Le coaching : un nouveau cadre pour la recommandation automatique en vue de modifications durables du comportement

Jules Vandeputte, Antoine Cornuéjols, Nicolas N. Darcel, F Delaere, Christine Martin

Le coaching : un nouveau cadre pour la recommandation automatique en vue de modifications durables du comportement

J. Vandeputte[1], A. Cornuéjols[1], N. Darcel[2], F. Delaere[3], Ch. Martin[1]

[1] UMR MIA-Paris, AgroParisTech, INRAe, Université Paris-Saclay. 16, rue Claude Bernard. Paris (France)
[2] UMR PNCA, AgroParisTech, INRAe, Université Paris-Saclay. 16, rue Claude Bernard. Paris (France)
[3] Danone Nutricia Research. Palaiseau (France)

jules.vandeputte@agroparistech.fr

Résumé

Cet article introduit un nouveau scénario de recommandation : le coaching. Dans ce scénario, l'objectif est de d'aider un utilisateur à modifier durablement ses propres préférences, dans un contexte de choix répétés. À chaque fois que l'utilisateur exprime un choix, le coach peut lui suggérer une modification afin de le guider vers de meilleures habitudes. Nous montrons que la meilleure stratégie du coach dépend des caractéristiques de l'utilisateur. Six stratégies de coaching, dans lesquelles le coach apprend les caractéristiques de l'utilisateur en cours d'interactions ont été comparées.

Mots-clés

Recommandation. Apprentissage. Apprentissage par renforcement.

Abstract

This article introduces a new recommendation scenario called coaching. In this scenario, the goal is to help the user modifying his consumption habits lastingly. Each time the user U expresses a choice, the coach can suggest a modification in order to guide U towards better habits. We show that the best coaching strategy depends on the user's characteristics. Six coaching strategies, in which the coach learns the user's characteristics during interactions, have been compared on an illustrative example.

Keywords

Recommending Systems. Machine Learning. Reinforcement Learning.

1 Introduction

Les développements récents, notamment de méthodes telles que l'apprentissage profond par renforcement, ont amené ces dernières années à de nombreuses avancées dans le domaine des systèmes de recommandation. Ils sont ainsi largement utilisés afin d'aider les utilisateurs de diverses plate-formes à gérer la surcharge d'information à laquelle ils font face. S'il existe quelques études s'intéressant aux effets de tels systèmes sur les préférences de l'utilisateur, la vaste majorité des travaux se concentre sur la manière d'apprendre au mieux ces préférences, afin de fournir à l'utilisateur à chaque instant t la ou les propositions les plus susceptibles d'être acceptée(s). Ces systèmes de recommandation visent donc une maximisation du nombre de recommandations acceptées par l'utilisateur à chaque instant t sans que la notion d'histoire des préférences ne soit prise en compte. Cependant, une recommandation à un instant t peut avoir un effet sur les choix futurs de l'utilisateur.

Le but de cet article est de proposer un nouveau cadre permettant d'étudier et de concevoir des systèmes de recommandation dont l'objectif est d'accompagner un utilisateur dans un processus de modification durable de ses préférences. Ce pourrait par exemple être le cas d'un utilisateur souhaitant améliorer son régime nutritionnel en apprenant progressivement à modifier ses choix d'aliments.

Le principe est de raisonner en terme de trajectoire parcourue par l'utilisateur dans l'espace des préférences, le rôle du système de recommandation étant de faire parcourir à l'utilisateur une trajectoire l'amenant, depuis son habitude initiale de choix, à une habitude de choix meilleure, voire optimale, selon une certaine fonction de score. Une originalité de l'approche proposée ici est que le système de recommandation ne propose pas des items à l'utilisateur, mais s'appuie sur les préférences exprimées par celui-ci, son choix d'items à l'instant t, pour proposer éventuellement une modification acceptable de ce choix. En ceci, ce type de systèmes de recommandation mérite d'être appelé « système de *coaching* », car il agit comme un coach personnalisé qui analyse les choix de l'utilisateur pour suggérer des modifications bénéfiques à terme, comme le coach d'un sportif guiderait ce dernier pour améliorer ses performances.

Cet article est organisé comme suit : la section 2 définit le coaching, et la manière dont le coach et l'utilisateur interagissent. La section 3 analyse les travaux pertinents par rapport au nouveau cadre proposé. La section 4 illustre à l'aide d'un exemple simple le problème de coaching et montre en particulier que la stratégie optimale du coach dépend des caractéristiques de l'utilisateur, d'où l'intérêt d'un coaching personnalisé. Plusieurs méthodes d'apprentissage sont comparées expérimentalement dans la section 5 sur un cas simplifié mais représentatif des problèmes de coaching

possibles. La section 6 évoque des pistes pour adapter le système de coaching à des scénarios plus généraux. Finalement, la section 7 tire des leçons de cette étude novatrice dans les systèmes de recommandation.

2 Le coaching : un nouveau cadre pour la recommandation

Le coaching consiste à chercher une trajectoire acceptable par l'utilisateur pour le conduire vers une habitude de choix meilleure. Afin de formaliser ceci, il faut définir les notions de préférence de choix et de score.
Formellement, nous considérons un ensemble $\mathcal{I}$ d'items représentant des choix possibles pour l'utilisateur, et nous supposerons, dans le travail présenté ici, qu'un score peut être attribué à chaque item $i \in \mathcal{I}$ indépendamment des autres. Il s'agit donc d'un score additif. D'autres types de score seraient envisageables, comme nous l'évoquons en section 6

$$Sc : \begin{cases} \mathcal{I} \to \mathbf{R} \\ i \mapsto \text{score}(i) = Sc_i \end{cases}$$

Dans la suite, afin de simplifier l'exposé, nous supposons que l'utilisateur ne propose qu'un seul item à chaque pas de temps. Nous représentons alors une habitude de choix à un instant t par un vecteur de probabilités défini sur l'ensemble des items définis précédemment : $\Pi_t = (\pi_t(i))_{i\in\mathcal{I}}$, $\pi_t(i)$ représentant la probabilité de choix de l'item i à t. On peut alors associer à un tel vecteur Π_t une espérance de gain selon le score :

$$V[\Pi_t] = \sum_{i\in\mathcal{I}} \pi_t(i) \cdot \text{score}(i) \tag{1}$$

2.1 L'interaction entre l'utilisateur et le coach : un jeu itéré à deux joueurs

La modification durable des habitudes de choix d'un utilisateur suppose des interactions entre l'utilisateur et le coach prenant place dans le temps. Il est naturel de modéliser ce processus par un jeu itéré à deux joueurs : C le coach et U l'utilisateur. Nous proposons ici un mécanisme d'interaction en quatre temps.

1. U fait à C une proposition d'item, par exemple i, en utilisant son vecteur de préférences Π_t.
2. C analyse la proposition de U, et suggère, s'il le juge utile, une proposition de modification $i \to j$, à partir de ses connaissances de la valeur des items, et de son estimation de la capacité de U à accepter la proposition.
3. U accepte ou refuse la proposition de substitution fournie par C.
4. Si U accepte la proposition de C, il modifie, en fonction de sa capacité d'apprentissage, son vecteur de préférences Π_t de sorte à proposer de lui même plus fréquemment l'item recommandé. Sinon U ne modifie pas le vecteur de préférence. C'est ainsi que l'on rend compte du fait que U peut apprendre au fil de ses interactions avec C, l'idée étant que U, s'il accepte la modification $i \to j$ proposée par C, est davantage prêt à choisir j à l'avenir au lieu de i.

Une suggestion de substitution $i \to j$ sera plus ou moins acceptable ou réalisable selon l'utilisateur U concerné.

2.2 Modèlisation de l'utilisateur

L'utilisateur est caractérisé par trois comportements :

1. Le **choix** d'un item i selon son vecteur de probabilité instantané Π_t.
2. Son **acceptation ou refus** de la proposition de substitution par le coach. Nous supposerons que ce comportement est contrôlé par une matrice $\mathbf{M}$: $\mathcal{I}\times\mathcal{I} \to [0,1]$ exprimant pour chaque couple d'item $(i,j) \in \mathcal{I}^2$ la substituabilité entre i et j (ie. la probabilité que la substitution $i \to j$ soit acceptée). Dans la suite de cet article, nous supposerons cette matrice constante. Elle serait cependant susceptible de dépendre de t.
3. **Apprentissage** par mise à jour de son vecteur de probabilité Π_t quand la substitution $i \to j$ est acceptée. Afin de traduire une transmission de probabilité de l'item i à l'item j, on modélise cet apprentissage de la manière suivante :

$$\forall t \in \mathcal{T} : \begin{cases} \Pi_{t+1}(i) \;=\; (1-\lambda)\,\Pi_t(i) \\ \Pi_{t+1}(j) \;=\; \Pi_t(j) \;+\; \lambda\,\Pi_t(i) \end{cases} \tag{2}$$

Le paramètre λ représente le taux d'apprentissage de l'utilisateur. Si $\lambda = 0$, alors l'utilisateur n'apprend rien et ne modifie pas ses préférences au fil des interactions. Si $\lambda = 1$, l'utilisateur est un apprenant « parfait », et après chaque substitution $i \to j$ acceptée il transfère l'intégralité de sa probabilité de proposer i vers sa probabilité de proposer j.

2.3 Modélisation du coach

Nous définissons le coach par sa fonction de choix de substitution qui peut évoluer avec le temps $c_t : i \in \mathcal{I} \to j \in \mathcal{I}$. Cette fonction de choix dépend de :

1. La fonction de score Sc qu'il connaît.
2. L'estimation des caractéristiques de U : Π_t, $\mathbf{M}$ et λ.

On fait ici l'hypothèse que le coach représente l'utilisateur à l'aide de ces trois caractéristiques. Celles-ci étant propres à chaque utilisateur, le coach devra chercher à les estimer au fil de ses interactions avec ce dernier.

2.4 Comment évaluer un coach

Le coaching a pour but de faire suivre à l'utilisateur U une trajectoire dans l'espace des préférences, c'est-à-dire, ici, dans l'espace des vecteurs de probabilité Π_t. L'évaluation d'une stratégie de coaching, c'est-à-dire de fonction de choix c_t au cours du temps, doit donc se définir par rapport à ces trajectoires.
Notons $\Pi^\star$ les préférences optimales selon la fonction de score Sc Leur valeur associée est : $V^\star = V(\Pi^\star)$, la valeur maximale atteignable. Par ailleurs, nous noterons $V(\Pi_0)$ l'espérance de score associée au vecteur de préférence initial de l'utilisateur.

Plusieurs options sont envisageables. Nous en mentionnons trois ici :

1. On cherche à guider U vers un vecteur de préférence Π tel que : $V(\Pi) \geq \gamma V^\star$ avec $\gamma \in [0,1]$. La performance du coach est ainsi mesurée en terme de *nombre moyen d'interactions* $\overline{T}_\gamma$ pour atteindre ce niveau de performance à partir du vecteur initial Π_0.
2. Une mesure duale consiste à mesurer le *gain de performance moyen* $\overline{V}_T = \text{moyenne}\big(V(\Pi_T) - V(\Pi_0)\big)$ après T interactions.
3. Il est également possible d'envisager un critère défini sur l'ensemble de la trajectoire suivie par l'utilisateur, par exemple le *gain cumulé sur l'ensemble de la trajectoire* par rapport à l'espérance de gain initiale : $G(T) = \sum_{t=1}^{T}\big(V(\Pi_t) - V(\Pi_0)\big)$.

Dans la suite de cet article, nous nous concentrerons sur le deuxième critère. Il permet en effet des comparaisons aisées, notamment dans le cas d'un utilisateur réel, ayant avec le système un nombre d'interactions limité.

3 Travaux reliés

La recommandation en vue de modifier durablement les préférences d'un consommateur a été peu étudiée jusqu'à présent, et les travaux s'attaquant à cette question ont principalement porté sur la conception d'interfaces informatiques incitant l'utilisateur à les utiliser et à en suivre les indications. Ces travaux ressortent davantage de l'étude des caractéristiques psychologiques mises en jeu lors des interactions avec des interfaces homme-machines (voir par exemple [7]).
Nous distinguons ici les travaux relatifs à la recommandation, et ceux donnant une modélisation mathématique permettant d'étudier le coaching comme un problème d'optimisation.

Point de vue des systèmes de recommandation

Dans [3], les auteurs explorent les effets durables des systèmes de recommandation sur les habitudes de consommation, et montrent qu'ils peuvent engendrer une homogénéisation des comportements. Quelques travaux ont également été publiés sur la recommandation avec pour but de modifier le comportement. Les auteurs de [4] ont proposé en 2012 un algorithme fondé sur une approche de recommandation intra-personnelle : plutôt que de s'intéresser aux comportement des autres utilisateurs, comme dans une approche de type filtrage collaboratif, les auteurs étudient les différents comportements typiques des utilisateurs pour baser leur recommandation. L'algorithme met en lien le comportement de l'utilisateur et la mesure du but recherché, afin de proposer à l'utilisateur des modifications qu'il serait susceptible d'accepter.
Une autre approche, basée sur les modèles de Rasch a également été étudiée dans plusieurs publications [8, 9]. Dans [9] notamment, les auteurs classent les buts poursuivis en fonction de leur difficulté définie selon une échelle de Rasch. Cela permet de conduire les utilisateurs à satisfaire des buts successifs, en commençant par les plus simples. Cette approche se base sur l'hypothèse que plus une recommandation est simple à suivre, et donc fréquemment suivie, plus l'utilisateur va se l'approprier.
Dans [6] sont distingués deux cas : celui où la modification de préférences est initiée par l'utilisateur, et celui où elle est initiée par le système. Ce travail soulève l'importance de la nature progressive de l'évolution des préférences et donc de la nécessité d'en tenir compte en proposant des choix acceptables par l'utilisateur.

Point de vue apprentissage par renforcement

À chaque étape t du coaching, le système de recommandation se trouve face au problème du choix de l'item j à suggérer comme substitution pour l'item i proposé par l'utilisateur, celui-ci ayant été décidé selon le vecteur de probabilité Π_t. Ce choix vise à optimiser le critère de performance décrit en section 2.4 et on a donc :

$$\forall i \in \mathcal{I} : c_t(i) = \underset{j \in \mathcal{I}}{\text{ArgMax}}\, \textit{Perf} \tag{3}$$

Cependant, le critère *Perf*, qui peut être $\overline{V}_T$ ou $\overline{T}_\gamma$ ou $G(T)$, est difficile à optimiser. Il résulte en effet d'un processus doublement stochastique : le choix de l'item i_t à chaque instant t par U, régi par la distribution de probabilité Π_t, et l'acceptation de la substitution $i_t \rightarrow c_t(i_t)$ par U résultant de la matrice de probabilité $\mathbf{M}$, ce qui entraîne un gain instantané et une modification potentielle de Π_t.
L'utilisateur peut être considéré comme l'environnement face auquel le coach doit faire ses choix pour optimiser le critère *Perf*. Cet environnement est markovien car, dans la modélisation proposée, la matrice $\mathbf{M}$ et le coefficient λ sont constants et Π_{t+1} est seulement fonction de Π_t.
Le coaching est donc un problème de décision dans un processus markovien avec un environnement, l'utilisateur U, imparfaitement connu. Le coach est ainsi face à un dilemme exploration vs. exploitation.
Différentes techniques ont été conçues pour optimiser ce compromis. Nous citerons en particulier le problème du *bandit multi-bras* [5], dans lequel l'agent (i.e. le coach) doit choisir à chaque instant un bras (i.e. une recommandation) de manière à optimiser un gain cumulé. Chaque bras est associé à une récompense stochastique. Cependant, ce problème correspond à un environnement stationnaire. C'est pourquoi a été inventé le problème du *bandit contextuel* [1], dans lequel il peut y avoir transition entre des bandits qui soit contrôlée par le bras sélectionné à chaque instant. Le problème du *bandit turbulent* enrichit ce cadre en permettant que les propriétés des bras évoluent également avec le temps. Il est possible plus généralement de considérer la tâche du coach comme celle d'un *apprentissage par renforcement* [10] dans lequel le coach apprend par essais et erreurs les caractéristiques de son environnement et ainsi ce qui correspond ici à sa fonction de choix c_t. De fait, il faut recourir aux *processus de décision markoviens partiellement observables* (POMDP) pour rendre pleinement compte du problème du coaching puisque le coach n'a à chaque instant qu'une connaissance incomplète de l'état de son environnement, l'utilisateur, dont il n'observe que l'item choisi.

Sans développer plus avant ici les correspondances formelles avec le problème du coaching, nous utiliserons dans la suite des techniques tirées de l'apprentissage par renforcement pour aborder le problème du coaching.

4 Étude analytique d'un cas simple

On se propose ici d'étudier de manière analytique, dans un cas simple, le problème de coaching décrit en section 2. L'objectif est de montrer que la meilleure fonction de choix du coach dépend des caractéristiques de l'utilisateur et du critère de performance visé.
On considère un espace d'items à trois éléments $\mathcal{I}$ dont les scores associés sont donnés :

$$\mathcal{I} = \{i_1, i_2, i_3\}, \quad \text{avec : } \begin{cases} score(i_1) = 5 \\ score(i_2) = 20 \\ score(i_3) = 50 \end{cases}$$

et un utilisateur U dont les habitudes initiales sont définies par :

$$\Pi_0 = (1, 0, 0)^\top$$

qui indique qu'initialement U choisit toujours l'item i_1

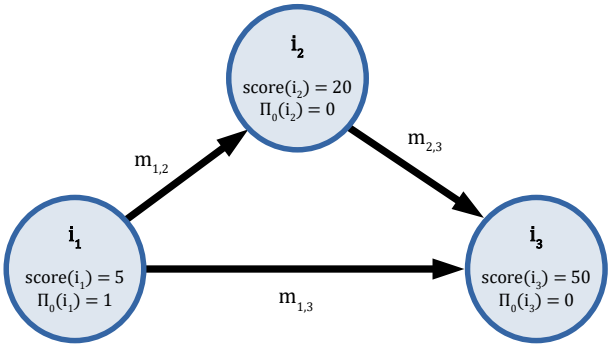

FIGURE 1 – Graphe présentant le scénario simplifié au pas de temps $t = 0$.

On suppose aussi que le coach C connaît Π_0, et maintient un modèle de U basé sur λ et $\mathbf{M}$, avec :

$$\mathbf{M} = \begin{pmatrix} m_{1,1} & m_{1,2} & m_{1,3} \\ m_{2,1} & m_{2,2} & m_{2,3} \\ m_{3,1} & m_{3,2} & m_{3,3} \end{pmatrix} \quad \text{et} \quad \lambda \in [0, 1]$$

La figure 1 résume les données du problème. Comme, par définition, le coach connaît la fonction de score, il est capable d'identifier l'unique vecteur de préférence optimal $\Pi^\star = (0, 0, 1)^\top$, dont l'espérance de gain associée est : $V(\pi^\star) = 50$.
Pour les choix i_2 et i_3 de U, la fonction de recommandation du coach est évidente : $c_t(i_2) = i_3$ et $c_t(i_3) = i_3, \forall t$. Mais quelle est la meilleure recommandation quand U choisit i_1 ?

Examinons la performance associée à chacun des deux choix possibles : i_2 ou i_3.

1. Le coach choisit la substitution directe $i_1 \to i_3, \forall\, t$.
L'espérance du vecteur de préférence pour U à $t + 1$ en fonction de sa valeur en t est donnée par la formule :

$$\begin{aligned} \mathbb{E}[\Pi_{t+1}(i_1)] &= (1 - m_{1,3})\,\Pi_t(i_1) \;+\; m_{1,3}(1 - \lambda)\,\Pi_t(i_1) \\ &= \Pi_t(i_1) \;-\; m_{1,3}\,\lambda\,\Pi_t(i_1) \\ &= \Pi_t(i_1)\,(1 - m_{1,3}\,\lambda) \end{aligned}$$

d'où par récurrence : $\mathbb{E}[\Pi_t(i_1)] \;=\; \Pi_0(i_1)\,(1 - m_{1,3}\lambda)^t$.
De plus, les seuls items considérés dans ce cas étant i_1 et i_3 on a $\Pi_t(i_3) = 1 - \Pi_t(i_1)$, d'où :

$$\forall t \in \{0, \dots, T\} : \begin{cases} \mathbb{E}[\Pi_t(i_1)] = (1 - m_{1,3}\lambda)^t \\ \mathbb{E}[\Pi_t(i_2)] = 0 \\ \mathbb{E}[\Pi_t(i_3)] = 1 - (1 - m_{1,3}\lambda)^t \end{cases} \tag{4}$$

2. Le coach choisit la substitution indirecte $i_1 \to i_2$.
Par le même raisonnement que précédemment, on a :

$$\mathbb{E}[\Pi_t(i_1)] \;=\; \Pi_0(i_1)\,(1 - m_{1,2}\lambda)^t$$

Par ailleurs :

$$\begin{aligned} \mathbb{E}[\Pi_{t+1}(i_2)] &= \Pi_t(i_2) \;+\; m_{1,2}\,\lambda\,\Pi_t(i_1) \;-\; m_{2,3}\,\lambda\,\Pi_t(i_2) \\ &= \Pi_t(i_2)\,(1 - m_{2,3}\,\lambda) \;+\; m_{1,2}\,\lambda\,\Pi_t(i_1) \end{aligned}$$

qui est la version discrète de l'équation différentielle :

$$\frac{\mathrm{d}\,\Pi_t(i_2)}{\mathrm{d}t} \;=\; -\ln(1 - \lambda\, m_{1,2})\,\Pi_t(i_1) + \ln(1 - \lambda\, m_{2,3})\,\Pi_t(i_2)$$

pour laquelle la solution est connue :

$$\begin{aligned} \Pi_t(i_2) \;&= \Pi_0(i_2)\,(1 - \lambda\, m_{2,3})^t \\ &\quad + \Pi_0(i_1)\,\frac{-\ln(1 - m_{1,2}\,\lambda)\big((1 - m_{1,2}\,\lambda)^t - (1 - m_{2,3}\,\lambda)^t\big)}{-\ln(1 - m_{2,3}\,\lambda) + \ln(1 - m_{1,2}\,\lambda)} \end{aligned}$$

Ici $\Pi_0(i_2) = 0$, on a donc finalement $\forall t \in \{0, \dots, T\}$:

$$\begin{cases} \mathbb{E}[\Pi_t(i_1)] \;=\; \Pi_0(i_1)\,(1 - m_{1,2}\lambda)^t \;=\; (1 - m_{1,2}\lambda)^t \\ \mathbb{E}[\Pi_t(i_2)] \;=\; \frac{-\ln(1 - m_{1,2}\,\lambda)\left((1 - m_{1,2}\,\lambda)^t - (1 - m_{2,3}\,\lambda)^t\right)}{-\ln(1 - m_{2,3}\,\lambda) + \ln(1 - m_{1,2}\,\lambda)} \\ \mathbb{E}[\Pi_t(i_3)] = 1 - \mathbb{E}[\Pi_t(i_1)] - \mathbb{E}[\Pi_t(i_2)] \end{cases} \tag{5}$$

À chaque instant t ; le coach estime la matrice $\mathbf{M}$ à partir de l'observation des interactions avec U. Soit cette estimation valant par exemple :

$$\widehat{\mathbf{M}} = \begin{pmatrix} 1 & 0.15 & 0.1 \\ 0 & 1 & 0.2 \\ 0 & 0 & 1 \end{pmatrix}$$

On peut alors comparer les choix possibles pour le coach : $c_t(i_1) \;=\; i_3$ (Eq. (4)) et $c_t(i_1) \;=\; i_2$ (Eq. (5)), $\forall t$. Il est apparent que les expressions (4) et (5) ont leurs valeurs qui dépendent du facteur d'apprentissage λ de U et du nombre de pas d'interactions t.
Selon la valeur T du nombre maximal d'interactions, il est possible que le choix optimal $c_t(i_1)$ dépende de λ. Ici par exemple, il existe une valeur critique λ_c du paramètre λ telle que si $\lambda > \lambda_c$ (resp. $\lambda \leq \lambda_c$) il faut choisir $c_t(i_1) = i_2$ (resp. $c_t(i_1) = i_3$) pour optimiser la performance $\overline{V}_T$.

T	λ_c
10	1
20	≈ 0.6028
50	≈ 0.2615
100	≈ 0.1346
1000	≈ 0.0138

TABLE 1 – Valeurs de λ_c en fonction de T.

On observe (voir la table 1) que plus le nombre d'itérations T est important, plus la valeur de λ_c diminue, c'est -à-dire que l'apprentissage par U rend le chemin indirect $i_1 \to i_2 \to i_3$ plus intéressant que le chemin direct $i_1 \to i_3$. Dans cet exemple, pour un T donné, on voit que l'estimation des valeurs de λ et de $\mathbf{M}$, si elle est bien faite, permet au coach de déterminer le meilleur choix de recommandation $c_t(\cdot)$.
Il nous faut maintenant examiner comment le coach peut estimer ces valeurs, ou bien directement le meilleur choix $c_t(i)$ lorsque l'utilisateur propose l'item i à l'instant t.

5 Évaluation expérimentale

Dans cette section, nous comparons expérimentalement des stratégies classiques de la littérature pour répondre au dilemme exploitation vs. exploration du coach, notamment inspirées de l'apprentissage par renforcement et des problèmes de *bandit multi-bras*.

5.1 Présentation des stratégies testées

Quatre stratégies classiques et une variante ont été testées dans nos expériences et comparées également à une stratégie de choix aléatoire de substitution qui sert de référence.

1. La stratégie *gloutonne* recommande la substitution qui maximise l'espérance de gain instantanée de score :

$$c_t(i) \;=\; \underset{j\in\mathcal{I}}{\text{ArgMax}}\big\{m_{i,j}\,\big(\text{score}(j)-\text{score}(i)\big)\big\}$$

En fonction du comportement observé chez l'utilisateur (ie. refus ou acceptation), le coach met à jour ses estimations des probabilités d'acceptation $m_{i,j}$ de U selon :

$$\widehat{m}_{i,j}^{t+1} = \begin{cases} \frac{\widehat{m}_{i,j}^{t}+1}{n_{i,j}} & \text{si la propositon } i\to j \text{ est acceptée} \\ \frac{\widehat{m}_{i,j}^{t}}{n_{i,j}} & \text{sinon} \end{cases}$$

avec $\widehat{m}_{i,j}^{t}$ l'estimation de la valeur de $m_{i,j}$, et $n_{i,j}$ le nombre de fois ou la recommandation $i \to j$ a été proposée à l'utilisateur.

2. La stratégie *UCB* (Upper Confidence Bound) [2] se base sur un calcul de la récompense moyenne empirique, à laquelle est ajouté un terme dépendant du nombre de fois où une option a été testée, de sorte que moins une option a été testée, plus elle est favorisée. À chaque itération t, i étant la proposition de U, le coach choisit :

$$c_t(i) \;=\; \underset{j\in\mathcal{I}}{\text{ArgMax}}\left\{\mu_{i,j} + C\sqrt{\frac{\ln(N_i)}{n_{i,j}}}\right\}$$

où $\mu_{i,j}$ est la moyenne du gain score(j) - score(i) obtenu jusqu'à l'instant t lorsque la substitution $i \to j$ a été suggérée, C est une constante, N_i le nombre de fois où l'item i a été proposé par U, et $n_{i,j}$ le nombre de fois où la substitution $i \to j$ a été proposée par C.

3. La stratégie dite d'*échantillonnage de Thomson* cherche à estimer les paramètres $m_{i,j}$ de la matrice $\mathbf{M}$ via une distribution de probabilité. Nous avons retenu ici la loi bêta. Ainsi chaque coefficient estimé $\widehat{m}_{i,j}^{t}$ est associé à une distribution $\mathbb{P}_{i,j}^{t} = Beta(\alpha_t, \beta_t)$ qui est mise à jour à chaque fois que la substitution $i \to j$ est proposée. Le choix de C s'opère selon :

$$c_t(i) \;=\; \underset{j\in\mathcal{I}}{\text{ArgMax}}\left\{\widehat{m}_{i,j}^{t}\,\big(\text{score}(j)-\text{score}(i)\big)\right\}$$

Les paramètres α et β de $\mathbb{P}_{i,j}$ sont ensuite mis à jour en fonction du comportement utilisateur :

$$\begin{cases} \alpha_{t+1} = \alpha_t + Accept_t \\ \beta_{t+1} = \beta_t + (1 - Accept_t) \end{cases}$$

avec $Accept_t$ = 1 si la substitution est acceptée et 0 sinon. Et :

$$\widehat{m}_{i,j}^{t+1} = \mathbb{E}[Beta(\alpha_{t+1}, \beta_{t+1}]$$

Cette stratégie permet d'avoir une estimation plus performante de la valeur de $m_{i,j}$, particulièrement pour un faible nombre d'itérations.

4. L'algorithme du *Q-learning* [10] cherche directement à estimer le gain $Q[i,j]$ à attendre quand $c_t(i) = j$. À chaque fois que la substitution $i \to j$ est suggérée, $Q[i,j]$ est mise à jour selon :

$$Q[i,j] := (1-\alpha)\,Q[i,j] + \alpha\big(g_{i,j} + \gamma \underset{j'\in\mathbf{I}}{\text{ArgMax}}\, Q[i',j']\big)$$

où α est un taux d'apprentissage, $g_{i,j}$ est le gain observé (éventuellement nul si j est refusé), γ est le facteur d'atténuation de prise en compte des gains dans le futur, $i' = j$ si la substitution $i \to j$ a été acceptée par U, et $i' = i$ sinon.
Les valeurs classiques pour le taux d'apprentissage α, de 0.1, et le facteur d'actualisation $\gamma = 0.9$ sont employées dans nos expériences.
Nous avons également introduit une variante du Q-learning, appelée λ-*Q-learning*, dans laquelle le taux d'actualisation d'apprentissage γ est proportionnel au taux d'apprentissage λ (supposé connu) de l'utilisateur. Cette variante est motivée par l'hypothèse que lorsque U apprend vite, son vecteur de préférence associé Π_t varie plus rapidement, et il est alors intéressant pour le coach de regarder loin dans le futur, alors que si $\lambda = 0$, cela n'a aucune utilité.

5.2 Le scénario de test

Afin de tester les capacités des différentes stratégies à conduire à une performance maximale, nous avons défini un scénario simple dans lequel U choisit i_0 à l'étape $t = 0$ et trois recommandations différentes $c_t(i_0)$ doivent être comparées (voir figure 2).
• Le choix de l'item i_4 par U est associé au score maximal : score(i_4) = 70, mais pour atteindre i_4 il faut passer par le

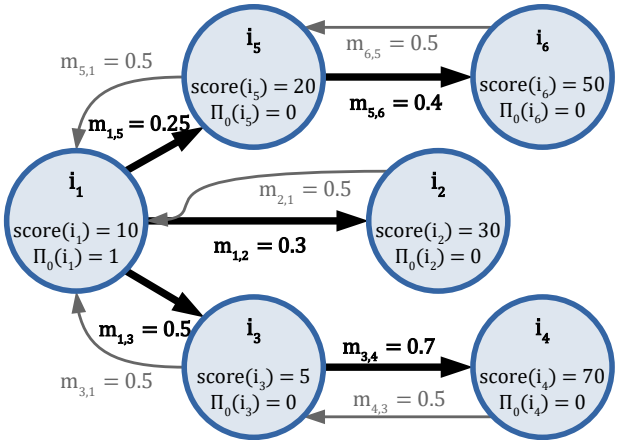

FIGURE 2 – Graphe présentant le scénario étudié dans le cadre des expériences au pas de temps initial $t = 0$.

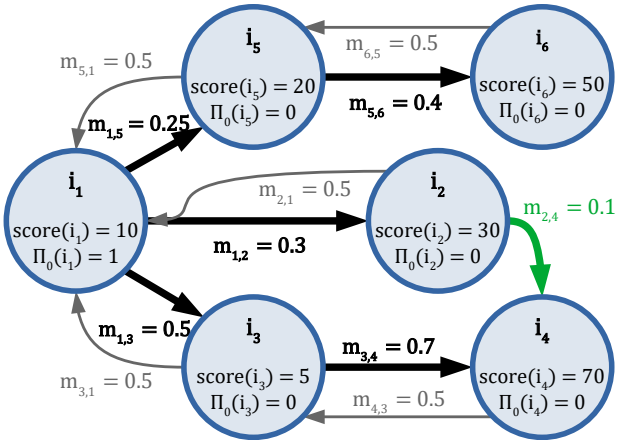

FIGURE 3 – Graphe présentant le scénario modifié au pas de temps initial $t = 0$.

chemin $i_1 \rightarrow i_3 \rightarrow i_4$ avec un score de i_3 faible : score(i_3) = 5 < score(i_1) = 10.
• Le chemin $i_1 \rightarrow i_5 \rightarrow i_6$ conduit à i_6 de score = 50 < score(i_4), en passant par i_5 de score = 20 > score(i_1).
• Finalement, le chemin le plus court $i_1 \rightarrow i_2$ est associé au gain immédiat le plus élevé : score(i_2) = 30.

5.3 Évaluation

Pour l'évaluation des résultats de simulation, nous étudions le gain de performance moyen $\overline{V}_T$ après T itérations (section 2.4). Les tests sont menés sur 200 utilisateurs simulés, et avec T = 1000 interactions avec le coach. Tous les utilisateurs simulés partagent les mêmes paramètres M, Π_0 et λ. Cependant, parce que l'acceptation ou le refus des propositions du coach est stochastique, les trajectoires dans l'espaces des vecteurs de préférence sont variables.

5.4 Résultats

5.4.1 Effet du paramètre λ

Nous testons ici l'effet du coefficient d'apprentissage λ caractérisant l'utilisateur. La table 2 donne les valeurs de $\overline{V}_T$ avec $T = 1000$ pour des valeurs de $\gamma \in \{0.005, 0.01, 0.05, 0.2, 0.4, 0.7, 1.0\}$. Les valeurs rapportées résultent de 200 simulations sur des utilisateurs U dont les caractéristiques sont données dans la figure 2.
Globalement, on observe que la stratégie du Q-learning est celle qui se comporte le mieux pour un large intervalle de valeurs de λ, approximativement dans [0.1, 1], et ce d'autant plus que l'écart-type observé est très contrôlé par rapport aux autres stratégies en compétition. En revanche, pour les valeurs de λ faibles, dans $[0, 0.1]$, les méthodes qui mettent à jour explicitement les coefficients de la matrices M l'emportent sur le Q-learning.
Deux leçons peuvent être tirées. D'une part, le choix de la meilleure stratégie dépend de la propension de U à apprendre, contrôlée par le paramètre λ. D'autre part, heureusement, cette dépendance est cependant limitée. Il suffit de savoir dans quelle grande gamme de valeurs λ s'inscrit pour savoir quelle stratégie favoriser.

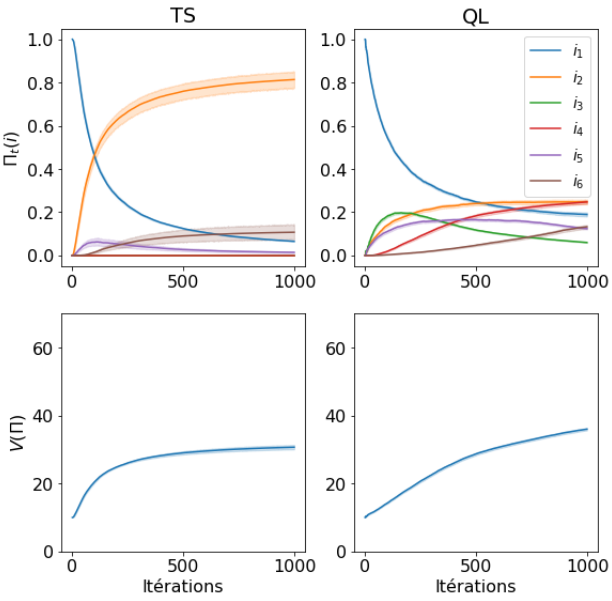

FIGURE 4 – Évolution des probabilités de choix des items et de l'espérance associée dans le cas $\lambda = 0.05$ pour les méthodes Q-Learning (à droite) et Échantillonnage de Thompson (à gauche). La courbe du haut présente la probabilité finale de choix de chaque item en fonction de T. La courbe du bas présente la valeur de $\overline{V}_T$ en fonction de T.

Pour mieux comprendre la différence de comportement entre les méthodes qui considèrent explicitement la matrice M et celles qui ne le font pas, comme le Q-learning qui est une méthode « model-free », il est intéressant de considérer la manière dont elles explorent l'espace des préférences au cours des interactions. La figure 4 comparant le Q-learning et la méthode de Thomson montre immédiatement que le Q-learning explore bien davantage l'espace des préférences, en maintenant les probabilités de choix des items à un niveau assez élevé, tandis que la méthode de Thomson converge bien plus rapidement. Cependant, le Q-

λ		Aléatoire	Gloutonne	Thomson	UCB	Q-learning	λ Q-learning
0.005	μ	13.4	19.8	**21.8**	**21.8**	10.4	18.7
	σ	0.39	2.45	0.43	0.34	0.16	0.83
0.01	μ	16.6	**26.7**	25.0	24.7	10.9	21.9
	σ	0.62	3.24	0.71	0.36	0.29	0.90
0.05	μ	28.1	30.2	29.6	29.4	21.0	**36.9**
	σ	1.88	6.34	2.98	0.68	2.53	2.29
0.2	μ	31.8	32.5	31.7	36.1	39.0	**48.6**
	σ	3.98	10.76	5.28	3.98	5.25	3.69
0.4	μ	31.1	30.1	33.9	39.7	47.9	**53.4**
	σ	4.76	14.13	6.97	6.23	6.74	4.76
0.7	μ	31.7	33.4	34.0	43.1	54.3	**55.9**
	σ	11.47	19.56	7.69	10.98	9.55	8.62
1.0	μ	34.1	35.5	34.0	42.4	**66.3**	64.7
	σ	26.74	30.54	8.02	16.72	11.53	12.44

TABLE 2 – Table des moyennes μ et écart-types σ de $\overline{V}_T$ pour $T = 1000$ calculée à partir de 200 simulations avec un utilisateur caractérisé par les paramètres fournis dans la figure 2.

λ		Aléatoire	Gloutonne	Thomson	UCB	Q-learning	λ Q-learning
0.005	μ	13.5	19.9	**24.4**	24.2	10.4	19.0
	σ	0.40	2.25	1.10	0.73	0.15	0.99
0.01	μ	16.6	26.6	**33.0**	31.8	11.0	22.8
	σ	0.66	3.30	1.61	1.40	0.33	1.18
0.05	μ	28.8	31.3	**55.8**	54.5	21.9	39.5
	σ	1.96	6.08	3.96	2.21	2.65	2.33
0.2	μ	32.1	33.0	**64.1**	63.5	41.6	52.0
	σ	4.04	9.94	5.87	2.93	5.17	3.40
0.4	μ	31.2	33.1	64.7	**65.1**	50.1	56.6
	σ	6.21	14.85	7.02	3.89	6.54	3.98
0.7	μ	32.7	31.6	64.9	**65.9**	57.2	60.7
	σ	11.44	19.22	7.88	5.02	8.79	5.49
1.0	μ	31.1	34.0	64.4	66.3	**67.9**	67.8
	σ	26.45	30.06	9.00	7.79	8.12	7.17

TABLE 3 – Table des moyennes μ et écart-types σ de $\overline{V}_T$ pour $T = 1000$ calculée à partir de 200 simulations avec un utilisateur caractérisé par les paramètres fournis dans la figure 3.

learning atteint des niveaux de performance $\overline{V}_T$ plus élevés pour un nombre d'interactions assez grand.

On notera que quand $\lambda = 1$, caractérisant un utilisateur qui adopte instantanément les suggestions du coach, la méthode du Q-learning est la seule qui permette d'approcher la performance $\overline{V}_T$ maximale qui est ici de 70.

La section suivante explore les différences entre stratégies en fonction des caractéristiques de la matrice de substituabilité qui contrôle les trajectoires possibles dans l'espace des vecteurs de préférence.

5.4.2 Effet de la matrice de substituabilité $\mathbf{M}$

Le contexte décrit par la figure 2 correspond à un cas difficile, puisque pour atteindre le choix de l'item i_4 associé au score le plus élevé, il faut d'abord passer par le choix de l'item i_3 dont le score est moins grand que le score de l'item i_1 choisi au départ par U. Le gradient de score n'informe donc pas sur la valeur potentielle de chaque item.

Ce gradient résulte à la fois des scores associés à chaque item et de la matrice $\mathbf{M}$ qui contrôle les chemins possibles dans l'espace des préférences. Dans l'exemple de la figure 3, $m_{2,4}$ a une valeur positive, ici 0.1, ce qui permet d'atteindre le choix de i_4 en passant par le choix de i_2. Le chemin $i_1 \to i_2 \to i_4$ est associé à des gradients tous positifs, et cela devrait permettre aux méthodes reposant sur cette information de suggérer les substitutions correspondantes et de guider l'utilisateur vers la préférence pour i_4.

La table 3 montre que la stratégie du Q-learning, qui ne repose pas directement sur l'évaluation du gradient de score sur les substitutions possibles, est peu sensible à cette modification de $\mathbf{M}$. En revanche, les méthodes telles que Thomson et UCB en tirent pleinement parti et permettent d'obtenir des performances $\overline{V}_{1000}$ plus élevées, notablement pour des valeurs de λ faibles, ici dans $[0, 0.2]$.

Il est présomptueux de tirer des leçons générales d'exemples particuliers. Cependant on peut conjecturer que la méthode du Q-learning est à favoriser quand la matrice $\mathbf{M}$ est clairsemée, avec de nombreuses valeurs nulles et, gé-

néralement, quand la fonction de score sur les items combinée aux substitutions possibles détermine des gradients de score négatifs sur les chemins qui conduisent aux performances maximales. À contrario, les méthodes tirant parti directement de l'information de gradient seront à favoriser quand la matrice $\mathbf{M}$ est non clairsemée et quand les gradients sont informatifs.
La meilleure stratégie à adopter pour le problème de coaching dépend bien à la fois de la fonction de score définie sur les items, de la matrice $\mathbf{M}$ et de la valeur de λ.

6 Vers un cadre plus général

Le concept de coaching décrit dans ce papier est limité car il suppose qu'à chaque instant l'utilisateur ne choisit qu'un item et que la fonction de score est définie sur chaque item indépendamment des autres. Cependant, dans de nombreux contextes, l'utilisateur doit choisir une combinaison d'items, par exemple les plats constituant un repas, et le score n'est pas additif mais prend en compte les interactions entre les items choisis, voire l'historique des choix, par exemple ce qui a été consommé sur une semaine.
Dans ce cas, il est plus difficile de définir une distribution de probabilité sur l'espace des préférences et la fonction de score n'est plus directement associée à chaque item. La définition de stratégies adaptées à ce cadre plus général est un objectif de nos travaux en cours.
Par ailleurs, il faut noter que l'estimation des caractéristiques des utilisateurs, ici discutée avec un seul utilisateur, peut bénéficier d'un apprentissage sur un ensemble d'utilisateurs, en tirant profit de la ressemblance mesurée entre eux, à l'instar de techniques de recommandation collaborative. Cela fait partie des développements futurs envisagés.

7 Conclusion

Cet article a présenté un nouveau scénario de recommandation dans lequel l'objectif est de modifier durablement les préférences de l'utilisateur à partir de l'observation répétée de ses choix. Au lieu de faire des recommandations à l'utilisateur, l'agent s'appuie sur les choix exprimés par celui-ci pour lui suggérer des modifications possibles. Nous appelons *coaching* ce processus itéré et personnalisé de recommandation. Nous avons proposé une formalisation de ce scénario comme un jeu itéré à deux joueurs. Nous avons défini plusieurs critères permettant d'évaluer la performance du coaching et montré qu'il n'y a pas de stratégie de coaching optimale pour tous les utilisateurs, mais qu'il fallait tenir compte des caractéristiques de ceux-ci pour optimiser la stratégie de coaching.
Les expériences réalisées sur des scénarios simples mais représentatifs des problèmes possibles ont permis de comparer plusieurs approches inspirées de la littérature sur l'optimisation du compromis exploration vs. exploitation et de tirer des leçons sur le meilleur choix en fonction des caractéristiques des problèmes.
Le coaching correspond à un large spectre de problèmes de recommandation pour lequel ce travail représente un premier pas.

Remerciements

Nous remercions Cécile Caumette et Hugo Vaysset pour leur contribution à l'état de l'art et à la conception de certaines expériences.

Références

[1] Robin Allesiardo. *Bandits Manchots sur Flux de Données Non Stationnaires*. PhD thesis, Université Paris-Saclay, 2016.

[2] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2) :235–256, 2002.

[3] Allison JB Chaney, Brandon M Stewart, and Barbara E Engelhardt. How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 224–232, 2018.

[4] Robert G Farrell, Catalina M Danis, Sreeram Ramakrishnan, and Wendy A Kellogg. Intrapersonal retrospective recommendation : lifestyle change recommendations using stable patterns of personal behavior. In *Proceedings of the First International Workshop on Recommendation Technologies for Lifestyle Change (LIFESTYLE 2012), Dublin, Ireland*, page 24. Citeseer, 2012.

[5] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[6] Yu Liang. Recommender system for developing new preferences and goals. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 611–615, 2019.

[7] Dorian Peters, Rafael A Calvo, and Richard M Ryan. Designing for motivation, engagement and wellbeing in digital experience. *Frontiers in psychology*, 9 :797, 2018.

[8] Mustafa Radha, Martijn C Willemsen, Mark Boerhof, and Wijnand A IJsselsteijn. Lifestyle recommendations for hypertension through rasch-based feasibility modeling. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*, pages 239–247, 2016.

[9] Hanna Schäfer and Martijn C Willemsen. Rasch-based tailored goals for nutrition assistance systems. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 18–29, 2019.

[10] Richard S Sutton and Andrew G Barto. *Reinforcement learning : An introduction*. MIT press, 2018.