}$	$0,74_{\pm 0,03}$
	YSE	$0,5_{\pm 0,002}$	$0,06_{\pm 0,05}$	$0,00_{\pm 0,00}$	$0,03_{\pm 0,02}$	$0,50_{\pm \varepsilon}$	$0,97_{\pm 0,02}$	$0,98_{\pm 0,02}$
	RS-ASM	$0,56_{\pm 0,002}$	$0,16_{\pm 0,01}$	$0,02_{\pm 0,01}$	$0,06_{\pm 0,01}$	$0,86_{\pm 0,01}$	$0,92_{\pm \varepsilon}$	$0,95_{\pm \varepsilon}$
	Poker Hand	$0,5_{\pm \varepsilon}$	$0,002_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,86_{\pm 0,01}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	Covertype	$0,36_{\pm 0,01}$	$0,01_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	Mushrooms	$0,52_{\pm 0,003}$	$0,012_{\pm 0,001}$	$0,01_{\pm \varepsilon}$	$0,02_{\pm 0,01}$	$0,98_{\pm \varepsilon}$	$0,99_{\pm \varepsilon}$	$0,99_{\pm \varepsilon}$
	Students	$0,42_{\pm \varepsilon}$	$0,05_{\pm 0,01}$	$0,01_{\pm \varepsilon}$	$0,02_{\pm 0,01}$	$0,02_{\pm \varepsilon}$	$0,96_{\pm 0,01}$	$0,97_{\pm \varepsilon}$
	Statlog	$0,55_{\pm 0,003}$	$0,02_{\pm 0,007}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,25_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	Yeast	$0,29_{\pm 0,01}$	$0,18_{\pm 0,01}$	$0,01_{\pm \varepsilon}$	$0,01_{\pm \varepsilon}$	$0,08_{\pm 0,001}$	$0,81_{\pm 0,02}$	$0,85_{\pm 0,02}$
	Adult	$0,76_{\pm 0,001}$	$0,004_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,92_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	Food	$0,74_{\pm 0,03}$	$0,25_{\pm 0,01}$	$0,11_{\pm 0,10}$	$0,55_{\pm 0,33}$	$0,93_{\pm 0,03}$	$0,97_{\pm 0,01}$	$0,98_{\pm \varepsilon}$
TS	*Contrôle*	$0,25_{\pm 0,001}$	$0,46_{\pm 0,05}$	$0,09_{\pm \varepsilon}$	$0,11_{\pm \varepsilon}$	$0,14_{\pm \varepsilon}$	$0,29_{\pm \varepsilon}$	$0,61_{\pm 0,01}$
	YSE	$0,5_{\pm 0,02}$	$0,04_{\pm \varepsilon}$	$0,01_{\pm 0,01}$	$0,02_{\pm 0,02}$	$0,50_{\pm 0,05}$	$0,98_{\pm 0,02}$	$0,99_{\pm 0,01}$
	RS-ASM	$0,59_{\pm 0,002}$	$0,02_{\pm 0,005}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,99_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	Poker Hand	$0,5_{\pm 0,001}$	$10^{-5}{}_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	Covertype	$0,49_{\pm 0,003}$	$10^{-5}{}_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	Mushrooms	$0,52_{\pm 0,002}$	$0,001_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,03_{\pm 0,03}$	$0,90_{\pm 0,06}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	Students	$0,427_{\pm \varepsilon}$	$0,008_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,01_{\pm \varepsilon}$	$0,99_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	Statlog	$0,56_{\pm 0,002}$	$0,003_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,99_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	Yeast	$0,31_{\pm \varepsilon}$	$0,01_{\pm 0,005}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,01_{\pm 0,01}$	$0,98_{\pm \varepsilon}$	$0,99_{\pm 0,01}$
	Adult	$0,76_{\pm \varepsilon}$	$10^{-4}{}_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	Food	$0,81_{\pm \varepsilon}$	$0,01_{\pm \varepsilon}$	$0,01_{\pm \varepsilon}$	$0,99_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
ε*-Greedy*	*Contrôle*	$0,251_{\pm 0,001}$	$0,54_{\pm 0,02}$	$0,03_{\pm \varepsilon}$	$0,05_{\pm \varepsilon}$	$0,06_{\pm \varepsilon}$	$0,29_{\pm 0,01}$	$0,83_{\pm 0,02}$
	YSE	$0,5_{\pm 0,001}$	$0,25_{\pm 0,04}$	$0,11_{\pm 0,02}$	$0,12_{\pm 0,02}$	$0,50_{\pm \varepsilon}$	$0,87_{\pm 0,02}$	$0,89_{\pm 0,02}$
	RS-ASM	$0,54_{\pm 0,003}$	$0,2_{\pm 0,02}$	$0,02_{\pm \varepsilon}$	$0,05_{\pm \varepsilon}$	$0,83_{\pm \varepsilon}$	$0,89_{\pm \varepsilon}$	$0,92_{\pm \varepsilon}$
	Poker Hand	$0,42_{\pm 0,01}$	$0,2_{\pm 0,005}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,38_{\pm 0,03}$	$0,83_{\pm 0,03}$	$0,93_{\pm 0,02}$
	Covertype	$0,42_{\pm 0,002}$	$0,19_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,13_{\pm 0,01}$	$0,84_{\pm 0,02}$	$0,92_{\pm 0,02}$
	Mushrooms	$0,51_{\pm \varepsilon}$	$0,19_{\pm \varepsilon}$	$0,05_{\pm 0,01}$	$0,10_{\pm 0,01}$	$0,78_{\pm \varepsilon}$	$0,91_{\pm \varepsilon}$	$0,96_{\pm \varepsilon}$
	Students	$0,413_{\pm 0,003}$	$0,19_{\pm \varepsilon}$	$0,06_{\pm \varepsilon}$	$0,06_{\pm \varepsilon}$	$0,07_{\pm \varepsilon}$	$0,87_{\pm \varepsilon}$	$0,88_{\pm \varepsilon}$
	Statlog	$0,547_{\pm 0,002}$	$0,19_{\pm 0,001}$	$0,08_{\pm 0,01}$	$0,09_{\pm 0,01}$	$0,88_{\pm 0,01}$	$0,91_{\pm \varepsilon}$	$0,92_{\pm \varepsilon}$
	Yeast	$0,27_{\pm 0,001}$	$0,23_{\pm 0,004}$	$0,01_{\pm \varepsilon}$	$0,02_{\pm 0,01}$	$0,06_{\pm 0,04}$	$0,76_{\pm 0,04}$	$0,80_{\pm 0,04}$
	Adult	$0,71_{\pm \varepsilon}$	$0,19_{\pm \varepsilon}$	$0,00_{\pm \varepsilon}$	$0,60_{\pm \varepsilon}$	$0,88_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$	$1,00_{\pm \varepsilon}$
	Food	$0,77_{\pm \varepsilon}$	$0,19_{\pm \varepsilon}$	$0,11_{\pm 0,01}$	$0,88_{\pm \varepsilon}$	$0,92_{\pm \varepsilon}$	$0,95_{\pm \varepsilon}$	$0,97_{\pm \varepsilon}$

TABLE C.2 – Résultats des algorithmes de *MAB* sur plusieurs jeux de données ($\varepsilon = 0,0009$)

		Mesures globales		Distribution de la Précision Individuelle				
		Acc	Div	10%	Q_1	Med	Q_3	90%
LinUCB	*Contrôle (vt)*	$0,749_{\pm\varepsilon}$	$0,88_{\pm 0,07}$	$0,35_{\pm 0,09}$	$0,52_{\pm 0,04}$	$0,88_{\pm 0,04}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	RS-ASM (vt)	$0,629_{\pm\varepsilon}$	$0,33_{\pm 0,01}$	$0,01_{\pm\varepsilon}$	$0,06_{\pm 0,01}$	$0,92_{\pm 0,01}$	$0,97_{\pm\varepsilon}$	$0,99_{\pm\varepsilon}$
	YSE (vt)	$0,6_{\pm\varepsilon}$	$0,99_{\pm 0,01}$	$0,00_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
CTS	*Contrôle (vt)*	$0,751_{\pm 0,001}$	$0,59_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$0,75_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	RS-ASM (vt)	$0,63_{\pm 0,001}$	$0,46_{\pm 0,02}$	$0,00_{\pm\varepsilon}$	$0,02_{\pm\varepsilon}$	$0,95_{\pm\varepsilon}$	$0,99_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	YSE (vt)	$0,6_{\pm\varepsilon}$	$0,99_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
EXP4.P	*Contrôle (vt)*	$0,744_{\pm 0,01}$	$0,59_{\pm 0,01}$	$0,00_{\pm\varepsilon}$	$0,73_{\pm 0,02}$	$0,99_{\pm 0,01}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
	RS-ASM (vt)	$0,58_{\pm\varepsilon}$	$0,04_{\pm\varepsilon}$	$0,00_{\pm 0,02}$	$0,01_{\pm 0,03}$	$0,95_{\pm 0,07}$	$0,98_{\pm 0,05}$	$0,99_{\pm 0,03}$
	YSE (vt)	$0,5_{\pm\varepsilon}$	$0,004_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$0,50_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$

TABLE C.3 – Résultats des algorithmes de *CMAB* sur jeu de données avec vecteur tronqué (*vt*) ($\varepsilon = 0,0009$)

	Mesures globales		Distribution de la Précision Individuelle				
	Acc	Div	10%	Q_1	Med	Q_3	90%
UCB2	$0,586_{\pm 0,03}$	$0,30_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$0,09_{\pm 0,01}$	$0,64_{\pm 0,04}$	$0,99_{\pm 0,01}$	$1,00_{\pm\varepsilon}$
EXP3	$0,557_{\pm 0,02}$	$0,28_{\pm 0,01}$	$0,02_{\pm 0,01}$	$0,11_{\pm 0,03}$	$0,73_{\pm 0,03}$	$0,91_{\pm 0,01}$	$0,95_{\pm 0,01}$
TS	$0,592_{\pm 0,003}$	$0,01_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$0,99_{\pm 0,01}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
ε-Greedy	$0,516_{\pm 0,006}$	$0,55_{\pm\varepsilon}$	$0,04_{\pm 0,01}$	$0,34_{\pm 0,02}$	$0,49_{\pm 0,01}$	$0,84_{\pm 0,02}$	$0,91_{\pm 0,01}$
LinUCB	$0,73_{\pm\varepsilon}$	$0,85_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$0,55_{\pm\varepsilon}$	$0,95_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
CTS	$0,685_{\pm\varepsilon}$	$0,79_{\pm\varepsilon}$	$0,03_{\pm\varepsilon}$	$0,44_{\pm 0,\varepsilon}$	$0,86_{\pm\varepsilon}$	$0,97_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
EXP4.P	$0,52_{\pm\varepsilon}$	$0,11_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$0,01_{\pm\varepsilon}$	$0,84_{\pm\varepsilon}$	$0,97_{\pm\varepsilon}$	$0,99_{\pm\varepsilon}$

TABLE C.4 – Résultats des algorithmes de *MAB* et *CMAB* sur le jeu de données *RS-ASM (ns)* ($\varepsilon = 0,0009$)

	Mesures globales		Distribution de la Précision Individuelle				
	Acc	Div	10%	Q_1	Med	Q_3	90%
UCB2	$0,74_{\pm 0,003}$	$0,03_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$0,12_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
EXP3	$0,51_{\pm 0,01}$	$0,23_{\pm 0,01}$	$0,00_{\pm\varepsilon}$	$0,06_{\pm\varepsilon}$	$0,75_{\pm\varepsilon}$	$0,92_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
TS	$0,75_{\pm 0,003}$	$10^{-}3_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$0,08_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$	$1,00_{\pm\varepsilon}$
ε-Greedy	$0,70_{\pm 0,004}$	$0,14_{\pm 0,01}$	$0,00_{\pm\varepsilon}$	$0,27_{\pm 0,03}$	$0,89_{\pm 0,04}$	$0,95_{\pm 0,02}$	$1,00_{\pm\varepsilon}$
CMABs	$0,33_{\pm\varepsilon}$	$0,95_{\pm\varepsilon}$	$0,00_{\pm\varepsilon}$	$0,13_{\pm\varepsilon}$	$0,29_{\pm\varepsilon}$	$0,50_{\pm\varepsilon}$	$0,67_{\pm\varepsilon}$

TABLE C.5 – Résultats des algorithmes de *MAB* et *CMAB* sur le jeu de données *Jester* (non contextuel) ($\varepsilon = 0,0009$)

ANNEXE D

INFORMATIONS COMPLÉMENTAIRES AU PROJET *Event-AI*

D.1 Extrait anonymisé d'un fichier journal de connexions *Wifilib*

Exemple de fichier journal de connexions anonymisées au format *JSON*


```
{
  "materiel": {
        "typeOs": "iOS",
        "materielId": 1811709,
        "dateInsertion": "2017-01-08T13:30:20+0100",
        "versionApp": null,
        "application": false,
        "dateDerniereConnexion": "2017-01-08T14:34:39+0100",
        "versionCode": null,
        "device": null,
        "dateDerniereNotification": null,
        "type": null
    },
    "timestamp": "2017-01-08T13:46:50+0100",
    "client": {
        "ville": "Fondcombe",
        "dateNaissance": "1982-11-09T00:00:00+0100",
        "etranger": false,
        "sexe": "H",
        "domaineMail": "gmail.com",
        "resident": 2,
        "age": 37,
        "statut": "ACTIF",
        "mailValide": true,
        "uuid": "2d6f161f-c834-4240-9db8-a59a472df773",
        "groupeBorneId": 146,
        "dateInscription": "2017-01-08T13:30:19+0100",
        "modeInscription": "MAIL",
```

```
    "codePostal": null,
    "langue": "fr",
    "visiteur": 1,
    "telephoneValide": false,
    "anciennete": 0,
    "langueNavigateur": "fr-fr",
    "userAgent": "Mozilla/5.0 (iPhone; CPU iPhone OS 10_2 like Mac OS X)

    AppleWebKit/602.3.12 (KHTML, like Gecko) Mobile/14C92"
},
"session": {
    "calledStationId": "@@@",
    "acctOutputOctets": "63598377",
    "framedIpAddress": "@@@",
    "acctMultiSessionId": "f8e71e014c6c90fd612b39db587230690031",
    "acctSessionTime": "1100",
    "ruckusSsid": "_WifiLib_HAUT_DEBIT_GRATUIT",
    "acctOutputPackets": "54147",
    "acctSessionId": "58723069-014C6000",
    "sessionId": 390899521,
    "acctTerminateCause": "@@@",
    "acctInputPackets": "36301",
    "connectInfo": "CONNECT 802.11a/n",
    "acctInputOctets": "5484329",
    "acctStatusType": "Stop"
},
"commande": {
    "nombrePubliciteVue": 0,
    "statut": "ACTIVE",
    "notification": false,
    "dateProchainePublicite": "2017-01-08T23:30:19+0100",
    "dateFin": null,
    "commandeId": 5803225,
    "transactionId": "1de9d3f6",
    "dateDebut": "2017-01-08T19:13:55+0100"
},
"offre": {
    "montantHt": 0,
    "isRecurrent": false,
    "profilPub": "PRF-FWD-PUB",
    "profilHorsReseau": "",
```

```
        "priorite": 8,
        "offreId": 14,
        "nombreMaterielAutorise": 2,
        "libelle": "Forfait GRATUIT",
        "profil": "PRF-OPN-ALL",
        "isPayante": false,
        "isPeriode": true,
        "montantTtc": 0,
        "nombrePublicite": 10000,
        "recurrencePublicite": "36000",
        "description": "Forfait GRATUIT"
    },
    "id": 390899521,
    "borne": {
        "nom": "@@@",
        "ville": "Fondcombe",
        "zone": "@@@",
        "adresseMac": "F8-E7-1E-01-4C-60",
        "statut": false,
        "nomMagasin": "@@@",
        "adresse": "@@@",
        "localisation": "10.6606521675,-12.9192602512",
        "adressseMAC": "@@@",
        "codePostal": "@@@",
        "borneId": 3095,
        "codeVip": "6320171413",
        "publicite": true,
        "publiciteGeolocalisee": false,
        "meshRole": 2,
        "modele": "T300"
    }
}
```

D.2 Visualisation des connexions aux points d'accès *Wifilib* dans la ville d'Angers par l'outil *Ur-MoVe*

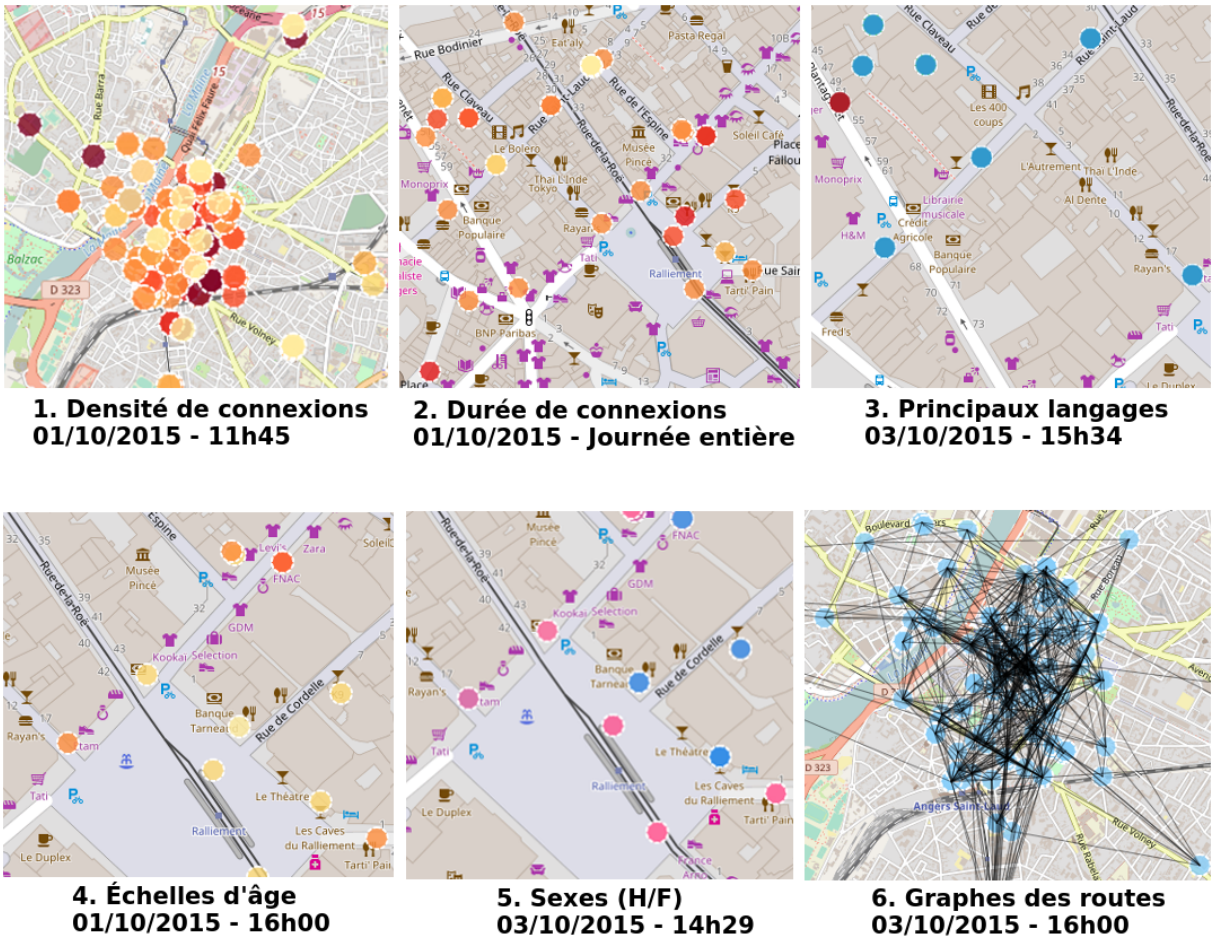


FIGURE D.1 – Contexte brut issu des connexions *Wifilib* à Angers et visualisable par l'outil *Ur-MoVe*

D.3 Politique de confidentialité (*RGPD*) de l'application *scéno*

Politique de confidentialité (RGPD)

Sécurité et protection des données personnelles de l'application mobile *scéno*

Définitions

- **L'Éditeur** : La personne, physique ou morale, qui édite les services au public connecté (mail : contact@sceno.fr).
- **L'Application** : L'application mobile et l'ensemble de ses services, proposés par l'Éditeur.
- **L'Utilisateur** : La personne utilisant l'application et les services.

Nature des données collectées

Dans le cadre de l'utilisation de l'Application, l'Éditeur est susceptible de collecter les catégories de données suivantes concernant ses Utilisateurs :

- Données d'état-civil, d'identité, d'identification... (date de naissance, sexe, ville de résidence,...)
- Données relatives à la vie personnelle (habitudes de vie, goûts et préférences, hors données sensibles ou dangereuses)
- Données relatives à la vie professionnelle (Catégorie socio-professionnelle)
- Données de localisation (déplacements, données GPS, GSM...)

Communication des données personnelles à des tiers

Pas de communication à des tiers
Vos données ne font l'objet d'aucune communication à des tiers. Vous êtes toutefois informés qu'elles pourront être divulguées en application d'une loi, d'un règlement ou en vertu d'une décision d'une autorité réglementaire ou judiciaire compétente.

Information préalable pour la communication des données personnelles à des tiers en cas de fusion / absorption

Collecte de l'opt-in (consentement) préalable à la transmission des données suite à une fusion / acquisition
Dans le cas où nous prendrions part à une opération de fusion, d'acquisition

ou à toute autre forme de cession d'actifs, nous nous engageons à obtenir votre consentement préalable à la transmission de vos données personnelles et à maintenir le niveau de confidentialité de vos données personnelles auquel vous avez consenti.

Finalité de la réutilisation des données personnelles collectées

La gestion des avis des personnes sur la pertinence des recommandations d'événements culturels qui leur sont faites.
Ainsi, les données seront utilisées à des fins de recherches scientifiques en apprentissage automatique à l'Université d'Angers (LERIA) et à l'ESEO (ERIS). Ces recherches se font dans le cadre du projet Event-AI. Event-AI est un projet financé par RFI (Atlanstic 2020) et labellisé par le pôle de compétitivité Images & Réseaux.

Agrégation des données

Agrégation avec des données non personnelles
Nous pouvons publier, divulguer et utiliser les informations agrégées (informations relatives à tous nos Utilisateurs ou à des groupes ou catégories spécifiques d'Utilisateurs que nous combinons de manière à ce qu'un Utilisateur individuel ne puisse plus être identifié ou mentionné) et les informations non personnelles à des fins d'analyses scientifiques, de profilage, à des fins de communications en congrès, conférences, articles de journal, ou chapitres de livre.

Collecte des données d'identité

Inscription et identification préalable pour la fourniture du service
L'utilisation de l'Application nécessite une inscription et une identification préalable. Vos données nominatives (nom, prénom, ville de résidence, date de naissances...) sont utilisées afin d'informer notre système automatique de recommandations d'événements culturels. Ces données individuelles ne seront traitées que par un algorithme et ne feront pas l'objet de visualisations ou de traitements manuels (i.e., par l'humain). Vous ne fournirez pas de fausses informations nominatives et ne créerez pas de compte pour une autre personne sans son autorisation. Vos informations devront toujours être exactes et à jour afin de ne pas dégrader le service de recommandations personnalisées qui vous est fourni.

Collecte des données d'identification

Utilisation de l'identifiant de l'utilisateur uniquement pour l'accès aux services
Nous utilisons vos identifiants électroniques seulement pour et pendant l'exécution du système de recommandations et du service de visualisation d'événements.

Géolocalisation

Géolocalisation à des fins de fourniture du service
Nous collectons et traitons vos données de géolocalisation afin de vous fournir des recommandations d'événements culturels mais aussi leur visualisation géolocalisée. Nous pouvons être amenés à faire usage des données personnelles dans le but de déterminer votre position géographique en temps réel. Conformément à votre droit d'opposition prévu par la loi n°78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés, vous avez la possibilité, à tout moment, de désactiver les fonctions relatives à la géolocalisation.
Géolocalisation à des fins de croisement
Nous collectons et traitons vos données de géolocalisation afin de permettre à notre système d'identifier les points de croisement dans le temps et dans l'espace avec d'autres Utilisateurs de l'application. Ceci nous permet de vous proposer des recommandations d'événements culturels correspondant à votre profil croisé avec des Utilisateurs de profil similaire au vôtre. Conformément à votre droit d'opposition prévu par la loi n°78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés, vous avez la possibilité, à tout moment, de désactiver les fonctions relatives à la géolocalisation. Vous reconnaissez alors que le système ne sera plus en mesure de vous présenter des recommandations d'événements personnalisées.
Géolocalisation avec mise à disposition des partenaires pour référencement et agrégation (avec opt-in)
Nous pouvons collecter et traiter vos données de géolocalisation avec nos partenaires (Université d'Angers et ESEO). Nous nous engageons à anonymiser les données utilisées. Conformément à votre droit d'opposition prévu par la loi n°78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés, vous avez la possibilité, à tout moment, de désactiver les fonctions relatives à la géolocalisation.

Collecte des données du terminal

Collecte des données de profilage et des données techniques à des fins de fourniture du service
Certaines des données techniques de votre appareil sont collectées automatiquement par l'application. Ces informations incluent notamment votre numéro utilisateur, configuration matérielle, configuration logicielle... La collecte de ces données est nécessaire à l'émission des recommandations et à la visualisation.
Collecte des données techniques à des fins scientifiques et statistiques
Les données techniques de votre appareil sont automatiquement collectées et enregistrées par notre système, à des fins statistiques et de recherches scientifiques. Ces informations nous aident à personnaliser et à améliorer continuellement votre expérience sur notre Application. Nous ne collectons ni ne conservons aucune donnée nominative (nom, prénom, adresse...) éventuellement attachée à une donnée technique. Les données collectées ne seront pas revendues à des tiers.

Cookies

Durée de conservation des cookies
Conformément aux recommandations de la CNIL, la durée maximale de conservation des cookies est de 13 mois au maximum après leur premier dépôt dans le terminal de l'Utilisateur, tout comme la durée de la validité du consentement de l'Utilisateur à l'utilisation de ces cookies. La durée de vie des cookies n'est pas prolongée à chaque connexion. Le consentement de l'Utilisateur devra donc être renouvelé à l'issue de ce délai.
Finalité cookies
Les cookies peuvent être utilisés pour des fins statistiques notamment pour optimiser les services rendus à l'Utilisateur, à partir du traitement des informations concernant la fréquence de connexion, la personnalisation des recommandations ainsi que les opérations réalisées et les informations consultées. Vous êtes informé que l'Éditeur est susceptible de déposer des cookies sur votre terminal. Le cookie enregistre des informations relatives à l'utilisation du service (les événements que vous avez consultés, la date et l'heure de la consultation...) que nous pourrons lire lors de vos visites ultérieures.
Opt-in pour le dépôt de cookies
Nous n'utilisons pas de cookies. Si nous devions en utiliser à l'avenir, vous en seriez informé préalablement et auriez la possibilité de désactiver ces cookies.

Conservation des données techniques

Durée de conservation des données techniques
Les données techniques sont conservées pour la durée strictement nécessaire à la réalisation des finalités visées ci-avant.

Délai de conservation des données personnelles et d'anonymisation

Conservation des données pendant la durée d'utilisation de l'Application
Conformément à l'article 6-5°de la loi n°78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés, les données à caractère personnel faisant l'objet d'un traitement ne sont pas conservées au-delà du temps d'utilisation de l'Application.
Conservation des données anonymisées au delà de l'utilisation de l'Application / après la suppression du compte
Nous conservons les données personnelles pour la durée strictement nécessaire à la réalisation des finalités décrites dans les présentes CGU. Au-delà de cette durée, elles seront anonymisées et conservées à des fins exclusivement statistiques et ne donneront lieu à aucune exploitation, de quelque nature que ce soit.
Suppression des données après suppression du compte
Des moyens de purge de données sont mis en place afin d'en prévoir la suppression effective dès lors que la durée de conservation ou d'archivage nécessaire à

l'accomplissement des finalités déterminées ou imposées est atteinte. Conformément à la loi n°78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés, vous disposez par ailleurs d'un droit de suppression sur vos données que vous pouvez exercer à tout moment en prenant contact avec l'Éditeur.
Suppression des données après 3 ans d'inactivité
Pour des raisons de sécurité, si vous ne vous êtes pas connecté à l'Application pendant une période de trois ans, vous recevrez une notification vous invitant à vous connecter dans les plus brefs délais, sans quoi vos données seront supprimées de nos bases de données.

Suppression du compte

Suppression du compte à la demande
L'Utilisateur a la possibilité de supprimer son Compte à tout moment, par simple demande à l'Éditeur.
Suppression du compte en cas de violation des CGU
En cas de violation d'une ou de plusieurs dispositions des CGU ou de tout autre document incorporé aux présentes par référence, l'Éditeur se réserve le droit de mettre fin ou restreindre sans aucun avertissement préalable et à sa seule discrétion, votre usage et accès aux services, à votre compte et à l'Application.

Indications en cas de faille de sécurité décelée par l'Éditeur Information de l'Utilisateur en cas de faille de sécurité

Nous nous engageons à mettre en œuvre toutes les mesures techniques et organisationnelles appropriées afin de garantir un niveau de sécurité adapté au regard des risques d'accès accidentels, non autorisés ou illégaux, de divulgation, d'altération, de perte ou encore de destruction des données personnelles vous concernant. Dans l'éventualité où nous prendrions connaissance d'un accès illégal aux données personnelles vous concernant stockées sur nos serveurs ou ceux de nos prestataires, ou d'un accès non autorisé ayant pour conséquence la réalisation des risques identifiés ci-dessus, nous nous engageons à :

1. Vous notifier l'incident dans les plus brefs délais.
2. Examiner les causes de l'incident et vous en informer.
3. Prendre les mesures nécessaires dans la limite du raisonnable afin d'amoindrir les effets négatifs et préjudices pouvant résulter dudit incident.

Limitation de la responsabilité
En aucun cas les engagements définis au point ci-dessus relatifs à la notification en cas de faille de sécurité ne peuvent être assimilés à une quelconque reconnaissance de faute ou de responsabilité quant à la survenance de l'incident en question.

Transfert des données personnelles à l'étranger

Pas de transfert en dehors de l'Union européenne
L'Éditeur s'engage à ne pas transférer les données personnelles de ses Utilisateurs en dehors de l'Union Européenne.

Modification des CGU et de la politique de confidentialité

En cas de modification des présentes CGU, engagement de ne pas baisser le niveau de confidentialité de manière substantielle sans l'information préalable des personnes concernées
Nous nous engageons à vous informer en cas de modification substantielle des présentes CGU, et à ne pas baisser le niveau de confidentialité de vos données de manière substantielle sans vous en informer et obtenir votre consentement.

Droit applicable et modalités de recours

Application du droit français (législation CNIL) et compétence des tribunaux Les présentes CGU et votre utilisation de l'Application sont régies et interprétées conformément aux lois de France, et notamment à la Loi n°78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés. Le choix de la loi applicable ne porte pas atteinte à vos droits en tant que consommateur conformément à la loi applicable de votre lieu de résidence. Si vous êtes un consommateur, vous et nous acceptons de nous soumettre à la compétence non-exclusive des juridictions françaises, ce qui signifie que vous pouvez engager une action relative aux présentes CGU en France ou dans le pays de l'UE dans lequel vous vivez. Si vous êtes un professionnel, toutes les actions à notre encontre doivent être engagées devant une juridiction en France. En cas de litige, les parties chercheront une solution amiable avant toute action judiciaire. En cas d'échec de ces tentatives, toutes contestations à la validité, l'interprétation et / ou l'exécution des présentes CGU devront être portées même en cas de pluralité des défendeurs ou d'appel en garantie, devant les tribunaux français.

Portabilité des données

Portabilité des données L'Éditeur s'engage à vous offrir la possibilité de vous faire restituer l'ensemble des données vous concernant sur simple demande. L'Utilisateur se voit ainsi garantir une meilleure maîtrise de ses données, et garde la possibilité de les réutiliser. Ces données devront être fournies dans un format ouvert et aisément réutilisable.

D.4 Copies d'écran de l'application mobile *scéno*

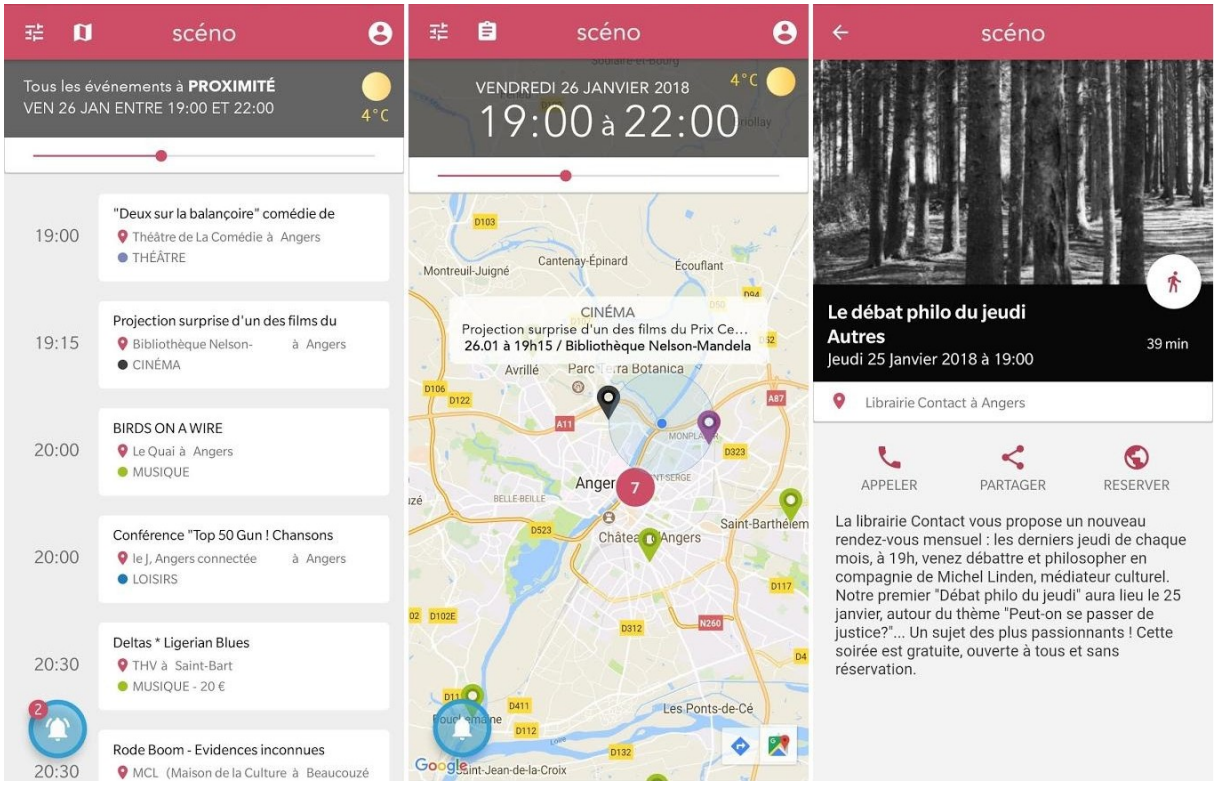


FIGURE D.2 – Principaux écrans de l'application *scéno*

FIGURE D.3 – Application *scéno* : zoom sur l'information de prévision météorologique en fonction du moment de la journée (période de 3h)

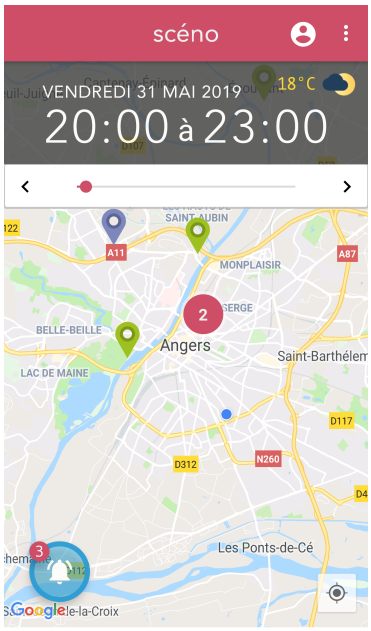


FIGURE D.4 – Application mobile *scéno* : écran de visualisation des événements culturels selon la géolocalisation

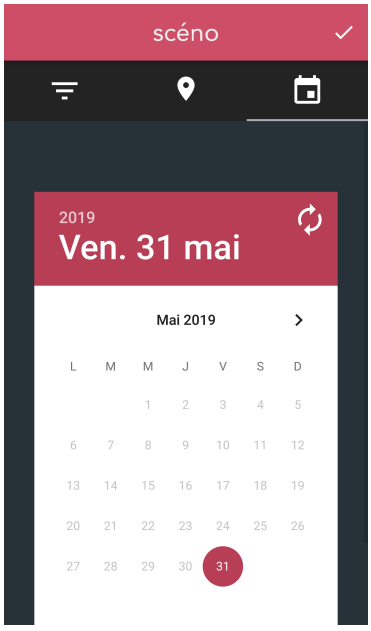


FIGURE D.5 – Application mobile *scéno* : module de calendrier

FIGURE D.6 – Application mobile *scéno* : écran de visualisation du détail d'un événement avec possibilité d'appel, partage, réservation et lancement du module de navigation jusqu'à destination

FIGURE D.7 – Application mobile *scéno* : différents filtres d'événements à afficher sur la carte

FIGURE D.8 – Application mobile *scéno* : exemple de recommandation

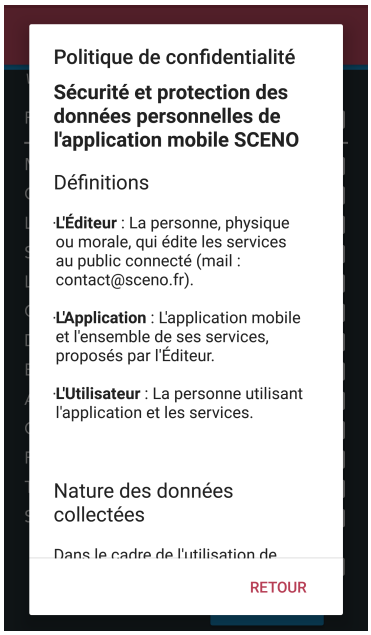


FIGURE D.9 – Application mobile *scéno* : Extrait de validation des conditions générales d'utilisations des données (*RGPD*)

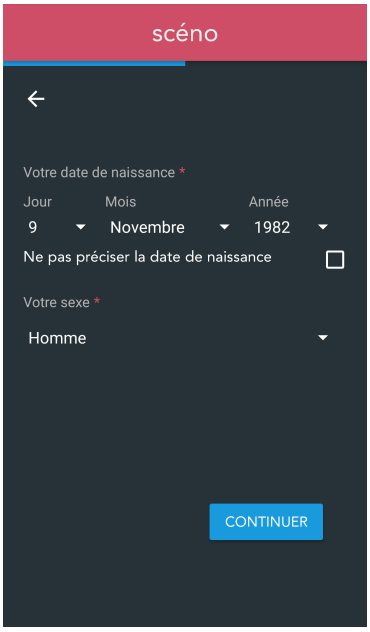


FIGURE D.10 – Application *scéno* : écran de renseignement du profil concernant l'âge et le sexe

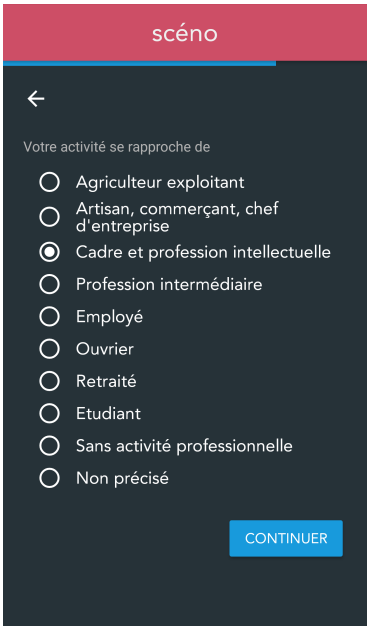


FIGURE D.11 – Application *scéno* : écran de renseignement du profil concernant la catégorie socio-professionnelle

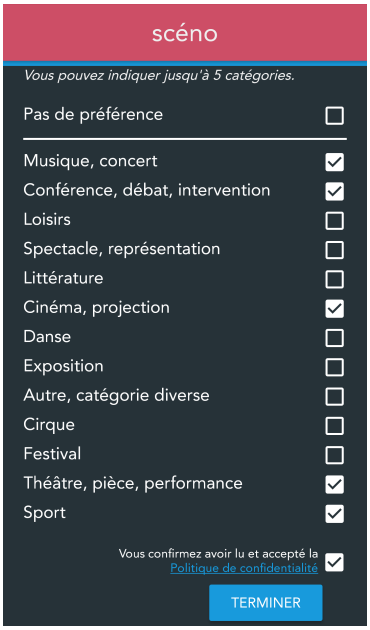


FIGURE D.12 – Application *scéno* : écran de renseignement du profil concernant les préférences utilisateurs

FIGURE D.13 – Application *scéno* : écran de paramétrage du rayon de recherche

ANNEXE E

LIENS DES PAGES WEB ARCHIVÉES

TABLE E.1 – Liens des pages archivées du site `http://www.wifilib-clustering.info/`

Page	Lien de la page archivée
Accueil	https://web.archive.org/web/http://www.wifilib-clustering.info/
Clusters Angers	https://web.archive.org/web/http://www.wifilib-clustering.info/cities/Angers.html
Clusters Nantes	https://web.archive.org/web/http://www.wifilib-clustering.info/cities/Nantes.html
Clusters Rennes	https://web.archive.org/web/http://www.wifilib-clustering.info/cities/Rennes.html
Clusters Paris	https://web.archive.org/web/http://www.wifilib-clustering.info/cities/Paris.html
Clusters Orléans	https://web.archive.org/web/http://www.wifilib-clustering.info/cities/Orleans.html
Clusters Reims	https://web.archive.org/web/http://www.wifilib-clustering.info/cities/Reims.html
Clusters Auxerre	https://web.archive.org/web/http://www.wifilib-clustering.info/cities/Auxerre.html
Clusters Strasbourg	https://web.archive.org/web/http://www.wifilib-clustering.info/cities/Strasbourg.html
Clusters Mulhouse	https://web.archive.org/web/http://www.wifilib-clustering.info/cities/Mulhouse.html
Clusters Lyon	https://web.archive.org/web/http://www.wifilib-clustering.info/cities/Lyon.html
Clusters Marseille	https://web.archive.org/web/http://www.wifilib-clustering.info/cities/Marseille.html
Clusters Montpellier	https://web.archive.org/web/http://www.wifilib-clustering.info/cities/Montpellier.html
Clusters Toulouse	https://web.archive.org/web/http://www.wifilib-clustering.info/cities/Toulouse.html
Clusters Hossegor	https://web.archive.org/web/http://www.wifilib-clustering.info/cities/Hossegor.html
Clusters « Terre du milieu »	https://web.archive.org/web/http://www.wifilib-clustering.info/middle-earth/MDLE.html
Clusters aléatoires « Terre du milieu »	https://web.archive.org/web/http://www.wifilib-clustering.info/middle-earth/MDLERandom.html

TABLE E.2 – Liens des pages archivées des sites web cités dans le mémoire

Site web	Lien de la page archivée
FOLDOC	https://web.archive.org/web/http://foldoc.org/
Cedric Cnam	https://web.archive.org/web/http://cedric.cnam.fr/vertigo/Cours/RCP216/coursSimilariteRecommandation.html
Open Data Angers	https://web.archive.org/web/20190717065041/https://data.angers.fr/pages/home/
Open Weather Map	https://web.archive.org/web/https://openweathermap.org/api
Kaggle (RSASM)	https://web.archive.org/web/https://www.kaggle.com/assopavic/recommendation-system-for-angers-smart-city
Wifilib	https://web.archive.org/web/https://www.wifilib.com/index.html
Livehoods	https://web.archive.org/web/http://livehoods.org/
Scikit-Learn	https://web.archive.org/web/http://scikit-learn.org/stable/
Étude préliminaire	https://web.archive.org/web/https://github.com/mabresearchstudy/mabaccuracy
Atlanstic 2020	https://web.archive.org/web/https://atlanstic2020.fr
scéno	https://web.archive.org/web/https://sceno.fr/
UCI Machine Learning	https://web.archive.org/web/https://archive.ics.uci.edu/ml/index.php
UC Berkeley (Jester)	https://web.archive.org/web/http://eigentaste.berkeley.edu/dataset/
Open Street Map	https://web.archive.org/web/https://www.openstreetmap.fr/
Wikipedia (LOTR)	https://web.archive.org/web/https://en.wikipedia.org/wiki/The_Lord_of_the_Rings

TABLE E.3 – Liens des pages archivées pour les applications mobiles

Application	Lien de la page archivée
scéno	https://web.archive.org/web/https://play.google.com/store/apps/details?id=sceno.android.pfe.com.sceno&hl=fr
Vivre à Angers	https://web.archive.org/web/https://play.google.com/store/apps/details?id=fr.angers.app
Angers Map	https://web.archive.org/web/https://play.google.com/store/apps/details?id=fr.patrickrgn.angersmap

Titre : Recommandation contextuelle de services : Application à la recommandation d'événements culturels dans la ville intelligente

Mot clés : Apprentissage par renforcement, Bandits-manchots, Systèmes de recommandation, Contexte, Diversité, Précision individuelle

Resumé : Les algorithmes de bandits-manchots pour les systèmes de recommandation sensibles au contexte font aujourd'hui l'objet de nombreuses études. Afin de répondre aux enjeux de cette thématique, les contributions de cette thèse sont organisées autour de 3 axes : 1) les systèmes de recommandation ; 2) les algorithmes de bandits-manchots (contextuels et non contextuels) ; 3) le contexte. La première partie de nos contributions a porté sur les algorithmes de bandits-manchots pour la recommandation. Elle aborde la diversification des recommandations visant à améliorer la précision individuelle. La seconde partie a porté sur la capture de contexte, le raisonnement contextuel pour les systèmes de recommandation d'événements culturels dans la ville intelligente, et l'enrichissement dynamique de contexte pour les algorithmes de bandits-manchots contextuels.

Title : Context-aware recommendation systems for cultural events recommendation in Smart Cities

Keywords : Reinforcement learning, Multi-Armed Bandit, Recommendation system, Context, Diversity, Individual Accuracy

Abstract : Nowadays, Multi-Armed Bandit algorithms for context-aware recommendation systems are extensively studied. In order to meet challenges underlying this field of research, our works and contributions have been organised according to three research directions : 1) recommendation systems ; 2) Multi-Armed Bandit (MAB) and Contextual Multi-Armed Bandit algorithms (CMAB) ; 3) context. The first part of our contributions focuses on MAB and CMAB algorithms for recommendation. It particularly addresses diversification of recommendations for improving individual accuracy. The second part is focused on context acquisition, on context reasoning for cultural events recommendation systems for Smart Cities, and on dynamic context enrichment for CMAB algorithms.