

UNIVERSITÉ NATIONALE DU VIETNAM À HANOÏ
INSTITUT DE LA FRANCOPHONIE INTERNATIONALE



Rapport de TPE

Option : Systèmes Intelligents et Multimédia (SIM)

Promotion : XXIII

RAPPORT FINAL

L'apprentissage par renforcement pour un système de recommandation contextualisé

Rédigé par :
ADOUM Okim Boka

Encadrant :
Dr. Ho Tuong Vinh

Année académique : 2018 - 2019

1 Remerciement

Par ces lignes, je voudrais manifester ma profonde gratitude et reconnaissance à tous ceux qui, d'une façon quelconque, ont contribué à ma formation à l'I.F.I (Institut Francophone Internationale) à Hanoï au Vietnam. Je tiens aussi à remercier tous ceux qui ont contribué à la réalisation de ce travail de recherche. Ainsi ces sincères remerciements s'adressent :

- A toute la Direction de IFI
- A mon encadreur Dr. Ho Tuong Vinh
- Au responsable de Master1 M. Nguyen Hong Quang
- A ma femme Khadidja Hanane DAOULA
- A mes collègues et amis

2 Résumé

Ce travail de recherche a pour objectif d'étudier les approches d'apprentissage par renforcement pour la construction d'un système de recommandation contextualisé. Nous formulons le problème de la recommandation interactive en tant que bandit multi-bras contextuel. Le système apprend les préférences des utilisateurs et recommande de nouveaux objets et reçoit leurs évaluations. Nous montrons que l'apprentissage par renforcement résout le problème du compromis exploitation-exploration et du problème du démarrage à froid. De même nous avons exploré l'approche basée sur le contenu ainsi que l'approche de filtrage collaboratif et les deux produisent des résultats de recommandation satisfaisante. Pour ce faire, premièrement, nous avons fait l'analyse du sujet, question de se forger de son idée. secundo, nous avons fait la recherche bibliographique ou nous avons répertorié les éléments constructifs de système de recommandation et de l'apprentissage par renforcement. Ensuite, nous avons proposé des éventuelles solutions qui concours à la résolution de ce problème. Enfin, nous avons entamé la réalisation pratique tout en présentant les résultats des analyses de façon continu.

Mots clés : Système de recommandation, apprentissage par renforcement, problème de bandit multibras.

3 Abstract

This research project aims to study reinforcement learning approaches for the construction of a contextualized recommendation system. We formulate the problem of interactive recommendation as a contextual multi-arm bandit. The system learns user preferences and recommends new items and receives their ratings. We show that reinforcement learning solves the problem of exploit-exploration compromise and the problem of cold start. Similarly, we explored the content-based approach and the collaborative filtering approach and both produce satisfactory recommendation results. To do this, in the first, we did the analysis of the subject, to form this idea. And then as a secondly we did the bibliographic research or we listed the constructive elements of recommendation system and reinforcement learning. Then we proposed possible solutions that help solve this problem. Finally, we started the practical realization while presenting the results of the analyzes in a continuous way.

Keywords : Recommendation system, reinforcement learning, multi-band bandit problem.

Table des matières

1	Remerciement	1
2	Résumé	2
3	Abstract	3
4	Introduction générale	7
5	Analyse du sujet	8
5.1	Contexte	8
5.2	Objectifs du sujet	8
5.3	Problématique.	9
5.4	Travaux à réaliser	9
5.5	Difficultés à prévoir	10
5.6	Conclusion	10
6	Recherche bibliographique	11
6.1	Etude de l'existant	11
6.2	Apprentissage par renforcement	11
6.2.1	Aperçu de l'apprentissage par renforcement	11
6.2.2	Les composants d'un système d'apprentissage par renforcement	12
6.3	Modèle de connaissance	13
6.4	Modèle du contexte	14
6.4.1	Exploitation vs Exploration	14
6.4.2	Qu'est-ce qu'un bandit à plusieurs bras ?	15
6.4.3	Définition mathématique	16
6.4.4	Processus de Decision Markovien (MDP)	16
6.5	Algorithme de ϵ -greedy (glouton)	17
6.6	Limites de confiance supérieures (UCB) [5]	17
6.7	Inégalité de Hoeffding	18
6.8	Algorithme UCB1	19
6.9	Algorithme de UCB Bayésien	19
6.10	Filtrage collaboratif, problème bandit de Bernoulli, biclustering et Processus de Décision Markovien (MDP)	19
6.11	Conclusion	20
7	Choix de la solution	21
8	Réalisation pratique, Expérimentations, Analyse des résultats	22
8.1	cas de système de recommandation avec filtrage collaboratif	22
8.1.1	Qu'est-ce que Google Colaboratory et quels en sont les avantages ?	22

8.1.3	Chargement et prévisualisation des données	23
8.1.4	Simple statistique sur les données	24
8.1.5	Hypotheses et model de l'apprentissage automatique	25
8.1.6	Création de données d'entraînement	25
8.1.7	Graphe d'entraînement et de perte	25
8.1.8	Analyse des résultats et évaluation	26
8.1.9	système de recommandation	26
8.2	cas de système de recommandation avec apprentissage par renforcement	26
8.2.1	Modélisation contextuelle des bandits multiples	26
8.2.2	Extraction et sélection attributs	27
8.2.3	Résultat et expérimentation	27
8.3	Conclusion	29
9	Conclusion générale	30
	Références	31

Table des figures

1	Les Sources de référence qui font l'exploit des systèmes de recommandation . . .	13
2	Les travaux de recherche sur le quel s'inspire notre travail	14
3	Un exemple concret du dilemme exploration/exploitation : où manger ? (Source de l'image : diapositive de cours UC Berkeley AI, conférence 11.)	15
4	Illustration du fonctionnement d'un bandit à multiples bras de Bernoulli. Les probabilités de récompense sont inconnues du joueur.	16
5	contenu du fichier vidéo	23
6	contenu du fichier ratings	23
7	contenu du fichier tags	23
8	contenu du fichier gnome-tags	24
9	contenu du fichier gnome-scores	24
10	perte d'entraînement	26
11	perte de test	26
12	Vu d'un système de recommandation de film en tant que problème de bandit multi-armés contextualisé	27
13	Espace de définition des films	27
14	vu de films sous évaluation des utilisateurs	27
15	Regret moyen cumulatif pour chaque algorithme	28
16	Comparaison des évaluations moyennes des utilisateurs pour chaque algorithme .	28
17	Précision moyenne pour chaque algorithme	29
18	Durée moyenne d'exécution pour chaque algorithme	29

4 Introduction générale

Un système de recommandation (S.R) est l'ensemble des moyens qui permet de faire le filtrage des informations dans le but de pouvoir les suggérer à un utilisateur et/ou groupe utilisateur susceptibles de les apprécier. pour ceux, en se basant sur la similarité des informations déjà apprécier dans le passé, par la similarité des objets appréciés par les autres utilisateurs et/ou groupes d'utilisateurs par l'expérience utilisateur, ou par son comportement. C'est aussi une technique qui consiste à prédire une appréciation face à un objet à un utilisateur dans le sens qu'il serait en mesure d'apprécier. L'objectif principal de cette recherche est d'implémenter une stratégie qui permettra au S.R d'explorer et d'exploiter automatiquement les objets fraîchement intégrés dans la base de données (les items, les utilisateurs, les interactions utilisateurs). pour mener à bien ce projet, nous travaillerons respectivement selon les points dont **l'analyse de sujet** ; nous permettra de bien comprendre la problématique, **étude bibliographique** ; nous permettra de cerner le contour du sujet, de suite nous **proposition de la solution**, puis nous finirons par une **implémentassions** et **test**.

5 Analyse du sujet

Dans cette partie, nous présenterons le contexte, la problématique et les travaux à réaliser question de forger l'idée du sujet.

5.1 Contexte

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, des films, des musiques, des informations de voyage personnalisées à des touristes ou des suggestions d'amis dans les réseaux sociaux.

Certains systèmes de recommandations font parties des applications dites de Mobile Crowd Sensing and Computing (MCSC) qui permettent de collecter une grande variété de données à partir des capteurs embarqués dans les téléphones mobiles des utilisateurs. Les systèmes de recommandation appliqués aux MCSC prennent tout leur sens au vu de la multitude d'informations disponibles dans ces villes intelligentes parmi lesquelles il est très difficile de faire un choix. Par ailleurs, les terminaux mobiles sont capable de capter le contexte des utilisateurs, donnant la possibilité au système de fournir des recommandations en fonction de ce dernier. Un tel système de recommandation contextuel pourrait ainsi recommander des activités d'intérieur s'il pleut et des activités d'extérieur en cas de beau temps. Plusieurs méthodes ont été utilisées pour mettre en œuvre des systèmes de recommandation :

- **approche collaboratrice ;**
- **approche basée sur le contenu ;**
- **approche hybride ;**
- **approche basé sur l'apprentissage par renforcement [3].**

Une autre approche consiste à considérer la recommandation contextuelle comme un problème de bandits multibras contextuel. Dans ce cas le système de recommandation peut s'appuyer sur des méthodes d'apprentissage par renforcement pour le résoudre. Des algorithmes tels que Upper Confidence Bound (UCB), Bayesian UCB, Random, Greedy ou Lin-UCB sont utilisés. Pour cela, le dernier étant spécifiquement dédié aux recommandations contextuelles.

5.2 Objectifs du sujet

Pour mieux cadrer notre travail, nous nous fixons les objectifs ci-dessous :

Objectif général

- Implémenter une stratégie d'exploration/exploitation dans un S.R ;
- Mettre en place un système de recommandation en se basant sur l'apprentissage par renforcement ;
- S'imprégner du monde de recherche

Objectif spécifique

- Réaliser un système qui permet de recommander des films à un utilisateur et/ou groupe d'utilisateur en fonction de son profil ainsi que ses préférences ;
- Réaliser un S.R des films avec le data-set Movielens 20M ;
- fixer un scénario d'utilisation, qui doit être à la fois cohérent avec un S.R et aussi avec une architecture de l'apprentissage par renforcement.

5.3 Problématique.

la problématique que soulève notre thématique est de définir un critère qui permet à un utilisateur de passer d'une approche de recommandation à une autre. En d'autres termes, A quel moment de recommandation l'approche apprentissage par renforcement est nécessaire pour un système de recommandation ?

5.4 Travaux à réaliser

Notre travail est d'ordre théorique et pratique.

Travaux théoriques :

Les travaux théoriques que nous réaliserons seront focalisés sur l'analyse du sujet, l'état de l'art et la proposition des éventuelles solutions.

Ainsi, pour mener à bien cette première partie qui est socle de notre travail, nous allons nous focalisés aux trois points suivants :

- Étudier l'état de l'art sur les systèmes de recommandations et plus particulièrement ceux qui tiennent compte du contexte :
 - recenser les approches utilisées ;
 - recenser les éléments de contextes utilisés.
- Étudier comment les méthodes d'apprentissage par renforcement permettent-elles de mettre en œuvre un système recommandation. Pour cela il sera nécessaire d'étudier les aspects suivants :
 - la façon dont le problème de recommandation peut être considéré comme un problème de bandits multibras (Multi Armed Bandits – MAB) et de bandits multibras contextuels (Contextual Multi Armed Bandit – CMAB) ;
 - les différents algorithmes tels que UCB, Lin-UCB etc. qui permettent de résoudre des problèmes de type MAB et CMAB. Dans le cas des algorithmes UCB et Lin-UCB, l'évaluation de la recommandation fournie à un utilisateur est supposée être

- retournée aussitôt que l'utilisateur reçoive la recommandation. Or, dans la pratique, les utilisateurs doivent attendre de se faire un avis avant d'évaluer une recommandation ;
- e.g. : on ne peut donner son avis sur un film qui nous a été recommandé qu'après l'avoir vu. L'objectif de ce travail est de proposer une évolution de l'algorithme Lin-UCB capable de prendre en compte des retours différés de la part des utilisateurs.
 - pour des raisons de complémentarité des algorithmes, nous allons faire recours à des algorithmes de greedy ou de random[6] ou de bayes etc. pour prendre en compte les manquements des algorithmes UCB et Lin-UCB
 - les possibilités des mesures et d'intégration de l'approche par filtrage collaboratif, de l'approche basée sur le contenu et l'approche hybride.
 - Proposition des éventuelles solutions qui concourent à la mise en oeuvre de projet.

Travaux pratiques :

Dans cette partie nous nous intéresserons à l'implémentation et au test selon les points ci-dessous :

1. Implémenter la solution proposée ;
2. Tester les résultats ;
3. Évaluer la solution.

5.5 Difficultés à prévoir

Aux regards de nos différentes analyses liées à notre thématique, nous aimerions prévoir les difficultés qui sont entre autres :

- Manque d'expérience dans le domaine d'apprentissage par renforcement, d'apprentissage automatique et les systèmes de recommandations
- La non maîtrise des plate-formes de développement des applications orientées Intelligence Artificielle(IA).

5.6 Conclusion

Cette partie dénommée analyse du sujet nous a permis de cerner le contour du sujet, de mieux appréhender et de comprendre les attentes de ce sujet. Ainsi dans les lignes suivantes nous allons faire l'étude bibliographique, celle-ci nous permettra de mûrir davantage les connaissances dans ce domaine.

6 Recherche bibliographique

Dans cette partie de notre TPE, nous allons faire des recherches sur les travaux existants en utilisant les bases de données de revues scientifiques et des moteurs de recherche académiques. Dans le sens de vouloir cerner les algorithmes potentiels, les méthodes de résolutions, les environnements d'application, et outils existants.

6.1 Etude de l'existant

Pour mieux cerner le contour de notre sujet, nous avons utilisé les bases de données des revues scientifiques et des moteurs de recherches académiques. Pour ceux, nous avons exploré respectivement à un degré près des articles, des thèses de recherches, des rapports d'étude, d'analyses, de synthèses question de vouloir connaître les résultats, des expériences, des travaux similaires des autres chercheurs qui ont abordé ce domaine de recherche dans le sens d'évaluer les solutions possibles, réutiliser au mieux ce qui existe, éviter de refaire les mêmes choses puis de profiter des expériences et des idées. Lors de nos recherches, nous avons procédé au croisement des mots et/ou groupe des mots suivants : apprentissage par renforcement, système de recommandation, système de recommandation contextualisé, reinforcement learning, recommendation system, contextualized recommendation system. Parmi les bases de données et les moteurs de recherche académiques consultés, nous avons :

- Nous avons Google Scholar, a travers son site <http://scholar.google.com> [en date 17/12/2018] a donné en utilisant « reinforcement learning » 2 550 000 resultats, puis en croisant avec « recommendation system » à donné 111 000 resultats, et en combinant une fois de plus avec le mot « contextualized » nous avons obtenu 23 500 resultats. Ainsi en faisant des recherches avec les mots « apprentissage par renforcement » nous a donnée 121 0000 resultats, et en faisant le croisement avec le mot « système de recommandation » nous a donné 34 700 resultats et puis en ajoutant à ce croisement le mot contextualisés nous avons eu 5 600 resultats;¹
- En utilisant le moteur de recherche https://www.google.fr/advanced_search (nous avons eu 96200 resultats en utilisant le groupe de mots « système de recommandation contextualisé», et en faisant le croisement des mots de la façon suivante « reinforcement learning and recommendation systems » nous avons eu 106 000 resultats).²

6.2 Apprentissage par renforcement

6.2.1 Aperçu de l'apprentissage par renforcement

L'apprentissage par renforcement (Reinforcement learning (RL) en anglais) est un sous-ensemble de l'apprentissage automatique (Machine learning (ML)). Cependant, les algorithmes

1. <https://scholar.google.com/>

2. https://www.google.fr/advanced_search

utilisés dans ML sont de nature prédictive, et le terme «apprentissage» dans ce type d'apprentissage automatique fait référence au fait que l'algorithme est alimenté en données. L'apprentissage automatique prédictif se divise en deux catégories :

- apprentissage supervisé ;
- l'apprentissage non supervisé.

Par contre, l'apprentissage par renforcement est différente de l'apprentissage automatique prédictive. Dorénavant, nous notons que les méthodes d'apprentissage supervisé et/ou non supervisé sont des classificateurs «à un seul plan» qui tentent de faire une prédiction unique à partir des données . En revanche, RL est conçu pour mettre en œuvre une pensée cognitive d'ordre supérieur. Plus précisément, RL cherche à trouver des décisions optimales profitant à l'utilisateur sur le long terme, même si cela nécessite de prendre des mesures indésirables à court terme. Dans l'apprentissage par renforcement, un agent d'informatique apprend en interagissant directement avec son environnement et enregistre les récompenses qu'il reçoit au fur et à mesure. L'objectif d'un agent RL est simplement d'identifier les actions qui mèneront à la récompense future maximale et cumulative. Cette approche est similaire à la façon dont le cerveau utilise la dopamine comme signal pour guider le comportement des animaux et des humains. Selon le concept de RL, l'agent apprend directement à partir de son environnement, il ne nécessite pas d'énormes quantités de données de formation préexistantes, spécifiques à un domaine, à apprendre. Au contraire, il génère ses propres données en interagissant avec son environnement et en se servant de son expérience pour tirer de leçon.

6.2.2 Les composants d'un système d'apprentissage par renforcement

En générale, l'apprentissage par renforcement est caractérisé par les composants suivants :

- **Un agent**

C'est un système informatique capable de percevoir son environnement et d'agir de manière autonome dans cet environnement afin d'atteindre ses objectifs.

- **Un environnement**

C'est un ensemble des éléments (biotiques ou abiotiques) qui entourent un individu ou une espèce et capable de subvenir à ses besoins.

- **Perception**

Le module "Perception" détecte les situations typiques dans lesquels il peut intervenir. Son rôle est de détecter l'opportunité d'une intervention.

- **Action**

Le module "Action" est un sous-module qui contient les actions à être activés pour réagir à une situation donnée. Il actionne l'environnement et lui-même.

6.3 Modèle de connaissance

Notre modèle de connaissance s'inspire de :

- succès des outils de recommandation tels que Netflix³ qui nous suggère des films en utilisant l'approches hybrides, Amazon⁴ qui recommande des livres à ses usagés en utilisant l'approche basée sur le filtrage collaboratif, Pandora⁵ qui intègre un module de recommandation musicale en utilisant l'approches basée sur le contenu.

Nous résumons cela dans le tableau ci-dessous.


Application	Site	Approche	Objectifs
	www.amazon.com	● Approches basées sur le filtrage collaboratif	● outil de marketing ciblé
	www.netflix.com	● Approches hybrides	● un service de location de films en ligne
	www.pandora.com	● Approches basées sur le contenu	● recommandation musicale

FIGURE 1 – Les Sources de référence qui font l'exploit des systèmes de recommandation

- Des traveaux de recherche sous les references cité en reference [Sungwoon 2018] présenté en janvier 2018 sous le thème « Reinforcement Learning based Recommender System using Biclustering Technique », des multiples traveaux de recherches [Lil'Log 2018] présenté en janvier 2018 sous le thème «The Multi-Armed Bandit Problem and Its Solutions », le 05 mai 2018 sous le thème « Implementing Deep Reinforcement Learning Models with Tensorflow + OpenAI Gym.», le 05 Mai 2019 sous le thème «Domain Randomization for Sim2Real Transfer.» et en juin 2019 sous le «Meta Reinforcement Learning.» par l'auteur Lilian Weng.

Quelques sources de connaissances utilisées ainsi que leur particularité sont regroupées dans le tableau ci-dessous :

-
3. <https://www.netflix.com/vn/>
 4. <https://www.amazon.fr/>
 5. <https://www.pandora.com>

Présenté par :	Université/ Conférence	Thème	Approches	Résultat
Thèse 2016, Jonathan LOUËDEC	U. Toulouse	Stratégies de bandit pour les systèmes de recommandation	<ul style="list-style-type: none"> . L'approche Multiple-Play Bandit (MPB) ; . Approches inspirées des bandits à tirages simples . Approches inspirées des modèles de clics de la recherche d'information 	<ul style="list-style-type: none"> . Pour recommander simultanément une liste de m objets, le premier objet est sélectionné selon la règle utilisée par le bandit. . variante de la stratégie capable de recommander plusieurs objets à chaque instant, Cette espérance est tirée aléatoirement selon la loi de probabilité Bêta associée à cet objet.
Thèse, 16/04/18 Idir BENOQUARET	U. Technologie Compiègne	Un système de recommandation contextuel et composite pour la visite personnalisée de sites culturels	<ul style="list-style-type: none"> . Les approches basées sur le contenu . Approche générale . Les approches basées sur le filtrage collaboratif . Les Approches Hybrides 	<ul style="list-style-type: none"> . analyser le contenu des items candidats à la recommandation . Approche basée contenu : les profils des items et les profils des utilisateurs. . approche basée sur le partage d'opinions entre les utilisateurs . La recommandation basée sur le contenu et la recommandation collaborative ont souvent été considérées comme complémentaires
Article, 23/01/18 Lilian Weng	Google scholar	The Multi-Armed Bandit Problem and Its Solutions	<ul style="list-style-type: none"> ϵ-Greedy Algorithm UCB1 <u>Bayesian UCB</u> 	Comparaison des différents stratégie en utilisant Approche Multi-bras pour résoudre le problème d'exploration-exploitation
Article, 05/05/18 Lilian	Google Scholar	Implementing Deep Reinforcement Learning Models with Tensorflow	Tensorflow, OpenAI Gym	Politique de gradient, REINFORCE

FIGURE 2 – Les travaux de recherche sur le quel s'inspire notre travail

6.4 Modèle du contexte

Le modèle de contexte répond aux études qui ont prouvées que le problème de recommandation peut être considéré comme un problème de bandits multibras (Multi Armed Bandits – MAB) et de bandits multibras contextuels (Contextual Multi Armed Bandit – CMAB) et au article de Lilian Weng (Jan 23, 2018) sous le thème (The Multi-Armed Bandit Problem and Its Solutions).

6.4.1 Exploitation vs Exploration

Le dilemme exploration/exploitation existe dans de nombreux aspects de notre vie. supposons que nous sommes un habitué de restaurant et que nous y allons tous les jours dans un seul restaurant, nous sommes confiant de à ce que nous obtiendrons, mais nous perdrons nos chances de découvrir une option encore meilleure. Mais dans le cas contraire ou si nous essayons de nouveaux endroits tout le temps, nous devrons probablement manger de la nourriture déplaissante de temps en temps. De même, les internautes tentent d'équilibrer les annonces connues les plus attrayantes et les nouvelles annonces susceptibles d'être encore plus efficaces [4].

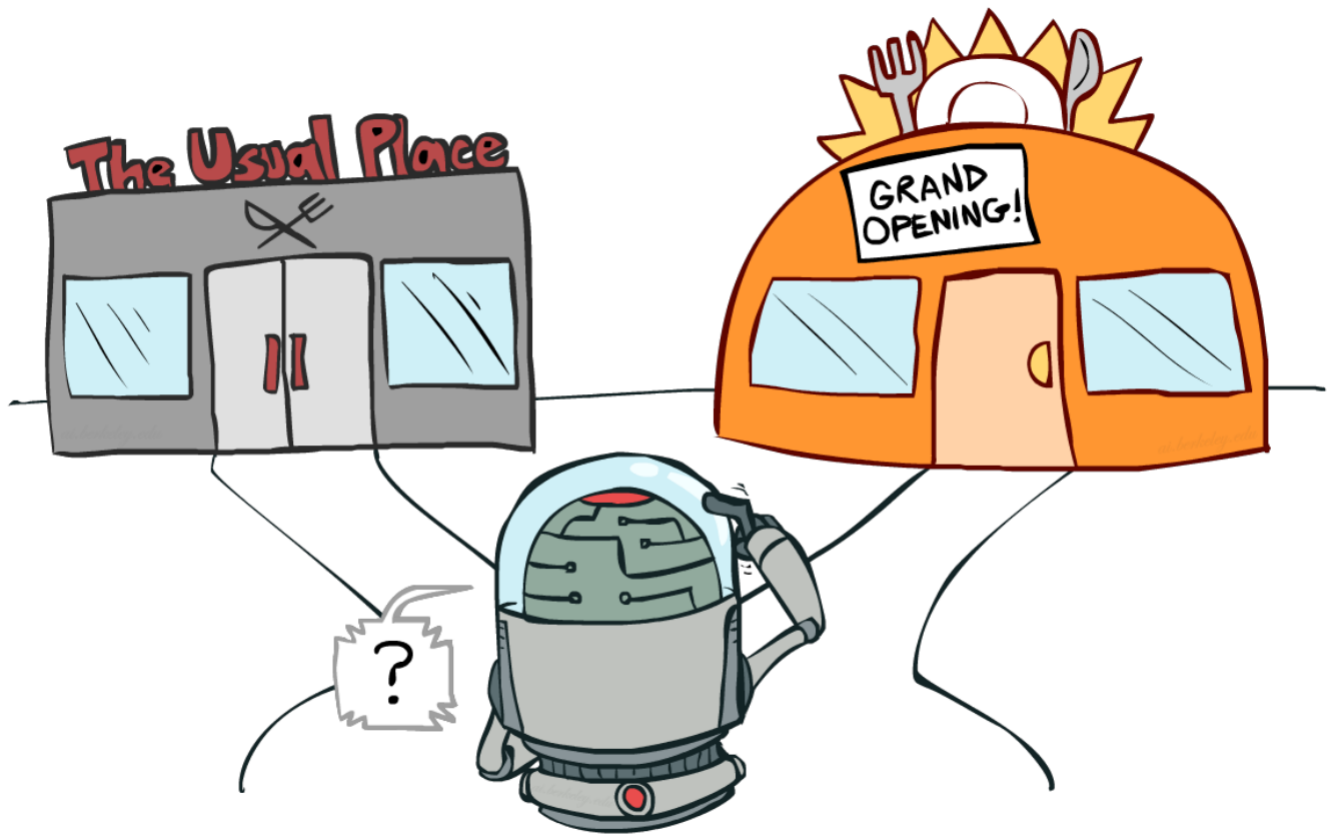


FIGURE 3 – Un exemple concret du dilemme exploration/exploitation : où manger ? (Source de l'image : diapositive de cours UC Berkeley AI, conférence 11.)

6.4.2 Qu'est-ce qu'un bandit à plusieurs bras ?

Illustration avec le problème de jeux de chance lié aux machines à sous. Le problème des bandits à plusieurs bras est un problème classique qui illustre bien le dilemme exploration/exploitation. Dans ce poste, nous ne discuterons que de la possibilité de disposer d'un nombre infini d'essais. La limitation d'un nombre d'essais introduit un nouveau type de problème d'exploration. Par exemple, si le nombre d'essais est inférieur au nombre de machines à sous, nous ne pouvons pas essayer toutes les machines pour estimer la probabilité de récompense. Donc nous devrions nous comporter intelligemment par rapport à un ensemble de connaissances acquis et le temps d'essaye impartie.

Ce langage est très simple, mais la nécessité d'alternance place/transition peut très vite faire exploser la taille du diagramme. Il a l'inconvénient de présenter une sémantique assez pauvre, dû à sa simplicité de représentation.

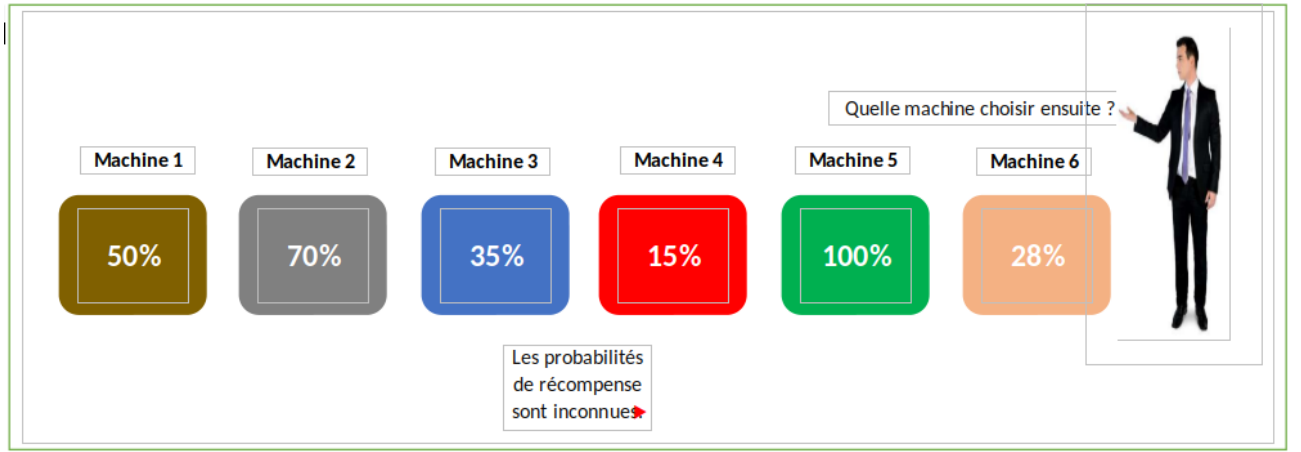


FIGURE 4 – Illustration du fonctionnement d'un bandit à multiples bras de Bernoulli. Les probabilités de récompense sont inconnues du joueur.

Une approche naïve peut consister à continuer à jouer avec une seule machine pendant plusieurs tours de manière à estimer éventuellement la «vraie» probabilité de récompense selon la loi des grands nombres. Cependant, cela représente un gaspillage considérable et ne garantit certainement pas la meilleure récompense à long terme.

6.4.3 Définition mathématique

Scientifiquement, on peut voir le problème de bandit à plusieurs bras de la façon suivante :

D'après la littérature, Un bandit à multiples bras de Bernoulli peut être décrit comme un tuple de $\langle \mathcal{A}, \mathcal{R} \rangle$.

ou :

- Nous avons K machines avec des probabilités de récompense $\{\theta_1 \dots \theta_k\}$;
- A chaque temps t que nous effectuons une action a sur une machine à sous et recevons une récompense r .
- A est un ensemble des actions. chaque action fait référence à l'interaction avec une machine à sous. La valeur de l'action a est la récompense attendue, on a $Q(a) = E [a || r] = \theta$. si l'action a_t en un temps t est sur i^{me} machine alors $Q(a_t) = \theta_i$
- R est une fonction de récompense. Dans le cas du bandit de Bernoulli, nous observons une récompense r de manière stochastique. A l'instant t $r_t = R(a_t)$ renvoie une récompense égale à 1 pour une probabilité $Q(a_t)$ et 0 dans le cas contraire.

6.4.4 Processus de Decision Markovien (MDP)

Définition 1 MDP un Processus de Decision Markovien est un tuple $M = (S; \bar{s}; A; P; R)$ ou :

- S est l'ensemble des états ;
- $\bar{s} \in S$ est un état initial ;
- A est l'ensemble des actions ;
- $P : S \times A \rightarrow \text{Dist}(S)$ est une fonction de transition probabiliste (partielle) ;

- $R = (R_S; R_A)$ est une structure de récompense où $R_S : S \rightarrow \mathbb{R}$ est une fonction de récompense de l'état et $R_A : S \times A \rightarrow \mathbb{R}$ une fonction de récompense d'action.

Stratégies Bandit :

Sur la base de choix d'une des types des explorations citées sur les points ci-dessous :

- cas où il n'y a pas d'exploration : l'approche la plus naïve et la mauvaise ;
- Exploration par hasard ;
- exploration intelligente avec de préférences l'incertitude.

Il existe plusieurs façons de résoudre le problème de bandit aux armes multiples en utilisant les algorithmes **greedy (glouton)**, **Limites de confiance supérieures (UCB)**, **Inégalité de Hoeffding**, **UCB1**, **UCB Linéaire**, **UCB Bayésien** ;

6.5 Algorithme de ϵ -greedy (glouton)

L'algorithme ϵ -greedy effectue la meilleure action la plupart du temps, mais effectue de temps en temps une exploration aléatoire. La valeur d'action est estimée en fonction de l'expérience passée en faisant la moyenne des récompenses associées à l'action cible a que nous avons observées jusqu'à présent (jusqu'au pas de temps actuel t) :

$$\hat{Q}_t(a) = \frac{1}{N_t(a)} \sum_{t=1}^t r_t 1[a_t = a]$$

où 1 est une fonction d'indicateur binaire et $N_t(a)$ est le nombre de fois où l'action a a été sélectionnée jusqu'à présent,

$$N_t(a) = \sum_{t=1}^t 1[a_t = a]$$

En accordant à l'algorithme de ϵ -greedy une petite probabilité ϵ , menons une action aléatoire, sinon (en temps normal la probabilité doit être égale $1-\epsilon$), nous sélectionnons la meilleure action que nous avons apprise jusqu'à présent :

$$\hat{a}_t^* = \operatorname{argmax}_{a \in \mathcal{A}} \hat{Q}_t(a)$$

6.6 Limites de confiance supérieures (UCB) [5]

L'exploration aléatoire nous donne l'occasion d'essayer des options que nous ne attendons pas. Mais cependant, en raison des caractères aléatoires, il est possible que nous explorions de mauvaise action que nous avons confirmée par le passé (malchance!). Pour éviter une exploration aussi inefficace, une approche consiste à diminuer le paramètre ϵ dans le temps et de même une autre approche consiste à montrer optimisme quant aux options très incertaines et à privilégier les actions pour lesquelles nous n'avons pas encore estimé la valeur. En d'autres termes, nous privilégions l'exploration des actions ayant une forte potentielle d'avoir une valeur

optimale. Limites de confiance supérieures (UCB) algorithme mesure cette potentialité par une limite de confiance supérieure de la valeur de la récompense,

$$\hat{U}_t(a)$$

de sorte que la valeur réelle soit inférieure à valeur limite

$$Q(a) \leq \hat{Q}_t(a) + \hat{U}_t(a)$$

avec une grande probabilité. la limite supérieur

$$\hat{U}_t(a)$$

est une fonction de

$$N_t(a)$$

Pour un plus grand nombre d'essais

$$N_t(a)$$

devrait nous donner une plus petite limite

$$\hat{U}_t(a)$$

. Dans l'algorithme UCB, nous sélectionnons toujours l'action la plus gourmande pour maximiser la limite de confiance supérieure :

$$\hat{a}_t^{UCB} = \operatorname{argmax}_{a \in \mathcal{A}} \hat{Q}_t(a) + \hat{U}_t(a)$$

Maintenant, la question est de savoir comment estimer la limite de confiance supérieure.

6.7 Inégalité de Hoeffding

En théorie des probabilités, l'inégalité de Hoeffding est une inégalité de concentration concernant les sommes de variables aléatoires indépendantes et bornées. C'est un théorème applicable à toute distribution bornée.

Soit $X_1 \dots X_t$ l'ensemble des variables aléatoires indépendantes et identiquement distribuées (i.i.d) à valeur bornée dans l'intervalle $[0,1]$. La moyenne

$$\bar{X}_t = \frac{1}{t} \sum_t^t X_t$$

Alors pour $u > 0$, on a :

$$P \left[E \left[X \right] > \bar{X}_t + u \right] \leq e^{-2tu^2}$$

Pour une action \mathbf{a} donnée, nous considérons :

- $r_t(a)$ est désigné comme ensemble des variables aléatoires ;
- $Q(a)$ comme la vraie moyenne ;
- $\hat{Q}_t(a)$ en tant que moyenne de l'échantillon ;
- $\mathbf{u} = U_t(a)$ est comme limites de confiance supérieure.

Ainsi, on aura :

$$P \left[E \left[X \right] > \bar{X}_t + U_t(a) \right] \leq e^{-2tU_t(a)^2}$$

Nous voulons choisir une limite avec de fortes chances afin que la vraie moyenne soit la moyenne de l'échantillon + la limite de confiance supérieure. Ainsi, $e^{-2tU_t(a)^2}$ devrait être une faible probabilité. Supposons que nous sommes d'accord avec un petit seuil p .

$$e^{-2tU_t(a)^2} = p, \text{ ainsi } U_t(a) = \sqrt{\frac{-\log p}{2N_t(a)}}$$

6.8 Algorithme UCB1

Une heuristique consiste à réduire le seuil p dans le temps, pour ceux nous souhaitons établir une estimation de limite de plus confiance avec plus des observations de récompenses. Pour $p=t^{-4}$, nous obtenons l'algorithme dénommé UCB1.

$$U_t(a) = \sqrt{\frac{2\log p}{N_t(a)}} \text{ et } \hat{U}_t^{UCB1} = \operatorname{argmax}_{a \in \mathcal{A}} \hat{Q}_t(a) + \sqrt{\frac{2\log p}{N_t(a)}}$$

6.9 Algorithme de UCB Bayésien

Dans l'algorithme UCB ou UCB1, nous ne pourrions en aucun cas supposer une priorité sur la distribution des récompenses et nous devons donc nous fier à l'inégalité de Hoeffding pour obtenir une estimation très généralisée. Si nous parvenons à connaître la distribution à l'avance, nous pourrions faire une meilleure estimation consolidée.

6.10 Filtrage collaboratif, problème bandit de Bernoulli, biclustering et Processus de Décision Markovien (MDP)

Vu la croissance exorbitante des données sur l'internet, les besoins en systèmes de recommandation augmentent. De plus, De nos jours, les systèmes de recommandation sont intégrés dans tous les domaines en fonction des intérêts et des objectifs de l'entreprise. Par le passé, le système de recommandation a connu un grand succès avec une méthode appelée filtrage collaboratif (CF). Ce dernier est l'une des techniques les plus populaires dans le domaine de recommandation. L'objectif de CF est de faire une prédiction personnalisée des préférences des utilisateurs en utilisant les informations relatives de départ et d'autres des utilisateurs ayant les mêmes intérêts.

De ce fait, le filtrage collaboratif présente des limites. C'est limites ont occasionné d'autres recherches qui ont abouti à des résultats que nous résumerons dans les lignes suivantes :

- L'un des inconvénients de CF est qu'il ne considère que l'un des deux dimensions (utilisateurs ou objet). C'est ce qui rend difficile la détection de modèles qui devraient être capturés en prenant en compte les deux dimensions ;
- De plus, la matrice de données qu'un système de recommandation est de grande dimension, car il contient un grand nombre d'objets disponibles, dont beaucoup ne sont jamais

exploités ;

Ces deux faits ont conduit au développement de systèmes de recommandation basés sur le biclustering [2].

- Un autre inconvénient de CF est qu'il est statique, donc il est généralement pas possible de refléter la réponse d'un utilisateur en temps réel ;

Ce point a suscité la nécessité d'un système de recommandation basé sur le MDP.

6.11 Conclusion

Cette partie, consacré à l'étude bibliographique, nous a permis d'étudier les principaux approches, les différents algorithmes utilisés et leur mode de fonctionnement en système de recommandation. Dans la partie suivante, nous allons présenter et justifier le choix de la solution que nous adopterons pour réaliser notre projet.

7 Choix de la solution

Suite à l'analyse et à l'étude bibliographique du sujet, vu l'importance et l'intérêt que présente le système de recommandation, vue l'explosion d'informations, les besoins en système de recommandation s'élargissent. Beaucoup des recherches ont été développées dans le domaine et diverses approches ont été suggérées pour fournir des recommandations significatives, utiles aux utilisateurs. Dont nous pouvons les cités dans les lignes suivantes :

L'une des approches proposées consiste à considérer un système de recommandation en tant que problème de processus de décision de Markov (PDM) et à tenter de le résoudre en utilisant l'apprentissage par renforcement (RL). Cependant, les méthodes existantes basées sur RL présentent un inconvénient évident. Pour résoudre un MDP dans un système de recommandation, un problème s'est posé avec des grands nombres d'actions discrètes, ce qui amène RL à une classe de problèmes plus large.

Autre approche est de formuler un système de recommandation comme un jeu gridworld, ainsi en utilisant la technique de biclustering peut réduire considérablement l'espace d'état et d'action. Cella, nous ramènera à nouveau le système de recommandation dans RL. Pour ceux, l'utilisation de biclustering non seulement réduit l'espace, mais améliore également la qualité de la recommandation et traite efficacement le problème du démarrage à froid.

Suite aux points ci-dessus, nous recommandons l'environnement de développement et Algorithmes ci-dessous pour notre travail :

Environnement

- GYM;
- Library Tensorflow;
- Library OpenAI Gym;
- Jupyter Notebook;
- Library Anaconda;
- Google colaboratory;
- Language Python;
- Jeux de données : Movielens Dataset.

Algorithmes

- Algorithmes de gradients politiques;
- Markov decision process (MDP)
- Limites de confiance supérieures (UCB)
- Algorithme ϵ -greedy, UCB Bayésien, UCB linéaire;
- Approche collaborative;
- Approche basé sur le contenu;
- Factorisation matricielle (pour représenter la structure de nos données de façon plus concise et pertinente).

8 Réalisation pratique, Expérimentations, Analyse des résultats

Cette partie concerne la mise en œuvre de notre recherche. Elle consiste à effectuer le codage, faire des tests et analyser les résultats. Nous implémenterons quelques approches de recommandation et quelques algorithmes. Nous présenterons les différents résultats sous des captures d'écrans avant de faire le bilan de notre étude.

Pour mieux s'approprier de la technologie, nous avons implémenté ce travail de recherche en deux (2) parties. Une première partie dénommée « **système de recommandation avec filtrage collaboratif** » nous permettra de comprendre le fonctionnement de RS basé sur filtrage collaboratif et de même analyser notre data set. La seconde partie qui concerne l'objet de notre travail dénommée « **système de recommandation avec apprentissage par renforcement** »

8.1 cas de système de recommandation avec filtrage collaboratif

Pour la réalisation de cette partie, nous avons utilisé Google colabollatory. Nous avons sollicité dernier pour des raisons de notre jeux de données est de taille considérable et notre machine n'arrive pas à le supporter lors de l'entraînement [1].

8.1.1 Qu'est-ce que Google Colaboratory et quels en sont les avantages ?

Google Colaboratory ou Colab, un outil Google simple et gratuit pour vous initier au Deep Learning ou collaborer avec vos collègues sur des projets en science des données.

Colab permet :

- d'améliorer vos compétences de codage en langage de programmation Python ;
- de développer des applications en Deep Learning en utilisant des bibliothèques Python populaires telles que Keras, TensorFlow, PyTorch et OpenCV ;
- d'utiliser un environnement de développement (Jupyter Notebook) qui ne nécessite aucune configuration.

Mais la fonctionnalité qui distingue Colab des autres services est l'accès à un processeur graphique GPU, totalement gratuit ! Des informations détaillées sur le service sont disponibles sur la page **FAQ de Colab**.

8.1.2 data set Movielens 20M⁶

est un jeux de données stable de référence de films. il compte 27278 films, 20000263 évaluations et au moins 138000 utilisateurs. il contient six (6) fichiers, à savoir movies.csv, ratings.csv, tags.csv, genome-tags.csv, genome-scores.csv, links.csv.

Nous avons téléchargé le data set Movielens 20M et nous l'avons chargé dans notre Drive pour l'utiliser dans google colab.

6. <https://grouplens.org/datasets/movielens/20m/>

8.1.3 Chargement et prévisualisation des données

```
[ ] print('movies.csv: ')
movies = pd.read_csv(os.path.join(DATA_PATH, DATA_SET_NAME, 'movies.csv'), index_col=None)
movies.describe()
movies.head(5)
```

movies.csv:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy
1	2	Jumanji (1995)	Adventure Children Fantasy
2	3	Grumpier Old Men (1995)	Comedy Romance
3	4	Waiting to Exhale (1995)	Comedy Drama Romance
4	5	Father of the Bride Part II (1995)	Comedy

FIGURE 5 – contenu du fichier vidéo

```
print('ratings.csv: ')
ratings = pd.read_csv(os.path.join(DATA_PATH, DATA_SET_NAME, 'ratings.csv'), index_col=None)
ratings.describe()
ratings.head(5)
```

ratings.csv:

	userId	movieId	rating	timestamp
0	1	2	3.5	1112486027
1	1	29	3.5	1112484676
2	1	32	3.5	1112484819
3	1	47	3.5	1112484727
4	1	50	3.5	1112484580

FIGURE 6 – contenu du fichier ratings

```
print('tags.csv: ')
tags = pd.read_csv(os.path.join(DATA_PATH, DATA_SET_NAME, 'tags.csv'), index_col=None)
tags.describe()
tags.head(5)
```

tags.csv:

	userId	movieId	tag	timestamp
0	18	4141	Mark Waters	1240597180
1	65	208	dark hero	1368150078
2	65	353	dark hero	1368150079
3	65	521	noir thriller	1368149983
4	65	592	dark hero	1368150078

FIGURE 7 – contenu du fichier tags


```
print('genome-tags.csv: ')
genome_tags = pd.read_csv(os.path.join(DATA_PATH, DATA_SET_NAME, 'genome-tags.csv'), index_col=None)
genome_tags.describe()
genome_tags.head(5)
```

genome-tags.csv:

	tagId	tag
0	1	007
1	2	007 (series)
2	3	18th century
3	4	1920s
4	5	1930s

FIGURE 8 – contenu du fichier *gnome-tags*

```
print('genome-scores.csv: ')
genome_scores = pd.read_csv(os.path.join(DATA_PATH, DATA_SET_NAME, 'genome-scores.csv'), index_col=None)
genome_scores.describe()
genome_scores.head(5)
```

genome-scores.csv:

	movieId	tagId	relevance
0	1	1	0.02500
1	1	2	0.02500
2	1	3	0.05775
3	1	4	0.09675
4	1	5	0.14675

FIGURE 9 – contenu du fichier *gnome-scores*

8.1.4 Simple statistique sur les données

Cette statistique nous aidera à mieux comprendre les données.

Le nombre de films : 27278

Le nombre d'évaluations : 20000263

valeur minimale de notation : 0.5

valeur maximale de notation : 5.0

Le nombre d'utilisateurs dans ratings.csv : 138493

Le nombre minimum de notes par utilisateur dans ratings.csv : 20

Le nombre maximum de notations par utilisateur dans ratings.csv : 9254

Le nombre de films dans ratings.csv : 26744

Le nombre minimum de notes par film dans ratings.csv : 1

Le nombre maximum de cotes par film dans cotes.csv : 67310

La longueur de genome_scores.csv : 11709768 Valeur maximale de pertinence de genome_scores.csv : 1.0 valeur minimale de pertinence de genome_scores.csv : 0.0002499999999999997247

Le nombre de films dans genome_scores.csv : 10381 Le nombre minimum de tags par film dans genome_scores.csv : 1128 Le nombre maximum de tags par film dans genome_scores.csv : 1128

Nombre de films dans genome_scores.csv et ratings.csv : 10370. Occupent 53,0 % du fichier ratings.csv.

Nombre de notes où movieId est dans genome_scores.csv : 19800443. Occupe 99.0% du fichier ratings.csv.

8.1.5 Hypotheses et model de l'apprentissage automatique

Hypothèse

- Les balises de genome-tags.csv constituent l'ensemble complet des espaces vectoriels de balises. Les autres balises ne figurant pas dans genome-tags.csv sont des combinaisons linéaires de balises dans le genome-tags.csv ;
- La fonctionnalité des films peut être parfaitement représentée par des balises dans genome-tags.csv, telles que le vecteur de pertinence dans genome_scores.csv. Le vecteur de pertinence dans genome_scores.csv est correct et peut représenter la fonctionnalité de films ;
- Ignorer la qualité des films ;
- Nous ne pouvons pas obtenir d'autres informations sur les films en dehors de l'ensemble de données. Donc, nous n'utilisons pas links.csv ;
- La durée de sortie des films n'affecte pas ;
- L'horodatage dans ratings.csv : et tags.csv n'affecte pas.

8.1.6 Création de données d'entraînement

Nous avons subdivisé 19800443 données évaluées et classifiées (du fichier genome_scores.csv selon movieId) en (80%) de données d'entraînement et en (20%) de données de test. Nous pouvons constater que les données test contient 99,86% des utilisateurs. le chargement des données pretraités se feront dans la cellule de code de la section suivant.

8.1.7 Graphe d'entraînement et de perte

A la fin de l'entraînement de notre modèle, ci-dessous les deux graphes qui schématisent la perte dans la phase de l'entraînement de notre jeux de données et la perte dans la phase de test.

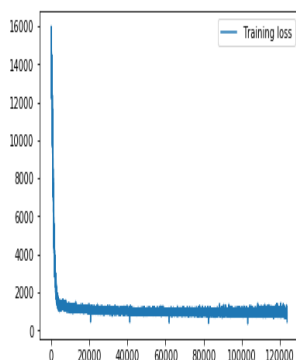


FIGURE 10 – *perte d'entraînement*

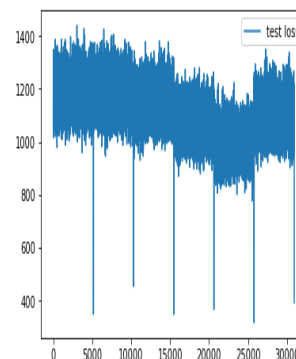


FIGURE 11 – *perte de test*

8.1.8 Analyse des résultats et évaluation

Nous pouvons voir que la perte de test diminue puis augmente lentement, Cela suppose que nous obtiendrons un meilleur résultat en modifiant les paramètres. Par contre la perte d'entraînement a chuté dès le début de l'entraînement de notre modèle. ce qui prouve que notre jeux de données est fortement stable.

8.1.9 système de recommandation

Sur la base du modèle présenté, nous avons créer un système de recommandation capable de :

- Donner une liste de films liés à un film donné en entré ;
- prédire les films préférés des utilisateurs a travers leur identifiant ;
- Recommander des films de souhait à un utilisateur par le billet d'un filme qu'il a regardé précédemment.

8.2 cas de système de recommandation avec apprentissage par renforcement

8.2.1 Modélisation contextuelle des bandits multiples

Le problème de la recommandation de film peut être vu comme un problème de bandits multi-armés textuelisé où chaque bras est un film et un utilisateur peut le tirer (est recommandé de le regarder). Le contexte de chaque bras (film) est un vecteur de caractéristiques conservant les préférences de l'utilisateur et les fonctionnalités du film. la récompense (évaluation) obtenue en tirant le bras (regardant le film) dépend de ce contexte. Lorsque la note est retournée, les préférences de l'utilisateur sont mises à jour en fonction du film caractéristiques et la cote. Cela soulève une préoccupation : comme le nombre de films augmente géant, certains films peuvent ne pas être regardés. Pour être sur de les films ne sont pas répétés trop souvent, un paramètre de nouveauté doit être utilisé⁷.

7. <https://github.com/PierreGe/RL-movie-recommender>

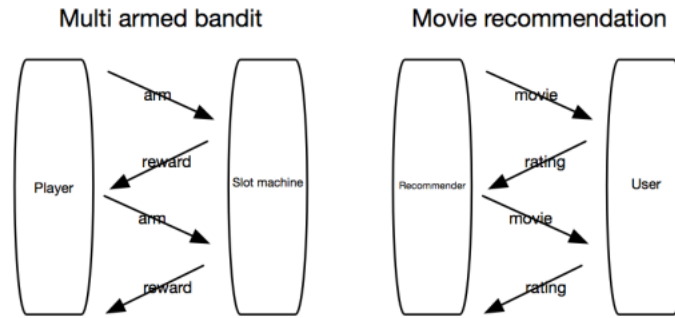


FIGURE 12 – *Vu d’un système de recommandation de film en tant que problème de bandit multi-armés contextualisé*

8.2.2 Extraction et sélection attributs

premièrement Nous avons extrait le jeux de données Movilens 20M afin quel soit plus clair. Pour le soucie de réduire la complexité d’utilisation de nos algorithmes, nous avons eu à unir les différents fichiers qui constitue notre jeux de données Movilens 20M en un deux fichiers **csv**. Ces fichiers sont constitués uniquement des attributs qui ont des impacts directs sur un système recommandation.

secundo Pour éviter de se confronter au problème de transformation de données vu la taille de notre jeux de données en terme d’évaluation, de même en tenant compte de la capacité de notre ordinateur, nous avons choisi les 1000 films ayant plus de voix et de même nous avons choisi tous les utilisateurs ayant au moins 50 avis sur ces films.

	rating	votes	Action	Adventure	Animation	Biography	Comedy	Crime	Drama	Family	...	2007	2008	2009	2010	2011	2012	2013	2014	:
title																				
Before Sunrise (1995)	0.81	190365	0	0	0	0	0	0	1	0	...	0	0	0	0	0	0	0	0	0

1 rows × 52 columns

FIGURE 13 – *Espace de définition des films*

	user	rating	link
movie			
The Shawshank Redemption (1994)	20255559	1.0	tt0111161

FIGURE 14 – *vu de films sous évaluation des utilisateurs*

8.2.3 Résultat et expérimentation

Nous avons construit un système de recommandation contextualisé basé sur l’apprentissage par renforcement de film complètement fonctionnelle et tient compte l’exploration de données en temps réel.

Ci-dessous les résultats traduisent en des graphes d'une expérimentation pour 1000 films suivant des utilisateurs qui ont au moins fait 50 évaluations ces films. Nous avons utilisé une probabilité de récompense initialisée de manière aléatoire selon les différents algorithmes utilisés.

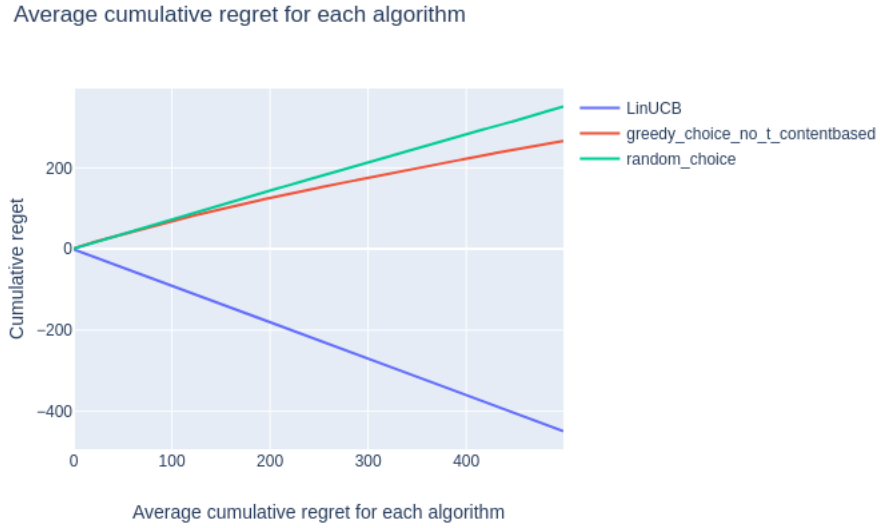


FIGURE 15 – *Regret moyen cumulatif pour chaque algorithme*

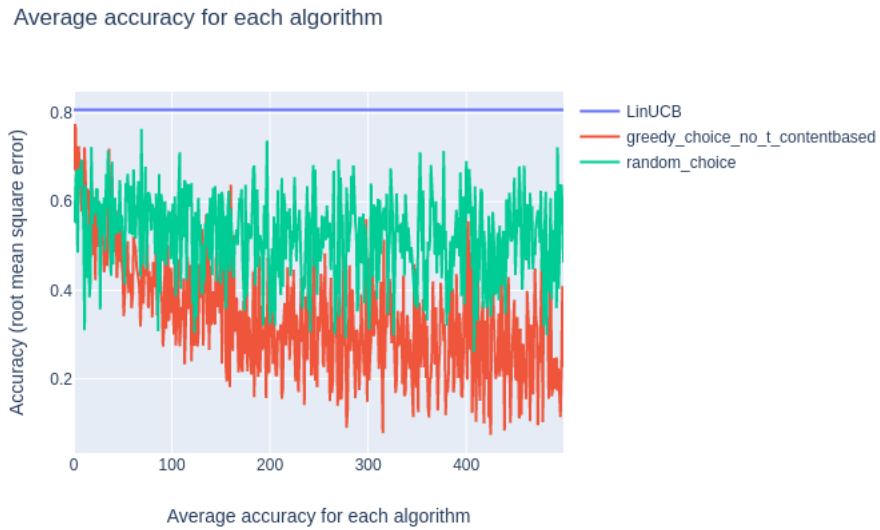


FIGURE 16 – *Comparaison des évaluations moyennes des utilisateurs pour chaque algorithme*

Average accuracy for each algorithm

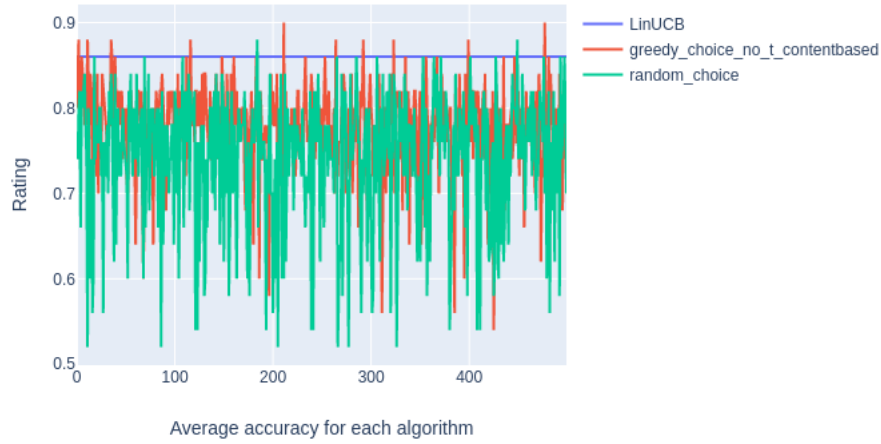


FIGURE 17 – Précision moyenne pour chaque algorithme

Average running time for each algorithm

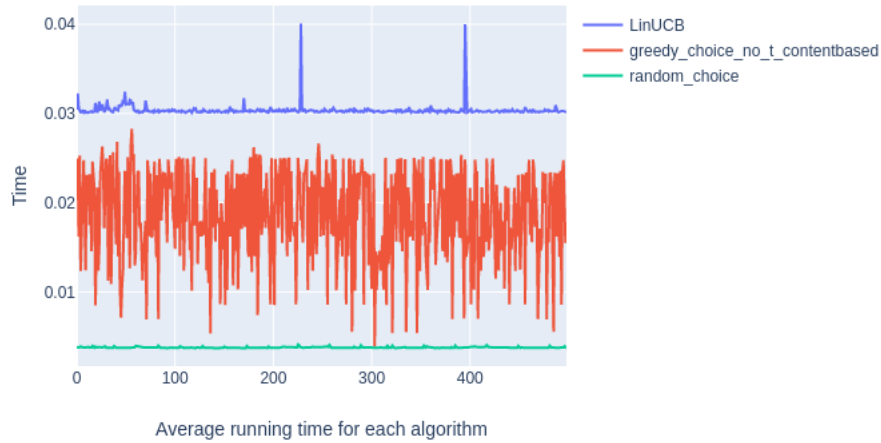


FIGURE 18 – Durée moyenne d'exécution pour chaque algorithme

8.3 Conclusion

Notre solution est implémentée à l'aide des librairies anaconda, dans Jupyter Notebook suivant les algorithmes de Limites de confiance supérieures (UCB), Algorithme ϵ -greedy, UCB Bayésien, UCB linéaire. De même nous pu cohabiter l'approche apprentissage par renforcement avec approche basé sur le contenu et avec approche basé sur le filtrage collaboratif. Bien que nous n'avons pas encore pu ressortir le graphe comparatif de ces trois approches dans notre modèle. Le bilan de notre TPE dont la conclusion générale intervient à la page suivante.

9 Conclusion générale

Tout au long de ce travail de recherche, nous avons pu cerner les efforts des uns et des autres dans le sens de vouloir toujours aller vers le meilleurs. Plusieurs thèses, articles, études, analyses, conférences et propositions sont publiés dans ce sens. Pour ceux, nous dirons que ce travail personnel encadré a été pour nous l'occasion d'une émulation intellectuelle dans le domaine de la recherche, Machine learning, Reinforcement learning, Deep learning, système de recommandation ...

Nous avons pu apprendre à faire un travail de recherche mais aussi à persévérer dans les difficultés qui se sont présenteront à nous.

Nous envisageons une autre occasion ou nous présenterons :

- nos recommandations sur les processus de mis en place de système de recommandation basé sur l'apprentissage par renforcement ;
- le graphe de fonctionnement de Système de recommandation basé sur l'apprentissage par renforcement, basé sur le contenu et basé sur le filtrage collaboratif dans un espace. ;
- Library OpenAI Gym ;
- Le résultat de RS dans l'environnement GYM avec la librairie OpenAI Gym.

Références

- [1] Idir Benouaret. Un système de recommandation sensible au contexte pour la visite de musée. In *CORIA*, pages 515–524, 2015.
- [2] Sungwoon Choi, Heonseok Ha, Uiwon Hwang, Chanju Kim, Jung-Woo Ha, and Sungroh Yoon. Reinforcement learning based recommender system using biclustering technique. *arXiv preprint arXiv :1801.05532*, 2018.
- [3] Nick Golovin and Erhard Rahm. Reinforcement learning architecture for web recommendations. In *International Conference on Information Technology : Coding and Computing, 2004. Proceedings. ITCC 2004.*, volume 1, pages 398–402. IEEE, 2004.
- [4] Jonathan Louëdec. *Stratégies de bandit pour les systèmes de recommandation*. PhD thesis, 2016.
- [5] Jonathan Louëdec, Laurent Rossi, Max Chevalier, Aurélien Garivier, and Josiane Mothe. Algorithme de bandit et obsolescence : un modèle pour la recommandation. 2016.
- [6] Per Martin-Löf. The definition of random sequences. *Information and control*, 9(6) :602–619, 1966.