# Système de recommandation avec prise en compte de la prévision de disponibilité des catégories de produits

Armel Jacques Nzekon Nzeko'o, Hamza Adamou, Maurice Tchuente

▶ **To cite this version:**

Armel Jacques Nzekon Nzeko'o, Hamza Adamou, Maurice Tchuente. Système de recommandation avec prise en compte de la prévision de disponibilité des catégories de produits. Revue Africaine de Recherche en Informatique et Mathématiques Appliquées, 2022, Volume 36 - Special issue CRI 2021, Volume 36 - Special issue CRI 2021 (36), 10.46298/arima.9156 . hal-03591997v2

**HAL Id: hal-03591997**
**https://inria.hal.science/hal-03591997v2**

Submitted on 1 Jun 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ARIMA

épisciences

# Système de recommandation avec prise en compte de la prévision de disponibilité des catégories de produits

**Armel Jacques NZEKON NZEKO'O**[*1,2], **Hamza ADAMOU**[1,2], **Maurice TCHUENTE**[1,2]

[1]Université de Yaoundé I, Département d'Informatique, BP 812, Yaoundé, Cameroun
[2]Sorbonne Université, IRD, UMI 209 UMMISCO, F-93143, Bondy, France

*E-mail : armel.nzekon@facsciences-uy1.cm

*DOI : 10.46298/arima.9156*
*Soumis le 1 Mars 2022 - Publié le 20 mai 2022*
Volume : **36** - Année : **2022**
Numéro spécial : **CRI 2021**
Éditeurs : *René Ndoundam, Eric Badouel, Maurice Tchuente, Paulin Melatagia*

**Résumé**

Recommander des articles appropriés aux utilisateurs est crucial dans les plateformes de commerce électronique et de streaming. Dans certaines situations, un client a une préférence pour un produit en fonction des caractéristiques du produit et du contexte temporel courant. Il est donc judicieux de prendre en compte ces aspects afin d'améliorer la qualité des recommandations. Dans cet article, nous proposons des systèmes de recommandation basés sur la prédiction de disponibilité des catégories des produits en fonction du contexte temporel. En effet, le classement des recommandations Top-N proposées par le système de recommandation initial est mis à jour de manière à favoriser les produits aux catégories prédites disponibles. Par ailleurs, nous proposons un algorithme pour le choix du contexte temporel adéquat à la considération de la prédiction de disponibilité des catégories. Des expérimentations sont menées sur quatre jeux de données et des comparaisons sont effectuées sur les résultats de trois systèmes de recommandation de base avec et sans intégration des prédictions de disponibilité, suivant les métriques d'évaluation Hit-ratio, MAP et F1-score. On constate que dans 75% des cas, pour avoir la meilleure performance, il est nécessaire d'intégrer les prédictions de disponibilité des catégories. Ce gain peut même aller à plus de 12% quel que soit le jeu de données. Tout ceci confirme la pertinence de notre contribution.

**Mots-Clés**

Système de recommandation, Prédiction de disponibilité des ressources, Recommandation Top-N

## I INTRODUCTION

Les sites de e-commerce proposent divers produits (chansons, vidéos, vêtements, etc) à leurs utilisateurs. Le nombre toujours croissant de produits pose un problème de choix aux utilisateurs. Les systèmes de recommandation répondent à ce problème en filtrant les produits pour présenter uniquement les plus susceptibles d'intéresser les utilisateurs [1].

L'une des tâches les plus populaires dans le domaine des systèmes de recommandation est la recommandation top-N [13]. Dans ce cas, le système propose une liste ordonnée de $N$ produits les plus probables d'avoir l'attention de l'utilisateur. Nous nous attardons sur cette tâche, car elle peut être abordée en utilisant des données implicites disponibles en grand volume à travers l'historique des actions des utilisateurs sur les produits.

Les approches classiques des systèmes de recommandation sont le filtrage collaboratif et le filtrage basé sur le contenu. Le premier utilise les similarités entre les utilisateurs, et le second, les informations liées aux caractéristiques des produits [4, 31]. L'autre approche est celle des systèmes de recommandation hybrides qui résultent de la combinaison des deux précédentes et dont le but est de tirer profit des atouts de chacune [11]. Cependant, dans de nombreux cas, il serait important de prendre en compte des informations contextuelles supplémentaires telles que la saison, l'heure ou le lieu ; par exemple, pour les vêtements, il est inapproprié de recommander des vêtements lourds pendant une période chaude (été, saison sèche).

Par ailleurs, pour pallier la limite précédente, des auteurs proposent d'intégrer d'autres informations contextuelles dans les techniques classiques de filtrage collaboratif et de filtrage basé sur le contenu. L'une des dimensions les plus considérées est le temps, dont l'une des stratégies courantes est de faire varier l'importance des données en fonction de leur contexte temporel (saison, moment de la journée) [30, 5]. A cet effet, nous proposons des systèmes de recommandation hybrides qui intègrent les contextes temporels via la prédiction de disponibilité des catégories de produits en fonction du contexte temporel.

La prédiction de disponibilité des ressources est utilisée dans les systèmes de calculs de machines volontaires afin de déterminer quelles machines seront utilisables à un moment donné pour effectuer un sous-ensemble d'opérations préalablement distribuées, et retourner leurs résultats qui participent à la réalisation d'un plus gros calcul, de manière à ce que tout le système simule une super-machine de haute performance [3, 37]. Nous l'utilisons ici, pour déterminer quelles catégories de produits peuvent être choisies durant une plage de temps donnée. Pour chaque catégorie, nous calculons un score qui représente les chances que des produits de cette catégorie soient achetés durant une période donnée. Les scores obtenus permettent de mettre à jour les classements des recommandations top-N.

Le reste de cet article est organisé comme suit : la section II rappelle les principes de base des systèmes de recommandation Top-N, la section III présente le modèle de prédiction de disponibilité des catégories de produits. L'usage des scores de prédiction des disponibilités des catégories des produits dans le processus de recommandation est décrit en section IV. Les expérimentations et les résultats sont présentés dans la section V. On conclut en section VI.

## II SYSTÈMES DE RECOMMANDATION TOP-N

La recommandation top-N propose à chaque utilisateur $u$ une liste des $N$ produits les plus susceptibles de l'intéresser ; ces produits sont classés par ordre décroissant de préférence. Cela suppose que pour toute paire utilisateur-produit $(u, i)$, le système de recommandation est capable de donner une estimation de la préférence de l'utilisateur $u$ pour le produit $i$.

Les plateformes de recommandation top-N sont très fréquentes sur le web. Leur mise en œuvre concerne divers types d'objets comme les chansons sur Last.fm, Yahoo ! music, Apple music [22], les films et les vidéos sur youtube, netflix, hulu [13], les liens sociaux sur facebook, linkedin, twitter [42], les produits physiques sur amazon, ebay [27] ou les journaux d'information sur Google news, Yahoo ! news [39].

Tout ceci fait des recommandations top-N la tâche la plus abordée en pratique [13], d'où l'intérêt de contribuer aux approches qui permettent leur mise en œuvre : filtrage collaboratif (sous-section 2.1), filtrage collaboratif enrichi par des informations basées sur le contenu (sous-section 2.2), et système de recommandation Top-N avec dynamique temporelle (sous-section 2.3).

## 2.1 Filtrage collaboratif

L'approche par filtrage collaboratif suppose que « les utilisateurs qui ont eu les mêmes préférences dans le passé auront les mêmes préférences dans le futur ». C'est l'approche la plus utilisée et la plus étudiée dans la littérature, notamment au travers des techniques basées sur la mémoire comme les k-plus proches voisins [21], les graphes de recommandation [6], et les techniques basées sur un modèle d'apprentissage automatique comme les réseaux de neurones [18], les réseaux bayésiens ou la factorisation matricielle [19].

### *2.1.1 Techniques basées sur la mémoire*

Les techniques de recommandation basées sur la mémoire reposent sur deux principales étapes : la première pour déterminer les similarités (corrélations) entre les utilisateurs (ou les produits) et la seconde pour calculer les prédictions. Dans cette catégorie, les techniques les plus utilisées sont les $K$-plus proches voisins (KNN) [21, 40], et les graphes de recommandation [6, 42].

Les graphes de recommandation sont facilement interprétables et fournissent un cadre naturel et intuitif pour différents types d'applications. C'est pour ces raisons que nous avons choisi un graphe de recommandation comme système de recommandation de base dans notre travail et c'est uniquement ce type de systèmes que nous présentons dans la suite de cette section.

#### Graphe de recommandation

Les systèmes de recommandation basés sur les graphes ont les mêmes étapes que les autres systèmes basés sur la mémoire comme KNN : une étape pour établir les corrélations entre les utilisateurs et les produits (le graphe), et une seconde étape pour le calcul des recommandations (algorithme de calcul des recommandations à partir du graphe).

**Construction du graphe.** Le graphe de recommandation qui est construit à partir de la matrice des notes binaires est le graphe biparti classique (BIP). Dans ce graphe, chaque utilisateur ainsi que chaque produit est représenté par un nœud. Lorsqu'un utilisateur $u$ marque un intérêt positif pour un produit $i$, le nœud de $u$ est relié au nœud de $i$ par une arête bidirectionnelle $(u, i)$. Ce graphe est fréquent pour le calcul des recommandations à l'exemple des travaux de Baluja et al. [6] sur la recommandation des vidéos sur Youtube et ceux de Yan et al. sur la recommandation des tweets [42].

**Calcul des recommandations.** Dans le cas du graphe biparti classique (BIP), l'hypothèse pour le calcul des recommandations repose sur la proximité de l'utilisateur cible $u$ avec les prochains produits qu'il va sélectionner. Ainsi, l'objectif est de recommander N produits que $u$ n'a pas encore sélectionnés et qui sont les plus proches de lui dans le graphe. Suivant ce principe, la plupart des algorithmes de calcul des recommandations sont basés sur une marche aléatoire dans laquelle le nœud source est celui associé à $u$ : Injected Preference Fusion (IPF) [41], score de Katz [15], HITS [20] ou le PageRank personnalisé [17].

### *2.1.2 Techniques basées sur un modèle*

Les techniques basées sur la mémoire utilise toutes les données disponibles à chaque fois qu'il faut recommander, et donc le modèle entier doit être en mémoire, ce qui est un souci si la mémoire est limitée. Dans ce cas, les techniques basées sur un modèle d'apprentissage peuvent être

une bonne alternative. En effet, ces dernières construisent des modèles de classification et/ou de régression de l'apprentissage automatique à partir de l'ensemble des données connues. Le modèle résultant intègre des informations latentes déduites des données brutes de telle sorte qu'il devient possible de calculer de nouvelles recommandations sans utiliser toutes des données.

Les techniques de recommandation basées sur un modèle reposent sur deux principales étapes : la première est l'apprentissage du modèle de prédiction et la seconde est le calcul des recommandations [10]. La plupart des systèmes de recommandation de cette catégorie reposent sur des modèles de réduction de dimensionnalité ou de clustering dans le but d'écarter les utilisateurs ou les produits non représentatifs. Dans cette catégorie, on retrouve des modèles de réseaux de neurones [18], les machines à vecteurs de supports (SVM), et les modèles les plus utilisés sont ceux de la factorisation matricielle comme SVD (Singular Value Decomposition) [23] et ALS (Matrix factorization with Alternating Least Squares) [19].

**Factorisation matricielle**

Dans cette section nous présentons le principe de fonctionnement des systèmes de recommandation qui reposent sur la factorisation matricielle, et plus précisément celui de la décomposition en valeurs singulières (SVD). De tels systèmes constituent une référence dans le domaine des systèmes de recommandation depuis la victoire de Koren au Grand Prix Netflix 2009 [24, 25]. Cette réalité justifie l'intérêt que nous accordons à la factorisation matricielle.

**Décomposition en valeurs singulières.** Pour l'application de cette méthode, on part de la matrice binaire $R$ de l'intérêt que chaque utilisateur accorde aux produits ($R[u,i] = 1$ si $u$ aime $i$ et 0 sinon), pour calculer un ensemble de $k$ facteurs latents à partir desquels on caractérise à la fois les produits et les utilisateurs.

La valeur $k$ doit être inférieure ou égale au rang de la matrice des notes ; pour $r = rang(A)$, $k \leq r$. Une fois que la valeur de $k$ est fixée, la matrice $R^{|U|\times|I|}$ peut être approximée par $\widetilde{R}$ le produit de trois matrices construites à partir des $k$ vecteurs propres de la matrice $R \cdot R^T$ (resp. $R^T \cdot R$) et des $k$ plus grandes valeurs propres de $R$. La matrice $\widetilde{R}$ est calculée comme suit :

$$\widetilde{R} = \widetilde{U} \times \Sigma \times \widetilde{I}^T$$

— $\widetilde{U}$ est une matrice de taille $|U| \times k$ qui décrit les utilisateurs par $k$ facteurs latents. Les colonnes de $\widetilde{U}$ sont les vecteurs propres des plus grandes valeurs propres de $R \cdot R^T$.
— $\Sigma$ est une matrice carrée diagonale de taille $k \times k$. Les valeurs sur la diagonale principale sont les $k$ plus grandes valeurs propres de la matrice $R \cdot R^T$ dans l'ordre décroissant.
— $\widetilde{I}$ est une matrice de taille $|I| \times k$ qui décrit tous les produits par $k$ facteurs latents. Les colonnes de $\widetilde{I}$ sont les vecteurs propres des plus grandes valeurs propres de $R^T \cdot R$.

La valeur $k$ est choisie pour assurer une bonne approximation de $R^{|U|\times|I|}$ par $\widetilde{R}$.

**Calcul des recommandations.** Le principe ici est de recommander à un utilisateur $u$ les produits qui lui sont similaires dans le nouvel espace vectoriel à $k$ dimensions. Ainsi, pour estimer la préférence qu'un utilisateur $u$ peut avoir pour un produit $i$, on calcule le produit des vecteurs de $u$ et $i$ dans le nouvel espace vectoriel. Cette estimation est donnée par :

$$preference(u,i) = \sum_{f=1}^{k} \widetilde{U}_{u,f} \times \widetilde{I}_{f,i}$$

Les N premiers produits que l'utilisateur cible $u$ n'a pas encore sélectionnés et qui sont associés aux plus grandes valeurs de $preference(u, i_*)$ lui sont recommandés.

### *2.1.3 Limites du filtrage collaboratif*

Dans le principe du filtrage collaboratif, lorsqu'il faut recommander de nouveaux produits à un utilisateur cible $u$, on lui propose des produits que des utilisateurs qui sont similaires à $u$ ont aimé. Ce principe impose une notion de similarité qui donne naissance aux limites suivantes : le démarrage à froid (difficulté de recommander des produits à un nouvel utilisateur), le problème des utilisateurs $Gray-sheep$ (difficulté de recommander de bon produits aux utilisateurs dont les préférences divergent grandement de celles des autres) et la sensibilité aux attaques de faux profils (la présence de faux profiles similaire à $u$ influence les recommandations faites à $u$).

Les techniques pures du filtrage collaboratif reposent soit sur des matrices explicites des notes que les utilisateurs accordent aux produits, soit sur des matrices binaires construites à partir de l'historique des actions des utilisateurs sur les produits. Dans les deux situations, plusieurs autres types de données sont ignorés. On peut citer les textes disponibles pour décrire les caractéristiques des produits, et les horodatages des actions des utilisateurs sur les produits.

La prise en compte de telles données supplémentaires peut pallier à certaines limites du filtrage collaboratif comme le manque de données : l'usage des catégories des produits permet de proposer des produits de catégories similaires à ceux que l'utilisateur cible a sélectionnés dans le passé. Ainsi, même les produits récemment ajoutés au catalogue de la plateforme peuvent être recommandés. De plus, le fait de considérer les horodatages qui sont très souvent enregistrés dans l'historique des actions des utilisateurs, peut permettre d'intégrer le fait que les préférences des utilisateurs changent avec le temps, et améliorer la qualité des recommandations.

## 2.2 Filtrage collaboratif enrichi par des informations basées sur le contenu

Les systèmes de recommandation qui prennent en compte des informations sur le contenu sont conçus pour les cas dans lesquels les produits peuvent être décrits par des caractéristiques diverses. Cette condition est vérifiée dans de nombreuses plateformes dédiées aux films, chansons et livres, pour lesquels on a des données sur les genres, les auteurs et les mots-clés.

Il est donc judicieux d'intégrer les informations du contenu dans les techniques du filtrage collaboratif pour pallier le manque de données et améliorer la prise en compte des goûts et préférences des utilisateurs. A cet effet, cette section présente un aperçu du filtrage basé sur le contenu (sous-section 2.2.1) et des systèmes de recommandation hybrides (sous-section 2.2.2).

### *2.2.1 Filtrage basé sur le contenu*

Les techniques du filtrage basé sur le contenu supposent que les catégories de produits préférées d'un utilisateur dans le passé seront les mêmes dans le futur. Ainsi, la recommandation basée sur le contenu tente de comparer des produits en utilisant leurs caractéristiques (genre, acteurs, éditeur, auteur) pour recommander de nouveaux produits similaires à ceux précédemment sélectionnés par l'utilisateur cible [28].

Le filtrage basé sur le contenu impose des étapes de base : la phase de pré-traitement des données (extraction des mots-clés ou des informations qu'on peut catégoriser à partir des données disponibles), la phase d'apprentissage des profils des utilisateurs et des produits (le profil d'un utilisateur/produit est un sac de mots-clés et leurs fréquences dans une représentation en espace vectoriel), et enfin la phase de recommandation de nouveaux produits aux utilisateurs (pour chaque couple $(u, i)$, un score $F(u, i)$ qui indique à quel point les caractéristiques de $i$ correspondent aux préférences de $u$, et les N meilleurs sont recommandés) [2].

Le principal avantage du filtrage basé sur le contenu est sa capacité à recommander des nouveaux produits du catalogue. Ceci permet de résoudre le problème du démarrage à froid des nouveaux produits qu'on rencontre avec le filtrage collaboratif. Par ailleurs, le filtrage basé sur le contenu est utile lorsqu'il y a très peu de données ou lorsque l'ensemble des produits change rapidement, mais pas l'ensemble de leurs étiquettes ou catégories.

Cependant, les techniques de filtrage basé sur le contenu souffrent de la sur-spécialisation des recommandations, car ces techniques suggèrent toujours des produits similaires à ceux déjà connus de l'utilisateur et ne peuvent donc pas fournir de recommandations nouvelles ou diverses qui puissent surprendre l'utilisateur [11].

#### *2.2.2 Systèmes de recommandation hybrides*

Pour pallier l'un des inconvénients du filtrage basé sur le contenu, à savoir la faible diversité des recommandations, cette approche peut être combinée au filtrage collaboratif en raison de sa capacité à exploiter les caractéristiques des produits dans le processus de recommandation. Cette combinaison permet de pallier le problème de démarrage à froid des produits, le manque de données dont souffre le filtrage collaboratif, de produire des recommandations diverses contrairement au filtrage basé sur le contenu, et d'augmenter la couverture des recommandations (possibilité d'atteindre des produits qui n'ont pas été sélectionnés par des voisins de l'utilisateur cible, mais qui ont des catégories en commun avec ses précédentes sélection).

Il y a plusieurs manières de combiner le filtrage collaboratif et le filtrage basé sur le contenu pour obtenir un système de recommandation hybride [1] : implémenter séparément les deux approches et combiner leurs recommandations [8], incorporer certaines caractéristiques basées sur le contenu dans une technique du filtrage collaboratif [9], incorporer certaines caractéristiques de collaboration dans une technique basée sur le contenu [34], et construire un modèle qui intègre à la fois des caractéristiques de collaboration et du contenu [32].

### 2.3 Système de recommandation Top-N avec dynamique temporelle

Les précédentes approches de recommandation ne tiennent pas compte de la dynamique temporelle des actions des utilisateurs. Ceci veut dire qu'elles ne font pas de différence entre les données récentes et les données plus anciennes et ne font également aucune différence entre le comportement des utilisateurs en fonction des moments de la journée ou des saisons. La prise en compte de tels aspects a pourtant un impact considérable sur la qualité des résultats.

Les systèmes de recommandation qui tiennent compte du temps ont diverses façons de considérer et d'intégrer la dynamique temporelle. La plupart font varier l'importance des données utilisées : soit cette importance décroit dans le temps (sous-section 2.3.1), soit cette importance varie en fonction du contexte temporel (sous-section 2.3.2).

#### *2.3.1 Décroissance de l'importance des données en fonction du temps*

Dans ce type de systèmes de recommandation, le temps est considéré comme étant une variable quantitative continue. La fonction $F(u, i, t)$ qui permet d'estimer la préférence de l'utilisateur $u$ pour le produit $i$ à l'instant $t$. Les systèmes de cette catégorie peuvent être classés en trois sous-catégories : découpage du temps en tranches (préférence à court terme), combinaison des préférences à long et à court terme et enfin l'usage des fonctions de décroissance temporelle.

**Découpage du temps en tranches.** On considère que l'importance des données est éphémère et donc devient obsolète au bout d'un certain temps. Ainsi, une fois que la taille des tranches de temps est fixée, chaque donnée est utilisée durant une seule tranche de temps et est ignorée les tranches suivantes. Cette stratégie ne capture que les préférences à court terme [12, 38].

**Combinaison des préférences à long et à court termes.** Le procédé précédent ignore les préférences à long terme des utilisateurs (celles qui perdurent dans le temps). Pour pallier cette limite, des systèmes de recommandation sont construits à la fois avec des données anciennes qui contiennent les préférences à long terme, et des données récentes qui représentent les préférences à court terme. Ensuite, le système combine ces deux types de préférences en accordant des poids différents aux données en fonction de l'horizon temporel qui leur correspond [41, 35].

**Fonctions de décroissance temporelle.** Dans les deux précédentes catégories, on procède à un découpage du temps en tranches, imposant une conception discontinue du temps. Cependant, le processus suivant lequel les produits se démodent peut être progressif et continu. Ainsi, plusieurs travaux intègrent des fonctions de décroissance temporelle qui permettent de pondérer les données de telle sorte que leur poids diminue au fil du temps. Ce qui est fidèle au fait que les récentes sélections des utilisateurs reflètent mieux leurs goûts et préférences actuels [14, 23].

### *2.3.2 Variation des préférences en fonction du contexte temporel*

L'importance des données suivant le contexte temporel s'illustre facilement par des exemples : les vêtements les plus intéressants en hiver sont chauds tandis que ceux intéressants en été sont légers. De même, les chansons qu'on aime écouter en période de travail ne sont pas les mêmes qu'on écoute dans la soirée. Au vu de telles situations, la prise en compte du contexte temporel sur l'intérêt accordé à un produit est importante pour améliorer la qualité des recommandations.

La dimension temporelle peut être modélisée comme une variable qualitative liée au contexte temporel, ce qui permet de traiter les données différemment en fonction des valeurs contextuelles associées. A cet effet, on a deux principales familles pour le calcul des recommandations : la première est celle du pré-filtrage et la seconde celle du post-filtrage contextuels.

Dans le pré-filtrage, pour calculer $F(u, i, t)$ la préférence de l'utilisateur $u$ pour le produit $i$ à l'instant $t$, un filtre $f(u, i, t)$ est appliqué pour pénaliser les données qui ne sont pas pertinentes pour le contexte temporel de $t$. Le filtre le plus trivial ignore toutes les données dont le contexte temporel est différent de $t$ [26]. Concernant le post-filtrage contextuel, $f(u, i, t)$ est utilisé pour adapter $F(u, i)$ l'estimation de la préférence de l'utilisateur $u$ pour le produit $i$ lorsque le temps n'est pas pris en compte. La valeur de $F(u, i, t)$ est alors fonction de $F(u, i)$ et $f(u, i, t)$ [30].

Dans cet article, les systèmes de recommandation proposés correspondent au filtrage collaboratif enrichi par des informations basées sur la prévision de disponibilité des catégories des produits suivant la stratégie post-filtrage de la variation des préférences en fonction du contexte temporel. On a donc des systèmes de recommandation hybrides avec dynamique temporelle.

## III PRÉDICTION DE DISPONIBILITÉ DES CATÉGORIES DES PRODUITS

Un système de prédiction de disponibilité des ressources est utilisé pour rendre les calculs beaucoup plus fiables dans les systèmes de calcul haute performance [37]. Ce système permet de déterminer si une ressource de calcul sera disponible ou non dans une plage de temps. Cette approche de prédiction de disponibilité est appliquée aux catégories de produits en trois phases : le pré-traitement des données (section 3.1), ensuite la phase de construction du modèle de prédiction (section 3.2) et enfin la prédiction des disponibilités des catégories (section 3.3).

### 3.1 Pré-traitement des données

Les données manipulées sont des flots de liens de la forme $\{(t_k, u_k, i_k, c_k)\}_{k=1..L}$ où $(t_k, u_k, i_k, c_k)$ signifie qu'à l'instant $t_k$, l'utilisateur $u_k$ a sélectionné/acheté le produit $i_k$ qui est de catégorie $c_k$. Pour la prédiction de disponibilité des catégories, une décomposition du temps en tranches de longueur $d$ est adoptée (la durée totale du flot de liens est divisée en tranches toutes de taille $d$). Ce découpage peut correspondre par exemple à un moment de la journée (matin, après-midi, soirée, nuit) ou à une saison (été, printemps, automne, hiver).

Dans la suite, on associe à chaque catégorie un vecteur binaire avec autant de bits que de tranches de temps. Le bit d'une tranche $t_k$ d'une catégorie $c_k$ est à $1$ si au moins $\gamma$ liens de la tranche $t_k$ sont étiquetés par $c_k$, et à $0$ sinon. Le paramètre $\gamma$ est fixé de telle sorte que les proportions des $0$ et des $1$ du jeu de données soient le plus proches possible de $50\%$ afin de garantir l'équilibre de classes des jeux d'apprentissage. Ainsi, une bonne valeur de $\gamma$ permet d'éviter que l'algorithme d'apprentissage du modèle de prédiction de disponibilité des catégories soit biaisé vers la classe dominante et conduire à des prédictions potentiellement moins robustes.

L'application de l'algorithme de prédiction de disponibilité sur le vecteur binaire d'une catégorie, nécessite un ensemble d'exemples $\{(x_i, y_i)\}_{i=1..n}$. Chaque couple $(x_i, y_i)$ est constitué de $x_i$, une suite de bits de taille $til$, et $y_i$, le bit qui suit directement $x_i$ dans le vecteur binaire de la catégorie $c_k$ concernée. Les exemples se chevauchent avec décalage d'une unité de temps à chaque fois comme présenté sur la figure 1.

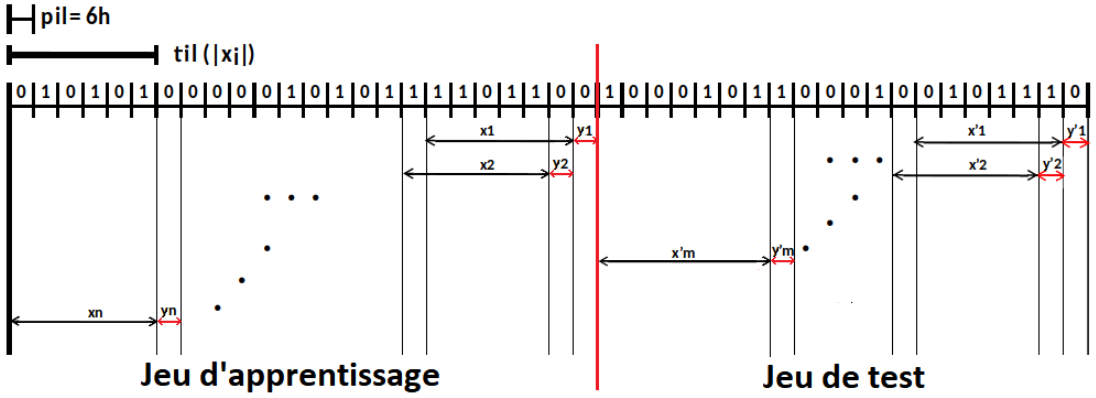


FIGURE 1 – Exemple de segmentation du vecteur binaire d'une catégorie. Le contexte temporel c'est moment de la journée et donc les tranches de temps $t_k$ sont de durée $6h$.

### 3.2 Modèle de prédiction de disponibilité

Pour chaque catégorie, une fois que tous les exemples sont construits à partir du flot de liens, une partie de ces exemples est utilisée pour l'entraînement du modèle et le reste comme jeu de test. Afin de prédire une valeur $y_i$ inconnue à partir d'une valeur connue $x_i$, le modèle de prédiction choisi est le classifieur naïf de Bayes [36] car il est facile à mettre en œuvre et permet d'obtenir rapidement les résultats. Ce classifieur est basé sur l'application du théorème de *Bayes* avec l'hypothèse d'indépendance entre chaque paire d'attributs.

Étant donné un vecteur d'attributs $x_i = (x_{i_1}, x_{i_2}, \ldots, x_{i_{til}})$ qui représente la ieme séquence de bits de taille $til$ de la catégorie concernée $c_k$, chacun des bits est associé à l'apparition (1) ou non (0) de $c_k$ durant la tranche de temps $t_k$, et la valeur de la prédiction de disponibilité $\hat{y}_i$ de $c_k$ dans la tranche de temps de taille $pil$ qui suit directement $x_i$ est la valeur qui maximise la *probabilité a posteriori* $P(y_i \mid x_{i_1}, x_{i_2}, \ldots, x_{i_{til}})$.

$$\hat{y}_i = \arg\max_{y_i} P(y_i) \prod_{j=1}^{p} P(x_{i_j} \mid y_i)$$

où $P(y_i)$ est la *probabilité a priori*, fréquence relative de la classe $c_k$ dans les données et $P(x_{i_j} \mid y_i)$ la probabilité conditionnelle de classe estimée à partir des données [36].

### 3.3 Prédiction de disponibilité des catégories des produits

Rendu à cette étape, il faut appliquer le modèle précédent pour prédire la disponibilité de chaque catégorie dans chaque instance du contexte temporel considéré. La valeur de $pil$ correspond donc à la durée du contexte temporel choisi. Par exemple, si le contexte temporel considéré est *"moment de la journée"*, alors $pil$ = *6 heures* et toutes les 6 heures, il faut une prédiction de disponibilité de chaque catégorie. La taille de $til$ doit permettre d'avoir suffisamment d'exemples pour prédire $y_i$ en $pil$ et d'avoir une bonne performance en précision.

## IV PRÉDICTIONS DE DISPONIBILITÉS ET RECOMMANDATION TOP-N

Cette section présente comment les systèmes de recommandation peuvent être combinés à la prédiction de disponibilité des catégories de produits. Nous commençons par présenter la stratégie de combinaison des systèmes de recommandations top-N à la prédiction de disponibilité des catégories des produits, nous décrivons les composantes de la combinaison à réaliser et clôturons par la présentation de la procédure de choix du contexte temporel le plus adéquat.

### 4.1 Principe de combinaison des systèmes de recommandations top-N et de la prédiction de disponibilité des catégories des produits

Dans cette section, nous décrivons comment calculer $R_{ui}^{t_k}$, l'estimation finale de la probabilité que l'utilisateur $u$ sélectionne le produit $i$ pendant la tranche de temps $t_k$.

À cet effet, on considère qu'après l'exécution d'un système de recommandation top-N, on obtient la matrice $M_{ui}^{t_k}$ qui contient toutes les estimations des préférences des utilisateurs pour les différents produits pendant la tranche de temps $t_k$. Autrement dit, $M_{ui}^{t_k}[u, i]$ quantifie la préférence de l'utilisateur $u$ pour le produit $i$.

On admet également que $T$, $U$, $I$ et $C$ sont respectivement les ensembles des tranches de temps $t_k$, des utilisateurs $u$, des produits $i$ et des catégories de produits $c$. La prédiction de disponibilité de chaque catégorie est appliquée autant de fois qu'il y a de tranches de temps dans $T$ et on obtient la matrice $M_{ct}$ qui contient les valeurs $M_{ct}[c, t_k] \in [0, 1]$ comme probabilité d'apparition de la catégorie $c$ dans la tranche de temps $t_k$.

L'influence des prédictions de disponibilité des catégories d'un produit $i$ dans la tranche $t_k$ est contenue dans la cellule $M_{it}[i, t_k]$ de la matrice $M_{it}$, dont les valeurs sont déduites de l'équation $M_{it} = M_{ic} \cdot M_{ct}$, où $M_{ic}$ est la matrice qui contient les relations d'appartenance des produits aux catégories, $M_{ic}[i, c] = 1$ si le produit $i$ est de catégorie $c$ et 0 sinon.

Pour intégrer l'influence des prédictions de disponibilité des catégories des produits dans les systèmes de recommandation, on effectue une somme pondérée dont la somme des poids des deux aspects est égale à 1. Ainsi, un paramètre $\beta \in [0, 1]$ est considéré pour calibrer l'influence de la prédiction des disponibilités dans la mise à jour des recommandations top-N, tandis que $(1 - \beta)$ calibre l'importance des résultats du système de recommandation de base.

La probabilité finale $R_{ui}^{t_k}$ que l'utilisateur $u$ sélectionne le produit $i$ dans la tranche de temps $t_k$ est calculée avec l'équation suivante :

$$R_{ui}^{t_k}[u,i] = M_{ui}^{t_k}[u,i] \cdot ((1-\beta) + \beta \cdot M_{it}[i,t_k])$$

Si $\beta = 0$ alors $R_{ui}^{t_k}[u,i] = M_{ui}^{t_k}[u,i]$, et si $\beta = 1$ alors $R_{ui}^{t_k}[u,i] = M_{ui}^{t_k}[u,i] \cdot M_{it}[i,t_k]$.

Finalement, le système de recommandation top-N renvoie les N premiers produits $i$ classés dans l'ordre décroissant des valeurs de $R_{ui}^{t_k}[u,i]$ pour l'utilisateur $u$ dans la tranche $t_k$.

## 4.2 Composantes des combinaisons systèmes de recommandation top-N et prédiction de disponibilité des catégories des produits

Dans ce travail, chaque système de recommandation qui intègre la prédiction de disponibilité a deux composantes, une première qui est un modèle de calcul des recommandations $\{POP, BIP, ALS\}$ et une seconde qui est un mode de prise en compte de la prédiction de disponibilité $\{PCc, PCn, PC1\}$. On a un total de 09 combinaisons considérées et dont les détails sur les mots-clés sont donnés ci-après.

### *4.2.1 Composante système de recommandation*

Les systèmes de base utilisés sont $POP$, $BIP$ [29] et $ALS$ [19].

**POP.** Le premier système de recommandation propose les N produits les plus populaires que l'utilisateur cible n'a pas encore sélectionnés. Ainsi $M_{ui}^{t_k}[u,i]$ est égal au nombre de fois que le produit $i$ a été sélectionné par les utilisateurs depuis l'instant initial $t0$, jusqu'à l'instant qui précède directement la tranche de temps $t_k$.

**BIP.** Correspond aux recommandations obtenues en exécutant le *PageRank personnalisé* sur le graphe bipartite simple construit à partir de toutes les actions de sélection des produits par les utilisateurs. Dans ce cas, $M_{ui}^{t_k}[u,i]$ est égal à la valeur du *PageRank personnalisé* du nœud du produit $i$, lorsque l'injection des préférences est faite sur le nœud de l'utilisateur $u$. Cet algorithme de recommandation nécessite l'utilisation d'un paramètre $\alpha \in [0,1]$ qui permet de calibrer le taux de personnalisation.

**ALS.** Ce système de recommandation repose sur la factorisation matricielle. La particularité de ALS est l'usage de la méthode des moindre carrés alternés (*Alternating Least Squares*) à chaque itération pour la mise à jour des coefficients des matrices $\widetilde{U}^{t_k}$ et $\widetilde{I}^{t_k}$ dans le but de minimiser l'erreur d'approximation des valeurs de la matrice binaire initiale des préférences utilisateurs. On a : $M_{ui}^{t_k}[u,i] = \sum_{f=1}^{n_f} \widetilde{U}_{u,f}^{t_k} \times \widetilde{I}_{f,i}^{t_k}$ où $n_f$ est le nombre de facteurs latents, $\widetilde{U}^{t_k}$ et $\widetilde{I}^{t_k}$ sont respectivement les matrices utilisateur-facteurs latents et item-facteurs latents construites sur la base des données recueillies jusqu'à la tranche de temps $t_k$.

### *4.2.2 Composante prédiction de disponibilité des catégories*

Pour la prédiction de disponibilité des catégories, on a les composantes suivantes :

**PCc.** Les prédictions de disponibilité des catégories calculées sont prises en compte sans aucune forme de normalisation.

**PCn.** Les valeurs des prédictions de disponibilité des catégories d'un produit sont normalisées. La matrice $M_{ic}$ qui met en relation les produits et les catégories est normalisée pour que $\sum M_{ic}[i,.] = 1$ afin de ne pas favoriser les produits qui sont reliés à plusieurs catégories.

**PC1.** Il s'agit d'un cas extrême où nous supposons que toutes les catégories sont toujours présentes dans toutes les tranches de temps. Par conséquent, il n'est pas nécessaire de recourir à un algorithme de calcul des prédictions de disponibilité. Nous avons donc $M_{ct}[c,t_k] = 1, \forall c, \forall t_k$.

### 4.3 Choix du contexte temporel le plus adéquat

Avant de réaliser la combinaison des systèmes de recommandations top-N et la prédiction des disponibilités des catégories des produits, il est important de choisir le contexte temporel adéquat qui permet de fixer la taille des tranches de temps considérées pour la prédiction de disponibilité. Ce contexte temporel est celui qui respecte le mieux les deux conditions suivantes : la condition d'hétérogénéité de la répartition des occurrences de chaque catégorie entre les valeurs du contexte temporel (section 4.3.1), et la dissimilarité des apparitions de chaque catégorie suivant les valeurs du contexte temporel (section 4.3.2).

La figure 2 illustre les critères d'évaluation des deux conditions considérées. Les mesures $H(ct_k, C)$ et $r_{vp_{max}}$ utilisées pour évaluer respectivement la première condition et la seconde condition sont décrites respectivement dans la sous-section 4.3.1 et la sous-section 4.3.2.

(a) $ct_k$

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $c_1$ | 1/4 | 1/4 | 1/4 | 1/4 |
| $c_2$ | 1/4 | 1/4 | 1/4 | 1/4 |
| $c_3$ | 1/4 | 1/4 | 1/4 | 1/4 |
| $c_4$ | 1/4 | 1/4 | 1/4 | 1/4 |

$H(ct_k, C) = 1$

$r_{vp_{max}} = 1$

**Critère 1 : Mauvais**

**Critère 2 : Mauvais**

(b) $ct_k$

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $c_1$ | 0 | 1/3 | 1/3 | 1/3 |
| $c_2$ | 1/3 | 0 | 1/3 | 1/3 |
| $c_3$ | 1/3 | 1/3 | 0 | 1/3 |
| $c_4$ | 1/3 | 1/3 | 1/3 | 0 |

$H(ct_k, C) = 0.89$

$r_{vp_{max}} = 0.33$

**Critère 1 : Mauvais**

**Critère 2 : Bon**

(c) $ct_k$

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $c_1$ | 0 | 1 | 0 | 0 |
| $c_2$ | 0 | 1 | 0 | 0 |
| $c_3$ | 0 | 1 | 0 | 0 |
| $c_4$ | 0 | 1 | 0 | 0 |

$H(ct_k, C) = 0$

$r_{vp_{max}} = 1$

**Critère 1 : Bon**

**Critère 2 : Mauvais**

(d) $ct_k$

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $c_1$ | 1 | 0 | 0 | 0 |
| $c_2$ | 0 | 1 | 0 | 0 |
| $c_3$ | 0 | 0 | 1 | 0 |
| $c_4$ | 0 | 0 | 0 | 1 |

$H(ct_k, C) = 0$

$r_{vp_{max}} = 0.33$

**Critère 1 : Bon**

**Critère 2 : Bon**

FIGURE 2 – Illustration des cas extrêmes de la répartition des apparitions des catégories $c_j$ des produits suivant les valeurs $v_z$ d'un contexte temporel $ct_k$. Le cas (a) représente le pire des cas et (d) le meilleur des cas pour la prédiction des disponibilités des catégories des produits.

#### *4.3.1 Hétérogénéité de la répartition des occurrences de chaque catégories entre les valeurs d'un contexte temporel*

La première condition à vérifier est la suivante : les occurrences de chaque catégorie ne doivent pas être uniformément réparties entre les valeurs du contexte temporel idéal. Autrement dit, un contexte temporel n'est pas utile pour une catégorie si les occurrences de cette catégorie sont uniformément réparties entre les valeurs de ce contexte temporel. Et donc, le contexte temporel qui est le plus utile pour toutes les catégories est le meilleur choix suivant ce critère.

Considérons une liste de catégories de produits $C = \{c_j\}_{j=1..n}$ et une liste de contextes temporels $CT = \{ct_k\}_{k=1..|CT|}$ où chaque contexte temporel $ct_k$ a $m_k$ valeurs possibles $\{v_z\}_{z=1..m_k}$. Pour évaluer la première condition de choix, on peut se servir des mesures d'homogénéité. Ces mesures permettent de déterminer l'écart de la distribution des valeurs d'une variable cible par rapport à la distribution uniforme. Ainsi, pour estimer à quel point un contexte temporel $ct_k$ est important pour prédire une catégorie $c_j$, on peut se rassurer que la mesure d'homogénéité de la répartition des occurrences de la catégorie $c_j$ entre les valeurs du contexte $ct_k$ est minimale.

La mesure d'homogénéité considérée dans cet article est l'entropie de Shannon $h()$ [33], et sa valeur pour une catégorie $c_j$ et un contexte temporel $ct_k$ est donnée par l'équation suivante.

$$h(c_j, ct_k) = -\sum_{z=1}^{m_k} f(c_j, v_z) \cdot log_b(f(c_j, v_z))$$

$f(c_j, v_z)$ retourne la probabilité que la catégorie $c_j$ soit présente durant la valeur $v_z$ du contexte temporel $ct_k$. Et on a $0 \leq h(c_j, ct_k) \leq log_b(m_k)$, et $b = 2$ car $c_j$ est présent ou pas durant $v_z$.

Pour un jeu de données, on calcul l'entropie moyen de chaque contexte temporel $ct_k$ pour toutes les catégories des produits $c \in C$ du jeu de données en utilisant l'équation ci-dessous.

$$H(ct_k, C) = \frac{\sum_{j=1}^{|C|} h(c_j, ct_k)}{|C| \cdot log_2(m_k)}$$

Suivant le premier critère, $H(ct_k, C)$ est minimal pour le meilleur contexte temporel.

#### *4.3.2 Dissimilarité entre les catégories des produits suivant leurs apparitions durant les valeurs des contextes temporels*

La seconde condition a vérifier est la suivante : toutes les catégories doivent être dissimilaires lorsqu'on observe les proportions des apparitions de chacune d'elles entre les valeurs du contexte temporel évalué. En effet, un contexte temporel $c_k$ n'est pas du tout utile pour la prédiction des disponibilités des catégories des produits si ces dernières apparaissent exactement de la même façon et au même proportions durant les valeurs de $c_k$.

Pour évaluer ce second critère pour un contexte temporel $c_k$, on applique une analyse factorielle des correspondances entre les variables qualitatives "catégorie de produits" et "contexte temporel $c_k$" [16]. Ce qui nous permet d'avoir une liste de valeurs propres non nulles $\{vp_j\}$. Si on a une seule valeur propre non nulle, alors $c_k$ n'est pas adéquat car toutes les catégories sont similaires suivant leur apparitions durant les valeurs $v_z$ de $c_k$. Par contre, si on a $min(|C|-1, m_k-1)$ valeurs propres et que toutes les valeurs propres sont égales alors nous sommes dans un cas idéal où toutes les catégories apparaissent différemment suivant les valeurs $v_z$ de $c_k$.

En appliquant ce critère, le contexte temporel le plus adéquat est celui pour lequel la plus grande valeur propre $vp_{max}$ a la plus petite proportion $vp_{max}/\sum vp_j$ comparés aux autres $c_k$ candidats.

#### *4.3.3 Algorithme pour le choix du contexte temporel adéquat dans une liste de candidats*

Pour choisir le contexte temporel le plus adéquat $ct^*$ dans une liste de contextes temporels candidats $CT = \{c_k\}_{k=1..|CT|}$, pour la prédiction des disponibilités de chacune des catégories de produits $c_j \in C$, nous proposons l'algorithme IV.1 qui repose sur les deux critères des sous-sections précédentes. Le paramètre $s$ est le seuil max acceptable pour le ratio $vp_{max}/\sum vp_j$.

**Algorithme IV.1** Choix du contexte temporel le plus adéquat ($CT$, $C$, $s$)

```
candidats ← {}
Pour chaque contexte temporel ct_k ∈ CT faire
        candidats[ct_k] ← H(ct_k, C)
Fin pour
candidats_trier ← Tri_des_cles_suivant_ordre_croissant_des_valeurs(candidats)
Pour chaque ct_k pris en ordre dans candidats_trier faire
        {vp_j} ← Analyse_factorielle_des_correspondances(ct_k, C)
        vp_max ← max({vp_j})
        Si (vp_max / ∑{vp_j}) ≤ s alors
                Retourner ct_k    // on a le meilleur contexte temporel de la liste CT
        Fin si
Fin pour
Retourner −1    // aucun contexte temporel n'est acceptable
```

Pour appliquer l'algorithme du choix du contexte temporel le plus adéquat, la difficulté est celle de la valeur à attribuer au seuil $s$. Aborder ce challenge est une perspective de notre travail.

## V EXPÉRIMENTATIONS ET RÉSULTATS

Cette section présente d'abord les jeux de données utilisés, puis les opérations effectuées pour la prédiction des disponibilités des catégories des produits. La section se poursuit par la description du protocole d'évaluation des recommandations top-N avec dynamique temporelle, et est clôturée par la présentation des résultats et commentaires.

### 5.1 Jeux de données utilisés

Nous utilisons quatre jeux de données accessibles publiquement sur internet : movielens-2k [1], movielens-ls [2], ciao [3] et retailrocket [4]. Les deux premiers proviennent de MovieLens une plateforme de streaming de films. Le suivant provient de Ciao, une plateforme où les utilisateurs donnent des avis sur des produits de domaines varié comme la santé ou l'électronique. Le dernier provient de la plateforme de e-commerce Retailrocket pour l'achat des produits divers.

La table 1 présente les détails sur les flots de liens utilisés dans les expérimentations. $\|L\|$ est le nombre de liens, $\|U\|$, $\|I\|$ et $\|C\|$ sont les cardinaux des ensembles d'utilisateurs, des films ou produits et des genres ou catégories des films, tandis que $Nb.(u,i)$ et $Nb.(i,c)$ sont respectivement le nombre de liens distincts utilisateur-produits et produit-catégories.

| | Date début | Nb. jours | $\|L\|$ | $\|U\|$ | $\|I\|$ | $\|C\|$ | Nb. (u,i) | Nb. (i, c) |
|---|---|---|---|---|---|---|---|---|
| Movielens-2k | 1997-09-17 | 4 128 | 2 240 215 | 2 113 | 10 109 | 20 | 855 598 | 20 670 |
| Movielens-ls | 1996-03-29 | 8 214 | 274 480 | 610 | 9 724 | 20 | 100 836 | 22 046 |
| Ciao | 2000-06-01 | 3 967 | 35 996 | 2 248 | 16 861 | 6 | 35 834 | 16 864 |
| Retailrocket | 2015-05-03 | 138 | 2 645 857 | 1 236 032 | 185 246 | 23 | 1 924 478 | 195 990 |

TABLE 1 – Descriptions des jeux de données.

### 5.2 Application de la prédiction de disponibilité des catégories des produits

Rendu à cette étape, nous commençons par déterminer le contexte temporel le plus adéquat à utiliser comme taille des tranches de temps pour la prédiction des disponibilités des catégories des produits. Ensuite, nous présentons l'application de la prédiction des disponibilités sur des extraits des jeux de données en fonction du contexte temporel choisi.

#### *5.2.1 Application des critères de choix du contexte temporel adéquat*

Nous considérons les sept contextes temporels suivants : *'Moment de la journée'-MDJ* (4 $\times$ 06 heures), *'Jour de la semaine'-JDS*, *'Période de la semaine'-PDS* (3 $\times$ 56 heures), *'Jour du mois'-JDM*, *'Mois'-M* (12 $\times$ 732 heures), *'Période du mois'-PDM* (4 $\times$ 183 heures) et *'Saison'-S* (4 $\times$ 3 mois). La table 2 présente les valeurs de $H(ct_k, C)$ et du ratio $(vp_{max}/\sum vp_j)$ pour chaque contexte temporel et chaque jeu de donnée. Certaines valeurs de Retailrocket ne sont pas disponibles car la durée des données disponibles rend ces valeurs non-cohérentes.

Suite à la lecture de la table 2.a, le contexte temporel *'Moment de la journée'-MDJ* semble être le meilleur pour tous les jeux de données car ce dernier est associé aux valeurs minimales de $H(ct_k, C)$. Néanmoins, il est nécessaire d'observer le second critère avant de conclure.

1. https ://grouplens.org/datasets/hetrec-2011/
2. https ://grouplens.org/datasets/movielens/latest/
3. https://www.cse.msu.edu/~tangjili/trust.html
4. https://www.kaggle.com/retailrocket/ecommerce-dataset

(a) Valeurs de $H(ct_k, C)$

| | MDJ | JDS | PDS | JDM | M | PDM | S |
|---|---|---|---|---|---|---|---|
| Movielens-2k | **0.928** | 0.965 | 0.978 | 0.957 | 0.958 | 0.977 | 0.970 |
| Movielens-ls | **0.967** | 0.989 | 0.984 | 0.980 | 0.988 | 0.990 | 0.995 |
| Ciao | **0** | 0.997 | 0.973 | 0.998 | 0.992 | 0.996 | 0.995 |
| Retailrocket | **0.922** | 0.984 | 0.987 | - | - | - | - |

(b) Valeurs de $(vp_{max}/\sum vp_j)$ en (%)

| | MDJ | JDS | PDS | JDM | M | PDM | S |
|---|---|---|---|---|---|---|---|
| Movielens-2k | 99.13 | 99.70 | 99.78 | 99.18 | 91.74 | 98.32 | 86.17 |
| Movielens-ls | 74.22 | 50.90 | 76.30 | 64.57 | 59.71 | 68.34 | 72.08 |
| Ciao | 100 | 56.85 | 72.52 | 38.59 | 43.06 | 67.86 | 58.92 |
| Retailrocket | 66.70 | 77.23 | 79.74 | - | - | - | - |

TABLE 2 – Valeurs des deux critères de choix pour chaque contexte temporel et chaque jeu de données.

Lorsqu'on observe les données de Ciao, on constate que l'entropie du contexte temporel *'Moment de la journée'-MDJ* est nulle, ce qui est très avantageux suivant le premier critère de choix du meilleur contexte temporel. Cependant, dans la table 2.b, on constate qu'il y a une seule valeur propre non nulle ce qui fait de *'Moment de la journée'-MDJ* un contexte temporel inutile pour la prédiction des disponibilités des catégories dans le jeu de données Ciao. Après ce constat, on identifie le second contexte temporel suivant le premier critère dans Ciao. On a donc *'Période de la semaine'-PDS* comme meilleur candidat dans Ciao avec $H(ct_k, C) = 0.973$.

On peut conclure que *'Moment de la journée'-MDJ* est le meilleur choix de contexte temporel pour les jeux de données : Movielens-2k si $s \geq 99.13$, Movielens-ls si $s \geq 74.22$, et Retailrocket si $s \geq 66.70$. Par ailleurs, pour Ciao on peut choisir *'Période de la semaine'-PDS* si $s \geq 72.52$.

### *5.2.2 Application de la prédiction des disponibilités en utilisant le meilleur contexte temporel*

Pour appliquer la prédiction de disponibilité, nous considérons $|T|$ tranches de temps de durée égale à celle du meilleur contexte temporel. Ensuite 80% des tranches de temps sont utilisées pour le jeu d'apprentissage et les 20% restant comme jeu de test. La table 3 présente les détails sur les sous-flots de liens $L_s$ extraits des jeux de données considérés. Un filtre est appliqué aux utilisateurs et produits ; $u \in L_s$ (resp. $i \in L_s$) si le nombre d'occurrences de $u$ (resp. $i$) dans le grand flot de liens $L$ est supérieur à $min_u$ (resp. $min_i$).

| | Nb. jours | $\|T\|$ | $min_u$ | $min_i$ | $\|\|L_s\|\|$ | $\|\|U_s\|\|$ | $\|\|I_s\|\|$ | $\|\|C_s\|\|$ | Nb. (u,i) | Nb. (i, c) |
|---|---|---|---|---|---|---|---|---|---|---|
| Movielens-2k | 345 | 1 380 | 350 | 50 | 188 331 | 608 | 3 244 | 19 | 68 157 | 7 969 |
| Movielens-ls | 345 | 1 380 | 1 | 1 | 20 438 | 57 | 3 501 | 20 | 7 368 | 8 587 |
| Ciao | 3 220 | 1 380 | 1 | 1 | 22 871 | 1 675 | 14 073 | 6 | 22 816 | 14 075 |
| Retailrocket | 135 | 540 | 15 | 15 | 198 757 | 3 816 | 21 180 | 21 | 112 202 | 22 234 |

TABLE 3 – Extraits des données utilisées pour la prédiction des disponibilités et les recommandations.

Pour déterminer le modèle de prédiction de disponibilité à utiliser, les valeurs des paramètres $\gamma$ et $til$ varient : $\gamma \in \{1, 2, 3, 4, 5, 10, 30, 50\}$ et $til \in \{12 \times t\}_{t=1..20}$. Ces deux paramètres sont choisis de telle sorte que les probabilités $p(\gamma, x_k = 0)$ d'avoir 0 et $(p(\gamma, x_k = 1))$ d'avoir 1 dans les vecteurs d'apparition des catégories dans chaque tranche de temps $t_k$ en utilisant le filtre $\gamma$ soient proches de 50%, et que la précision du modèle de prédiction des disponibilités soit maximal. La table 4 contient les meilleurs résultats à l'issue des expérimentations.

Dans la table 4, $precision > max(p(\gamma, x_k = 0), p(\gamma, x_k = 1))$ pour Movielens-2k, Ciao et Retailrocket, ce qui veut dire que le modèle retenu est meilleur qu'un modèle basique qui prédit toujours 1 (resp. 0). On fait le constat inverse pour Movielens-ls, ce qui est justifiable par le déséquilibre du jeu d'apprentissage où $p(\gamma, x_k = 0) > p(\gamma, x_k = 1)$ même pour $\gamma = 1$.

| | $\gamma$ | $til$ | $p(\gamma, x_k = 0)$ | $p(\gamma, x_k = 1)$ | $precision$ |
|---|---|---|---|---|---|
| Movielens-2k | 2 | 12 | 0.49 | 0.51 | **0.70** |
| Movielens-ls | 1 | 240 | 0.88 | 0.12 | 0.82 |
| Ciao | 3 | 12 | 0.45 | 0.55 | **0.75** |
| Retailrocket | 1 | 96 | 0.48 | 0.52 | **0.67** |

TABLE 4 – Paramètres et précision des modèles de prédiction de disponibilités des catégories.

### 5.3 Protocole d'évaluation des recommandations Top-N

Une fois que la détermination des meilleurs modèles de prédiction des disponibilités des catégories des produits est achevée, nous procédons à l'exécution de ces derniers pour calculer la matrice $M_{it}$ de l'influence des prédictions de disponibilité des catégories de chaque produit $i$ dans la tranche de temps $t_k$ du jeu de test telle qu'illustrée sur la figure 1.

Autrement dit, si $|T|$ est le nombre total de tranches de temps, alors les $N_{test} = (|T| \times 80\%) - til$ dernières tranches du jeu de données sont utilisées pour calculer $M_{it}$ de la prédiction des disponibilités et pour évaluer les résultats de tous les systèmes de recommandation top-N considérés.

#### *5.3.1 Estimation des meilleures performances des systèmes de recommandation*

Dans le but de comparer tous les systèmes de recommandation considérés dans ce travail, il est important de pouvoir estimer la meilleure performance de chacun de ces systèmes. Malheureusement, comme il n'est pas possible de tester tous les paramétrages, pour chacun des paramètres nous fixons un sous ensemble de valeurs dans lequel la valeur du paramètre considéré va varier.

| **Paramètre** | **Description** | **Ensemble fixé de valeurs** |
|---|---|---|
| $\alpha$ | Facteur d'amortissement du PageRank personnalisé | {0.1, 0.3, 0.5, 0.7, 0.9} |
| $nf$ | Nombre de facteurs latents de ALS | {5, 10, 20, 30, 50, 100, 500} |
| $\beta$ | Calibre l'influence de la prédiction de disponibilité | {0.1, 0.3, 0.5, 0.7, 0.9, 1} |

TABLE 5 – Ensembles fixés de valeurs pour la variation des valeurs des paramètres.

Le tableau 5 contient les valeurs fixées pour les paramètres : $\alpha$ du PageRank personnalisé appliqué au graphe biparti classique BIP, $nf$ nombre de facteurs latents de la factorisation matricielle ALS, et $\beta$ qui permet de calibrer l'influence de la prédiction de disponibilité des catégories des produits dans les systèmes de recommandation.

#### *5.3.2 Métriques d'évaluation des systèmes de recommandation*

Pour évaluer la performance de chaque système de recommandation top-N, après chacune des $N_{test}$ tranches de temps de test, les métriques d'évaluation suivantes sont utilisées :

— **Hit-Ratio (HR) :** proportion d'utilisateurs ayant reçu au moins une bonne recommandation du système (taux d'utilisateurs satisfaits par le service de recommandation),

$$HR@N = \frac{\sum_{u \in U} (hit_N(u) > 0)}{|U|}$$

où $hit_N(u)$ retourne le nombre de bonnes recommandations faites à $u$ sur une liste de $N$ produits recommandés.

— **Mean Average Precision (MAP) :** proportion de produits pertinents en tenant compte de la position des produits parmi les N recommandés,

$$MAP@N = \frac{\sum_{u \in U} AP_N(u)}{|U|}$$

avec $AP_N(u) = \frac{1}{hit_N(u)} \sum_{k=1}^{N} \frac{hit_k(u)}{k} \times h(k)$ où $AP_N(u)$ désigne la précision moyenne des recommandations top-N proposées à $u$ et $h(k) = 1$ si le produit à la position $k$ est une bonne recommandation et 0 sinon.

— **F1-Score (F1) :** compromis entre la précision et le rappel afin que le score-F1 soit plus robuste que la précision et le rappel. Pour un utilisateur $u$, *Precision* = $\frac{hit_N(u)}{N}$, *Recall* = $\frac{hit_N(u)}{I_{new}(u)}$, et $F1 = 2 \cdot \frac{Precision \times Recall}{Precision + Recall} = 2 \cdot \frac{hit_N(u)}{I_{new}(u)+N}$. On déduit que le calcul du score-F1 pour l'ensemble des utilisateurs est donné par :

$$F1@N = \frac{\sum_{u \in U} 2 \times hit_N(u)}{\sum_{u \in U} (I_{new}(u)+N)}$$

avec $I_{new}(u)$ le nombre de nouveaux produits sélectionnés par $u$.

Pour chaque métrique $Me \in \{HR, MAP, F1\text{-}score\}$, après avoir déterminé toutes les valeurs $Me_k$ de $Me$ associées aux tranches de temps du jeu de test, la moyenne pondérée de toutes ces valeurs est calculée. Le poids de chaque valeur $Me_k$ est la valeur $deno_{Me_k}$ du dénominateur de cette métrique dans la tranche de temps concernée. On a l'équation suivante :

$$\overline{Me} = \frac{\Sigma_k Me_k \times deno_{Me_k}}{\Sigma_k deno_{Me_k}}$$

## 5.4 Résultats et commentaires

Le tableau 6 contient les résultats en top-5, top-10 et top-50 obtenus pour les jeux de données Movielens-2k, Movielens-ls, Ciao et Retailrocket. Chaque valeur est donnée en pourcentage. La colonne $am.$ indique le taux d'amélioration ou de détérioration des performances du système de recommandation de base par la composante PCc, PCn ou PC1. A cet effet, la couleur bleu indique une amélioration, et la couleur rouge une détérioration des performances. Par ailleurs, la couleur verte indique la meilleure performance sans ex-aequo dans un bloc.

### *5.4.1 Les meilleures performances*

Les résultats de la table 6 sont répartis dans 108 blocs où un système de recommandation de base est comparé aux systèmes qui résultent des combinaisons, en utilisant une métrique d'évaluation et une valeur spécifique de top-N. La table 7 contient l'extrait des 108 meilleurs systèmes de recommandation des blocs de la table 6 et les améliorations réalisées.

Les systèmes de recommandation qui constituent notre contribution sont ceux avec les composantes PCc et PCn. En effet, PC1 correspond au cas où toutes les catégories sont considérées disponibles tout le temps, ce qui équivaut à un cas classique de système de recommandation hybride. Ainsi, pour évaluer efficacement notre contribution, dans la table 7 les systèmes de recommandation de base et ceux avec PC1 sont prioritaires en cas d'égalité.

Lorsque le meilleur système de recommandation d'un bloc intègre la composante PCc, la cellule correspondante dans la table 7 est verte, si c'est PCn la cellule est bleu et si c'est la composante PC1 la cellule est rouge. Dans le pire des cas, lorsque le système de base a la meilleure performance, la cellule est blanche et contient un tiret.

**Performances globales.** En comparant les performances des quatre classes de systèmes de recommandation (système de base, PC1, PCc et PCn) dans la table 7, on constate que les systèmes de base sont meilleurs 11.1% (12/108), PC1 13.9% (15/108), PCc 39.8% (43/108) et PCn 35.2% (38/108). Les résultats de PC1, PCc et PCn montrent que dans 88.9% des cas la considération les catégories des produits pour améliore les systèmes de recommandation de base. Par ailleurs, dans 75% des cas l'intégration de la prédiction des disponibilités des catégories (PCc, PCn) conduit à un résultat meilleur. Ce qui confirme la pertinence de nos travaux.

| MOVIELENS 2K Moment Journée | Hit Ratio | | | | | | MAP | | | | | | F1-Score | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | H@5 | am. | H@10 | am. | H@50 | am. | M@5 | am. | M@10 | am. | M@50 | am. | F@5 | am. | F@10 | am. | F@50 | am. |
| POP | 12.9 | - | 16.98 | - | 34.21 | - | 8.19 | - | 8.59 | - | 8.19 | - | 3.56 | - | 3.39 | - | 2.42 | - |
| POP-PCc | 14.12 | 9.5 | 17.29 | 1.8 | 33.9 | -0.9 | 9.31 | 13.7 | 9.48 | 10.4 | 9.06 | 10.6 | 4.02 | 12.9 | 3.45 | 1.8 | 2.41 | -0.4 |
| POP-PCn | 13.17 | 2.1 | 17.19 | 1.2 | 34.32 | 0.3 | 8.09 | -1.2 | 8.41 | -2.1 | 8.09 | -1.2 | 3.61 | 1.4 | 3.52 | 3.8 | 2.44 | 0.8 |
| POP-PC1 | 14.01 | 8.6 | 17.03 | 0.3 | 33.53 | -2.0 | 9.44 | 15.3 | 9.66 | 12.5 | 9.16 | 11.8 | 3.95 | 11.0 | 3.41 | 0.6 | 2.4 | -0.8 |
| BIP | 14.23 | - | 19.67 | - | 37.18 | - | 9.14 | - | 9.53 | - | 9.17 | - | 4.03 | - | 4.13 | - | 2.96 | - |
| BIP-PCc | 14.44 | 1.5 | 20.15 | 2.4 | 37.65 | 1.3 | 9.96 | 9.0 | 10.41 | 9.2 | 9.91 | 8.1 | 4.2 | 4.2 | 4.18 | 1.2 | 2.97 | 0.3 |
| BIP-PCn | 14.28 | 0.4 | 19.62 | -0.3 | 37.92 | 2.0 | 9.2 | 0.7 | 9.54 | 0.1 | 9.2 | 0.3 | 4.03 | - | 4.09 | -1.0 | 2.99 | 1.0 |
| BIP-PC1 | 14.65 | 3.0 | 19.83 | 0.8 | 37.18 | - | 9.87 | 8.0 | 10.3 | 8.1 | 9.77 | 6.5 | 4.2 | 4.2 | 4.11 | -0.5 | 2.93 | -1.0 |
| ALS | 8.14 | - | 12.11 | - | 29.61 | - | 4.29 | - | 4.59 | - | 4.82 | - | 2.12 | - | 2.38 | - | 2.01 | - |
| ALS-PCc | 8.3 | 2.0 | 12.85 | 6.1 | 29.35 | -0.9 | 4.33 | 0.9 | 4.85 | 5.7 | 5.0 | 3.7 | 2.18 | 2.8 | 2.48 | 4.2 | 2.06 | 2.5 |
| ALS-PCn | 7.88 | -3.2 | 12.16 | 0.4 | 29.56 | -0.2 | 4.15 | -3.3 | 4.54 | -1.1 | 4.76 | -1.2 | 2.13 | 0.5 | 2.36 | -0.8 | 1.99 | -1.0 |
| ALS-PC1 | 8.3 | 2.0 | 12.69 | 4.8 | 28.87 | -2.5 | 4.58 | 6.8 | 4.85 | 5.7 | 4.98 | 3.3 | 2.12 | - | 2.4 | 0.8 | 1.98 | -1.5 |

| MOVIELENS LS Moment Journée | Hit Ratio | | | | | | MAP | | | | | | F1-Score | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | H@5 | am. | H@10 | am. | H@50 | am. | M@5 | am. | M@10 | am. | M@50 | am. | F@5 | am. | F@10 | am. | F@50 | am. |
| POP | 0.0 | - | 8.33 | - | 33.33 | - | 0.0 | - | 1.19 | - | 1.5 | - | 0.0 | - | 1.32 | - | 1.59 | - |
| POP-PCc | 8.33 | +∞ | 16.67 | 100 | 33.33 | - | 1.67 | +∞ | 2.71 | 127 | 2.41 | 60.7 | 2.2 | +∞ | 2.68 | 103 | 1.59 | - |
| POP-PCn | 8.33 | +∞ | 16.67 | 100 | 41.67 | 25.0 | 8.33 | +∞ | 9.52 | 700 | 10.2 | 580 | 2.25 | +∞ | 2.68 | 103 | 1.59 | - |
| POP-PC1 | 0.0 | - | 8.33 | - | 25.0 | -25 | 0.0 | - | 0.83 | -31 | 0.99 | -34 | 0.0 | - | 1.32 | - | 1.27 | -21 |
| BIP | 8.33 | - | 8.33 | - | 33.33 | - | 4.17 | - | 4.17 | - | 4.73 | - | 2.2 | - | 1.32 | - | 1.27 | - |
| BIP-PCc | 8.33 | - | 16.67 | 100 | 33.33 | - | 2.78 | -34 | 3.01 | -28 | 3.71 | -22 | 2.2 | - | 2.68 | 103 | 1.27 | - |
| BIP-PCn | 16.67 | 100 | 25.0 | 200 | 41.67 | 25.0 | 8.33 | 99.8 | 10.19 | 144 | 10.19 | 115 | 4.49 | 104 | 4.03 | 205 | 1.59 | 25.2 |
| BIP-PC1 | 8.33 | - | 8.33 | - | 33.33 | - | 1.67 | -60 | 1.67 | -60 | 1.6 | -67 | 2.2 | - | 1.32 | - | 1.27 | - |
| ALS | 8.33 | - | 8.33 | - | 33.33 | - | 2.78 | - | 2.78 | - | 1.64 | - | 2.2 | - | 1.32 | - | 1.91 | - |
| ALS-PCc | 8.33 | - | 8.33 | - | 25.0 | -25 | 2.78 | - | 2.78 | - | 2.0 | 22.0 | 2.2 | - | 1.34 | 1.5 | 1.59 | -17 |
| ALS-PCn | 16.67 | 100 | 25.0 | 200 | 33.33 | - | 8.33 | 199 | 8.33 | 199 | 9.12 | 456 | 4.49 | 104 | 4.03 | 205 | 1.91 | - |
| ALS-PC1 | 8.33 | - | 8.33 | - | 16.67 | -50 | 2.08 | -26 | 2.08 | -26 | 1.04 | -37 | 2.2 | - | 1.32 | - | 0.95 | -51 |

| CIAO Periode Semaine | Hit Ratio | | | | | | MAP | | | | | | F1-Score | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | H@5 | am. | H@10 | am. | H@50 | am. | M@5 | am. | M@10 | am. | M@50 | am. | F@5 | am. | F@10 | am. | F@50 | am. |
| POP | 0.5 | - | 0.78 | - | 2.04 | - | 0.16 | - | 0.19 | - | 0.24 | - | 0.16 | - | 0.14 | - | 0.08 | - |
| POP-PCc | 0.55 | 10.0 | 0.78 | - | 2.06 | 1.0 | 0.19 | 18.8 | 0.23 | 21.1 | 0.27 | 12.5 | 0.18 | 12.5 | 0.14 | - | 0.09 | 12.5 |
| POP-PCn | 0.55 | 10.0 | 0.78 | - | 2.08 | 2.0 | 0.19 | 18.8 | 0.23 | 21.1 | 0.27 | 12.5 | 0.18 | 12.5 | 0.14 | - | 0.09 | 12.5 |
| POP-PC1 | 0.48 | -4.0 | 0.76 | -2.6 | 2.08 | 2.0 | 0.15 | -6.3 | 0.19 | - | 0.24 | - | 0.16 | - | 0.14 | - | 0.09 | 12.5 |
| BIP | 0.42 | - | 0.84 | - | 2.85 | - | 0.21 | - | 0.26 | - | 0.34 | - | 0.14 | - | 0.15 | - | 0.12 | - |
| BIP-PCc | 0.57 | 35.7 | 0.97 | 15.5 | 3.04 | 6.7 | 0.26 | 23.8 | 0.3 | 15.4 | 0.38 | 11.8 | 0.18 | 28.6 | 0.18 | 20.0 | 0.13 | 8.3 |
| BIP-PCn | 0.55 | 31.0 | 0.94 | 11.9 | 3.02 | 6.0 | 0.25 | 19.0 | 0.3 | 15.4 | 0.39 | 14.7 | 0.18 | 28.6 | 0.17 | 13.3 | 0.12 | - |
| BIP-PC1 | 0.42 | - | 0.82 | -2.4 | 2.9 | 1.8 | 0.21 | - | 0.26 | - | 0.34 | - | 0.14 | - | 0.15 | - | 0.12 | - |
| ALS | 0.63 | - | 1.01 | - | 3.04 | - | 0.28 | - | 0.32 | - | 0.4 | - | 0.21 | - | 0.19 | - | 0.13 | - |
| ALS-PCc | 0.67 | 6.3 | 1.05 | 4.0 | 3.0 | -1.3 | 0.3 | 7.1 | 0.34 | 6.3 | 0.42 | 5.0 | 0.23 | 9.5 | 0.19 | - | 0.13 | - |
| ALS-PCn | 0.67 | 6.3 | 1.05 | 4.0 | 3.0 | -1.3 | 0.3 | 7.1 | 0.34 | 6.3 | 0.42 | 5.0 | 0.23 | 9.5 | 0.19 | - | 0.13 | - |
| ALS-PC1 | 0.65 | 3.2 | 1.03 | 2.0 | 3.06 | 0.7 | 0.28 | - | 0.33 | 3.1 | 0.41 | 2.5 | 0.22 | 4.8 | 0.19 | - | 0.13 | - |

| RETAILROCKET Moment Journée | Hit Ratio | | | | | | MAP | | | | | | F1-Score | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | H@5 | am. | H@10 | am. | H@50 | am. | M@5 | am. | M@10 | am. | M@50 | am. | F@5 | am. | F@10 | am. | F@50 | am. |
| POP | 1.42 | - | 2.28 | - | 7.41 | - | 0.66 | - | 0.75 | - | 0.94 | - | 0.33 | - | 0.34 | - | 0.42 | - |
| POP-PCc | 2.85 | 100 | 3.42 | 50.0 | 7.69 | 3.8 | 0.97 | 47.0 | 1.1 | 46.7 | 1.07 | 13.8 | 0.93 | 181 | 0.74 | 117 | 0.46 | 9.5 |
| POP-PCn | 2.56 | 80.3 | 3.42 | 50.0 | 8.26 | 11.5 | 1.28 | 93.9 | 1.24 | 65.3 | 1.37 | 45.7 | 0.81 | 145 | 0.78 | 129 | 0.49 | 16.7 |
| POP-PC1 | 1.71 | 20.4 | 2.28 | - | 7.69 | 3.8 | 0.67 | 1.5 | 0.77 | 2.7 | 0.98 | 4.3 | 0.39 | 18.2 | 0.35 | 2.9 | 0.46 | 9.5 |
| BIP | 4.84 | - | 5.98 | - | 18.52 | - | 2.32 | - | 2.36 | - | 2.66 | - | 1.35 | - | 1.19 | - | 1.41 | - |
| BIP-PCc | 5.13 | 6.0 | 6.84 | 14.4 | 18.23 | -1.6 | 2.88 | 24.1 | 2.91 | 23.3 | 2.98 | 12.0 | 1.72 | 27.4 | 1.67 | 40.3 | 1.39 | -1.4 |
| BIP-PCn | 4.84 | - | 6.55 | 9.5 | 18.52 | - | 2.84 | 22.4 | 2.85 | 20.8 | 3.01 | 13.2 | 1.56 | 15.6 | 1.63 | 37.0 | 1.41 | - |
| BIP-PC1 | 5.13 | 6.0 | 6.27 | 4.8 | 18.52 | - | 2.44 | 5.2 | 2.54 | 7.6 | 2.82 | 6.0 | 1.48 | 9.6 | 1.25 | 5.0 | 1.39 | -1.4 |
| ALS | 4.84 | - | 7.41 | - | 18.8 | - | 2.22 | - | 2.59 | - | 2.83 | - | 1.35 | - | 2.05 | - | 1.35 | - |
| ALS-PCc | 4.27 | -12 | 8.26 | 11.5 | 19.94 | 6.1 | 2.12 | -4.5 | 2.6 | 0.4 | 2.86 | 1.1 | 1.21 | -11 | 2.0 | -2.4 | 1.42 | 5.2 |
| ALS-PCn | 4.56 | -5.8 | 9.12 | 23.1 | 20.23 | 7.6 | 1.96 | -12 | 2.62 | 1.2 | 2.82 | -0.4 | 1.33 | -1.5 | 2.08 | 1.5 | 1.5 | 11.1 |
| ALS-PC1 | 4.27 | -12 | 7.69 | 3.8 | 19.37 | 3.0 | 2.3 | 3.6 | 2.78 | 7.3 | 2.99 | 5.7 | 1.19 | -12 | 2.04 | -0.5 | 1.38 | 2.2 |

TABLE 6 – Résultats pour les jeux de données Movielens-2k, Movielens-ls, Ciao et Retailrocket.

**Performances en fonction des métriques.** Du point de vue métrique d'évaluation des recommandation, le premier constat est qu'aucun système de base n'est meilleur en MAP et donc que les systèmes combinés avec PC1, PCc et PCn améliorent toujours suivant la métrique MAP qui tient compte de la position des produits bien recommandés dans la liste des recommandations Top-N. Les systèmes combinés avec PCc et PCn sont meilleurs dans 72.2% (26/36) des cas en Hit-ratio, 77.8% (28/36) en MAP et 75% (27/36) en F1-score. On conclut que notre contribution permet de ranger les bonnes recommandations dans les premières positions du Top-N.

| MOVIELENS 2K Moment Journée | Hit Ratio | | | | | | MAP | | | | | | F1-Score | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | H@5 | am. | H@10 | am. | H@50 | am. | M@5 | am. | M@10 | am. | M@50 | am. | F@5 | am. | F@10 | am. | F@50 | am. |
| POP | PCc | 9.5 | PCc | 1.8 | PCn | 0.3 | PC1 | 15.3 | PC1 | 12.5 | PC1 | 11.8 | PCc | 12.9 | PCn | 3.8 | PCn | 0.8 |
| BIP | PC1 | 3.0 | PCc | 2.4 | PCn | 2.0 | PCc | 9.0 | PCc | 9.2 | PCc | 8.1 | PC1 | 4.2 | PCc | 1.2 | PCn | 1.0 |
| ALS | PC1 | 2.0 | PCc | 6.1 | - | - | PC1 | 6.8 | PC1 | 5.7 | PCc | 3.7 | PCc | 2.8 | PCc | 4.2 | PCc | 2.5 |

| MOVIELENS LS Moment Journée | Hit Ratio | | | | | | MAP | | | | | | F1-Score | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | H@5 | am. | H@10 | am. | H@50 | am. | M@5 | am. | M@10 | am. | M@50 | am. | F@5 | am. | F@10 | am. | F@50 | am. |
| POP | PCc | +∞ | PCc | 100 | PCn | 25.0 | PCn | +∞ | PCn | 700 | PCn | 580 | PCn | +∞ | PCc | 103 | - | - |
| BIP | PCn | 100 | PCn | 200 | PCn | 25.0 | PCn | 99.8 | PCn | 144 | PCn | 115 | PCn | 104 | PCn | 205 | PCn | 25.2 |
| ALS | PCn | 100 | PCn | 200 | - | - | PCn | 199 | PCn | 199 | PCn | 456 | PCn | 104 | PCn | 205 | - | - |

| CIAO Periode Semaine | Hit Ratio | | | | | | MAP | | | | | | F1-Score | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | H@5 | am. | H@10 | am. | H@50 | am. | M@5 | am. | M@10 | am. | M@50 | am. | F@5 | am. | F@10 | am. | F@50 | am. |
| POP | PCc | 10.0 | - | - | PC1 | 2.0 | PCc | 18.8 | PCc | 21.1 | PCc | 12.5 | PCc | 12.5 | - | - | PC1 | 12.5 |
| BIP | PCc | 35.7 | PCc | 15.5 | PCc | 6.7 | PCc | 23.8 | PCc | 15.4 | PCn | 14.7 | PCc | 28.6 | PCc | 20.0 | PCc | 8.3 |
| ALS | PCc | 6.3 | PCc | 4.0 | PC1 | 0.7 | PCc | 7.1 | PCc | 6.3 | PCc | 5.0 | PCc | 9.5 | - | - | - | - |

| RETAILROCKET Moment Journée | Hit Ratio | | | | | | MAP | | | | | | F1-Score | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | H@5 | am. | H@10 | am. | H@50 | am. | M@5 | am. | M@10 | am. | M@50 | am. | F@5 | am. | F@10 | am. | F@50 | am. |
| POP | PCc | 100 | PCc | 50.0 | PCn | 11.5 | PCn | 93.9 | PCn | 65.3 | PCn | 45.7 | PCc | 181 | PCn | 129 | PCn | 16.7 |
| BIP | PC1 | 6.0 | PCc | 14.4 | - | - | PCc | 24.1 | PCc | 23.3 | PCn | 13.2 | PCc | 27.4 | PCc | 40.3 | - | - |
| ALS | - | - | PCn | 23.1 | PCn | 7.6 | PC1 | 3.6 | PC1 | 7.3 | PC1 | 5.7 | - | - | PCn | 1.5 | PCn | 11.1 |

TABLE 7 – Extraits des meilleurs systèmes de recommandation et des améliorations effectuées dans Movielens-2k, Movielens-ls, Ciao et Retailrocket. Vert pour PCc, Bleu pour PCn et Rouge pour PC1.

**Performances en fonction du système de base.** En nous focalisant sur les systèmes de base POP, BIP et ALS, on constate que le couple (PCc, PCn) permet d'améliorer POP dans 77.8% (28/36) des cas, BIP dans 86.1% (31/36) des cas et ALS dans 61.1% (22/36). On conclut que notre contribution est recommandé pour les graphes de recommandation (BIP) et devrait être affiné pour les systèmes de recommandation qui reposent sur la factorisation matricielle (ALS).

**Performances en fonction du jeu de données.** Une relecture de la table 7 avec une attention particulière sur les jeux de données permet de constater que PCc et PCn sont meilleurs dans 66.7% (18/27) des cas dans Movielens-2k, 88.9% (24/27) dans Movielens-ls, 74.1% (20/27) dans Ciao et 70.4% (19/27) dans Retailrocket. Tous ces ratios sont supérieurs à 65%, ce qui rassure que l'intégration de la prédiction des disponibilités des catégories a un fort potentiel d'amélioration des recommandations quel que soit le contexte des interactions utilisateur-produits.

**Performances en top-N.** En observant les résultats en fonction des top-N, on constate que l'intégration de la prédiction des disponibilités concoure à un meilleur résultat 75% (27/36) des cas en top-5, 83.3% (30/36) en top-10 et 66.7% (24/36) en top-50. Ainsi, notre contribution semble recommandable pour le top-5 et le top-10, ce qui est intéressant car en pratique, l'espace dédié aux recommandations est restreint à 5 ou 10 (moins de 10) propositions dans la plupart des plateformes qui proposent des recommandations top-N.

Lorsqu'on se restreint à ces deux cas (top-5 et top-10), on note que les variantes PCc et PCn issues de la prise en compte de la prévision de disponibilité des catégories de produits, donnent les meilleurs résultats dans la majorité des cas soit 79.2% (57/72) répartis ainsi qu'il suit : 61.1% (11/18) dans Movielens-2k, 100% (18/18) dans Movielens-ls, 83.3% (15/18) dans Ciao et enfin 72.2% (13/18) dans Retailrocket. Par ailleurs, la version de base est plus performante que toutes les variantes introduites ici dans uniquement 02.8% (2/72) des cas : ALS appliqué à Retailrocket avec les métriques Hit-Ratio@5 et F1-Score@5.

### *5.4.2 Les meilleures valeurs des paramètres des systèmes des recommandations manipulés*

Nous avons analysé les valeurs du paramètre $\beta$ de PCc, PCn et PC1, du paramètre $\alpha$ de BIP et du paramètres $n_f$ de ALS uniquement pour les systèmes combinés qui ont des meilleures performances que le système de base associé. Les systèmes de recommandation concernés sont au nombre de 57 dans Movielens-2k (dont 21 en BIP et 16 en ALS), 35 dans Movielens-ls (dont 11 en BIP et 9 en ALS), 52 dans Ciao (dont 18 en BIP et 18 en ALS) et enfin 62 dans Retailrocket (dont 20 en BIP et 16 en ALS). Les paragraphes suivants résument nos observations.

**Movielens-2k.** Nous recommandons $\beta \leq 0.3$ car c'est sous cette condition que dans 75.4% (43/57) des cas on a les meilleures performances. Pour ce qui est du paramètre $\alpha$, dans 95.2% (20/21) cas on a $\alpha \leq 0.3$ et pour $n_f$ on a $n_f = 50$ dans 81.3% (13/16) des cas.

**Movielens-ls.** Dans ce jeu de données, $\beta \geq 0.5$ dans 85.7% (30/35) des cas où on a des améliorations, $\alpha \geq 0.7$ dans 81.8% (9/11) des cas et $n_f$ n'a que deux valeurs observées, la valeur 10 dans 55.6% (5/9) des cas et la valeur 500 dans 44.4% (4/9) des cas restants.

**Ciao.** Les 52 systèmes de recommandations évoqués sont répartis entre les métriques d'évaluation comme suit : 19 pour Hit-ratio, 20 pour MAP et 13 pour F1-score. Le paramètre $\beta \leq 0.5$ dans 78.9% (15/19) des cas du Hit-ratio, $\beta \geq 0.5$ dans 90% (18/20) des cas MAP et $\beta \leq 0.5$ dans 69.2% (9/13) des cas F1-score. En ce qui concerne $\alpha$, on constate que $\alpha \geq 0.7$ dans 66.7% (12/18) des cas. Et enfin, $n_f$ n'a que deux valeurs observées, la valeur 30 dans 61.1% (11/18) des cas et la valeur 10 dans 38.9% (7/18) des cas restants.

**Retailrocket.** $\beta \leq 0.5$ dans 69.4% (43/62) des cas possibles. Et pour le cas des systèmes de recommandation BIP, $\alpha \leq 0.3$ dans tous les 9 cas du Hit-ration, et dans les 11 cas restants du Hit-ratio et du F1-score on a $\alpha \geq 0.7$. Le paramètre $n_f = 100$ dans tous les 16 cas ALS.

Les remarques faites sur les valeurs recommandées des paramètres dans les paragraphes précédents nous permettent d'affirmer que la configuration idéal dépend fortement du jeu de données. Ce qui veut dire qu'il serait important de proposer un protocole adéquat pour orienter le choix des meilleurs paramétrages. Ceci constitue une perspective de notre travail.

## VI CONCLUSION

Notre objectif dans cet article était de montrer que l'utilisation de la prédiction de disponibilité des catégories des produits peut améliorer les systèmes de recommandation Top-N. Pour ce faire, nous avons utilisé un modèle de prédiction de disponibilité des ressources basé sur le classifieur naïf de Bayes pour calculer pour chaque catégorie de produits la probabilité que des produits de cette catégorie soit acheté durant la période où les recommandations sont proposées. Nous avons ensuite considéré les systèmes de recommandations POP, BIP et ALS pour avoir les recommandations Top-N de base, dont le classement des produits est mis à jour en utilisant les scores des prédictions de disponibilité des catégories suivant les composantes PCc et PCn.

Les expérimentations sur quatre jeux de données avec les métriques d'évaluation Hit-ratio, MAP et F1-score montrent que dans 89% des cas il est nécessaire de considérer les catégories des produits pour améliorer les systèmes de recommandation de base. Par ailleurs, dans 75% des cas l'intégration de la prédiction des disponibilités des catégories (composantes PCc et PCn) conduit à un résultat meilleur. En outres, les composantes PCc et PCn concourent à une plus grande amélioration suivant la métriques MAP qui tient compte de la position des produits bien recommandés dans la liste des recommandations Top-N. Autrement dit, PCc et PCn permettent de ranger les bonnes recommandations dans les premières positions du Top-N.

Le classifieur de Bayes utilisé pour la prédiction de disponibilité des catégories ne tient pas compte de la périodicité du contexte temporel, mais uniquement de sa durée. Et donc impossible de différencier efficacement les contextes temporels *'Jour de la semaine'* ou *'Jour du mois'*. De plus, les contextes temporels de tailles variables ne sont pas considérés (petites et grandes saisons sèches ou de pluies). Ainsi l'utilisation d'autres techniques de prédictions de disponibilité constitue un futur chantier à explorer tout comme l'élaboration d'un protocole qui permet d'estimer les meilleures paramétrages des systèmes de recommandation considérés.

## RÉFÉRENCES


[1] Gediminas ADOMAVICIUS et Alexander TUZHILIN. « Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions ». In : *IEEE transactions on knowledge and data engineering* 17.6 (2005), p. 734-749.

[2] Charu C AGGARWAL. *Data mining : the textbook*. Springer, 2015.

[3] Artur ANDRZEJAK et Derrick KONDO. « Modeling and optimizing availability of non-dedicated resources ». In : *Desktop Grid Computing, Numerical Analy & Scient Comp. Series* (2012), p. 191-210.

[4] Marko BALABANOVIĆ et Yoav SHOHAM. « Fab : content-based, collaborative recommendation ». In : *Communications of the ACM* 40.3 (1997), p. 66-72.

[5] Linas BALTRUNAS et Xavier AMATRIAIN. « Towards time-dependant recommendation based on implicit feedback ». In : *Workshop on context-aware recommender systems (CARS'09)*. Citeseer. 2009, p. 25-30.

[6] Shumeet BALUJA et al. « Video suggestion and discovery for youtube : taking random walks through the view graph ». In : *Proceedings of the 17th international conference on World Wide Web*. ACM. 2008, p. 895-904.

[7] Daniel BERNARDES et al. « A social formalism and survey for recommender systems ». In : *Acm Sigkdd Explorations Newsletter* 16.2 (2015), p. 20-37.

[8] Daniel BILLSUS et Michael J PAZZANI. « User modeling for adaptive news access ». In : *User modeling and user-adapted interaction* 10.2-3 (2000), p. 147-180.

[9] Toine BOGERS. « Movie recommendation using random walks over the contextual graph ». In : *Proc. of the 2nd Intl. Workshop on Context-Aware Recommender Systems*. 2010.

[10] John S BREESE, David HECKERMAN et Carl KADIE. « Empirical analysis of predictive algorithms for collaborative filtering ». In : *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc. 1998, p. 43-52.

[11] Robin BURKE. « Hybrid recommender systems : Survey and experiments ». In : *User modeling and user-adapted interaction* 12.4 (2002), p. 331-370.

[12] Pedro G CAMPOS et al. « Simple time-biased KNN-based recommendations ». In : *Proceedings of the Workshop on Context-Aware Movie Recommendation*. ACM. 2010, p. 20-23.

[13] Paolo CREMONESI, Yehuda KOREN et Roberto TURRIN. « Performance of recommender algorithms on top-n recommendation tasks ». In : *Proceedings of the fourth ACM conference on Recommender systems*. ACM. 2010, p. 39-46.

[14] Yi DING et Xue LI. « Time weight collaborative filtering ». In : *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM. 2005, p. 485-492.

[15] Francois FOUSS et al. « An experimental investigation of graph kernels on a collaborative recommendation task ». In : *Sixth International Conference on Data Mining (ICDM'06)*. IEEE. 2006, p. 863-868.

[16] Michael GREENACRE. *Correspondence analysis in practice*. chapman et hall/crc, 2017.
[17] Taher H HAVELIWALA. « Topic-sensitive pagerank ». In : *Proceedings of the 11th international conference on World Wide Web*. ACM. 2002, p. 517-526.
[18] Xiangnan HE et al. « Nais : Neural attentive item similarity model for recommendation ». In : *IEEE Transactions on Knowledge and Data Engineering* 30.12 (2018), p. 2354-2366.
[19] Yifan HU, Yehuda KOREN et Chris VOLINSKY. « Collaborative filtering for implicit feedback datasets ». In : *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE. 2008, p. 263-272.
[20] Zan HUANG, Daniel ZENG et Hsinchun CHEN. « A comparison of collaborative-filtering recommendation algorithms for e-commerce ». In : *IEEE Intelligent Systems* 22.5 (2007), p. 68-78.
[21] George KARYPIS. « Evaluation of item-based top-n recommendation algorithms ». In : *Proceedings of the tenth international conference on Information and knowledge management*. ACM. 2001, p. 247-254.
[22] Peter KNEES et Markus SCHEDL. « A survey of music similarity and recommendation from music context data ». In : *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 10.1 (2013), p. 2.
[23] Yehuda KOREN. « Collaborative filtering with temporal dynamics ». In : *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2009, p. 447-456.
[24] Yehuda KOREN. « The bellkor solution to the netflix grand prize ». In : *Netflix prize documentation* 81.2009 (2009), p. 1-10.
[25] Yehuda KOREN, Robert BELL et Chris VOLINSKY. « Matrix factorization techniques for recommender systems ». In : *Computer* 8 (2009), p. 30-37.
[26] Sangkeun LEE et al. « Random walk based entity ranking on graph for multidimensional recommendation ». In : *Proceedings of the fifth ACM conference on Recommender systems*. ACM. 2011, p. 93-100.
[27] Greg LINDEN, Brent SMITH et Jeremy YORK. « Amazon. com recommendations : Item-to-item collaborative filtering ». In : *IEEE Internet computing* 1 (2003), p. 76-80.
[28] Raymond J MOONEY et Loriene ROY. « Content-based book recommending using learning for text categorization ». In : *Proceedings of the fifth ACM conference on Digital libraries*. ACM. 2000, p. 195-204.
[29] Armel Jacques NZEKON NZEKO'O, Maurice TCHUENTE et Matthieu LATAPY. « A general graph-based framework for top-N recommendation using content, temporal and trust information ». In : *Journal of Interdisciplinary Methodologies and Issues in Sciences* (2019).
[30] Umberto PANNIELLO et al. « Experimental comparison of pre-vs. post-filtering approaches in context-aware recommender systems ». In : *Proceedings of the third ACM conference on Recommender systems*. ACM. 2009, p. 265-268.
[31] Michael J PAZZANI et Daniel BILLSUS. « Content-based recommendation systems ». In : *The adaptive web*. 2007, p. 325-341.
[32] Andrew I SCHEIN et al. « Methods and metrics for cold-start recommendations ». In : *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM. 2002, p. 253-260.
[33] Claude Elwood SHANNON. « A mathematical theory of communication ». In : *The Bell system technical journal* 27.3 (1948), p. 379-423.
[34] Ian SOBOROFF et Charles NICHOLAS. « Combining content and collaboration in text filtering ». In : *Proceedings of the IJCAI*. T. 99. sn. 1999, p. 86-91.

[35] Yang SONG, Ali Mamdouh ELKAHKY et Xiaodong HE. « Multi-rate deep learning for temporal recommendation ». In : *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM. 2016, p. 909-912.

[36] Pang-Ning TAN, Michael STEINBACH et Vipin KUMAR. *Introduction to data mining*. Pearson Education India, 2016.

[37] Bing TANG et al. « Availability/network-aware mapreduce over the internet ». In : *Information Sciences* 379 (2017), p. 94-111.

[38] João VINAGRE et Alípio Mário JORGE. « Forgetting mechanisms for scalable collaborative filtering ». In : *Journal of the Brazilian Computer Society* 18.4 (2012), p. 271.

[39] Chong WANG et David M BLEI. « Collaborative topic modeling for recommending scientific articles ». In : *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2011, p. 448-456.

[40] Jun WANG, Arjen P DE VRIES et Marcel JT REINDERS. « Unifying user-based and item-based collaborative filtering approaches by similarity fusion ». In : *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM. 2006, p. 501-508.

[41] Liang XIANG et al. « Temporal recommendation on graphs via long-and short-term preference fusion ». In : *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2010, p. 723-732.

[42] Rui YAN, Mirella LAPATA et Xiaoming LI. « Tweet recommendation with graph co-ranking ». In : *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics : Long Papers-Volume 1*. Association for Computational Linguistics. 2012, p. 516-525.

## A REMERCIEMENTS

Nous tenons à remercier les reviewers du CRI'2021 pour leurs commentaires utiles qui ont contribué à améliorer la qualité de ce travail. Nous remercions également nos collègues de l'équipe complex-networks du Laboratoire d'Informatique de Paris 6 et ceux de l'équipe UMMISCO-Yaoundé pour leurs conseils et leur assistance logistique.