Vers l'utilisation de relations de préférence pour le filtrage collaboratif

Armelle Brun[1] Ahmad Hamad[1] Olivier Buffet[2] Anne Boyer[1]

[1] LORIA / Nancy-Université
[2] LORIA / INRIA

Campus Scientifique – BP 239 / 54506 Vandœuvre-lès-Nancy – France
{prénom.nom}@loria.fr

Résumé

Les systèmes de recommandation à base de filtrage collaboratif exploitent les préférences d'utilisateurs sur des ressources pour effectuer des recommandations. Ces préférences se présentent généralement sous la forme de votes quantitatifs. Cependant, fixer un vote pour un utilisateur n'est pas une tâche simple ; elle peut être influencée par de nombreux facteurs et les votes obtenus ne sont donc pas totalement fiables. Dans cet article nous proposons une nouvelle approche pour exprimer les préférences des utilisateurs sous forme de relations de préférence au lieu de votes. Nous utilisons les mêmes étapes que le filtrage collaboratif classique pour effectuer les recommandations et nous proposons dans ce cadre des mesures alternatives adaptées à l'exploitation de relations de préférence. Des premières expérimentations montrent le potentiel de cette nouvelle approche.

Mots Clef

Systèmes de recommandation, filtrage collaboratif, relation de préférence, utilité.

Abstract

Collaborative filtering based recommender systems exploit users preferences about items to provide recommendations to these users. These preferences are generally ratings. However, choosing a rating is not an easy task for any user ; the rating value may be influenced by many factors and the ratings are thus not completely trustworthy. In this article, we propose a new approach of expressing preferences, under the form of preference relations instead of ratings. Similar steps as in classical collaborative filtering are used and we propose new measures to exploit preference relations to compute recommendations. First experimentaions of this new approach have been conducted on a state of the art corpus of the recommender systems domain.

Keywords

Recommender systems, collaborative filtering, preference relation, utility.

1 Introduction

Un *système de recommandation* [1] a pour but de préconiser à un utilisateur des items (également appelés ressources) en lien avec ses goûts et ses attentes. L'objectif est à la fois de minimiser son temps passé à la recherche, mais aussi de lui suggérer des items pertinents qu'il n'aurait pas spontanément consultés et ainsi accroître sa satisfaction globale. Un item peut par exemple être une page web, un livre, un film, de la musique, etc. Les systèmes de recommandation ont vu leur popularité croître ces dernières années en raison de la démocratisation du web et de l'augmentation exponentielle de la quantité de ressources disponibles et accessibles ; de nombreux sites tels que *amazon* [1] intègrent un système de recommandation.

Pour recommander des items à un utilisateur, le système doit disposer d'un profil représentatif de ses préférences. Pour le construire il doit recueillir des informations sur celui-ci, soit directement (via un formulaire) ou indirectement (par analyse de traces).

Pour déterminer les items à recommander, plusieurs approches sont possibles : (i) l'approche par contenu [2] qui effectue des recommandations en comparant le contenu sémantique des ressources avec les goûts exprimés par l'utilisateur ; (ii) l'approche à base de connaissances [3] qui effectue des recommandations en exploitant les connaissances sur l'utilisateur et des heuristiques pré-établies ; et (iii) l'approche par filtrage collaboratif [4] qui effectue des recommandations par analyse à la fois des opinions de l'utilisateur sur les ressources qu'il a consultées ainsi que celles des autres utilisateurs sur les ressources qu'ils ont consultées. L'intérêt porté à cette approche s'est largement accru ces dernières années.

Dans cet article, nous nous intéressons à l'approche par filtrage collaboratif. Dans ce cadre, aucune information n'est disponible *a priori* sur une ressource. Seul son identifiant est connu (aucune indexation *a priori* n'est nécessaire). Un avantage du filtrage collaboratif est donc la possibilité de manipuler des ressources de toute langue, de tout media (texte, audio, vidéo), puisque ne réquerant aucune indexa-

[1] http ://www.amazon.com

tion de la ressource. De la même manière, pour un utilisateur seul l'ensemble des ressources qu'il a consultées ainsi que ses préférences concernant ces ressources sont connus. L'utilisateur n'a donc pas besoin d'être identifié nominativement. De plus, aucune donnée démographique le concernant n'est nécessaire. Un autre atout du filtrage collaboratif est donc le respect de la vie privée de l'utilisateur.

Dans une approche par *filtrage collaboratif*, les *préférences* d'un ensemble d'*utilisateurs* U pour un ensemble d'*items* I sont donc partiellement connues puisque limitées aux informations fournies par les utilisateurs. Pour recommander de nouvelles ressources à un utilisateur, le filtrage collaboratif exploite les préférences des autres utilisateurs et une approche consiste à utiliser les similarités entre utilisateurs pour effectuer des recommandations (voir [5] pour un panorama des différentes approches). Ces similarités sont des similarités de goûts puisqu'elles exploitent les préférences des utilisateurs sur les ressources.

Dans le cadre du filtrage collaboratif, la représentation la plus courante des préférences se fait sous la forme de votes quantitatifs associés aux ressources. Ils sont utilisés à la fois pour mesurer des similarités entre utilisateurs [6, 7] et pour estimer les votes que les utilisateurs donneraient aux ressources qu'ils n'ont pas vues. Ces votes estimés sont ensuite exploités pour effectuer des recommandations.

L'utilisation de votes a toutefois certains inconvénients. En effet, lorsqu'un utilisateur doit voter pour une ressource, il le fait en général sur une échelle de valeurs entières fixée et relativement réduite, donc imprécise (l'échelle de valeurs peut varier en fonction des applications, voir [8]). A vote équivalent attribué par un utilisateur, deux ressources peuvent avoir été appréciées de façon différente. Mais cette différence n'est pas reflétée par le vote. En outre, estimer la pertinence d'une ressource peut être difficile pour un utilisateur. En effet, le contexte, les ressources précédemment notées, etc. ont une influence non négligeable sur le choix du vote. Les votes ainsi obtenus sont donc imprécis et peu fiables, ce qui limite la qualité des mesures de similarité calculées ainsi que la qualité des recommandations.

Lors de l'étape de recommandation, le système estime un vote pour chaque ressource qu'un utilisateur u n'aura pas votée. Ce vote est dépendant à la fois des votes que les autres utilisateurs auront mis sur cette ressource et des mesures de similarité entre utilisateurs. Les ressources ayant obtenu les meilleurs votes seront celles qui seront recommandées à l'utilisateur. On peut alors remarquer que le but premier de cette étape de recommandation n'est pas de fournir des informations quantitatives sur les ressources telles que des votes, mais de fournir la liste (éventuellement) ordonnée des ressources que le système jugera être préférées par l'utilisateur u. Dans ce cadre, nous jugeons qu'une information qualitative telle qu'une liste ordonnée (par préférence) de ressources "préférables" est suffisante pour effectuer des recommandations.

Au vu des inconvénients de l'utilisation de votes pour exprimer les préférences, nous proposons de les remplacer par des *relations de préférence*. Dans ce cas, on ne demande plus à l'utilisateur d'attribuer un vote à une ressource mais d'exprimer qualitativement son intérêt par rapport à des ressources qu'il a déjà vues. Par exemple, il indiquera "je préfère la ressource j à la ressource i". Une relation de préférence peut s'avérer plus adéquate que des votes. En effet, dans une relation de préférence on ne retrouve pas le problème de la finesse de la discrétisation. Ainsi, si donner une estimation absolue est délicat, il semble plus robuste (et fiable) de demander à un utilisateur de comparer des ressources deux à deux. Au vu de cette facilité, nous nous attendons donc à obtenir un plus grand nombre de préférences utilisateur, améliorant ainsi le potentiel du système. De plus, l'approche par relations de préférence nous permettra de considérer comme similaires des utilisateurs qui n'attribuent pas les mêmes votes aux ressources mais qui les ordonnent dans le même ordre. Un défaut de l'utilisation de relations de préférence est qu'il faut comparer une nouvelle ressource à de nombreuses autres pour pouvoir la situer par rapport aux autres alors qu'attribuer un vote à une ressource était suffisant. De plus, une relation de préférence a l'inconvénient de ne pas contenir d'information quantitative. Nous pouvons également noter que bien souvent les préférences utilisateur sont partielles : on préfère une ressource i à une ressource j sur tel critère mais j est préférée à i sur tel autre critère. L'approche que nous choisissons ici est donc moins fine.

Des travaux sur les systèmes de recommandation ont déjà exploité les relations de préférence, mais sur des systèmes à base de contenu. Dans [9] l'objectif était de compléter des préférences inconnues.

Cet article est une première étude sur l'utilisation de relations de préférence pour le filtrage collaboratif. Il présente une étude comparative à la fois théorique et expérimentale de l'approche classique et de l'approche à base de relations de préférence.

La seconde section s'intéresse à deux manières d'exprimer des préférences : relations de préférence d'une part et fonctions d'utilité d'autre part. La troisième section présentera le filtrage collaboratif et les différentes étapes qui le consituent. Ces étapes seront détaillées dans la quatrième section sous deux angles, selon qu'on exploite des fonctions d'utilité ou des relations de préférence ; dans ce second cas nous présenterons nos propositions pour intégrer des relations de préférence dans un système de filtrage collaboratif. La section suivante est une validation expérimentale de l'approche proposée. Enfin, nous concluerons et présenterons des perspectives à ce travail.

2 Expression des préférences

La préférence est un "jugement ou sentiment par lequel on place une personne ou une chose au dessus des autres"[2]. On parle aussi classiquement de goûts. Nous présentons ci-dessous deux manières d'exprimer des préférences, que sont les relations de préférence et les fonctions d'utilité.

[2] Définition du Petit Robert

2.1 Relation de préférence

Une première façon de décrire les préférences d'un utilisateur est d'employer une *relation de préférence*, c'est-à-dire une *relation binaire* $i \preceq j$ sur I, qui est :
– réflexive : $\forall i \in I,\ i \preceq i$;
– antisymétrique : $\forall i, j \in I,\ (i \preceq j) \wedge (j \preceq i) \Rightarrow (i \simeq j)$;
– transitive : $\forall i, j, k \in I,\ (i \preceq j) \wedge (j \preceq k) \Rightarrow (i \preceq k)$;
– totale : $\forall i, j \in I,\ (i \preceq j) \vee (j \preceq i)$.
On notera que, avec cette définition :
– "j est strictement préféré à i" s'écrit $(i \preceq j) \wedge \neg(j \preceq i)$ et se note $i \prec j$.
– "i et j sont équivalents" ou "l'utilisateur est indifférent entre i et j" s'écrit $(i \preceq j) \wedge (j \preceq i)$ et se note $i \simeq j$.
Pour deux élements i, j, il y a trois possibilités : $i \prec j$, $j \prec i$ et $i \simeq j$, dont exactement une doit être vraie.
Il est important de noter que tous les couples de ressources $\{i, j\}$ sont comparables. Il n'y a pas d'indéterminisme. En fait, si tout couple de ressources $\{i, j\}$ est supposé comparable par l'utilisateur, cela ne signifie pas que, à un instant t, l'utilisateur connaisse effectivement ces ressources et sache à laquelle va sa préférence. En d'autres termes, bien que la relation soit totale, nous n'en avons qu'une vue incomplète. En revanche, cela signifie que, si l'utilisateur a vu toutes les ressources, il sait à laquelle va sa préférence. Il en ira de même avec les utilités.
Etant données toutes les propriétés vérifiées par une relation de préférence, on peut la représenter de façon simplifiée (en omettant les relations qui peuvent être déduites par transitivité) sous la forme d'une chaîne strictement ordonnée de classes d'équivalence. La figure 1-a montre une telle représentation pour une relation de préférence sur huit ressources $\{a, b, c, d, e, f, g, h\}$.

$$(a \simeq e)$$
$$\downarrow$$
$$(c \simeq f \simeq g)$$
$$\downarrow$$
$$(d)$$
$$\downarrow$$
$$(b \simeq h)$$

a- une relation de préférence (la flèche signifie "strictement préférés à")

i	u_1	u_2
a	4	10^3
b	1	-1
c	3	π
d	2	$\sqrt{2}$
e	4	10^3
f	3	π
g	3	π
h	1	-1

b- deux fonctions d'utilité compatibles avec la relation de préférence

FIG. 1 – Deux types de représentations de préférences d'un utilisateur sur un ensemble de huit ressources.

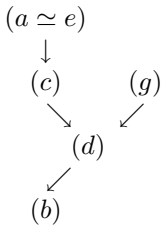
2.2 Utilité

Dans le cas précédent où les préférences/goûts d'un utilisateur u sont vus comme une relation binaire, ils sont une notion purement qualitative. On ne dit pas "à quel point" on préfère j à i.
Les préférences "quantitatives" classiquement exprimées par les utilisateurs (les votes par exemple) peuvent être modélisées sous forme d'une fonction d'*utilité* $ut : I \to \mathbb{R}$. On dira alors que la ressource j est préférée à la ressource i d'après la fonction d'utilité ut (c'est-à-dire $i \preceq_{ut} j$) si et seulement si $ut(i) \leq ut(j)$.
On peut observer ici qu'une fonction d'utilité donnée définit une unique relation de préférence. A l'inverse, une relation de préférence $\preceq$ peut correspondre à plusieurs fonctions d'utilité. Supposons par exemple que ut soit "compatible" avec $\preceq$ et que $f : \mathbb{R} \to \mathbb{R}$ soit strictement croissante, alors $f(ut)$ est aussi compatible avec $\preceq$. Ainsi, deux utilisateurs peuvent avoir des fonctions d'utilité différentes mais être toujours d'accord quant aux ressources qu'ils préfèrent, ce qui constitue un avantage de la représentation par relation de préférence. La figure 1-b montre deux fonctions d'utilité correspondant à la même relation de préférence (celle présentée en figure 1-a).

3 Filtrage collaboratif

En filtrage collaboratif, on exploite les préférences d'un utilisateur u ainsi que les préférences des autres utilisateurs pour estimer les préférences non connues de u et ainsi lui recommander des ressources qu'il n'a pas encore vues.
Pour estimer les préférences non connues d'un utilisateur, on peut soit exploiter une approche dite "modèle" (ou approche paramétrique) [10, 11], soit exploiter une approche dite "mémoire" (ou approche non paramétrique) qui exploitera typiquement des similarités (de préférence) entre utilisateurs [6]. C'est cette dernière approche sur laquelle nous porterons notre attention.
Pour un utilisateur $u \in U$, on notera ses préférences :
– $i \preceq_u j$ pour la relation de préférence qui lui est associée ;
– $ut_u(i)$ pour la fonction d'utilité qui lui est associée.
En filtrage collaboratif, les préférences d'un utilisateur sont en général incomplètement connues. On désignera la partie connue des préférences d'un utilisateur par son *profil* et on notera une information absente $(i \preceq_u j) =?$ dans le cas d'un profil de type "relation de préférence" et $ut_u(i) =?$ dans le cas d'un profil de type "fonction d'utilité".
La figure 2 montre trois profils associés aux préférences apparaissant dans la figure 1, chacun ayant perdu des informations différentes. Notons que, pour la relation de préférence, la ressource g n'est comparable ni à a, ni à e, ni à c, et que l'on n'a plus la moindre information sur les ressources f et h.
Quel que soit le type d'information exploité (relation de préférence ou fonction d'utilité), nous proposons d'utiliser le même processus classique de filtrage collaboratif, lequel se décompose en trois étapes :

1. **acquisition des profils utilisateurs :** chaque utilisateur renseigne ses préférences (goûts) sur un certain nombre de ressources ; ces informations constitueront son profil ;
2. **mesure de similarité entre utilisateurs :** à l'aide des profils des utilisateurs, on mesure la similarité $sim(u, u')$ entre ceux-ci ;



a- une relation de préférence incomplète

i	u_1	u_2
a	–	–
b	1	–
c	3	π
d	–	$\sqrt{2}$
e	–	10^3
f	3	π
g	3	–
h	1	-1

b- deux fonctions d'utilité incomplètes

FIG. 2 – Trois profils d'utilisateur associés aux trois représentations de préférences de la figure 1.

3. **recommandation à un utilisateur :** pour un utilisateur u donné, on utilise les profils des utilisateurs similaires u' pour "extrapoler" ses préférences et lui recommander de nouvelles ressources ; ce calcul peut être vu comme un système de vote dans lequel le poids des votants u' dépend de leur similarité avec u.

Chacune de ces étapes peut être mise en œuvre de différentes manières. Cet article présente pour chacune des deux approches reposant l'une, classique, sur des fonctions d'utilité et l'autre, nouvelle, sur des relations de préférence, la façon de mettre en œuvre ces étapes.

4 Mise en œuvre du filtrage collaboratif

4.1 Acquisition des profils

A base de fonction d'utilité. Quand on utilise des fonctions d'utilité, une première approche consiste à demander explicitement à l'utilisateur de voter pour un certain nombre de ressources. Ces ressources peuvent être proposées par le système, typiquement avec une sélection représentative appelée *ensemble jauge* [12], ou choisies par l'utilisateur, par exemple au cours de sa navigation sur un site web.
Toutefois, l'échelle de votes disponible est souvent réduite, ce qui limite la précision de l'évaluation (avec un petit nombre de classes d'équivalence). En outre, la manière de voter d'un utilisateur peut dépendre du contexte, évoluant par exemple selon son humeur ou selon les ressources dernièrement notées.
On remarquera qu'il faut poser $|I|$ questions pour connaître le profil complet d'un utilisateur.
On peut aussi chercher à estimer automatiquement l'intérêt porté à certaines ressources rencontrées lors de la navigation de l'utilisateur (en fonction du temps passé sur différentes pages, si la personne imprime la page, des liens suivis, etc.) [13]. Mais ce travail d'estimation indirecte est encore plus délicat à mener.

A base de relation de préférence. Dans le cas de l'utilisation d'une relation de préférence, une procédure de question-réponse similaire peut être employée, pour acquérir les préférences de l'utilisateur. Dans ce cas il faut lui demander, pour des couples de ressources $\{i, j\}$, laquelle il préfère à l'autre ou s'il est indifférent. Nous rappelons que trois réponses sont possibles : $i \prec j$, $j \prec i$ et $i \simeq j$.
L'acquisition du profil d'un utilisateur sous forme de relation de préférence n'est pas limitée à un nombre donné de classes d'équivalence. On a donc une plus grande précision des goûts de l'utilisateur comparé à l'utilisation d'une fonction d'utilité. Aussi, il semble plus facile, et donc plus robuste, d'évaluer des ressources de manière *relative* les unes aux autres –ce qui est fait ici– que de manière *absolue* –comme c'est le cas avec des fonctions d'utilité. La mesure effectuée sera moins sensible au contexte.
Un point délicat ici est le nombre de questions à poser pour compléter un profil. Si le profil contient actuellement n classes d'équivalence, une procédure dichotomique va requérir dans le pire cas $\lceil \log_2(n) \rceil$ questions pour ajouter une nouvelle ressource. En pratique, on pourra se restreindre à un petit nombre de questions, quitte à n'avoir qu'un profil incomplet, mais en général largement suffisant.

4.2 Similarité entre utilisateurs

Une fois les profils des utilisateurs obtenus, la seconde étape est de savoir mesurer, à l'aide de ces profils, la similarité entre les utilisateurs. Les mesures de similarité considérées ici respectent les propriétés suivantes, pour tout $u, v \in U$:
– $sim(u, v) \in [0; 1]$;
– $sim(u, v) = sim(v, u)$;
– $sim(u, v) = 1$ si et seulement si u et v ont le même profil commun ;
– $sim(u, v) = 0$ si u et v n'ont pas d'éléments de comparaison.

Fonction d'utilité. Soient u_1 et u_2 deux utilisateurs dont les profils spécifient leurs fonctions d'utilité respectives sur les sous-ensembles $I_1 \subseteq I$ et $I_2 \subseteq I$. On calcule alors classiquement la similarité à l'aide par exemple de la mesure cosine [7] par :

$$\cos_{ut}(ut_{u_1}, ut_{u_2}) = \frac{\sum_{i \in I_1 \cap I_2} ut_{u_1}(i).ut_{u_2}(i)}{\sqrt{\sum_{i \in I_1} ut_{u_1}(i)^2}.\sqrt{\sum_{i \in I_2} ut_{u_2}(i)^2}}.$$

Cette définition permet à la fois de tenir compte :
– de la proportion de ressources communes aux deux profils, et
– des différentes répartitions des votes donnés aux ressources.

Dans ce cas deux utilisateurs u_1 et u_2 seront considérés comme ayant des profils identiques si et seulement si ceux-ci sont "co-linéaires", c'est-à-dire : (1) $I_1 = I_2$ et (2) il existe une constante $k \in \mathbb{R}$ telle que, pour tout $i \in I_1$, $ut_{u_1}(i) = k.ut_{u_2}(i)$. Toutefois, de manière générale, le fait que les relations de précédence $\preceq_{u_1}$ et $\preceq_{u_2}$ (associées aux profils $ut_{u_1}()$ et $ut_{u_2}()$) soient identiques ne va pas amener à considérer u_1 et u_2 comme similaires.

Relation de préférence. Dans le cadre de l'exploitation de relations de préférence, on notera $\mathcal{I}_u$ l'ensemble de paires de ressources (i,j) pour lesquelles les valeurs $i \preceq_u j$ (et donc $j \preceq_u i$) sont présentes dans le profil de l'utilisateur u.

Nous proposons de définir $f_{u_1,u_2}(i,j)$ la fonction qui indique si deux utilisateurs u_1 et u_2 classent dans le même ordre les deux ressources i et j. Par exemple, à une paire de ressources (i,j) cette fonction associe la valeur 1 si les utilisateurs u_1 et u_2 ont les mêmes préférences en ce qui concerne i et j, et 0 sinon. Nous définissons une nouvelle mesure de similarité cosine adaptée aux relations de préférence :

$$\begin{aligned}&\cos_{\preceq}(u_1,u_2)\\ &= \frac{\sum_{(i,j)\in\mathcal{I}_1\cap\mathcal{I}_2} f_{u_1,u_2}(i,j)}{\sqrt{\sum_{(i,j)\in\mathcal{I}_1} f_{u_1,u_1}(i,j)}\cdot\sqrt{\sum_{(i,j)\in\mathcal{I}_2} f_{u_2,u_2}(i,j)}}\\ &= \frac{\sum_{(i,j)\in\mathcal{I}_1\cap\mathcal{I}_2} f_{u_1,u_2}(i,j)}{\sqrt{|\mathcal{I}_1|.|\mathcal{I}_2|}}.\end{aligned}$$

De manière comparable au cas des utilités, cette définition permet de tenir compte :

- de la proportion des paires de ressources communes aux deux profils, et
- des différentes répartitions des préférences entre ressources deux à deux.

On notera une "déformation" naturelle liée au passage d'ensembles de ressources à des ensembles de paires de ressources. Supposons par exemple qu'on dispose de deux profils d'utilisateurs tels que, pour tout $i \in I_1$, $ut_{u_1}(i) = 1$ et, pour tout $i \in I_2$, $ut_{u_2}(i) = 1$. $\mathcal{I}_1$ et $\mathcal{I}_2$ sont déduits de $ut_{u_1}()$ et $ut_{u_2}()$. On a alors :

$$\begin{aligned}\cos_{ut}(u_1,u_2) &= \frac{\sum_{i\in I_1\cap I_2} ut_{u_1}(i).ut_{u_2}(i)}{\sqrt{\sum_{i\in I_1} ut_{u_1}(i)^2}\cdot\sqrt{\sum_{i\in I_2} ut_{u_2}(i)^2}}\\ &= \frac{\sum_{i\in I_1\cap I_2} 1}{\sqrt{\sum_{i\in I_1} 1}\cdot\sqrt{\sum_{i\in I_2} 1}}\\ &= \frac{|I_1\cap I_2|}{\sqrt{|I_1|.|I_2|}}\end{aligned}$$

et

$$\begin{aligned}\cos_{\preceq}(u_1,u_2) &= \frac{\sum_{(i,j)\in\mathcal{I}_1\cap\mathcal{I}_2} f_{u_1,u_2}(i,j)}{\sqrt{|\mathcal{I}_1|.|\mathcal{I}_2|}}\\ &= \frac{\sum_{(i,j)\in\mathcal{I}_1\cap\mathcal{I}_2} 1}{\sqrt{|\mathcal{I}_1|.|\mathcal{I}_2|}}\\ &= \frac{|\mathcal{I}_1\cap\mathcal{I}_2|}{\sqrt{|\mathcal{I}_1|.|\mathcal{I}_2|}}\\ &\sim \left[\frac{|I_1\cap I_2|}{\sqrt{|I_1|.|I_2|}}\right]^2\\ &\sim \cos_{ut}(u_1,u_2)^2\end{aligned}$$

parce que $|\mathcal{I}_u| = |I_u|.(|I_u|-1)/2$. Au vu de ce résultat, on peut s'attendre à ce que le passage d'une mesure à l'autre se fasse par une déformation quadratique.

4.3 Recommandation à un utilisateur

Fonction d'utilité. Pour estimer l'utilité d'une ressource i pour un utilisateur u, l'approche classique calcule simplement une moyenne des utilités de cette ressource pour les autres utilisateurs. Dans cette moyenne, chaque "vote" d'un utilisateur u' sera pondéré par la similarité entre u et u'.

L'utilité de la ressource i pour l'utilisateur u est donc calculée comme suit :

$$\widetilde{ut}_u(i) = \frac{\sum_{u'\in U_i} sim(u,u').ut_{u'}(i)}{\sum_{u'\in U_i} sim(u,u')},$$

où U_i est l'ensemble des utilisateurs dont le profil contient la ressource i ($U_i = \{u' \in U;\ ut_{u'}(i) \neq ?\}$) et où $sim(u,u')$ est la mesure de similarité choisie.

En fait, les utilisateurs peu similaires risquent de ne faire qu'ajouter du bruit à l'estimation. Il est donc préférable de ne considérer dans la somme que les utilisateurs dont la similarité est supérieure à un seuil σ [7] :

$$\widetilde{ut}_u(i) = \frac{\sum_{u'\in U_{i,u,\sigma}} sim(u,u').ut_{u'}(i)}{\sum_{u'\in U_{i,u,\sigma}} sim(u,u')}, \qquad (1)$$

où $U_{i,u,\sigma}$ est l'ensemble des utilisateurs de U_i dont la similarité avec u est supérieure à σ ($U_{i,u,\sigma} = \{u' \in U_i;\ sim(u,u') > \sigma\}$).

Une fois ces estimations calculées, les ressources correspondant aux plus grandes utilités estimées sont recommandées. On peut fixer *a priori* soit le nombre de ressources à recommander, par exemple 10 ressources, soit la valeur d'utilité minimale pour les ressources à recommander. Il est à noter qu'on ne recommandera pas des ressources apparaissant dans le profil de l'utilisateur.

Relation de préférence. Recommander des ressources dans le cas de l'utilisation de relations de préférence consistera dans un premier temps à compléter la relation de préférence de l'utilisateur u, puis dans un second temps à lui recommander des ressources. La complétion de cette relation de préférence est un point plus délicat.

Une approche possible pourrait être :

1. de calculer, pour chaque paire (i,j), la probabilité que $i \preceq_u j$ soit vraie ; puis
2. de chercher une relation de préférence "la plus probable".

Cette deuxième étape peut être délicate puisqu'il ne suffit pas d'écrire "$(i \preceq_{\tilde{u}} j) = 1 \Leftrightarrow P(i \preceq_u j) > 0,5$". En effet, on risquerait d'obtenir une relation ne satisfaisant pas la propriété de réflexivité, de transitivité ou de complétude. C'est pour cette raison que nous avons exploité une alternative.

Nous proposons donc une autre approche consistant à estimer la position d'une ressource dans la relation de préférence cherchée. Il s'agit, en d'autres termes, de calculer une fonction de positionnement. Dans ce cas, nous allons, comme dans l'approche classique, exploiter les utilisateurs similaires u', calculer la position d'une ressource i dans chacun de leurs profils et estimer la position de i pour l'utilisateur u comme étant une moyenne pondérée des positions de i pour les utilisateurs u'.

Le profil d'un utilisateur u' étant généralement incomplet, nous proposons de mesurer la position d'une ressource i en comptant :

$\#^{\oplus}_{u',i}$ le nombre de ressources qui lui sont strictement préférées ;

$\#^{\simeq}_{u',i}$ le nombre de ressources (autres que i) qui sont équivalentes ;

$\#^{\ominus}_{u',i}$ le nombre de ressources auxquelles elle est strictement préférée.

Il reste alors un certain nombre de ressources non comparables avec i, lesquelles n'interviennent pas. Cela est dû au fait que le profil est incomplet. Certaines ressources ne sont pas comparables à toutes les ressources de la relation. Ces trois valeurs étant connues, la position de i (par rapport à la moyenne) est :

$$\widehat{ut}_{u'}(i) = \frac{-\#^{\oplus}_{u',i} + \#^{\ominus}_{u',i}}{\#^{\oplus}_{u',i} + \#^{\simeq}_{u',i} + \#^{\ominus}_{u',i}}.$$

Cette formule a plusieurs propriétés intéressantes :
- $\widehat{ut}_{u'}(i) \in [-1; +1]$;
- $\widehat{ut}_{u'}(i) = 0 \Leftrightarrow \#^{\oplus}_{u',i} = \#^{\ominus}_{u',i}$;
- $\widehat{ut}_{u'}(i) = +1 \Leftrightarrow i$ est la ressource préférée de u' ;
- $\widehat{ut}_{u'}(i) = -1 \Leftrightarrow i$ est la ressource la moins aimée de u'.

En exploitant ce calcul de position d'une ressource i au sein du profil d'un utilisateur quelconque u', on peut exploiter la formule de prédiction classiquement utilisée (celle de la section précédente) :

$$\widetilde{ut}_u(i) = \frac{\sum_{u' \in \hat{U}_{i,u,\sigma}} sim(u,u').\widehat{ut}_{u'}(i)}{\sum_{u' \in \hat{U}_{i,u,\sigma}} sim(u,u')},$$

où $\hat{U}_{i,u,\sigma}$ est l'ensemble des utilisateurs u' de U :
- pour lesquels i apparaît dans leur profil (est comparé à au moins une autre ressource) ; et
- dont la similarité avec u dépasse le seuil σ.

Mentionnons que plus la position d'une ressource est élevée, meilleure sera la ressource pour l'utilisateur. Une fois les positions estimées pour un utilisateur u, pour l'ensemble des ressources i absentes du profil, les ressources ayant les positions estimées les plus hautes seront recommandées à l'utilisateur, en suivant le même principe que dans la section 4.3.

5 Expérimentations

5.1 Données d'expérimentation

Idéalement, une comparaison expérimentale entre les approches "relations de préférence" et "utilité" devrait mettre en compétition les deux chaînes complètes correspondantes allant de l'entrée des préférences à la recommandation de ressources. Or il n'existe pas à notre connaissance de bases de préférences recueillies aussi bien sous la forme de relations de préférence que d'utilités. Mettre en place une telle base serait une tâche particulièrement laborieuse que nous n'avons pas nous-même menée. On ne pourra donc pas comparer directement la précision des approches dans l'un et l'autre cas.

Les bases les plus courantes exploitées dans les systèmes à base de filtrage collaboratif fournissent des utilités (votes). Nous choisissons donc d'exploiter une telle base pour générer des recommandations en utilisant à la fois des utilités et des relations de préférence. Pour obtenir les relations de préférence des utilisateurs, nous transformons les utilités disponibles pour chaque utilisateur sous la forme d'une relation de préférence. Nous sommes dans ce cas dans les pires conditions pour quantifier l'apport de notre approche : nous perdons des informations puisque nous partons d'utilités (et donc de données avec des imprécisions). De plus, retransformant ces utilités en relations de préférence, nous dégradons à nouveau le potentiel de l'approche, du fait de la perte de l'information quantitative contenue dans les utilités. Il n'est donc pas attendu d'améliorer la fiabilité des recommandations obtenues mais nous souhaitons prouver le potentiel de notre approche si nous obtenons des performances comparables.

En l'occurrence, nous avons travaillé avec la base de l'état de l'art MovieLens.[3] Cette base est composée d'un ensemble de préférences utilisateurs sur des films. Ces préférences se trouvent sous forme d'utilités (votes) qui sont des entiers entre 1 et 5.[4] La base recense 1682 utilisateurs, 943 ressources et 100k préférences. La base est divisée 2 parties, 80% des données sont utilisées pour apprendre le système de recommandation (les données d'apprentissage) et les 20% restant sont utilisés pour évaluer l'approche (les données de test). Après transformation des données sous forme de relations de préférence, nous pouvons noter que chaque relation de préférence obtenue à ce stade contiendra donc au plus 5 classes d'équivalence.

5.2 Mesures d'évaluation

Dans les approches classiques du filtrage collaboratif, le système de recommandation estime les utilités manquantes. Ces utilités sont ensuite comparées aux votes des données de test (que les utilisateurs ont effectivement mis). La qualité du système est évaluée à l'aide de la mesure MAE (Mean Absolute Error) qui calcule l'erreur moyenne

[3] http ://movielens.org

[4] Un vote de 1 signifie que la personne n'a pas aimé et 5 signifie que la personne a adoré le film.

sur les votes fournis par le système de recommandation. Plus une valeur de MAE est basse, et plus l'erreur sera faible, on cherche donc à minimiser la valeur de la MAE.
Dans notre approche exploitant des relations de préférence, aucun vote n'est manipulé ; aussi le système ne peut fournir de vote. Les performances ne pourront donc pas être évaluées en exploitant les votes des données de test. Nous proposons dans ce cas d'évaluer les performances du système à l'aide de la mesure de précision en comparant la liste des ressources préférées par les utilisateurs (celles du corpus de test qui ont obtenu les meilleurs votes) avec celles que le système a considérées comme étant les ressources préférées (celles ayant obtenu le meilleur score). Concrètement, nous évaluerons le rapport entre le nombre de ressources préférées par les utilisateurs que le système juge comme étant les préférées et le nombre de ressources effectivement préférées par les utilisateurs.
Nous évaluerons également notre approche en terme de rang moyen, ce qui nous permettra de quantifier la qualité du système sur l'ensemble des ressources effectivement appréciées par l'utilisateur.

5.3 Résultats

** début relecture** Nous choisissons donc d'évaluer notre approche en trois étapes. Dans un premier temps, nous utiliserons les relations de préférence uniquement dans le calcul de la similarité entre utilisateurs. Cela nous permettra de quantifier leur influence sur la qualité des recommandations. Dans un second temps, nous utiliserons les relations de préférence sur l'ensemble du processus de recommandation : à la fois lors du calcul de la similarité entre utilisateurs et pour effectuer des recommandations. Cela nous permettra de quantifier globalement l'influence des relations de préférence. Enfin, nous évaluerons les performances de l'approche à base de relations de préférence en termes de rang moyen, ce qui permettra de quantifier la qualité des recommandations sur l'ensemble des ressources que l'utilisateur a effectivement appréciées.

Similarité entre utilisateurs. Dans cette évaluation, nous exploitons les relations de préférence dans le processus classique de recommandation, mais uniquement sur l'étape de calcul de similarité entre utilisateurs (équation 1). Le processus sera exécuté dans un premier temps en instanciant la formule $sim(u, u')$ par $\cos_{ut}(u, u')$: la similarité entre utilisateurs calculée à partir d'utilités. Dans un second temps, elle sera instanciée par $\cos_{\preceq}(u, u')$: la similarité entre utilisateurs calculée à partir de relations de préférence.
Dans ce cas, l'évaluation des deux approches peut se faire en terme de MAE puisque les votes des données de test sont disponibles. On pourra alors quantifier l'impact de la mesure de similarité sur les performances. Les résultats en termes de MAE sont présentés dans le tableau 1.
Nous pouvons remarquer que l'erreur moyenne en prédiction est plus grande lorsque l'on utilise des relations de préférence. Cependant, cette dégradation est de moins de 3% et n'est donc pas véritablement significative. Cette dégradation était prévisible, puisque nous exploitons une mesure de similarité qui n'est pas basée sur les votes, et qui est utilisée pour estimer des votes. Les conditions d'expérimentation sont donc biaisées. La faible perte constatée dans ce cas montre la potentialité de notre approche.

TAB. 1 – MAE en fonction de l'approche utilisée pour la mesure de similarité entre utilisateurs

	MAE
Fonctions d'utilité	0,71
Relations de préférence	0,73

Recommandation à un utilisateur. Nous exploitons maintenant les relations de préférence sur l'ensemble du processus de recommandation. Comme nous l'avons précisé auparavant, la mesure de MAE ne peut plus être utilisée. Nous évaluons donc à l'aide de la mesure de précision. Dans ce cas, les conditions d'expérimentations sont moins biaisées.
Nous mesurons donc, parmi les ressources que l'utilisateur a jugées comme étant celles qu'il préférait, si ces ressources sont évaluées par le système de recommandation comme étant des ressources préférées, qu'il exploite les utilités ou des relations de préférence.
Les ressources que nous considérons comme étant préférées par un utilisateur sont celles qui sont les plus fortement votées. Dans le cas du corpus MovieLens, cela correspond aux votes 4 et 5. Nous évaluons donc la précision de notre approche sur deux ensembles de ressources : l'ensemble des ressources votées uniquement 5 et l'ensemble des ressources votées 4 ou 5. Les précisions associées sont présentées dans le tableau 2.

TAB. 2 – Précision des deux appproches

Approche	vote à 5	vote à 4 ou 5
Fonctions d'utilité	0,52	0,75
Relations de préférence	0,51	0,77

Nous pouvons remarquer que, sur les deux ensembles des ressources, les deux approches ont des précisions similaires. L'approche à base d'utilités a cependant une précision très légèrement supérieure dans le cas des ensembles de ressources votées à 5. En revanche, sur l'ensemble des ressources avec vote à 4 ou 5, l'approche à base de relations de préférence est légèrement supérieure à celle à base d'utilités.

Evaluation en termes de rang moyen. Rappelons que notre approche s'appuie sur la "position" des ressources les unes relativement aux autres. L'évaluation en termes de rang moyen est également une évaluation en termes de po-

sition. Elle est donc l'évaluation la mieux appropriée pour valider l'apport de notre approche.
Le rang moyen est calculé sur l'ensemble des ressources votées 5 par les utilisateurs dans les données de test. L'étude du rang moyen montre que, dans le cas des relations de préférence, il est de 9% plus faible que pour l'approche classique à base d'utilités. Les ressources sont donc mieux classées. Nous pouvons donc conclure que, dans ce cas, notre approche est meilleure.
Au vu des expérimentations menées, nous pouvons conclure que l'approche par relations de préférence, moins contraignante pour l'utilisateur, permet d'obtenir des performances similaires à celle utilisant des utilités. Ces résultats nous encouragent donc à poursuivre dans cette voie.

6 Conclusion et perspectives

Cet article propose une nouvelle approche des préférences dans les systèmes de recommandation à base de filtrage collaboratif. Nous exploitons des relations de préférence en remplacement des votes classiquement utilisés. Dans un premier temps, nous avons présenté les notions d'utilité et de relations de préférence que nous avons mises en relation. Nous avons proposé une adaptation des étapes du filtrage collaboratif afin de pouvoir prendre en compte des relations de préférence. Nous avons évalué cette approche sur un corpus de l'état de l'art que nous avons transformé de façon à représenter des relations de préférence. Notre approche obtient des performances similaires à l'approche classique (à base d'utilités) alors que le corpus utilisé pour les tests est en sa défaveur.
Les perspectives de ce travail sont nombreuses puisqu'il ouvre une voie nouvelle pour le filtrage collaboratif. Dans un premier temps, nous évaluerons la robustesse de cette nouvelle approche en supprimant des informations des relations de préférence afin d'observer l'évolution des performances et étudier la quantité d'information minimale suffisante pour effectuer des recommandations "acceptables" pour un utilisateur.
A plus long terme, nous mettrons en œuvre un système complet comprenant également l'étape d'acquisition des préférences utilisateurs sous forme de relations de préférence, étape cruciale qui nous manque ici mais qui est une tâche ardue.

Références

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems : A survey of the state-of-the-art. *IEEE Transactions on Knowledge and Data Engineering*, 17(6) :734–749, 2005.

[2] M. Pazzani and D. Billsus. *The Adaptive Web*, chapter Content-Based Recommendation Systems, pages 325–341. Springer Berlin / Heidelberg, 2007.

[3] R. Burke, K. Hammond, and E. Cooper. Knowledge-based navigation of complex information spaces. In *Proc. of the 13th National Conference on Artificial Intelligence (AAAI'96)*, pages 462–468, Menlo Park, Canada, 1996.

[4] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12) :61–70, 1992.

[5] L. Candillier, F. Meyer, and M. Boullé. Comparing state-of-the-art collaborative filtering systems. In *Proc. of 5th International Conference on Machine Learning and Data Mining in Pattern Recognition, MLMD'07*, pages 548–562, 2007.

[6] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl. GroupLens : An Open Architecture for Collaborative Filtering of Netnews. In *Proc. of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186, Chapel Hill, North Carolina, 1994. ACM.

[7] U. Shardanand and P. Maes. Social information filtering : algorithms for automating "word of mouth". In *Proc. of the ACM CHI'95 - Conference on Human Factors in Computing Systems*, volume 1, pages 210–217, 1995.

[8] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1) :5–53, 2004.

[9] L.G. Perez, M. Barranco, and L. Martinez. Building user profiles for recommender systems from incomplete preference relations. In *Proc. of the Fuzzy Systems Conference*, 2007.

[10] J. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 43–52, San Francisco, CA, 1998. Morgan Kaufmann.

[11] D.M. Pennock, E. Horvitz, S. Laurence, and C.L. Giles. Collaborative filtering by personality diagnosis : A hybrid memory- and model-based approach. In *Proc. of the 16th Conference on Uncertainty in Artificial Intelligence (UAI'00)*, pages 473–480, 2000.

[12] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste : A constant time collaborative filtering algorithm. *Information Retrieval Journal*, 4(2) :133–151, July 2001.

[13] S. Castagnos. *Modélisation de comportements et apprentissage stochastique non supervisé de stratégies d'interactions sociales au sein de systèmes temps réel de recherche et d'accès à l'information*. PhD thesis, Université Nancy 2, 2008.