

# Système de recommandations conscient du contexte pour départ froid avec évaluation financière des retombées

PHILIPPE ST-AUBIN<sup>1,2</sup>, BRUNO AGARD<sup>1,2</sup>

<sup>1</sup> Laboratoire en Intelligence des Données  
Département de mathématiques et génie industriel,  
École Polytechnique de Montréal, CP 6079, succursale Centre-Ville, Montréal, Québec, Canada  
philippe.st-aubin@polymtl.ca, bruno.agard@polymtl.ca

<sup>2</sup> Centre Interuniversitaire de Recherche sur les Réseaux d'Entreprise, la Logistique et le Transport (CIRRELT)

---

**Résumé** – Cet article présente une méthodologie pour faire de la recommandation consciente du contexte dans un cas où les items sont disponibles pour un temps limité. Il est proposé une manière d'inclure le contexte et les métadonnées des items pour parvenir à surmonter le problème du départ froid pour ces items. Le modèle suggéré permet de faire des recommandations basées sur le contenu, mais également sur la base du contexte géographique, temporel et financier de chaque item. Cela permet de faire des recommandations pertinentes même dans les cas où le contenu des items est inconnu pour un utilisateur. Une méthode pour mesurer l'impact financier d'un système de recommandations est également suggérée et utilisée pour permettre de comparer les systèmes testés. La méthodologie est finalement implémentée sur les données d'une entreprise de commerce en ligne.

**Abstract** - This article presents a methodology to make context aware recommendations in a case where the items are available for a limited time. A way to include context and item metadata is proposed to overcome the problem of cold start for items. The suggested model makes recommendations based on the content, but also based on geographical, temporal, financial context of each item. This allows the system to make relevant recommendations even in cases where the content of the items is unknown to a user. A method for measuring the financial impact of a recommendation system is also suggested and used to compare the systems tested. The methodology is finally implemented on real e-commerce data.

**Mots clés** – Système de recommandations, Système de recommandations conscient du contexte, Modélisation du contexte, Départ à froid, Évaluation des systèmes de recommandations

**Keywords** – Recommender system, Context-aware recommender systems, Context modeling, Cold-start, Recommender system evaluation

---

## 1 INTRODUCTION

Dans le contexte actuel de globalisation, les entreprises doivent rivaliser de créativité et d'incitatifs pour conserver leurs parts de marché. C'est pourquoi de plus en plus d'entreprises concentrent leurs efforts pour améliorer leurs relations avec les clients déjà présents. Pour ça, il est d'abord important qu'une entreprise comprenne et étudie ses clients. Par exemple [St-Aubin & Agard, 2017] démontré l'utilité et comparé l'effet de certaines méthodes permettant d'étudier les clients d'une entreprise de commerce en ligne.

Parmi les outils permettant de personnaliser les expériences et les communications entre une entreprise et ses clients se trouvent les systèmes de recommandations (SR).

Ces systèmes permettent de présenter aux clients les items les plus susceptibles de correspondre à leurs intérêts. Pour ce faire, ils se basent sur les historiques d'achats et information connus sur les clients pour pouvoir présenter aux clients les produits qui correspondent le mieux à leur profil [Dadouchi & Agard, 2017].

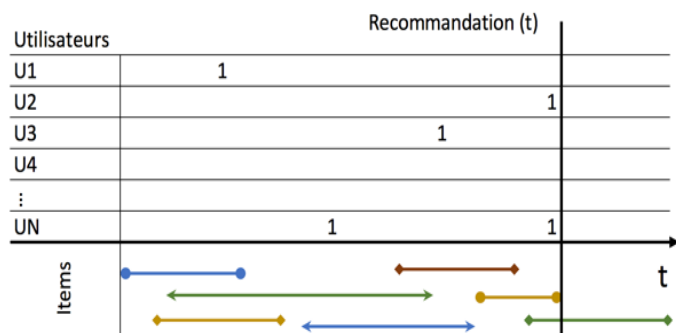
Or les méthodes classiques utilisées par les SR performant mal dans un contexte où les produits changent souvent. Il est également nécessaire d'appliquer certaines modifications aux méthodes en présence de contraintes sur la disponibilité des produits [Dadouchi & Agard, 2018], notamment dans des cas où des contraintes géographiques et temporelles s'appliquent. Dans ces cas, des méthodes adaptées doivent être développées.

Pour ces raisons et afin d'assurer un meilleur service aux clients, une méthodologie pour faire de la recommandation consciente du contexte est proposée.

La méthode proposée démontre son efficacité dans un contexte où les items sont disponibles pour un temps limité, et lorsqu'ils ont une composante géographique.

La Figure 1 présente le contexte de recommandation spécifique du cas d'étude. Les utilisateurs sont représentés de U1 à UN. Leurs achats passés sont identifiés par des 1 au moment de l'achat. Les disponibilités de chaque item sont représentées par des traits. La longueur du trait indique le temps de disponibilité de l'item. La couleur des traits identifie leur situation

géographique. Les couleurs bleu, vert, rouge et jaune identifient des zones géographiques différentes pour les items  $i_1, i_2, i_3, \dots$ . Les pointes des traits représentent le type d'items le cercle, la flèche et le losange représentent des catégories d'items différentes. La figure montre qu'étant données les disponibilités restreintes dans le temps et dans la zone géographique des items, il est possible que plusieurs types d'items ne soient pas disponibles au moment d'une recommandation dans une zone spécifique.



**Figure 1: Représentation de l'historique dans le temps, de la disponibilité géographique (couleur), du type d'item (flèche) et temporel (longueur du trait) au moment d'une recommandation pour les items  $i_1, i_2, i_3, \dots$**

Le système de recommandations proposé vise à faire des recommandations qui prennent en compte plusieurs facteurs contextuels comme le temps, la distance géographique et le type d'items. Le système résout donc le problème du départ à froid pour les items. Le départ à froid ou problème du nouvel item concerne la difficulté des SR à recommander les nouveaux items. Ce problème est dû au fait qu'aucune appréciation pour l'item n'ait encore été observée. La solution proposée se distingue toutefois des modèles basés purement sur le contenu puisqu'il arrive à faire des recommandations dans des cas où aucun item n'a de contenu similaire. Par exemple, si un utilisateur aime un certain type d'items, mais qu'aucun item de ce type n'est disponible, le système proposé arrive tout de même à fournir une recommandation personnalisée. Cette recommandation sera basée sur la pertinence contextuelle de l'item : par exemple, la proximité géographique, le prix, la saison, etc.

Enrichir la recommandation avec le contexte permet également d'éviter de tomber dans un cas de surspécialisation, puisque le contenu n'est pas le seul critère évalué permettant ainsi d'augmenter la sérendipité du système. C'est-à-dire la capacité du système à faire des recommandations variées et surprenantes pour l'utilisateur ce que les systèmes basés sur le contenu ont de la difficulté à faire [Lops & al., 2011].

Pour évaluer la performance du système, un moyen d'estimation des revenus engendrés par le système est proposé. Cette méthode permet de comparer des systèmes de recommandations sur la base des revenus générés estimés plutôt que sur la précision des recommandations.

Avant de poursuivre avec la méthodologie (section 4), une revue de la littérature sur les systèmes de recommandations sera présentée (section 2) suivie par l'explicitation de la contribution apportée par ce papier (section 3). Finalement, un cas d'étude présentera l'application de la méthodologie sur les données d'une entreprise réelle de commerce en ligne (section 5).

## 2 ÉTAT DE L'ART

Les SR ont été développés à partir des années 90. Ils visent à aider un utilisateur à trouver de l'information pertinente [Ricci & al., 2011]. En commerce, on précise la définition tel que les SR visent à prédire l'appréciation d'un utilisateur pour un item lui étant inconnu [Adomavicius & Tuzhilin, 2005]. Les principales méthodes de recommandations seront présentées, suivies d'un survol des méthodes de recommandations conscientes du contexte et finalement les métriques et méthodes d'évaluation des SR seront présentées.

### 2.1 Méthodes de recommandations

Il existe pour le moment 4 grandes méthodes de recommandations. Chacune d'elle possède des forces et des faiblesses et s'applique mieux selon certains contextes. Cette section présentera le principe derrière chacune de ces méthodes.

#### 2.1.1 Filtres collaboratifs

La méthode des filtres collaboratifs [Schafer & al., 2007] consiste à comparer les notes accordées aux items par les utilisateurs du système. On cherche à identifier des utilisateurs ayant accordé des notes similaires à des items communs. Les notes accordées pour des items inconnus d'un utilisateur sont ensuite déterminées par la moyenne des notes de cet item pondérées par la similitude des préférences entre utilisateurs. On est donc capable d'inférer la note qu'un utilisateur aurait donnée à un item qu'il n'a jamais vu en se basant sur le comportement d'utilisateurs similaires. Les indicateurs de similitudes les plus utilisés sont l'indicateur de corrélation de Pearson, le cosinus de la projection vectorielle ou la métrique de Jaccard [Resnick & al., 1994].

Il existe toutefois quelques limites à cette méthode puisqu'elle n'est pas en mesure de faire de bonnes recommandations en contexte de départ à froid. C'est-à-dire, pour de nouveaux utilisateurs ou de nouveaux produits, car il n'existe pour eux aucune donnée sur laquelle se baser pour le calcul des similitudes. Par le même principe, lorsqu'une très grande variété d'items est offerte beaucoup d'items se retrouvent sans notes et il est difficile de trouver des utilisateurs similaires [Adomavicius & Tuzhilin, 2005]. Ce type de problème est nommé le problème des données clairsemées.

#### 2.1.2 Méthode basée sur le contenu

La méthode basée sur le contenu consiste à recommander aux utilisateurs des items similaires à ceux qu'ils ont déjà achetés [Pazzani & Billsus, 2007]. Pour ce faire, un item est représenté par ses caractéristiques. Ainsi, un item similaire d'après ses caractéristiques à un item apprécié de l'utilisateur peut être recommandé.

La difficulté de cette méthode se trouve dans le fait qu'il faut posséder beaucoup d'informations sur les caractéristiques des items pour pouvoir mesurer la similarité de leurs caractéristiques. L'autre limite est que cette méthode a tendance à sur-spécialiser les recommandations pour les utilisateurs.

#### 2.1.3 Méthode des graphes

La troisième technique est basée sur les graphes et explore les liens entre les items pour recommander les plus populaires [Gori & Pucci, 2006]. Cette méthode est souvent utilisée dans le contexte de recommandation pour la recherche de documents pertinents comme des articles scientifiques ou des pages web. C'était d'ailleurs la méthode initialement employée par Google

[Page & al, 1999]. Dans le cas des articles scientifiques par exemple, les liens entre les articles sont les citations. On recommande donc les articles populaires (souvent cités) qui citent également des articles populaires.

Toutefois, en contexte de départ à froid, comme les nouveaux items ont peu de liens avec les autres items, il est difficile pour cette méthode de recommander les nouveaux items. La méthode a donc tendance à surévaluer les recommandations des items plus anciens au détriment des items plus récents.

#### 2.1.4 Factorisation de matrice

La quatrième méthode consiste à identifier des facteurs latents permettant d'expliquer les préférences des utilisateurs et les similarités sous-jacentes des items [Salakhutdinov & Mnih, 2008]. On peut réduire la dimension du problème en le projetant dans un système de moindre dimension. On peut alors représenter les items ainsi que les utilisateurs par un vecteur de leur appartenance aux facteurs latents. Les facteurs latents permettent d'identifier un nombre réduit de composantes permettant d'expliquer les résultats. Par exemple, une de ces composantes pourrait représenter le sexe des utilisateurs. On retrouverait donc qu'une des composantes du vecteur des utilisateurs représente leur appartenance à une sélection plus féminine ou plus masculine.

Pour appliquer cette méthode, on utilise principalement des méthodes de factorisation de matrice, notamment la factorisation en valeur singulière (SVD) [Paterek, 2007], ou des méthodes par descente de gradient ou des moindres carrés partiels alternés. Les méthodes de factorisation de matrices se sont avérées donner de bons résultats sur des ensembles de données connus comme sur l'ensemble de Netflix [Koren & al., 2009].

La méthode par factorisation de matrice présente toutefois également des faiblesses en contexte de départ à froid pour les nouveaux items ou les nouveaux utilisateurs sur lesquels on possède encore peu d'information.

La plupart des systèmes de recommandations peuvent se classer dans l'une ou l'autre de ces quatre grandes familles. On distingue ensuite les systèmes d'après le type de données auxquels ils ont accès. C'est-à-dire si les appréciations des utilisateurs pour les items sont explicites ou implicites. Dans le cas explicite, les appréciations ou notes sont directement données par l'utilisateur. Dans l'autre cas, le système doit les déduire du comportement de l'utilisateur. Dans le cas explicite, une ou l'autre des méthodes peut directement être appliquée sur la matrice des appréciations des utilisateurs pour les items. Dans l'autre cas, une étape supplémentaire de déduction de l'appréciation doit être ajoutée avant d'appliquer une méthode de recommandation.

La section suivante fera un survol des méthodes qui intègrent le contexte pour améliorer les recommandations.

#### 2.2 Système de recommandations conscient du contexte (CARS)

Pour tenter d'améliorer les performances des systèmes de recommandations, de nombreux chercheurs [Covington & al., 2016], [Kefalas & Manolopoulos, 2017] et [Yin & al., 2013] ont tenté d'inclure des éléments du contexte entourant les préférences des utilisateurs. Ce faisant, des informations supplémentaires sur les conditions accompagnant les choix d'un utilisateur sont utilisées pour améliorer les prévisions. Par exemple des variables de temps comme l'heure de la journée, la position géographique, dans quel lieu se trouve l'utilisateur, etc.

[Panniello & al., 2014] divise les CARS en trois méthodes différentes dont deux basées sur les notes accordées par les clients et une impliquant la modélisation du contexte à même l'attribution des notes :

1. Les méthodes de pré filtrage. Dans ces méthodes on applique un filtre contextuel aux notes avant de prévoir les notes manquantes.
2. Post filtrage, on calcule d'abord les notes, puis on ajuste les notes en fonction de leur pertinence dans un contexte donné.
3. Le contexte est modélisé à même l'algorithme générant les notes inconnues.

Les auteurs ont démontré que les méthodes 2 et 3 rendaient de meilleures précisions et qu'en termes de balance entre la précision et diversité, les RS avec le contexte modélisé à même l'algorithme étaient les RS les plus équilibrés.

#### 2.3 Métrique et évaluation des performances

Une difficulté demeure dans le développement d'un SR : son évaluation. Il est compliqué de déterminer la performance réelle qu'aura un SR puisque leurs effets sur la demande réelle des clients demeurent encore inconnus [Jannach & al., 2016]. De plus le problème contient de multiples dimensions ainsi que des considérations ergonomiques et des objectifs qui peuvent changer selon les contextes d'application. Notamment, le rôle des SR dans la recommandation d'items inusités pour un utilisateur. Ces items inusités permettent aux utilisateurs de découvrir de nouveaux items qui pourraient potentiellement leur plaire en plus d'aider l'entreprise à développer sa relation avec le client. Les sections suivantes présenteront donc certaines métriques ayant été développées pour permettre de mesurer certaines dimensions des performances des SR avec les méthodes utilisées pour évaluer ces performances.

##### 2.3.1 Métriques de performance

Les métriques diffèrent selon si des notes explicites des utilisateurs sont disponibles ou non. Si elles le sont, des métriques comme la racine de la moyenne des erreurs au carré (RMSE) ou la moyenne des erreurs absolues (MAE) peuvent être employées. Ces métriques permettent d'évaluer la proximité des notes prédites d'un utilisateur pour un item aux notes réelles que l'utilisateur a données à ces items. Les métriques quantifient donc de combien sur l'échelle de notation le modèle s'écarte des notes réelles de l'utilisateur [Shani & Gunawardana, 2011]. Ces méthodes s'appliquent donc sur des modèles de régression des notes. Dans d'autres cas, on évalue les modèles sur la base des N premiers item recommandés. On calcule la précision comme étant le nombre de recommandations réussies sur les N recommandations. Une autre métrique : le rappel, calcule la proportion des items sélectionnés par l'utilisateur couvert par les N recommandations. La mesure-F, intègre les deux dernières sous forme de moyenne harmonique [Sasaki, 2007], ce qui permet de concentrer les efforts sur l'optimisation d'une seule métrique plutôt que les deux à la fois.

Certains chercheurs se sont également intéressés à l'utilité de la recommandation. C'est-à-dire qu'ils ont développé une métrique permettant d'évaluer la probabilité qu'un utilisateur consulte la n<sup>ième</sup> recommandation [Gunawardana & Shani, 2009].

Il existe également d'autres dimensions aux métriques de performance des recommandations comme la couverture qui s'intéressent à la proportion d'items que le système de

recommandations est capable de recommander par rapport à l'ensemble des items disponibles. Puisque dans certains cas les systèmes se basent sur la popularité des items, plusieurs items pourraient être laissés de côté et jamais recommandés. Une autre des métriques de performance des recommandations mesure la sérendipité : la capacité d'un SR à faire des recommandations à la fois surprenantes et appréciées de l'utilisateur [McNee, 2006]. Certains auteurs proposent de modéliser le problème de la recommandation comme un problème d'optimisation où on tente de maximiser une fonction objectif qui contient les différents objectifs des acteurs en interaction avec le système [Rodriguez & al., 2012].

### 2.3.2 Évaluation des SR

Classiquement, pour les SR les méthodes d'évaluation des performances ont été fortement influencées par le prix Netflix lancé en 2009 [Jannach & al., 2016]. La procédure pour évaluer la performance d'un système consistait à retenir des appréciations connues et à tenter de les inférer à partir des autres disponibles. Or [Jannach & al., 2016] et [Covington & al., 2016] ont critiqué cette manière de procéder puisqu'inférer une appréciation manquante au hasard n'est pas équivalent à faire la prévision d'un vote. En effet, l'évolution de la relation de l'utilisateur avec le système ne peut pas être prise en compte. De plus, les appréciations ne sont pas manquantes au hasard donc un système performant sur les appréciations connues ne sera pas nécessairement performant sur un ensemble d'items inconnus de l'utilisateur.

## 3 CONTRIBUTION

Pour toutes les grandes familles de recommandations, sauf pour la méthode basée sur le contenu, il y a des faiblesses au niveau des recommandations pour les nouveaux items.

De plus, dans le contexte spécifique du cas d'étude, les méthodes basées sur le contenu classique ne pourraient pas non plus s'appliquer directement étant donné qu'il se pourrait qu'aucun item correspondant au contenu précédemment apprécié d'un utilisateur ne soit disponible. Ce faisant, le système ne serait pas en mesure de fournir des recommandations à l'utilisateur. On suggère donc d'élargir les méthodes basées sur le contenu pour considérer, en plus du contenu, le contexte des items. C'est-à-dire considérer les caractéristiques des items qui ne sont pas en lien avec sa description, par exemple sa position géographique, son prix et la période de l'année. En procédant ainsi, malgré le fait qu'aucun item similaire en contenu ne soit disponible, on pourra recommander un item similaire en contexte.

Une autre contribution se trouve est au niveau de la méthode d'évaluation du système. La méthode proposée permet de mesurer l'impact financier d'un système de recommandations avec une manière prédictive pour remplacer la méthode d'évaluation classique par retenue de notes.

## 4 MÉTHODOLOGIE

La méthodologie proposée pour construire le SR s'applique dans un contexte d'appréciation implicite. De plus, elle est conçue pour s'appliquer à un contexte où les items sont disponibles pour un temps limité et situés géographiquement. Elle se divise en deux étapes.

La première étape consiste à développer une fonction d'appréciation (4.1). La seconde étape permettra de mesurer la

performance du système d'un point de vue financier (4.2). Les sections suivantes détailleront les éléments des deux étapes.

### 4.1 Fonction d'appréciation

La fonction d'appréciation permet de créer un profil du comportement des utilisateurs. À partir de ce profil, il est possible d'inférer l'appréciation d'un nouvel item pour un utilisateur. Deux méthodes pour estimer l'appréciation d'un utilisateur sont testées. Une fonction linéaire (4.1.1), ainsi qu'un réseau de neurones (4.1.2). La première solution permet d'obtenir un modèle qu'il est possible d'interpréter. La seconde donne une fonction difficilement interprétable, mais qui, par la flexibilité du modèle, permet d'obtenir une meilleure précision.

Pour faire la recommandation, le système doit être en mesure de déterminer, pour chaque utilisateur, lesquels parmi les items disponibles obtiendraient les meilleures appréciations. Le comportement passé des utilisateurs sert de guide pour inférer l'appréciation.

Premièrement, les variables qui permettent de représenter les items doivent être sélectionnées et structurées. Ces variables sont celles en lien avec le contexte ainsi qu'avec le contenu. Par exemple, les variables descriptives du temps, de la localisation géographique, du prix, de la catégorie, de la sous-catégorie, du marchand et toute autre variable permettant de caractériser un item. Ensuite, les variables numériques doivent être transformées en variables catégorielles. Puis, pour permettre de modéliser les interactions entre les variables caractéristiques, de nouvelles variables représentant les combinaisons des variables caractéristiques sont créées. Celles-ci permettent de caractériser les items et les comportements des utilisateurs. Chacune d'elle représente donc une caractéristique ou l'interaction de caractéristiques. Un item possède autant de caractéristiques qu'il a de variables pour le représenter.

En construisant le profil des utilisateurs d'après les caractéristiques des items, il est possible de modéliser leur comportement dans le temps, dans l'espace, selon le prix, selon la nature des items. En prenant en compte les combinaisons de caractéristiques, on modélise également les influences des caractéristiques les unes par rapport aux autres.

Deux modèles de résolutions sont alors développés.

#### 4.1.1 Fonction linéaire

Pour inférer l'appréciation d'un utilisateur pour un nouvel item, une première fonction linéaire est élaborée.

La première étape consiste à noter individuellement l'appréciation d'un utilisateur pour chaque caractéristique d'un item. Les appréciations individuelles sont désignées par la probabilité que l'utilisateur adopte la valeur de la caractéristique de l'item. Ainsi, l'appréciation  $p_{u,i,k}$  représente la probabilité que l'utilisateur  $u \subset U$  adopte la même valeur  $c$  que celle de l'item  $i \subset I^d$  pour la caractéristique  $k \subset K$ . Avec  $U, I^d, K$ , l'ensemble des utilisateurs, l'ensemble des items disponibles et l'ensemble des caractéristiques respectivement. On formule :

$$p_{u,i,k} = \frac{\sum_{j=1}^{a^u} I(c_{j,k} = c_{i,k})}{a^u} \quad (1)$$

Avec

$$I(c_{j,k} = c_{i,k}) = \begin{cases} 1 & c_{j,k} = c_{i,k} \\ 0 & \text{sinon} \end{cases} \quad (2)$$



Ici,  $a^u$  représente le nombre total d'achats de l'utilisateur  $u$ . L'indice  $j$  parcourt l'ensemble de ses achats. Ainsi,  $c_{j,k}$  représente la valeur de la caractéristique  $k$  de l'achat  $j$  de l'utilisateur  $u$ . La variable  $c_{i,k}$  représente la valeur de la caractéristique  $k$  de l'item  $i$ .

On obtient donc pour chaque caractéristique  $k$  d'un item  $i$  la probabilité  $p$  que l'utilisateur  $u$  adopte la valeur  $c$  de l'item pour cette caractéristique. La Table 1 présente un résumé de la notation utilisée.

**Table 1 : Notation pour représenter le modèle**

Notation	Description
$U$	Ensemble des utilisateurs
$u$	Indice désignant un utilisateur
$I^d$	Ensemble des items disponibles
$i$	Indice désignant un item disponible
$K$	Ensemble des caractéristiques
$k$	Indice désignant une caractéristique
$a^u$	Nombre total d'achats de l'utilisateur $u$
$j$	Indice désignant un achat de l'utilisateur $u$
$c_{i,k}$	Valeur de l'item $i$ pour la caractéristique $k$
$c_{j,k}$	Valeur de l'achat $j$ pour la caractéristique $k$
$p_{u,i,k}$	Probabilité que l'utilisateur $u$ adopte la valeur de l'item $i$ pour la caractéristique $k$

Pour aider à la compréhension, les variables peuvent être représentées en tableau de données :

Items (I)	k1	k2	k3	...
I1	$c_{1,1}$	$c_{1,2}$	$c_{1,3}$	...
I2	$c_{2,1}$	$c_{2,2}$	$c_{2,3}$	...
I3	$c_{3,1}$	$c_{3,2}$	$c_{3,3}$	...
⋮	⋮	⋮	⋮	⋮

**Figure 2 : Variables du modèle liées aux items disponibles représentées en tableau de données**

La Figure 2 représente le tableau des items. La première colonne représente chaque item et les éléments de la première colonne représentent l'indice  $i$ . Les colonnes  $k1, k2, k3, \dots$  représentent les caractéristiques descriptives. Les éléments de ces colonnes sont les valeurs des items pour les caractéristiques.

Similairement, on peut représenter les variables du modèle en lien aux utilisateurs sur un tableau :

Utilisateurs (U)	j	k1	k2	k3	...
U1	1	$c_{1,1}$	$c_{1,2}$	$c_{1,3}$	...
U1	2	$c_{2,1}$	$c_{2,2}$	$c_{2,3}$	...
⋮	⋮	⋮	⋮	⋮	⋮
U1	$a^1$	$c_{a^1,1}$	$c_{a^1,2}$	$c_{a^1,3}$	...
U2	1	$c_{2,1}$	$c_{2,2}$	$c_{2,3}$	...
⋮	⋮	⋮	⋮	⋮	⋮
U2	$a^2$	$c_{a^2,1}$	$c_{a^2,2}$	$c_{a^2,3}$	...
⋮	⋮	⋮	⋮	⋮	⋮

**Figure 3 : Variables du modèle liées aux utilisateurs représentées en tableau de données**

La première colonne représente les utilisateurs avec en élément les indices  $u$ . La colonne  $j$  représente l'indice des achats de l'utilisateur et les colonnes  $k1, k2, k3, \dots$  les caractéristiques des utilisateurs. Les éléments  $c_{j,k}$  sont les valeurs des caractéristiques de chaque achat des utilisateurs.

Une note  $r_{u,i}$  de l'appréciation globale de l'utilisateur  $u$  pour l'item  $i$  est obtenue en prenant la somme des probabilités  $p_{u,i,k}$  sur l'ensemble des  $k$  et en ajoutant une correction pour la popularité de l'item  $\gamma_{i,pop}$ . Pour ajuster l'influence des caractéristiques sur les décisions des utilisateurs, un poids  $\alpha_k$  est ajouté pour pondérer l'effet de chacune des caractéristiques.

$$r_{u,i} = \sum_{k=1}^l \alpha_k p_{u,i,k} + \gamma_{i,pop} \quad (3)$$

Où  $l$  est l'ensemble des caractéristiques utilisées pour représenter l'appréciation d'un utilisateur pour un item.

On recommande les items en ordre décroissant de  $r_{u,i}$

L'avantage d'utiliser cette méthode est qu'elle permet de modéliser le comportement d'un utilisateur selon le contexte.

Par exemple, en prenant comme caractéristique la position géographique des items, on peut comparer la distance des items disponibles avec celles des items achetés dans le passé et accorder une note à la distance en fonction du comportement passé de l'utilisateur. De la même manière, les caractéristiques peuvent être combinées. Par exemple combiner à la distance la catégorie de l'item pour identifier si l'utilisateur se déplace dans des régions plus éloignées de ses achats habituels pour une certaine catégorie d'item ou encore à une certaine période de l'année si la distance et le temps sont combinés.

#### 4.1.2 Réseau de neurones

Pour améliorer la précision des résultats, une seconde stratégie est développée autour des réseaux de neurones. Les réseaux de neurones, malgré leur aspect boîte noire, sont de plus en plus utilisés dans de nombreuses applications réelles [Covington & al., 2016], [Wang & al., 2015]. Ils sont particulièrement utiles lorsque la relation entre les variables d'entrées et la variable de sortie est non linéaire.

Ainsi pour tirer profit des capacités d'ajustement du réseau de neurones, une partie du modèle linéaire (3.1.1) a été réutilisée. Pour pouvoir entraîner le modèle, il s'agit cette fois de construire un ensemble de données avec les  $p_{u,i,k}$  tel qu'illustré sur la Figure 4.

Utilisateurs (U)	Items (i)	k1	k2	k3	...
U1	I1	P <sub>1,1,1</sub>	P <sub>1,1,2</sub>	P <sub>1,1,3</sub>	...
U1	I2	P <sub>1,1,1</sub>	P <sub>1,1,2</sub>	P <sub>1,1,3</sub>	...
⋮	⋮	⋮	⋮	⋮	⋮
U2	1	P <sub>2,1,1</sub>	P <sub>2,1,2</sub>	P <sub>2,1,3</sub>	...
U2	2	P <sub>2,2,1</sub>	P <sub>2,2,2</sub>	P <sub>2,2,3</sub>	...
⋮	⋮	⋮	⋮	⋮	⋮

Figure 4 : Ensemble de données pour faire l'apprentissage

Le réseau est entraîné à prédire d'après les caractéristiques si l'utilisateur sélectionnera l'item. La recommandation est faite en ordonnant les items en ordre de probabilité d'achat décroissant. Pour choisir les hyper paramètres, de la validation croisée peut être faite pour maximiser des métriques comme la précision ou le rappel.

Dans certains cas, la décision pourrait être prise autour de l'impact financier du modèle. Pour ces cas, une métrique de performance permettant d'évaluer l'impact financier d'un système de recommandation a été développée et a servi à la sélection des paramètres et hyper paramètres pour les deux méthodes de recommandations.

#### 4.2 Métrique de performance

Pour optimiser la valeur des paramètres de la fonction d'appréciation et pour comparer les systèmes de recommandations, une métrique évaluant les revenus générés par le système est proposée.

Pour ce faire, une fonction compare les recommandations effectuées par le système de recommandations aux vrais items achetés et la valeur des items des recommandations réussies est attribuée aux revenus du système de recommandations. Le processus est illustré sur la Figure 5. Sur le tableau du haut sont représentés les achats réels des utilisateurs dans le temps. La couleur verte représente les événements qui sont réellement survenus. Le tableau du bas représente les recommandations du système pour chaque utilisateur. Les recommandations qui correspondent à un achat réel sont en vert. Les recommandations qui ne correspondent pas sont en rouge. Une seule recommandation par utilisateur est représentée sur la Figure 5. Toutefois, on prend en compte les  $N$  premières recommandations du système pour considérer une recommandation réussie. On attribue finalement la valeur des items des recommandations réussies aux revenus du système. L'évaluation des revenus du système est représentée sur fond noir à la Figure 5.

Utilisateurs	Achat( $t_1$ )	Achat( $t_2$ )	Achat( $t_3$ )
U1	I2	I32	I29
U2	I2	I48	I17
U3	I7	I2	I8
U4	I9	I9	I43
⋮	⋮	⋮	⋮
UN	I1	I5	I23

Recommandations	Item( $t_1$ )	Item( $t_2$ )	Item( $t_3$ )	t
R(U1)	I2	I40	I29	
R(U2)	I4	I2	I17	
R(U3)	I7	I5	I5	
R(U4)	I1	I9	I43	
⋮	⋮	⋮	⋮	⋮
R(UN)	I8	I6	I40	
Revenus (\$)	I2 + I7	I9	I29+I17+I43	

Figure 5 : Méthode d'évaluation des revenus d'un système de recommandations

Certaines hypothèses sont impliquées par cette manière de procéder. Par exemple, les recommandations à des temps précédents n'ont pas d'influence sur les achats au temps présent. L'autre hypothèse qui est faite est que tous les clients prennent connaissance des 20 items recommandés chaque jour.

Cette manière de procéder permet de mesurer la différence des revenus générés par différents systèmes. Ce qui permettra, éventuellement, à une entreprise d'évaluer la rentabilité d'un projet d'implémentation d'un nouveau système, ainsi que le retour sur investissement espéré.

## 5 CAS D'ÉTUDE

La méthodologie a été testée sur les données d'une entreprise de commerce en ligne et implémentée à l'aide du logiciel de programmation statistique R 3.3.3. L'entreprise se spécialise dans la vente de produits, d'activités, de services et d'événements à rabais. Elle a mis à disposition 3 années de données de son système de gestion de la relation client (CRM). Les données comprenaient environ 2 millions d'utilisateurs, 10 000 items en historique dont entre 50 et 200 items disponibles quotidiennement. L'information sur les items se résume aux codes postaux, à la catégorie (9 catégories), à la sous-catégorie (200 sous-catégories), au prix de base ainsi qu'au prix réduit. Il faut noter qu'un item peut posséder plusieurs codes postaux. On peut voir le format des données sur la Figure 6. La première colonne représente l'identifiant unique des items, la deuxième donne le code postal, on y voit ensuite les informations sur la catégorie, la sous-catégorie le prix de base et le prix réduit.

Item_ID	ZIP	Categorie	Sous-Categorie	Prix_base	Prix_reduit
1	J6V0E1	Food Shop	Chocolate Shops	30	18
1	K2T0K5	Food Shop	Chocolate Shops	30	18
2	H3C0K2	Restaurant	Cafe	15	5
3	H2J1X1	Shopping	Home Decor	207	60

Figure 6: Format des données sur les items

L'historique des achats des clients, ainsi que leur code postal était disponible. Le format de l'historique des achats est affiché sur la Figure 7.

User_ID	Deal_ID	Order_Date
1	1	2016-01-10
1	3	2016-02-14
2	8	2016-02-17
3	7	2016-02-22
3	4	2016-05-17

Figure 7 : Historique des achats clients

Il comprend l'identifiant de l'utilisateur, l'identifiant de l'item ainsi que la date de l'achat.

Pour construire le profil des utilisateurs, les caractéristiques sur la saison, la distance géographique, la catégorie, la sous-catégorie, le marchand émettant l'item, le prix de base, le montant de la réduction et les combinaisons des caractéristiques ont été utilisés. La Figure 8 montre une partie de l'ensemble de données obtenu après avoir noté l'appréciation de chacune des caractéristiques. La Figure 8 affiche l'appréciation des

utilisateurs 1 et 2 pour les items 1, 2 et 3. La caractéristique *pop* affiche le rang de popularité des items. L'appréciation des utilisateurs pour les items à une distance égale à la distance entre eux et l'item est représentée par *geo*. Les caractéristiques *Categorie* et *Sous-Categorie* représentent l'appréciation des utilisateurs sur le contenu des items.

User_ID	Item_ID	pop	geo	Categorie	Sous-Categorie
1	1	2	0,19	0,36	0,96
1	2	1	0,07	0,48	0,42
1	3	3	0,29	0,09	0,79
2	1	2	0,65	0,11	0,95

**Figure 8 : Appréciation des utilisateurs pour chaque caractéristique des items**

À partir de ces données, les deux modèles (tels que présentés en 3.1.1 et 3.1.2) ont été appliqués.

Dans le cas présent, un perceptron multi couches avec 34 neurones en entré, 2 couches cachées contenant 1 neurone et 2 neurones a permis de générer les meilleurs résultats. Pour les autres paramètres, ils ont été déterminés par validation croisée. Un taux d'apprentissage de 0,05 avec l'optimiseur Adam et une tangente hyperbolique comme fonction d'activation rendaient les meilleurs résultats.

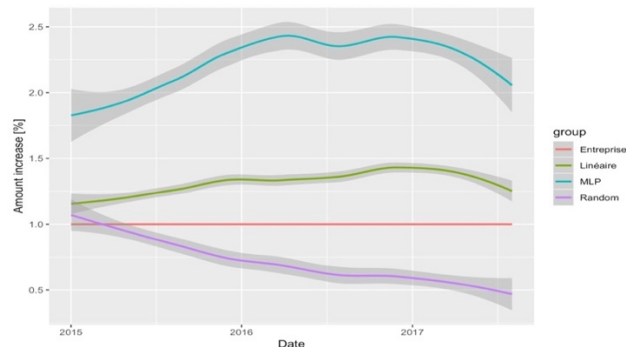
Pour l'apprentissage, les données de transactions de 58 jours bien distribuées à travers les années 2016 et 2017 ont été utilisées.

Seuls les utilisateurs ayant effectivement fait au moins un achat durant les journées étudiées ont été sélectionnés pour l'entraînement. Ainsi, la table des appréciations des caractéristiques des items a été générée pour les jours sélectionnés pour les utilisateurs actifs cette journée. Pour faciliter l'apprentissage, les observations sans achat ont été sous échantillonnées. Puisque les utilisateurs achètent beaucoup moins de produits qu'il y a de produits offerts les observations avec achat et sans achat étaient fortement déséquilibrés. Le sous-échantillonnage a permis au modèle d'apprendre plus facilement à discerner les observations positives des négatives. Autrement, étant donné le déséquilibre, prédire que l'utilisateur n'achèterait pas aurait été vrai près de 95% du temps.

Les paramètres ont été optimisés par validation croisée avec 5 groupes différents échantillonnés de l'ensemble d'entraînement. L'ensemble de test a été utilisé pour évaluer la performance du modèle.

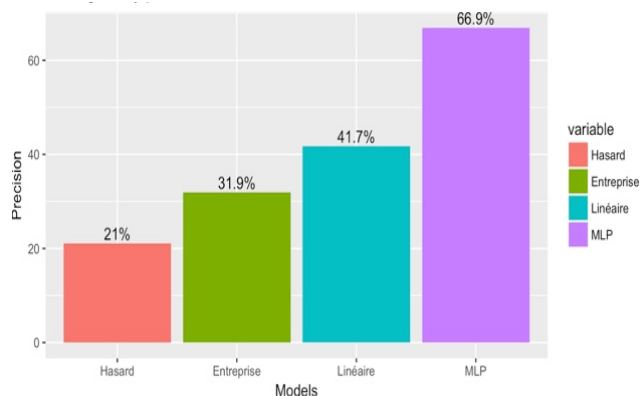
Finalement, la validation s'est faite avec en utilisant les autres jours entre 2016 et 2017 avec la métrique de performance financière et la précision telles que décrites en section 4.2 et 2.3.1 respectivement. Les performances de quatre modèles ont été évaluées. Celles du modèle linéaire et du réseau de neurones, ainsi que celles du modèle de l'entreprise et d'un modèle faisant des recommandations aléatoires.

La Figure 9 montre les courbes lissées des revenus quotidiens des différents modèles testés. Afin de conserver les revenus de l'entreprise partenaire confidentiels, les résultats ont été normalisés par ceux du modèle en *Entreprise*. On remarque que le modèle linéaire augmente en moyenne les revenus quotidiens de l'ensemble test de 32%. On observe également que la courbe du réseau de neurones fournit une hausse des revenus quotidiens chiffrée à une moyenne quotidienne de 123%.



**Figure 9 : Lissage des ventes par jour selon le modèle**

La Figure 10 montre la précision quotidienne moyenne selon les modèles. On constate que les résultats suivent la tendance observée sur la Figure 9.



**Figure 10 : Précision quotidienne moyenne des modèles**

Le modèle basé sur les réseaux de neurones obtient une précision d'environ 67%, tandis que le modèle linéaire obtient une précision de 42%.

## 6 CONCLUSION

Cet article avait pour but de proposer et démontrer l'efficacité de faire des recommandations basées sur le contexte (géographique, temporelle, prix, etc.) en plus du contenu d'un item pour faire des recommandations pertinentes à l'aide d'un système de recommandations.

Ce faisant, un système utilisant d'autres informations que le contenu des items est capable d'identifier les contextes d'achats favorables d'un utilisateur. Cela permet au système de faire des recommandations pertinentes pour un utilisateur même si le contenu des items est nouveau pour celui-ci.

Deux modèles ont été testés. Un modèle linéaire ainsi qu'un réseau de neurones. La performance des modèles a été mesurée à l'aide d'une métrique permettant de mesurer les retombées financières des modèles sur les ensembles de validation.

Les résultats ont été obtenus en testant les méthodes sur les données d'une entreprise de commerce en ligne qui offre des produits géolocalisés et disponibles pour un temps limité.

Les résultats sur les données de l'entreprise ont montré que le modèle linéaire améliorerait en moyenne les revenus de 30% par jours contre 125% par jours pour le modèle utilisant les réseaux de neurones.

Une amélioration possible de la méthode serait de prendre en compte les similitudes entre clients et entre le contenu des items. Cette amélioration permettrait de faire des recommandations collaboratives basées sur les similitudes entre clients et items. Une piste envisageable pour cela serait de projeter les clients et les items dans l'espace sémantique des termes permettant de décrire les items.

C'est d'ailleurs une piste qui sera poursuivie afin de tenter de générer des prévisions temporelles de la demande pour des clients individuels. L'idée étant de vérifier si les recommandations et prévisions qu'il est possible de générer avec les SR sont cohérentes avec les prévisions des systèmes experts utilisés pour la prise de décisions d'approvisionnement.

Ce problème sera d'ailleurs étudié plus en détail sur le cas industriel de Logistik Unicorp, une entreprise de gestion des programmes d'uniformes pour des organisations variées.

## 7 REMERCIEMENTS

Les auteurs expriment leur reconnaissance au partenaire industriel Tuango et Logistik Unicorp pour leur collaboration au projet et la fourniture des données. Les auteurs soulignent également le support financier de MITACS IT09078 et IT12058

## 8 RÉFÉRENCES

Adomavicius G., Tuzhilin A., Toward the next generation of recommender systems: a survey of the state of the art and possible extensions, *IEEE Transactions on Knowledge and Data Engineering* 17 (2005) 734–749.

Balabanović, M., & Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of the ACM*, vol. 40(3), pp. 66-72.

Baltrunas, L., & Amatriain, X. (2009). Towards time-dependant recommendation based on implicit feedback. In *Workshop on context-aware recommender systems (CARS'09)*. New York, NY, USA, 25 October.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, vol. 12(4), pp. 331-370.

Covington, P., Adams, J., & Sargin, E. (2016). Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems* (pp.191-198). ACM, Boston, MA, USA, 15-19 September.

Dadouchi, C., & Agard, B. (2017). État de l'art sur les systèmes de recommandations. *12ème Congrès International de Génie Industriel – CIGI 2017*, Compiègne, France, 3-5 May.

Dadouchi, C., & Agard, B. (2018). Lowering penalties related to stock-outs by shifting demand in product recommendation systems. *Decision Support Systems*, 114, pp. 61-69.

Gori, M. & Pucci, A. (2006). Research Paper Recommender Systems: A Random-Walk Based Approach. In 2006 IEEE/WIC/ACM *International Conference on Web Intelligence* (pp. 778 – 781). WI, Hong Kong, China, 18-22 December.

Gunawardana, A., & Shani, G. (2009). A survey of accuracy evaluation metrics of recommendation tasks. *Journal of Machine Learning Research*, 10, pp. 2935-2962.

Hu, Y., Koren, Y., & Volinsky, C. (2008) Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference* (pp. 263-272). IEEE, Pisa, Italy, 15-19 December.

Jannach, D., Resnick, P., Tuzhilin, A., & Zanker, M. (2016). Recommender systems—beyond matrix completion. *Communications of the ACM*, 59(11), pp. 94-102.

Kefalas, P., & Manolopoulos, Y. (2017). A time-aware spatio-textual recommender system. *Expert Systems with Applications*, 78, pp. 396-406.

Kelly, D., & Teevan, J. (2003). Implicit feedback for inferring user preference: a bibliography. In *Acm Sigir Forum* (vol. 37(2), pp. 18-28). ACM, Toronto, ON, Canada, 28-1 July, August.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8), pp. 30-37.

Lops, P., De Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In Ricci, F., Rokach L., Shapira B., Kantor, P. (eds) *Recommender systems handbook*, (pp. 73-105). Boston, MA, Springer.

McCarthy, J. F., & Anagnost, T. D. (1998). MusicFX: an arbiter of group preferences for computer supported collaborative workouts. In *Proceedings of the 1998 ACM conference on Computer supported cooperative work* (pp. 363-372). ACM, Seattle, WA, USA, 14-18 November.

McNee, S. M., Riedl, J., & Konstan, J. A. (2006). Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI'06 extended abstracts on Human factors in computing systems* (pp. 1097-1101). ACM, Montréal, QC, Canada, 22-27 April.

Nadolski, R. J., Van den Berg, B., Berlanga, A. J., Drachler, H., Hummel, H. G., Koper, R., & Sloep, P. B. (2009). Simulating light-weight personalised recommender systems in learning networks: A case for pedagogy-oriented and rating-based hybrid recommendation strategies. *Journal of Artificial Societies and Social Simulation*, 12(1), pp. 4-25.

Oard, D. W., & Kim, J. (1998). Implicit feedback for recommender systems. Technical Report. *Proceedings of the AAAI workshop on recommender systems*. Madison, WI, USA, 26-31 July.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The PageRank citation ranking: Bringing order to the web. Technical Report. *Stanford InfoLab*.

Panniello, U., Tuzhilin, A., & Gorgoglione, M. (2014). Comparing context-aware recommender systems in terms of



- accuracy and diversity. *User Modeling and User-Adapted Interaction*, 24(1-2), pp. 35-65.
- Paterek, A. (2007). Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop* (Vol. 2007, pp. 5-8). KDD, San Jose, Ca, USA, 12 August.
- Pazzani M.J., Billsus D. (2007) Content-Based Recommendation Systems. In: Brusilovsky P., Kobsa A., Nejdl W. (eds) *The Adaptive Web*. Lecture Notes in Computer Science, vol 4321. Springer, Berlin, Heidelberg.
- Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work* (pp. 175-186). ACM, Chapel Hill, NC, USA, 22-26 October.
- Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to recommender systems handbook. In Ricci, F., Rokach L., Shapira B., Kantor, P. (eds) *Recommender systems handbook*, (pp. 1-35). Boston, MA, Springer.
- Rodriguez, M., Posse, C., & Zhang, E. (2012). Multiple objective optimization in recommender systems. In *Proceedings of the sixth ACM conference on Recommender systems* (pp. 11-18). ACM, Dublin, Ireland, 9-13 September.
- Salakhutdinov, R., & Mnih, A. (2008). Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proceedings of the 25th international conference on Machine learning* (pp. 880-887). ACM, Helsinki, Fabianinkatu, Finland, 05-09 July.
- Sasaki, Y. (2007). The truth of the F-measure. *Teach Tutor mater*, 1(5), pp.1-5.
- Schafer J.B., Frankowski D., Herlocker J., Sen S. (2007) Collaborative Filtering Recommender Systems. In: Brusilovsky P., Kobsa A., Nejdl W. (eds) *The Adaptive Web*. Lecture Notes in Computer Science, vol 4321. Springer, Berlin, Heidelberg.
- Shani, G., & Gunawardana, A. (2011). Evaluating recommendation systems. In Ricci, F., Rokach L., Shapira B., Kantor, P. (eds) *Recommender systems handbook*, (pp. 257-297). Boston, MA, Springer.
- St-Aubin, P., & Agard, B. (2017) Data mining vs RFM en marketing : une étude comparative dans le secteur du e-commerce. *12ème Congrès International de Génie Industriel – CIGI 2017*, Compiègne, France, 3-5 May.
- Wang, H., Wang, N., & Yeung, D. Y. (2015). Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1235-1244). ACM, Sydney, NSW, Australia, 10-13 August.
- Wei, K., Huang, J., & Fu, S. (2007). A survey of e-commerce recommender systems. In *Service systems and service management, 2007 international conference on* (pp. 1-5). IEEE, Chengdu, China, 9-11 June.
- Yin, H., Sun, Y., Cui, B., Hu, Z., & Chen, L. (2013, August). LCARS: a location-content-aware recommender system. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* on (pp. 221-229). ACM, Chicago, Illinois, USA, 11-14 August.
- Zhou, X., Xu, Y., Li, Y., Josang, A., & Cox, C. (2012). The state-of-the-art in personalized recommender systems for social networking. *Artificial Intelligence Review*, 37(2), pp. 119-132.