Filtrage collaboratif sensible au contexte

Une approche basée sur LDA

Josiane Mothe[1], **Ambinintsoa Jocelyn Rakotonirina**[2]

1. *ESPE, Université de Toulouse, Université de Toulouse Jean Jaurès*
IRIT, UMR 5505 CNRS, 118 Route de Narbonne, Toulouse, France
Josiane.Mothe@irit.fr

2. *DMI, Université d'Antananarivo, Madagascar*
Mathématiques Informatique et Statistique Appliquées, MISA
BP 906 Ankatso
Ambinintsoa26@outlook.com

*RÉSUMÉ. Les systèmes de recommandations visent à proposer aux utilisateurs des items en lien avec leur consultation en cours et qui peuvent retenir leur intérêt. L'intérêt des utilisateurs dépend du contexte dans lequel ils se trouvent. Dans ce travail, nous proposons un système hybride CBCF (Context-aware Based Collaborative Filtering) qui combine le système de recommandations sensibles aux contextes et le filtrage collaboratif. Le contexte est ici défini comme l'objectif ou l'intention de l'utilisateur. Nous le modélisons par une approche LDA (*Latent Dirichlet Allocation*) qui génère un modèle de thèmes pour chaque intention. Nous avons évalué notre approche sur la collection Book-Crossing et montrons sa supériorité par rapport à plusieurs méthodes état de l'art.*

ABSTRACT. Recommender systems are designed to provide user with items related to their ongoing browsing and that may be of interest to them. User interest depends on the context. In this work, we propose a hybrid CBCF (Context-aware Based Collaborative Filtering) system combining context-sensitive and collaborative filtering. We define context as the objective or intent of the user. We model it by a LDA (Latent Dirichlet Allocation) approach which generates a topic model for each intention. We evaluated our approach using the Book-Crossing collection and demonstrated the superiority of our model over several state-of-the-art methods.

MOTS-CLÉS : système d'information, recherche d'information, accès à l'information, système de recommandation, latent dirichlet allocation, filtrage collaboratif, système de recommandation hybride.

KEYWORDS: information systems, information retrieval, recommender systems, latent dirichlet allocation, collaborative filtering, hybrid recommender system.

DOI:10.3166/ISI.23.1.89-109

1. Introduction

Depuis le début des années 1990, internet a changé la manière de consommer et de vendre : l'e-commerce est devenu un canal commun de commercialisation. Sur un site de e-commerce, l'enjeu pour les entreprises est d'attirer le plus possible de clients, de les aider à accéder rapidement aux items (produits, services, films, restaurants, etc.) pertinents et de transformer une visite sur le site en un achat.

Les systèmes de recommandations (SR) sont une solution pour recommander automatiquement des items aux utilisateurs qui peuvent être perdus dans un vaste choix. Lorsqu'un utilisateur consulte un document ou un item d'un site marchand, le SR propose à cet utilisateur d'autres documents ou items l'incitant à poursuivre sa navigation ou ses achats. Selon Nick Tsinonis, les systèmes de recommandation améliorent l'expérience client et augmentent le chiffre d'affaire des sites de e-commerce (30 % du chiffre d'affaire en 2011 chez Amazon.com au sein de RecSys.com).

Les systèmes de recommandation se basent soit sur le contenu des items et proposent à l'utilisateur des items proches de celui qu'il consulte, soit sur les consultations passées et proposent à l'utilisateur des items qui ont été consultés dans des sessions contenant aussi l'item qu'il consulte. Les systèmes les plus efficaces combinent ces deux approches.

Bien que de nombreux travaux se soient intéressés aux SR, certains défis restent à lever encore aujourd'hui. Le *démarrage à froid* désigne un manque d'information sur les nouveaux utilisateurs ou les nouveaux items (Schein *et al.*, 2002). Il est alors impossible de se baser sur les consultations passées pour réaliser les recommandations. La *rareté* ou la *parcimonie* des informations sur les consultations passées est une autre difficulté (Adomavicius et Tuzhilin, 2005), tout comme le manque, voire l'*absence de diversité* dans les recommandations des items (Chevalier *et al.*, 2016 ; Candillier *et al.*, 2011). Idéalement, un SR devrait être adapté à chaque utilisateur ou client. Le système devrait s'adapter également aux situations car souvent les données sur les entités (utilisateurs, produits, etc.) sont dynamiques et évoluent (Louëdec *et al.*, 2015).

Dans la littérature les SR sensibles aux contextes sont utilisés pour traiter ce caractère variable de la pertinence des items pour un utilisateur. Selon (Dey, 2001), un contexte désigne n'importe quelle information qui peut caractériser la situation d'une entité (personne, produit, localisation, etc.). Palmisano *et al.* (2008) ont analysé l'influence des informations contextuelles dans la prédiction des comportements et dans la modélisation des utilisateurs (l'étude définit les contextes comme le but ou l'intention d'achats des utilisateurs dans un SR). En fait, les auteurs ont étudié le comportement des utilisateurs qui est susceptible de changer dans différents contextes. En effet, pour un site de e-commerce, différents clients peuvent acheter un même produit mais pour différentes intentions ou dans différents contextes. Par exemple, un champagne peut être considéré comme un produit de

luxe adapté pour un cadeau, mais pour d'autres consommateurs, il s'agit d'un produit essentiel pour une fête. Si le champagne est vu par les utilisateurs comme une boisson de luxe, ils trouveront pertinents la recommandation d'autres produits de luxe, mais s'il est vu comme produit de fêtes, d'autres accessoires de fêtes seront pertinents. La notion de contexte est étudiée par ailleurs dans de nombreux domaines comme la prise en compte du contexte métier dans l'accès à l'information (Chaker *et al.*, 2013) ou la prise en compte du contexte dynamique dans les profils utilisateurs (Canut *et al.*, 2015). Les défis à relever sont en particulier :

– Comment caractériser un contexte (ici but ou intention) d'achat d'items dans un système de recommandations ?

– Comment adapter la recommandation aux contextes ?

– Comment reconnaitre un contexte lorsqu'il se produit ?

La partie contextuelle de la consultation d'un utilisateur est propre à chaque utilisateur et n'est pas explicitement donnée au système. En revanche, le système dispose de différentes caractéristiques associées à chaque item. Pour un site d'e-commerce ce pourront être la description, le prix, la catégorie du produit, etc.). La difficulté dans le cas d'un système de recommandations sensible aux contextes est de savoir comment caractériser un contexte d'achat d'items.

Palmisano *et al.* (2008) ont montré que, si à partir d'une source extérieure les informations contextuelles sont cachées, il est possible de les induire à partir des données non contextuelles (description, prix, catégorie, etc.) grâce aux méthodes de classification comme les modèles bayésiens. Cette méthode capture les dépendances internes entre les caractéristiques du modèle et les contextes implicites. Ces informations contextuelles conduisent à de meilleures prédictions du profil utilisateur et à des inférences contextuelles. Tavakol et Brefeld ont adopté une approche séquentielle basée sur la session pour détecter l'intention des utilisateurs (Tavakol et Brefeld, 2014). Ils ont défini la session de l'utilisateur comme une séquence d'items cliqués et ont utilisé fMDPs (*factored Markov Decision Processes*) pour la détection de thèmes à partir des attributs des items. Puis les recommandations sont traduites à partir des thèmes.

L'objectif de la caractérisation ou modélisation du contexte est donc de pouvoir la transformer en recommandations d'items.

Les études dans la plupart des SR utilisent les notes des utilisateurs pour trouver la similarité entre les items et ne considèrent pas le contexte dans lequel se trouve l'utilisateur au moment de noter. Dans cet article, nous proposons un SR sensible au contexte utilisant les descriptions des items pour trouver la similarité entre ces items. Pour atteindre cet objectif, nous combinons deux approches :

– la modélisation des thèmes, qui permet de rechercher l'intention des utilisateurs à partir des descriptions textuelles d'un ou plusieurs items successifs qu'ils ont consultés ;

– un système de filtrage collaboratif basé sur le thème de l'utilisateur (ou profil utilisateur) extrait précédemment.

La principale contribution de cet article est la création d'un système de recommandations sensible aux contextes utilisant la sémantique des mots pour trouver la similarité entre les utilisateurs. Le contexte est défini comme le but ou l'intention d'achat de l'utilisateur pour un site de e-commerce. Cet article est une version étendue de (Mothe et Rakotonirina, 2017).

La suite de cet article est structurée comme suit. La section 2 présente l'état de l'art. La section 3 introduit la motivation de la méthode que nous proposons, les jeux de données choisis, l'implémentation et l'évaluation de l'approche. La section 4 montre les résultats empiriques et les analyses. La section 5 conclut cet article.

2. État de l'art

Les travaux reliés au notre relève à la fois des systèmes de recommandation en général et de la prise en compte des contextes. Pour modéliser les contextes, nous proposons d'utiliser une modélisation selon les thèmes (au sens Latent Dirichlet Allocation – LDA). Nous présentons donc également LDA dans cette section.

2.1. Systèmes de recommandation

2.1.1. Principes généraux

Les systèmes de recommandations peuvent être définis comme des programmes qui tentent de recommander les éléments ou items (vidéos, images, documents textuels, produits ou services commerciaux, etc.) les plus appropriés à des utilisateurs particuliers (individus ou entreprises) en prédisant l'intérêt de l'utilisateur pour un item en se basant sur des informations connexes sur les items, les utilisateurs et les interactions entre les items et les utilisateurs (Bobadilla *et al.*, 2013).

2.1.2. Techniques utilisées

Trois types de SR ont été définis dans la littérature : l'approche basée sur le contenu, le filtrage collaboratif et l'approche hybride (Adomavicius *et al.*, 2005).

Les méthodes de recommandations *basées sur le contenu* recommandent à un utilisateur des items similaires à partir des caractéristiques ou propriétés des items. Un produit peut par exemple avoir des propriétés comme la marque, le prix, la couleur, etc. Cette approche génère les recommandations à partir de l'historique de préférence (items similaires visités) d'un utilisateur associé aux propriétés des items courants (Pazzani et Billsus, 2007). La méthode basée sur le contenu atteint ses limites en se confrontant à des problèmes complexes liés aux aspects sémantiques comme l'ambiguïté des termes ou celle de la polysémie. Par exemple, si un utilisateur aime un livre intitulé « Histoire de la Terre depuis les dinosaures », la technique cherchera seulement les items dont les attributs contiennent « Histoire », « Terre » et « Dinosaure ». D'autres livres comme « Les mammifères

de la préhistoire » ne seront pas recommandés mêmes s'ils sont pertinents pour l'utilisateur (Picot-Clémente, 2011).

Contrairement à l'approche basée sur le contenu à un seul utilisateur, les recommandations par *filtrage collaboratif* utilisent les préférences des autres utilisateurs similaires. Cette méthode essaye de former un groupe d'utilisateurs qui a les mêmes préférences. Ainsi, seuls les items les plus appréciés par le groupe sont pertinents (Adomavicius *et al.*, 2005). Dans la littérature plusieurs algorithmes de filtrage collaboratif ont été développés : récemment (Nilashi *et al.*, 2014), Netflix Prize (Bell *et al.*, 2008) et Grouplens (Konstan *et al*., 1997). Les problèmes comme le démarrage à froid et la rareté handicapent souvent la méthode de filtrage collaboratif. Les informations sur les nouveaux items et utilisateurs sont mal gérées par le système. En effet, le manque d'information sur les items et les utilisateurs rend la tâche difficile au système pour trouver des similarités entre eux (Yu *et al.*, 2004 ; Adomavicius *et al*., 2005).

Pour de meilleures performances et afin de combiner les meilleures caractéristiques de plusieurs techniques de recommandations, une *approche hybride* a été proposée. Selon (Burke, 2007), il y a sept mécanismes de base d'hybridation qui peuvent être utilisés dans les systèmes de recommandations : 1) pondérés (Mobasher *et al*., 2004), 2) mixtes (Smyth et Cotter, 2000), 3) de commutation (Billsus et Pazzani, 2000), 4) la combinaison de fonctionnalités (Wilson *et al.*, 2003), 5) l'augmentation de fonctionnalité (Sullivan *et al.*, 2004), 6) cascade (Burke, 2002) et 7) méta-niveau (Pazzani, 1999). L'approche hybride peut éviter certains problèmes comme le démarrage à froid et la rareté d'information.

Selon (Adomavicius et Tuzhilin, 2011), malgré un nombre considérable de recherches faites sur les SR, la plupart des approches se focalisent sur la recommandation des items les plus pertinents pour les utilisateurs sans prendre en compte les informations contextuelles (exemple : le temps, la localisation ou la compagnie d'autres personnes). (Adomavicius et Tuzhilin, 2011) ont montré que les informations contextuelles pertinentes ont des influences importantes sur un SR. Il est donc important d'étudier les SR sensibles aux contextes.

2.2. Systèmes de recommandation et contexte

La notion de contexte est intéressante pour les SR. Selon (Dey, 2001), un contexte est n'importe quelle information qui peut caractériser la situation d'une entité (personne, localisation, produit, etc.). (Ryan *et al.*, 1999) définissent le contexte comme l'identité de l'utilisateur, les ressources de l'environnement proche, la localisation de l'utilisateur et la période temporelle d'exécution de l'interaction. Selon (Berry et Linoof, 1997), les contextes sont définis comme des évènements qui caractérisent les phases de la vie d'un client et qui peuvent influencer ses préférences, son statut et sa valeur pour une entreprise. Des études comportementales en marketing ont montré que la prise de décision des clients

dépend des contextes dans lesquels ils se trouvent (Adomavicius *et al.*, 2005). En effet, selon les contextes comme la localisation, les saisons, l'humeur, etc. le même client peut choisir différents produits.

Plusieurs recherches ont été menées dans différents domaines pour évaluer l'impact des contextes dans les SR. Ces recherches ont été faites sur les contextes observables (localisation, compagnie, période, etc.) (Borras *et al.*, 2014 ; Lamsfus *et al.*, 2009) et les contextes non observables (identité d'un membre d'une famille) (Palmisano *et al.*, 2008).

Adomavicius *et al.* (2005) ont présenté un SR avec une méthode multidimensionnelle. Des contextes sont ajoutés à la fonction d'évaluation de dimension deux (R : User x Item → Rating). Ainsi, on obtient une fonction multidimensionnelle (R : User x Item x Contexte → Rating) qui inclut les informations contextuelles dans la prédiction des préférences des utilisateurs. Pour implémenter la méthode multidimensionnelle et tester sa performance, des données sur des films (notes) et des données contextuelles (localisation, période, compagnie) ont été collectées. Ces données contextuelles ne sont pas disponibles sur la collection de référence Movielens (movielens.umn.edu) généralement utilisée pour évaluer les SR, ni sur les autres données publiques. Par conséquent, un site internet spécifique a été créé et il a été demandé à des utilisateurs d'évaluer les films qu'ils ont vus ainsi que les informations contextuelles pertinentes. Les résultats montrent empiriquement une amélioration de la prédiction des films des systèmes sensibles aux contextes par rapport aux systèmes qui ne les incluent pas.

Selon (Borras *et al.*, 2014) les activités des voyageurs touristiques peuvent être variables en temps réel, il faut donc adapter les recommandations aux circonstances des voyages (exemple : il pleut ou pas, on est à l'intérieur ou à l'extérieur d'un musée). Ainsi, dans les applications qui utilisent la mobilité (tourisme, visite de musée, restauration, etc.), les SR sensibles aux contextes améliorent l'expérience utilisateur. L'approche développée par (Lamsfus *et al.*, 2009) utilise les contextes (localisation, période, météo courante) et propose des suggestions à tout instant en fonction des préférences d'activités du touriste. Par exemple, si un client s'attarde sur une activité qu'il rencontre sur la route et que le temps pour les autres activités est repoussé, alors les visites suivantes devraient être adaptées au plan initial.

Les informations contextuelles proviennent de plusieurs sources diversifiées. Ainsi, caractériser un contexte de recommandations d'items se différencie par rapport à ses origines. Selon (Adomavicius et Tuzhilin, 2011), il y a trois manières d'obtenir les informations contextuelles :

1) explicitement, en posant directement des questions aux utilisateurs (sondages) qui utilisent un site web ;

2) implicitement, par les informations sur les achats effectués, le nombre de clics, la localisation de l'utilisateur grâce aux smartphones (utilisé en tourisme, restauration) ;

3) par induction, en utilisant des modèles prédictifs (ou des classsifieurs). Par exemple dans un supermarché, il est difficile de connaître explicitement l'identité d'un membre d'une famille, qui réalise des achats ensemble avec une seule carte de paiement (ou un même compte pour un site d'e-commerce). Avec les méthodes d'induction utilisant les classifieurs Naïves Bayes et les réseaux bayésiens, (Palmisano *et al*., 2008) ont montré que, des informations contextuelles cachées (ici identité d'un membre) peuvent être induites à partir des données existantes (ici les items achetés).

Un défi se pose sur la façon d'obtenir les informations contextuelles. Nous avons choisi de nous appuyer sur la représentation LDA des items. Nous détaillons ce type de modélisation dans la section suivante.

2.3. Modélisation de thèmes

Selon (Blei, 2012) les modèles de thèmes sont des techniques d'apprentissage automatique et de statistiques qui analysent les mots des textes dans les documents pour découvrir les thèmes traités, comment ces thèmes sont connectés entre eux et comment ils changent au fil du temps. Les documents sont considérés comme des mélanges de thèmes où un thème est une distribution de probabilité sur les mots (Alghamdi et Alfalqi, 2015).

2.3.1. Modélisation de thèmes probabilistes

Les systèmes de recommandations sensibles aux contextes se limitent souvent aux problèmes des données contextuelles latentes. En effet, il est difficile de modéliser un contexte à partir des données contextuelles qui sont partiellement observables ou non observables. Capturer l'intention de l'utilisateur est un des plus grands défis à relever dans les moteurs de recherches et de recommandations. Une approche alternative pour capturer implicitement l'intention de l'utilisateur est le modèle de thèmes (Tavakol et Brefeld, 2014).

Le modèle de thèmes a été d'abord proposé dans le domaine de la recherche textuelle et ses puissantes propriétés de réduction de dimensions et de génération de thèmes cachés l'a rendu populaire également dans le domaine des systèmes de recommandations (Yuan *et al.,* 2015).

Tavakol et Brefeld (2014) ont étudié un système de recommandations de vente de vêtements en ligne basé sur les données implicites, principalement les clics sur les items. Une approche séquentielle basée sur la session est utilisée pour détecter l'intention des utilisateurs. Ils ont défini la session de l'utilisateur comme une séquence d'items cliqués et ont utilisé fMDPs pour la détection de thèmes à partir des attributs des items. La précision[1] des recommandations proposées est d'environ

1. La précision est le ratio : nombre d'items recommandés pertinents pour l'utilisateur par rapport au nombre d'items recommandés.

90 % sur les données collectées à partir de www.zalando.com, surpassant ainsi les méthodes de référence comme les méthodes de filtrage collaboratif.

Xie *et al*. (2014) ont proposé une nouvelle approche de recommandations probabiliste ne prenant pas en compte les contenus, inspiré du modèle LDA (*Latent Dirichlet Allocation*) (Blei *et al.*, 2003). Dans l'approche, les comportements collectés des utilisateurs sont des évènements probabilistes dans lesquels un utilisateur peut appartenir à plusieurs groupes d'utilisateurs et les utilisateurs dans chaque groupe ont différentes préférences collectées. Le processus de collecte est considéré comme deux processus probabilistes joints interférés par le groupe d'utilisateurs. Ainsi, chaque utilisateur est membre d'un groupe d'utilisateurs latent avec une certaine probabilité, tandis que chaque groupe d'utilisateurs collectera des items variés avec différentes probabilités. Sur trois collections de données, MovieLens (movielens.com), Netflix (netflix.com) et Last.fm (last.fm), les résultats ont montré que la méthode possède des performances compétitives non seulement sur la précision mais aussi sur la diversité des items recommandés.

Comme la description des items correspond à du texte non structuré dans notre collection de données (titre des livres), la méthode basée LDA sera utilisée dans cet article. En effet, pour appliquer fMDP par exemple, chaque produit doit avoir au préalable les mêmes types d'attributs (comme genre, couleur, taille, etc.) or seule la description du produit est disponible pour les données textuelles de l'item. De plus LDA est une des méthodes de modélisation de thèmes la plus récente et relativement simple à implémenter dans un système de recommandations (Blei, 2012 ; Yu *et al.*, 2012).

2.3.2. LDA dans les systèmes de recommandations

LDA a été largement étudié dans l'analyse de document (Griffiths et Steyvers, 2004 ; Fei-Fei et Perona, 2005), la catégorisation et le regroupement de documents (Wei et Croft ; 2006 ; Ramage *et al.*, 2009) et l'ordonnancement de systèmes de recherche d'information (Ionescu *et al.*, 2015). LDA a été introduit dans les SR afin d'analyser le contexte dans les méthodes basées sur le contenu (Yu *et al.*, 2012). Dans les systèmes de recommandations basées sur les tags, LDA est utilisé pour trouver la relation cachée entre les mots-clés des descriptions d'items et les tags d'items crées par l'utilisateur, de telle sorte que les items peuvent être recommandés en fonction des tags (Xie *et al.*, 2014 ; Krestel *et al.*, 2009 et Si et Sun, 2009).

3. Filtrage collaboratif basé sur LDA

Modéliser les contextes à partir d'informations contextuelles non observables comme l'intention d'achat de l'utilisateur est difficile. Nous avons choisi de capturer l'intention implicite de l'utilisateur par les thèmes associés aux items qu'il a consultés. La méthode LDA permet cette modélisation des thèmes.

Dans cet article, nous proposons une utilisation nouvelle du modèle LDA. Le résultat du modèle LDA est intégré dans un système de filtrage collaboratif pour trouver la similarité entre les items consultés par un utilisateur courant.

Nous définissons le profil d'un utilisateur en le représentant par son thème. Notre approche recommande alors des items pour lesquels les distributions de thèmes des descriptions (titres des livres dans notre collection) sont similaires au profil de l'utilisateur courant.

Ainsi, cet article propose un SR sensible aux contextes ; il s'agit d'un modèle de recommandation hybride qui combine la méthode de modélisation de thèmes basée sur LDA et la méthode de filtrage collaboratif. Notre approche est décomposée en trois étapes qui sont décrites dans les sous-sections suivantes.

3.1. Modèle LDA pour la représentation des thèmes des utilisateurs

La première étape implémente le modèle LDA à partir des descriptions/titres des items que les utilisateurs ont consultés. LDA est utilisé pour extraire la structure sémantique cachée dans les descriptions des items que les utilisateurs ont consultés, la distribution des mots sur les thèmes latents et le mélange des distributions des thèmes latents. Cela consiste à estimer la distribution de thèmes latents (noté Θ) pour chaque item et la distribution de mots (noté φ) pour chaque thème. Ces distributions vont permettre d'identifier la sémantique de l'espace de thèmes latents en les rapportant aux mots et aux items.

Dans la littérature, l'algorithme EM (*Expected Maximization*) (Blei *et al.*, 2003) et l'algorithme Gibbs sampling (Griffiths et Steyvers, 2004) sont les méthodes les plus utilisées pour l'estimation des paramètres (distributions) Θ et φ du modèle LDA. Cependant, l'algorithme EM est pénalisé par un grand nombre d'opérations à cause du grand nombre de documents ; il est donc plus lent à converger. L'algorithme Gibbs sampling permet de contourner cette difficulté. C'est pour cette raison que nous l'avons utilisé dans ce travail.

Le *Collapsed Gibbs Sampling* (Griffiths et Steyvers, 2004) est un algorithme d'échantillonnage qui permet l'estimation des paramètres d'un espace discret de grande dimension (Steyvers *et al.*, 2004).

Dans cet article, la méthode de *Gibbs sampling* est utilisée pour estimer les paramètres de LDA qui itèrent plusieurs fois sur chaque mot v pour extraire un nouveau thème k pour le mot basé sur la probabilité $p(z_i=k| v_i, z_{-i})$ comme suit :

$$p(z_i=k|v_i,z_{-i}) \propto (n_{d,k} + \alpha_k)\frac{n_{k,v}+\beta_v}{\sum_{v'} n_{k,v'}+\beta_{v'}} \quad (1)$$

Où

– $n_{k,v}$ calcule le nombre des affectations thème-mot,

– $n_{d,k}$ calcule le nombre des affectations document-thème,

– $z_{\_i}$ désigne toutes les affectations thème-mot et document-thème sauf pour l'affectation courantez_i pour le mot v_i,

– α et β sont les paramètres de Dirichlet utilisés comme des paramètres de lissage pour les calculs.

A partir de l'équation (1), les paramètres θ et φ du modèle LDA sont estimées comme suit (Griffiths et Steyvers, 2004) :

$$\theta_{d,k} = \frac{n_{d,k}+\alpha_k}{\sum_{v'} n_{k,v'}+\beta_{v'}} \quad (2)$$

$$\phi_{k,v} = \frac{n_{k,v}+\beta_v}{\sum_{v'} n_{k,v'}+\beta_{v'}} \quad (3)$$

3.2. Similarité entre items

Cette étape intègre les résultats fournis par LDA pour trouver la similarité entre items afin de prédire les préférences de l'utilisateur courant dans le filtrage collaboratif. L'estimation obtenue Θ est la distribution de thèmes latents pour chaque item, vu comme une matrice de similarité d'items par thème, et permet de calculer la similarité entre items. Chaque item possède sa propre distribution à partir de Θ.Pour mesurer la similarité entre deux items, différentes mesures peuvent être utilisées. Nous avons choisi d'utiliser le coefficient de corrélation de Pearson car d'après (Herlocker *et al.*, 2002), en général, les résultats sont meilleurs. Chaque item est représenté comme un vecteur de thèmes et le coefficient de corrélation entre deux items *i* et *j* ayant chacune une variance (finie), noté *Cor*(*i, j*) et définie par :

$$Cor(i, j) = \frac{Cov(i,j)}{\sigma_i \sigma_j} \quad (4)$$

où *Cov(i, j)* désigne la covariance de deux items *i* et *j*, σ_i et σ_j leurs écarts types. Le coefficient de corrélation est symétrique et prend ses valeurs entre -1 et +1.

3.3. Prédiction des préférences de l'utilisateur

Notre approche recommande des items pour lesquels les distributions de thèmes des descriptions des items sont similaires au profil utilisateur. L'objectif est de prédire les préférences de l'utilisateur courant aux items non consultés. Supposons que nous ayons l'historique des préférences des utilisateurs vus comme une matrice M, qui est la matrice d'évaluation employée dans le filtrage collaboratif. Nous regardons ensuite l'ensemble des items que l'utilisateur courant a déjà consultés et déterminons la similarité avec les autres items que l'utilisateur courant n'a pas encore vus en utilisant la matrice de similarité de l'étape précédente. En effectuant cela, les notes des items pour l'utilisateur courant peuvent être obtenues et serviront

à indiquer le degré de préférence de l'utilisateur courant pour les nouveaux items non consultés.

La prédiction des préférences $P_{u,i}$ pour un item *i*, pour l'utilisateur *u*, est basée sur la moyenne pondérée des préférences et des scores de similarité à partir de tous les autres items qui ont été notés par l'utilisateur u.

La formule est la suivante :

$$P_{u,i} = \sum_{j \in J} W_{u,j} * sim(i,j) \quad (5)$$

Où J est l'ensemble des items les plus similaires à l'item i et que l'utilisateur u a notés ; $W_{u,j}$ est le score donné par u pour l'item $j \in J$; $sim\ (i, j)$ la similarité entre les items i et j. La somme est calculée à partir de tous les items $j \in J$ notés par l'utilisateur u.

Les préférences calculées précédemment sont ordonnées par prédiction de pertinence décroissante et les N premières recommandations non notées par l'utilisateur courant sont recommandées. Notre approche a l'avantage de pouvoir prendre en compte l'oubli en considérant J non pas comme l'ensemble de tous les items déjà consultés mais les k derniers ou l'ensemble des items de la session courante, ou l'ensemble des items consultés dans la semaine courante, l'ensemble des items dans une catégorie donnée, etc. Nous laissons toutefois cette adaptation pour des travaux futurs. Dans la section 4, l'évaluation considère J comme l'ensemble des items déjà consultés par l'utilisateur.

4. Evaluation et résultats

4.1. Cadre d'évaluation : collection, mesures et références

Pour évaluer notre méthode, nous avons utilisé la méthode de validation utilisant 90 % des données pour l'entrainement du modèle et 10 % de données pour le test. Pour un utilisateur dans les données tests, nous tirons aléatoirement un item supposé être en cours de consultation et répétons cela pour 50 items. A partir de cet item, le système entrainé propose les items recommandés. Si un item recommandé est effectivement noté positivement (la note est supérieure ou égale à 5 dans l'intervalle de 1 à 10) par l'utilisateur dans la collection, la recommandation est considérée comme pertinente. Nous nous appuyons pour l'évaluation sur la collection Book-Crossing et des mesures d'évaluation présentées ci-dessous.

4.1.1. Collection

Cette collection[2] a été effectuée par Cai-Nicolas Ziegler (*via* une exploration automatique du web) pendant quatre semaines (en 2004) à partir de la communauté

2. Source : http://www2.informatik.uni-freiburg.de/~cziegler/BX/

Book-Crossing avec l'autorisation de Ron Hornbaker (Humankind Systems). Elle contient 278 858 utilisateurs (rendus anonymes mais avec des informations démographiques) fournissant 1 149 780 évaluations (explicites/implicites) sur environ 271 379 livres.

La collection **Book-Crossing** est constituée de trois parties :

– BX-Users : Cette collection contient des informations sur les utilisateurs. Les identifiants des utilisateurs (« ID-Utilisateur ») ont été anonymisés. Des données démographiques comme (« Localisation », « Age ») sont parfois fournies.

– BX-Books : Les livres sont identifiés par leur ISBN (*International Standard Book Number*) ou numéro international standard des livres. Certaines métadonnées comme (« Titre du livre », « Auteur du livre », « Années de Publication », « Editeur ») sont fournies.

– BX-Book-Ratings : Cette collection contient les informations de notations du livre. Les notes sont soit explicites, exprimées sur une échelle de 1 à 10 (valeurs plus élevées indiquant une appréciation plus élevée), soit implicites, identifiées par 0.

Dans nos expérimentations, nous nous sommes focalisés sur les données pour lesquelles une notation explicite était présente ; 397 247 notations respectent cette contrainte.

4.1.2. Mesures

En recherche d'information, la précision mesure la proportion de documents pertinents dans l'ensemble de documents restitués. Dans le cas d'un SR, cette mesure peut être adaptée en la proportion d'items pertinents recommandés dans l'ensemble des items recommandés[3].

$$\textbf{Précision} = \frac{\boldsymbol{RP(i)}}{\boldsymbol{R(i)}} \tag{6}$$

Où

– *RP(i)* est le nombre d'items recommandés et pertinents pour l'item *i* et,

– *R(i)* est le nombre de documents recommandés pour l'item *i*.

De la même façon, nous pouvons adapter la mesure de précision moyenne pour une requête définie en recherche d'information en la précision moyenne pour un item donné :

$$\textbf{AP}(\text{i}) = \frac{\sum_{r=1}^{R(i)}[P@r(i).rel(r)]}{P(i)} \tag{7}$$

Où :

3. Pinel-Sauvagnat et Mothe (2013) présentent un état de l'art des méthodes et mesures d'évaluation dans ces domaines.

– *R(i)* est le nombre d'items recommandés pour l'item *i*,

– r est le rang,

– *P@r(i)* est la précision lorsque les *r* premiers items sont recommandés pour l'item *i*,

– *rel(r)* vaut 1 si la recommandation au rang *r* est pertinente et 0 sinon.

La moyenne des précisions moyennes (*Mean Average Precision* ou MAP) est alors la moyenne arithmétique des précisions moyennes sur l'ensemble des items considérés.

$$\textbf{MAP} = \frac{\sum_{i=1}^{I} AP(i)}{I} \quad (8)$$

Avec *I* le nombre d'items à partir desquels on recherche les items à recommander.

Ces mesures considèrent deux niveaux de pertinence : un item est soit pertinent, soit non pertinent pour un item de départ donné. Par ailleurs, ces mesures sont orientées vers l'utilisateur qui souhaite d'abord des items pertinents ; le rappel[4] est donc moins important dans les SR.

Blei *et al*. (2003) ont proposé la « perplexité », une des mesures d'évaluations de modèle la plus utilisée. La perplexité décroît lorsque le log-vraisemblance du modèle augmente. D'après (Blei *et al.*, 2003) et (Rosen-Zvi *et al.*, 2004), une perplexité faible indique un modèle au pouvoir de généralisation plus élevé. Ainsi, en variant le nombre de thèmes latents, on observe l'évolution de la mesure de perplexité. Le minimum de la valeur de la mesure atteint sera le nombre de thèmes latents utilisé.

La mesure de perplexité des données tests pour *M* documents est :

$$\textbf{Perplexité}\ (\text{B\_test}) = exp(-\frac{1}{N_B}\sum_{d=1}^{M} log(P(wd)) \quad (9)$$

avec $N_B = \sum_{d=1}^{M} N_d$ N_d est le nombre de mots contenus dans le document *d*, $P(w_d)$ est la probabilité que le modèle génératif (au sens modèle de langue par exemple) assigne à un document d du corpus d'évaluation un terme w.

4.1.3. Méthodes de référence

Nous avons comparé les résultats de notre méthode à 3 modèles de référence.

TFIDF (*Term Frequency-Inverse Document Frequency*) (Salton, 1989) est une méthode de pondération de termes qui peut être incorporée dans un système de

4. En recherche d'information, le rappel est défini comme la proportion de documents pertinents effectivement retrouvés. Dans le contexte des SR, le rappel correspond à la proportion d'items pertinents pour l'utilisateur effectivement retrouvés.

filtrage collaboratif pour trouver la similarité entre items. Il s'agit donc d'une approche hybride.

UBCF (*User-Based Collaborative Filtering*) ou filtrage collaboratif basé Utilisateur est une approche qui, à partir d'un utilisateur courant u, recherche les utilisateurs qui sont similaires à cet utilisateur en fonction de la similarité des notes qu'ils ont fournies sur les items. UBCF recommande les items que ces utilisateurs similaires ont aimés (Ekstrand *et al.*, 2011).

IBCF (*Item Based Collaborative Filtering*) ou filtrage collaboratif basé item est une approche obtenue par la transposition de la matrice de similarité de la méthode UBCF. Alors que UBCF génère des prédictions basées sur les similarités entre les utilisateurs, IBCF génère des prédictions basées sur les similarités entre les items (Sarwar *et al.*, 2001).

4.2. Résultats et discussions

Perplexité

Le nombre de thèmes latents est un paramètre du modèle. La figure 1 montre la courbe de la perplexité en fonction du nombre de thèmes. Le tableau 1 détaille cette variation. A partir du résultat, on observe une courbe qui atteint un minimum en 180. Cette étude nous a conduits à retenir 180 sujets latents pour cette collection. Il aurait été intéressant cependant de vérifier qu'il ne s'agit pas d'un minimum local.

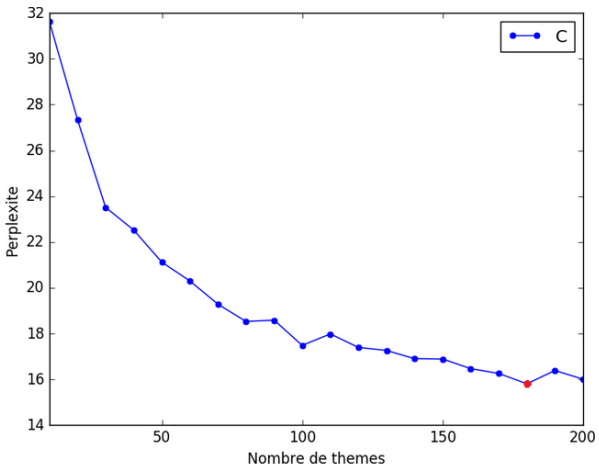

Figure 1. Courbe représentant la perplexité en fonction du nombre de thèmes latents

Dans la suite, le but de l'expérimentation est de comparer notre méthode CBCF avec les méthodes de référence TFIDF, IBCF et UBCF utilisant les jeux de données Book-Crossing.

Les figures 2 et 3 montrent les résultats des comparaisons de performance utilisant la Précision et la MAP. Le Rappel n'est pas utilisé car selon (Herlocker *et al.*, 2004), cette mesure est moins importante. D'après l'auteur, l'utilisateur ne se soucie probablement pas du nombre d'autres éléments pertinents. Ainsi nous rapportons la précision et la MAP en fonction du nombre d'items.

Ces résultats correspondent à une moyenne de précision obtenue en prenant 50 items initiaux de différents utilisateurs choisis aléatoirement à partir desquels le système propose des recommandations.

Nous obtenons des résultats MAP ~0,5 et précision ~0,6 ce qui correspond à de bons résultats par rapport aux autres méthodes de la littérature.

Dans figure 2, nous obtenons la précision en fonction du nombre d'items recommandés et que nous faisons varier de 5 à 20 items. Pour toutes les méthodes nous obtenons la meilleure performance de précision en utilisant 5 items recommandés et inversement pour 20 items recommandés. Pour 5 items recommandés, notre méthode CBCF enregistre un taux de performance de 58 % qui est supérieur à celui de TFIDF de 39 %. IBCF et UBCF enregistrent des précisions inférieures de 9 % et 6 % respectivement.

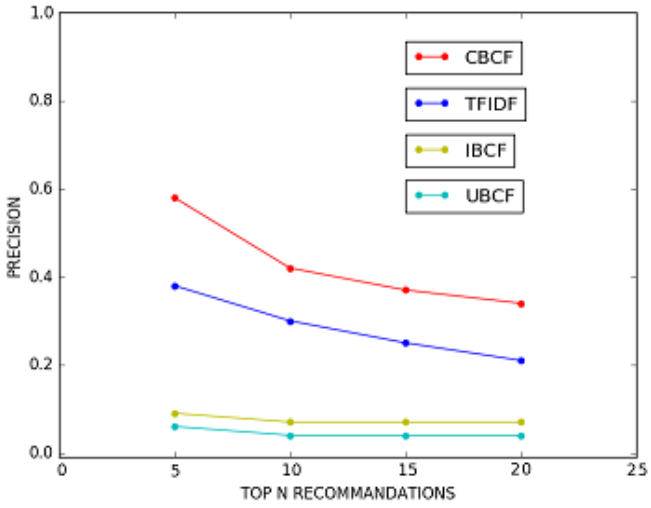

Figure 2. Précision en fonction du nombre d'items recommandés (moyenne sur 50 items de départ utilisant différentes approches - CBCF correspond à la méthode que nous présentons)

Dans figure 3, nous comparons la performance de la MAP des différentes approches. Nous observons le même type de résultats que dans la figure 1. CBCF a le meilleur taux avec 47 % suivi de TFIDF de 26 %. IBCF et UBCF ont des taux de 7 % et 5 % respectivement.

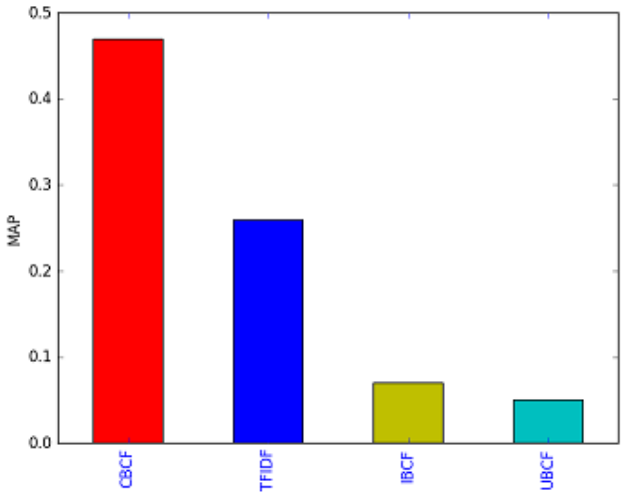

Figure 3. Comparaison de performance de la MAP-Moyennes sur 50 items (axe des Y) de départ en utilisant les différentes approches, CBCF, TFIDF, IBCF et UBCF (axe des X)

Dans figures 2 et 3, nous observons que les deux méthodes IBCF et UBCF utilisant des données explicites (notes) sont moins performantes. Ceci est certainement dû au fait que le filtrage collaboratif pur n'arrive pas à gérer le problème de démarrage à froid dans les deux méthodes.

CBCF et TFIDF hybride montrent de meilleures performances comparées à l'approche filtrage collaboratif grâce à leur propriété hybride et l'utilisation des données implicites (titres des livres) pour trouver la similarité entre items. Néanmoins, la méthode que nous proposons a de meilleure performance que TFIDF.

Les recommandations des items sont faites à partir des thèmes. Dans notre méthode nous caractérisons les thèmes à partir des clics ou requêtes successives.

La figure 4 représente la précision en considérant plusieurs requêtes successivement. En considérant les N premières recommandations, pour 2 requêtes la valeur de la précision augmente par rapport à 1 requête. Pour 3 requêtes le résultat est instable mais reste supérieur à 1 requête. Ainsi, en prenant en compte 2 requêtes, on peut voir que les précisions pour les recommandations sont plus élevées.

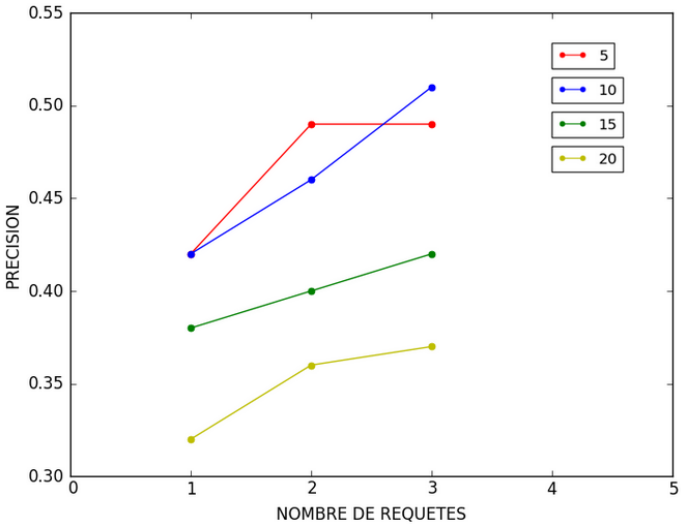

Figure 4. Comparaison de la précision pour les requêtes successives dans CBCF

5. Conclusion

Dans cet article, nous avons proposé un SR hybride combinant LDA et la méthode de filtrage collaboratif pour des données implicites. LDA permet de trouver la structure sémantique latente dans les titres des items (ici des livres) consultés, la distribution des mots sur les thèmes latents et le mélange des distributions des thèmes latents. Le résultat provenant de LDA est ensuite intégré dans un système de filtrage collaboratif basé sur la similarité des utilisateurs.

Par manque d'information complémentaire sur le contexte induisant la consultation de l'utilisateur, nous avons considéré l'item courant comme une description de ce contexte. Notre approche recommande alors des items pour lesquels les distributions de thèmes des titres de l'item courant est similaire au profil utilisateur courant.

Nous avons montré que notre approche donne de meilleurs résultats par rapport au modèle hybride TFIDF. Nous devons maintenant confronter notre modèle à d'autres modèles de la littérature.

Par ailleurs, dans notre implémentation de SR sensible aux contextes, nous avons induit le contexte (intention de l'utilisateur) à partir des titres des livres. Cependant, d'autres caractéristiques (auteurs, éditeurs, notes, etc.) des livres peuvent être utilisées pour induire d'autres contextes non observables et ainsi avoir de meilleure performance. De plus, des travaux additionnels peuvent être effectués en ajoutant des contextes observables comme la localisation de l'utilisateur, la période dans l'année, etc.

Bibliographie

Adomavicius G., Tuzhilin A. (2005). 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*, vol. 17, n° 6, p. 734-749.

Adomavicius G., Tuzhilin A. (2011). Contextaware recommender systems. *Recommender systemshandbook*, p. 217-253. Springer.

Alghamdi R. and Alfalqi, K. (2015). A survey of topic modeling in text mining. *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 6, n° 1.

Bell R. M., Koren Y., and Volinsky, C. (2008). The bellkor 2008 solution to the Netflix prize. *Statistics Research Department at AT&T Research*.

Berry M. J. and Lino, G. (1997). *Data mining techniques : for marketing, sales, and customer support*. John Wiley & Sons, Inc.

Billsus D. and Pazzani M. J. (2000). User modeling for adaptive news access. *User modeling and user-adapted interaction*, vol. 10, n° 2-3, p. 147-180.

Blei D. M., Ng A. Y., Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of machine Learning research*, vol. 3 (January), p. 993-1022.

Blei D. M. (2012). Probabilistic topic models. *Communications of the ACM, 55(4):77-84.*

Bobadilla J., Ortega F., Hernando A., and Gutièrrez A. (2013). Recommender systems survey. *Knowledge-Based Systems*, 46, p. 109-132.

Borras J., Moreno A., Valls A. (2014). Intelligent tourism recommender systems: A survey. *Expert Systems with Applications*, vol. 41, n° 16, p. 7370-7389.

Buntine W. (2009). Estimating likelihoods for topic models. *Asian Conference on Machine Learning*, p. 51-4. Springer.

Burke R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, vol. 12, n° 4 331-370.

Burke R. (2007). Hybrid web recommender systems. *The Adaptive Web*, Springer, p. 377-408.

Candillier L., Chevalier M., Dudognon D., Mothe J. (2012). Multiple Similarities for Diversity in Recommender Systems.International *Journal on Advances in Intelligent Systems*, International Academy, Research and Industry Association, vol. 5, n° 3&4, p. 234-246.

Candillier L., Chevalier M., Dudognon D., Mothe J. (2011). Diversity in Recommender Systems: Bridging the gap between users and systems, *International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services, CENTRIC 2011*, p. 48-58.

Canut M.-F., On-At S., Péninou A., Sèdes F. (2015). Enrichissement du profil utilisateur à partir de son réseau social dans un contexte dynamique : application d'une méthode de pondération temporelle. *INFormatique des Organisations et Systèmes d'Information et de Décision, INFORSID'15*, p. 15-30.

Chaker H., Chevalier M., Tricot A. (2013). Une approche de gestion de contextes métiers pour l'accès à l'information. *INFormatique des Organisations et Systèmes d'Information et de Décision, INFORSID'13*, p. 115-130.

Chevalier M., Dudognon D., Mothe J. (2016). ADORES: a diversity-oriented online recommender system. *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, p. 1075-1076..

Dey A. K. (2001). Understanding and using context. Personal and ubiquitous computing, vol. 5, n° 1, p. 4-7.

Dudognon D. (2014). Diversité et système de recommandation : application à une plateforme de blogs à fort trafic (convention CIFRE n 20091274). Thèse de doctorat, Université de Toulouse, Université Toulouse III-Paul Sabatier.

Ekstrand M. D., Riedl, J. T., Konstan, J. A., *et al.* (2011). Collaborative filtering recommender systems. *Foundations and Trends R in Human – Computer Interaction*, vol. 4, n° 2, p. 81-173.

Fei-Fei L., Perona, P. (2005). A bayesian hierarchical model for learning natural scene categories. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, p. 524-531.

Griffths T. L., Steyvers M. (2004). Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1), p. 5228-5235.

Herlocker J., Konstan J. A., Riedl J. (2002). An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms, *Information Retrieval*, vol. 5, n° 4, p. 287-310.

Herlocker J.L., Konstan J.A., Terveen L. G., Riedl J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, vol. 22, n° 1, p. 5-53.

Ionescu R. T., Chifu A. G., Mothe J. (2015). DeShaTo: describing the shape of cumulative topic distributions to rank retrieval systems without relevance judgments. In *International Symposium on String Processing and Information Retrieval*, p. 75-82. Springer International Publishing.

Konstan J. A., Miller B. N., Maltz D., Herlocker J. L., Gordon L. R., Riedl J. (1997). Grouplens : applying collaborative filtering to UseNet news. *Communications of the ACM*, vol. 40, n° 3, p. 77-87.

Krestel R., Fankhauser P., Nejdl W. (2009). Latent Dirichlet Allocation for tag recommendation. Proceedings of the third ACM conference on Recommender systems, p. 61-68.

Lamsfus C., Alzua-Sorzabal A., Martin D., Salvador Z., Usandizaga A. (2009). Human-centric ontology-based context modelling in tourism. *KEOD*, p. 424-434.

Louëdec J., Chevalier M., Garivier A., Mothe J. (2015). *Systèmes de recommandation et algorithmes de bandits: Notebook Ipython*. Tutoriel. http://www.math.univ-toulouse.fr/~jlouedec/demoBandits.html

Louëdec J., Chevalier M., Mothe J., Garivier A., Gerchinovitz S. (2015). A multiple-play bandit algorithm applied to recommender systems. *FLAIRS Conference*, p. 67-72.

Louëdec J., Chevalier M., Garivier A., Mothe J. (2015). *Algorithmes de bandits pour la recommandation à tirages multiples.Document numérique*, Hermès, vol. 18, n° 2&3, p. 59-79.

Mothe J., Rakotonirina A.J. (2017). *Filtrage collaboratif sensible au contexte - Une approche basée sur LDA. INFormatique des Organisations et Systèmes d'Information et de Décision (INFORSID)*, p. 113-126.

Mothe J., Ramiandriosa F., Rasolomanana M. (2018). *Automatic Keyphrase Extraction using Graph-based Methods. ACM Symposium on Applied Computing (SAC)*.

Mobasher B., Jin X., Zhou Y. (2004). Semantically enhanced collaborative filltering on the web. In *Web Mining: From Web to Semantic Web*, p. 57-76. Springer.

Nguyen C. P. (2010). Conception d'un système d'apprentissage et de travail pervasif et adaptatif fondé sur un modèle de scénario. Thèse de doctorat.

Nilashi M., Bin Ibrahim O., Ithnin N. (2014). Hybrid recommendation approaches for multi-criteria collaborative filtering. *Expert Systems with Applications*, vol. 41, n° 8, p. 3879-3900.

Palmisano C., Tuzhilin A., Gorgoglione M. (2008). Using context to improve predictive modeling of customers in personalization applications. *IEEE transactions on knowledge and data engineering*, vol. 20, n° 11, 1535-1549.

Pazzani M. J. (1999). A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, vol. 13, n° 5-6, p. 393-408.

Pazzani M. J., Billsus D. (2007). Content-based recommendation systems. *The Adaptive Web*, Springer, p. 325-341.

Picot-Clémente R. (2011). Une architecture générique de Systèmes de recommandation de combinaison d'items : application au domaine du tourisme. Thèse de doctorat, Université de Bourgogne.

Pinel-Sauvagnat K., Mothe J. (2013). *Mesures de la qualité des systèmes de recherche d'information.* Dans : *Ingénierie des Systèmes d'Information*, Hermès Science, Numéro spécial *Evaluation des systèmes d'information*, Hors Série, n° 3, p. 11-38.

Ramage D., Hall D., Nallapati R., Manning C. D. (2009). Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, vol. 1, p. 248-256. Association for Computational Linguistics.

Rosen-Zvi M., Griths T., Steyvers M., Smyth P. (2004). The author-topic model for authors and documents. *Proceedings of the 20th Conference on Uncertainty in artificial intelligence*, p. 487-494. AUAI Press.

Rakotonirina A. J. (2017). Filtrage Collaboratif Sensible au Contexte : une approche basée sur LDA, thèse de Master.

Ryan N., Pascoe J., Morse D. (1999). Enhanced reality fieldwork: the context aware archaeological assistant. *Bar International Series*, 750, p. 269-274.

Salton G., McGill, M. J. (1986). Introduction to modern information retrieval, Salton, G. (1989). *Automatic text processing: The transformation, analysis, and retrieval of. Reading*. Addison-Wesley.

Sarwar B., Karypis G., Konstan J., Riedl J. (2001). Itembased collaborative filtering recommendation algorithms. *Proceedings of the 10th International ACM Conference on World Wide Web*, p. 285-295.

Schein A. I., Popescul, A., Ungar, L. H., Pennock D. M. (2002). Methods and metrics for cold-start recommendations. *Proceedings of the 25th Annual International ACM SIGIR conference on Research and development in Information Retrieval*, p. 253-260.

Si X., Sun M. (2009). Tag-LDA for scalable real-time tag recommendation. *Journal of Computational Information Systems*, vol. 6, n° 1, p. 23-31.

Smyth B., Cotter P. (2000). A personalized television listings service. *Communications of the ACM*, 43, n° 8, p. 107-111.

Steyvers M., Smyth P., Rosen-Zvi M., Grifiths T. (2004). Probabilistic author-topic models for information discovery. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 306-315..

Sullivan D. O., Smyth B., Wilson D. (2004). Preserving recommender accuracy and diversity in sparse datasets. *International Journal on Artificial Intelligence Tools*, vol. 13, n° 1, p. 219-235.

Tavakol M., Brefeld U. (2014). Factored mdps for detecting topics of user sessions. *Proceedings of the 8th ACM Conference on Recommender Systems*, p. 33-40.

Wallach H. M., Mimno D. M., McCallum A. (2009). Rethinking LDA: Why priors matter. *Advances in neural information processing systems*, p. 1973-1981.

Wei X., Croft W.B. (2006). LDA-based document models for ad-hoc retrieval. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, p. 178-185.

Wilson D. C., Smyth B., and Sullivan D. O. (2003). Sparsity reduction in collaborative recommendation: A case-based approach. *International journal of pattern recognition and artificial intelligence*, vol. 17, n° 5, p. 863-884.

Xie W., Dong Q., Gao H. (2014). A probabilistic recommendation method inspired by Latent Dirichlet Allocation model. *Mathematical Problems in Engineering*.

Yu K., Schwaighofer A., Tresp V., Xu X., Kriegel H.-P. (2004). Probabilistic memory-based collaborative filtering. *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, n° 1, p. 56-69.

Yu K., Zhang B., Zhu H., Cao H., Tian J. (2012). Towards personalizedcontext-aware recommendation by mining context logs through topic models. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, p. 431-443.

Yuan J., Gao F., Ho Q., Dai W., Wei J., Zheng X., Xing E. P., Liu T.-Y., Ma W.-Y. (2015). LightLDA: Big topic models on modest computer clusters. *Proceedings of the 24th International ACM Conference on World Wide Web*, p. 1351-1361.