

---

# Exploitation des données « disponibles à froid » pour améliorer le démarrage à froid dans les systèmes de filtrage d'information

## Une approche expérimentale fondée sur les « espaces de communautés » et la classification par règles

**An-Te Nguyen, Nathalie Denos, Catherine Berrut**

Laboratoire CLIPS-IMAG

385, rue de la Bibliothèque

BP 53, 38041 Grenoble Cedex

{an-te.nguyen, nathalie.denos, catherine.berrut}@imag.fr

---

*RÉSUMÉ.* Dans les systèmes de filtrage d'information, les utilisateurs reçoivent des documents que leur recommande le système sur la base de leurs profils et/ou de leurs communautés, mais le problème du « démarrage à froid » conduit à des performances très pauvres pour les nouveaux utilisateurs. Nous améliorons ici les techniques classiques de démarrage à froid en exploitant les données « disponibles à froid » (par exemple âge, profession, lieu de résidence, etc.) afin d'associer automatiquement les meilleures communautés initiales aux nouveaux utilisateurs. Nous nous appuyons sur le modèle existant des « espaces de communautés », et nous utilisons un processus de classification par règles et définissons un processus de recommandations par niveau d'accord. L'évaluation montre que cette approche fournit des recommandations meilleures que celles obtenues suite à un processus de démarrage à froid classique.

*ABSTRACT.* With information filtering systems, users receive documents recommended on the basis of their profiles and/or communities. New users experience the “cold-start” problem: as their profiles and communities are very poor, the system performs very poorly. In this paper, we improve classical “cold-start” techniques by exploiting the data that are “readily available” (e.g. age, job, city, etc.), in order to automatically associate users with better initial communities. We rely on the existing “community spaces model”, and we use a rule-based classification process and define a recommendation process based on the “level of agreement”. The experiments lead to better recommendations than those obtained after a classical “cold-start” technique.

*MOTS-CLÉS:* systèmes de filtrage, démarrage à froid, espaces de communautés.

*KEYWORDS:* filtering systems, recommender systems, cold-start problem, community spaces.

---

*Cette recherche a été partiellement soutenue par le Ministère Délégué à la Recherche et aux Nouvelles Technologies, dans le programme ACI Masses de Données, projet #MD-33.*

## 1. Introduction

**Problématique.** Dans les systèmes de filtrage d'information, les utilisateurs reçoivent des documents que leur recommande le système sur la base de leurs profils et/ou de leurs communautés. Le profil et les communautés d'un utilisateur évoluent au cours du temps grâce aux interactions entre l'utilisateur et le système, notamment grâce aux évaluations produites par cet utilisateur. Lorsqu'il s'inscrit et commence à utiliser le système, le problème du « démarrage à froid » se pose, car son profil est encore très pauvre, voire inexistant, et ses communautés sont encore inconnues. Par conséquent, le système ne peut pas lui fournir des recommandations pertinentes (Maltz *et al.*, 1995).

Selon le type de filtrage d'information (collaboratif, basé sur le contenu, ou encore hybride), les réponses à ce problème varient, mais dans tous les cas, l'utilisateur est mis à contribution, et les tâches qui lui sont demandées sont généralement difficiles, longues et/ou fastidieuses. L'objectif de cet article est d'améliorer ces processus de gestion du démarrage à froid. Pour cela nous proposons d'exploiter les données « disponibles à froid » pour rattacher automatiquement le nouvel utilisateur aux meilleures communautés initiales et à terme pour lui associer un meilleur premier profil.

Les données « disponibles à froid » sont les informations sur l'utilisateur que l'on peut recueillir dès son inscription, et que l'utilisateur peut fournir avec une grande fiabilité, et sans effort particulier : par exemple, son âge, sa profession, son lieu de résidence. Les autres données qui participent au profil de l'utilisateur (Bouzghoub *et al.*, 2005) (évaluations des documents présentés, description du contenu des documents intéressants) sont plus difficiles à obtenir des utilisateurs.

**Notre approche.** Nous pensons qu'utiliser les données disponibles à froid peut permettre de compenser partiellement les données non fournies initialement par les utilisateurs, qui sont utiles dans le positionnement d'utilisateurs au sein des communautés. Notre objectif ici consiste à valider expérimentalement que cette première approche améliore significativement les performances des systèmes.

Ainsi, nous montrons comment positionner un nouvel utilisateur dans les communautés multicritères sans qu'il ait à fournir d'évaluations ou de descriptions. Pour cela, nous nous appuyons sur le modèle des « espaces de communautés » (Nguyen *et al.*, 2006) qui permet de prendre en compte les divers facteurs de pertinence exploitables dans les systèmes de filtrage d'information. Schématiquement, ce modèle permet à un utilisateur d'être membre de multiples communautés : une par espace de communautés, un espace de communautés étant associé à chaque facteur de rapprochement entre utilisateurs.

Les données « disponibles à froid » nous permettent d'attribuer au nouvel utilisateur une communauté pour elles-mêmes (communautés d'âge, de profession

etc.) mais également dans l'espace des communautés relatives au contenu, ou relatives aux évaluations passées. Ainsi le système déduit les communautés Contenu et Evaluation par un processus de classification par règles, qui s'appuie sur :

a) l'appartenance de l'utilisateur aux communautés relevant des espaces où l'information est bien connue (données disponibles à froid comme Age, Profession, etc.), et

b) l'ensemble des données relatives aux utilisateurs déjà intégrés dans le système, données qui révèlent la façon dont les différents attributs sont liés entre eux.

Les performances de cette approche sont évaluées selon leur capacité à fournir, sans effort de la part de l'utilisateur, des recommandations équivalentes à celles obtenues suite à un processus de démarrage à froid classique, qui lui, requiert un effort de l'utilisateur.

## **2. Etat de l'art sur le démarrage à froid**

Le démarrage à froid est le phénomène qui se produit en début d'utilisation du système, dans des situations critiques où le système manque de données pour procéder à un filtrage personnalisé de bonne qualité. On distingue trois types de problèmes de démarrage à froid (Burke, 2002) :

– le démarrage à froid pour un nouveau système (« new system ») (Middleton *et al.*, 2002), où les performances des systèmes sont très mauvaises en raison de l'absence d'informations sur lesquelles fonder le processus de filtrage personnalisé. Ce problème est généralement traité en exploitant des données externes, données dont on ne dispose pas toujours, selon le cadre applicatif (Middleton *et al.*, 2004).

– le démarrage à froid pour un nouveau document (« new item ») : c'est un problème spécifique à l'approche collaborative, pour laquelle les objets à recommander ne sont décrits que par les évaluations fournies par les utilisateurs. Ce problème est généralement traité en combinant une approche de filtrage basé sur le contenu avec l'approche collaborative (approche hybride) par exemple utilisant la similarité entre documents (Schein *et al.*, 2001) ou introduisant des agents intelligents qui évaluent les documents automatiquement (Good *et al.*, 1999).

– le démarrage à froid pour un nouvel utilisateur (« new user ») : le profil de l'utilisateur est inexistant et ses communautés sont encore inconnues, ce qui conduit à des recommandations de mauvaise qualité.

Nous nous intéressons ici au démarrage à froid pour un nouvel utilisateur. Selon le type de filtrage, les réponses à ce problème varient. A l'exception des contextes applicatifs où des sources externes d'informations sur les nouveaux utilisateurs existent, l'utilisateur est toujours mis à contribution. Selon les modalités de cette contribution, la tâche est plus ou moins difficile, plus ou moins fastidieuse, et les

résultats (profil et communautés résultant du processus de démarrage à froid) sont plus ou moins fiables.

Pour le filtrage collaboratif qui se base sur les communautés formées à partir des évaluations d'utilisateurs (Breese *et al.*, 1998), la technique classique consiste à demander à l'utilisateur d'évaluer un ensemble de documents que le système lui présente, ces documents n'ayant a priori aucun rapport privilégié avec les centres d'intérêt potentiels de l'utilisateur. Cette tâche est simple mais fastidieuse : par exemple, le système de recommandation de films MovieLens<sup>1</sup> exige de l'utilisateur au moins 15 évaluations avant de fournir des recommandations, mais les films proposés n'étant pas ciblés pour cet utilisateur, ce dernier peut ne rencontrer que des films qu'il n'aime pas, ou des films qu'il ne sait pas évaluer, comme par exemple des films dont il n'a jamais entendu parler. Ainsi l'utilisateur devra parcourir une liste parfois très longue avant d'atteindre ce nombre de 15 évaluations. Les travaux se consacrant à cette approche ont donc pour objectif de trouver les meilleurs documents à présenter aux nouveaux utilisateurs, et cela indépendamment de l'utilisateur considéré (Rashid *et al.*, 2002).

Pour le filtrage basé sur le contenu et le filtrage hybride (Burke, 2002), la réponse la plus courante consiste à demander à l'utilisateur de définir ses centres d'intérêt à partir d'une liste de termes et/ou d'exemples décrivant au mieux ses centres d'intérêt (Melville *et al.*, 2002, Claypool *et al.*, 1999). L'effort demandé à l'utilisateur est important : il doit procéder à une réflexion pour synthétiser ses centres d'intérêt sous la forme de termes, ou bien rechercher des exemples pertinents de documents. Cette dernière tâche peut être automatisée lorsque le système dispose de données externes sur les utilisateurs, par exemple pour des chercheurs académiques qui sont aussi auteurs de publications (Middleton *et al.* 2002, Middleton *et al.* 2004), mais en règle générale on ne dispose pas de telles données. Dans la plupart des cas, le profil résultant est incomplet et bruité.

Enfin, une dernière approche consiste à associer le nouvel utilisateur à un profil-type (« stéréotype ») parmi ceux prédéfinis. Le processus de construction de ces profils-type nécessite un ensemble de données d'apprentissage, puis le processus de démarrage à froid doit confronter à ces profils-type certaines informations relatives au nouvel utilisateur. Le système les obtient généralement en interagissant avec l'utilisateur, par exemple en lui posant une série de questions (Krulwich, 1997), mais aussi parfois en exploitant une source d'informations « démographiques » externe comme leurs pages Web personnelles (Pazzani, 1999). Pazzani étudie entre autres les performances de cette approche en la prenant pour unique base d'un système de recommandation « démographique », et nous retenons de son travail que les performances de recommandation via les données démographiques sont certes moins bonnes que celles de systèmes basés contenu ou collaboratifs, mais toutefois acceptables.

---

<sup>1</sup> <http://movielens.umn.edu>

Pour conclure, ces processus de démarrage à froid ont tous un coût pour l'utilisateur : d'une part ils requièrent un effort en termes de temps ou d'accomplissement d'une tâche plus ou moins difficile ; d'autre part, le profil et les communautés résultant de ce processus ne peuvent pas donner lieu à des recommandations de qualité. Ils conduisent donc à un rapport coût/bénéfice déficitaire pour les utilisateurs.

Suite à ce constat, nous proposons ici d'étudier la qualité obtenue avec une approche où l'on ne demande aucun effort à l'utilisateur (quelques questions « démographiques » dont la réponse est simple, immédiate et sûre), sachant qu'à terme cette approche est destinée à être combinée avec l'une ou l'autre des approches classiques de façon à leur conférer un meilleur rapport coût/bénéfice. Notre approche de démarrage à froid s'apparente donc à celle des profils-type, avec l'optique de limiter au maximum le nombre et la complexité des informations à demander aux utilisateurs.

### 3. Approche

#### 3.1. Approche par les espaces de communauté

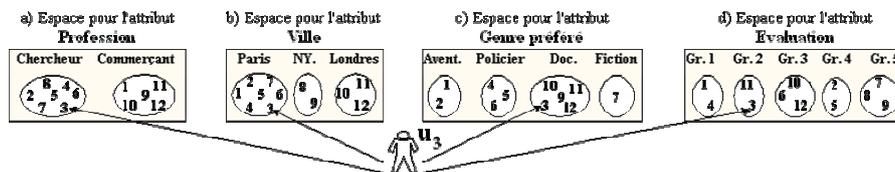
Des travaux antérieurs nous ont conduits à définir le modèle des « espaces de communautés » (Nguyen *et al.*, 2006) comme un cadre pour prendre en compte les divers facteurs de pertinence exploitables dans les systèmes de filtrage d'information. Nous présentons ici les éléments du modèle et les résultats utiles pour la suite.

Un utilisateur est positionné dans divers espaces de communautés, chaque espace  $\Omega_a$  étant associé à un « attribut »  $a$  (Age, Profession, Contenu, Evaluation, etc.), et reflétant un facteur possible de rapprochement entre les utilisateurs (même tranche d'âge, même catégorie socio-professionnelle, proximité des thèmes couvrant les centres d'intérêt, similarité dans l'évaluation des documents, etc.). Dans chacun de ces espaces de communautés  $\Omega_a$ , l'utilisateur appartient à une communauté particulière  $G_a$ . Typiquement, le système de filtrage peut s'appuyer sur chacune de ces communautés pour produire un flux de recommandations (approche collaborative). Etant donné la table de communautés  $T$  (voir **Figure 1**) construite à partir des espaces de communautés, le « vecteur de positionnement »  $P_u = T[u]$  de l'utilisateur  $u$  est composé des positions  $G_a$  de cet utilisateur dans les divers espaces de communautés  $\Omega_a$ , et chacune de ces positions constitue une source de recommandation potentielle.

A partir d'un vecteur de positionnement incomplet et des données relatives aux autres utilisateurs, nous complétons les valeurs manquantes de ce vecteur et

produisons une liste de recommandations pour chacun des espaces de communautés. Ainsi, pour positionner un nouvel utilisateur, nous initialisons d'abord le vecteur de positionnement avec ses données disponibles à froid. Par exemple pour les attributs Profession, Ville, Genre préféré et Evaluation, un nouvel utilisateur  $u$  verra son vecteur initialisé ainsi :  $P_u = (\text{Chercheur}, \text{Paris}, \_, \_)$ . Nous complétons ensuite ce vecteur de positionnement automatiquement en instanciant les valeurs manquantes, en vue de limiter l'effort de l'utilisateur pour passer le cap du démarrage à froid.

Utilisateur \ Attribut	Profession	Ville	Genre préféré	Evaluation
$u_1$	Commerçant	Paris	Aventure	Groupe 1
$u_2$	Chercheur	Paris	Aventure	Groupe 4
$u_3$	Chercheur	Paris	Documentaire	Groupe 2
$u_4$	Chercheur	Paris	Policier	Groupe 1
$u_5$	Chercheur	Paris	Policier	Groupe 4
$u_6$	Chercheur	Paris	Policier	Groupe 3
$u_7$	Chercheur	Paris	Fiction	Groupe 5
$u_8$	Chercheur	New York	Documentaire	Groupe 5
$u_9$	Commerçant	New York	Documentaire	Groupe 5
$u_{10}$	Commerçant	Londres	Documentaire	Groupe 3
$u_{11}$	Commerçant	Londres	Documentaire	Groupe 2
$u_{12}$	Commerçant	Londres	Documentaire	Groupe 3

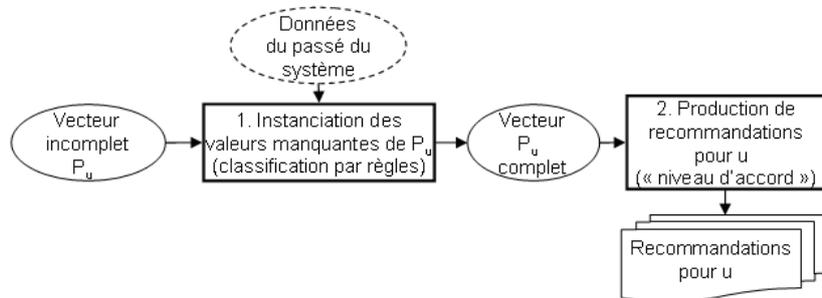


**Figure 1.** Table de communautés d'un système de recommandation de films et espaces de communautés associés à chaque attribut.

Dans la suite, nous décrivons le principe de l'instanciation des valeurs manquantes dans un vecteur de positionnement d'un nouvel utilisateur, qui est faite via un processus de classification par règle, puis le dispositif mis en place pour évaluer l'apport de cette approche.

### 3.2. Principe pour compléter un vecteur de positionnement

Compléter le vecteur de positionnement d'un utilisateur au sein des espaces de communautés se fait en 2 étapes (voir **Figure 2**).



**Figure 2.** Principe pour compléter un vecteur de positionnement.

**Etape 1. Classification par règles.** On complète le vecteur de positionnement en introduisant des valeurs estimées là où les valeurs sont manquantes. Ces valeurs sont estimées grâce à un processus de *classification par règles*.

Nous exploitons d’abord l’ensemble des vecteurs de positionnement dont le système dispose : ce sont les vecteurs des utilisateurs arrivés plus tôt dans le système, appelées « données du passé du système ». On extrait de ces données des connaissances qui permettent d’estimer la valeur manquante. Pour cela, nous avons choisi d’utiliser une méthode de classification par règles ou arbre de décision (Quinlan, 1993 ; C5.0 ; Liu *et al.*, 2000 ; CBA) en raison des bonnes performances de ces méthodes, et de leur capacité d’explication (nécessaire pour une future interaction avec l’utilisateur).

Ce type de méthode commence par une *phase d’apprentissage* s’appuyant sur les données disponibles pour construire un classificateur qui est un ensemble de règles,  $X \rightarrow d$  où  $X$  est un sous ensemble d’attributs et  $d$  est l’attribut choisi comme « décision », c’est-à-dire celui pour lequel on veut instancier la valeur manquante. La qualité d’une règle  $X \rightarrow d$  dans le classificateur est généralement mesurée par son support et sa confiance. Le support d’une règle est la proportion d’occurrences de cette règle sur la taille de l’ensemble d’apprentissage, et la confiance est la proportion d’occurrences de cette règle par rapport aux occurrences de la prémisse  $X$ .

La méthode se poursuit par une *phase de classification*, où on cherche une règle applicable  $X \rightarrow d$  dont la prémisse  $X$  est satisfaite par les données disponibles de l’utilisateur  $u$  considéré. La valeur  $d$  de la règle trouvée remplace alors la valeur manquante dans le vecteur de positionnement de  $u$ . Pour connaître les détails de ces méthodes, nous conseillons au lecteur de consulter les références suivantes : (Quinlan, 1993 ; Liu *et al.*, 2000).

Une fois le vecteur de positionnement complété, il nous faut évaluer sa qualité au niveau de production des recommandations pour l’utilisateur. Dans ce but d’évaluation, nous avons défini de façon ad hoc un *processus de recommandation*

*fondé sur le « niveau d'accord » au sein des communautés que nous décrivons ci-dessous.*

**Etape 2. Recommandations par « niveau d'accord ».** Le principe de cette méthode de production de recommandations est très simple : supposons que le nouvel utilisateur  $u$  appartient à la communauté  $G$  dans un espace de communautés donné. Alors, le nouvel utilisateur reçoit tous les documents jugés intéressants de façon unanime par la communauté  $G$ . Le niveau d'accord au sein de la communauté est alors maximal, puisqu'il y a unanimité.

Ce principe général peut être décliné de diverses manières, par exemple en limitant le « niveau d'accord » à un sous-ensemble de la communauté, comme par exemple aux membres les plus représentatifs de la communauté.

### **3.3. Application au jeu de données MovieLens**

Le jeu de données MovieLens<sup>2</sup> permet d'appliquer cette approche sur 5 espaces de communautés, à partir desquels nous pouvons générer, entre autres, 3 listes de recommandations : une liste issue de la combinaison des attributs disponibles à froid (Age, Profession, Géographie), et une liste par attribut déduit (Genre préféré et Evaluation).

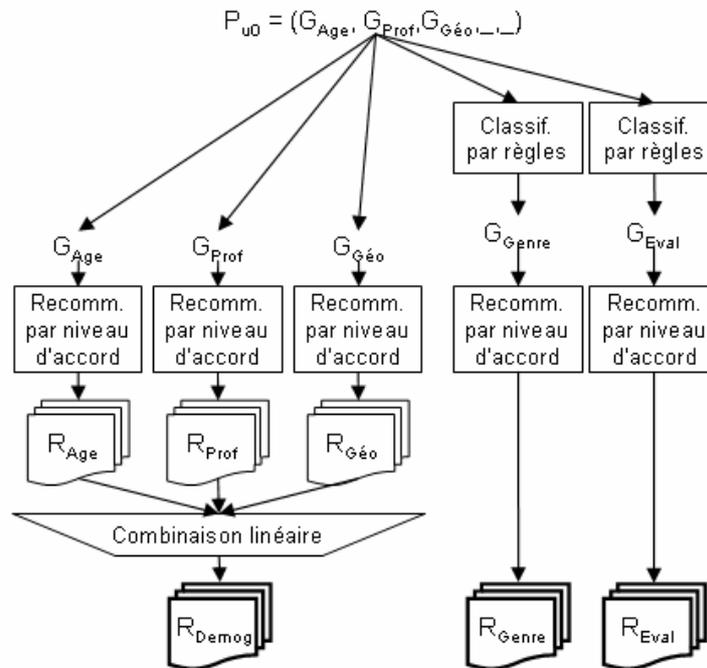
**Jeu de données.** Dans la continuité des travaux présentés dans (Nguyen *et al.*, 2005, Nguyen *et al.*, 2006), nous avons utilisé le jeu de données réelles MovieLens qui contient 100 000 évaluations données par 943 utilisateurs sur 1682 films relevant de 19 genres de film. Chaque évaluation contient les identificateurs de film et d'utilisateur, le score et le moment d'évaluation. Nous avons exploité 5 attributs permettant la formation de 5 espaces de communautés : Age, Profession, Géographie, Genre préféré et Evaluation.

**Application de l'approche.** A partir des 3 attributs disponibles à froid (Age, Profession, Géographie) pour un nouvel utilisateur  $u_0$ , nous pouvons produire divers types de recommandations, présentés dans la **Figure 3**.

$R_{Demog}$  : les recommandations s'appuient directement sur les communautés  $G_{Age}$ ,  $G_{Profession}$  et  $G_{Géographie}$  sans passer par l'étape de classification, ici inutile puisque les valeurs du vecteur  $P_u$  sont déjà connues pour ces attributs. On génère 3 ensembles de recommandations  $R_{Age}$ ,  $R_{Prof}$ ,  $R_{Géo}$  par niveau d'accord dans chacun des 3 espaces de communautés correspondants, puis on les combine linéairement avec des pondérations égales pour les 3 sources de recommandation.

---

<sup>2</sup> <http://movielens.umn.edu>



**Figure 3.** Production de diverses listes de recommandations à partir des données disponibles à froid.

$R_{Genre}$  : les attributs Age, Profession et Géographie sont utilisés en entrée de la classification par règles, qui associe à  $u_0$  une communauté  $G_{Genre}$  dans l'espace  $\Omega_{Genre}$ . Le processus de recommandation par niveau d'accord génère ensuite pour  $u_0$  une liste de recommandations  $R_{Genre}$  en rapport avec la position de l'utilisateur dans cette communauté.

$R_{Eval}$  : il en est de même pour  $R_{Eval}$  où l'espace  $\Omega_{Evaluation}$  remplace l'espace  $\Omega_{Genre}$ .

En pratique, voici comment nous avons réalisé la classification et la recommandation.

**Méthode de classification par règles.** A l'étape 1 (voir **Figure 2**), nous instancions les valeurs manquantes via un processus de classification par règles. Après avoir comparé diverses méthodes, nous avons choisi la méthode C5, successeur de la méthode populaire C4.5 (Quinlan, 1993).

**Recommandation ad hoc par « niveau d'accord ».** A l'étape 2 (voir **Figure 2**), nous produisons les recommandations par « niveau d'accord ». Pour chaque attribut (c'est-à-dire pour chaque espace de communautés), la méthode doit sélectionner les

films à recommander parmi ceux qui ont été évalués par la communauté dans laquelle  $u_0$  se situe. On procède à un premier filtre en ne considérant que les films évalués avec un score supérieur ou égal à un seuil  $S_{\text{score}}$ , puis à un deuxième filtre où on ne conserve que les films qui ont été ainsi évalués par une part suffisante des membres de la communauté, dont la limite est fixée par un autre seuil  $S_{\text{accord}}$ .

#### 4. Expérimentation

Nous avons évalué notre approche en comparant ses performances de recommandations à celles d'une approche classique de démarrage à froid obtenue par le recueil de 15 évaluations par utilisateur. Ces performances seront mesurées par rapport à des données de références. Pour cela nous utilisons 3 mesures différentes.

##### 4.1. Les données à évaluer et les données de référence

**Les listes de recommandations issues de notre approche  $R_{\text{Genre}}$ ,  $R_{\text{Eval}}$  et  $R_{\text{Demog}}$ .** Nous avons divisé l'ensemble des utilisateurs de MovieLens en deux catégories :

(i) « Nouveaux » utilisateurs ( $U_N$ ) : 77 utilisateurs qui se sont inscrits au cours du dernier mois (04/1998), et

(ii) Utilisateurs « existants » ( $U_E$ ) : 866 utilisateurs inscrits avant le dernier mois.

Pour les nouveaux utilisateurs ( $U_N$ ), nous avons calculé les 3 listes de recommandations  $R_{\text{Genre}}$ ,  $R_{\text{Eval}}$  et  $R_{\text{Demog}}$  avec les seuils  $S_{\text{Score}} = 4$  étoiles et  $S_{\text{accord}} = 25\%$  pour le processus de recommandation par niveau d'accord.

**La liste de recommandations  $R_{\text{Pearson}}$ .** Nous nous appuyons sur un exemple de système réel pour calibrer notre base de comparaison, celui du système MovieLens<sup>3</sup> pour lequel un nouvel utilisateur doit fournir, outre des informations personnelles, des évaluations sur au moins 15 des films proposés avant de recevoir des recommandations. Pour nous rapprocher de ces conditions, nous extrayons les 15 premières évaluations d'un utilisateur  $u$  de  $U_N$ , qui composent le profil. Nous générons ensuite les recommandations pour ce profil par le processus classique de recommandation selon la corrélation de Pearson (Breese *et al.*, 1998). On obtient alors  $R_{\text{Pearson}}$ , une nouvelle liste de recommandations analogue à celles que nous générons par notre approche.

**Les données de référence  $E_{\text{Réelles}}$ .** Les évaluations des utilisateurs de  $U_N$  non utilisées pour générer les listes de recommandations constituent l'ensemble des

---

<sup>3</sup> <http://movielens.umn.edu>

données de référence : c'est l'ensemble  $E_{R\acute{e}elles}$ . Cet ensemble nous sert de référence dans les calculs de performance.

#### 4.2. Mesures d'évaluation

Pour estimer la qualité des listes de recommandations pour un nouvel utilisateur  $u_0$  dans  $U_N$ , nous utilisons les mesures suivantes.

– *Erreur moyenne absolue* (MAE – Mean Absolute Error) (Rashid *et al.*, 2002) : cette mesure calcule la différence moyenne entre les prédictions de recommandations du système  $p_j$  et les scores  $e_j$  donnés réellement par l'utilisateur dans les évaluations. Dans cet article, nous avons utilisé les deux formules suivantes :

$$M_1 = \frac{1}{|E|} \sum_j |p_j - e_j| \quad [1]$$

$$M_2 = \frac{1}{|R|} \sum_j |p_j - e_j| \quad [2]$$

où  $E$  : ensemble des évaluations de référence,  
 $R$  : ensemble des recommandations à analyser (cf. 4.3).

La mesure [1] prend pour référence de normalisation l'ensemble  $E$  des évaluations réellement faites par les utilisateurs pour lesquels on produit des recommandations. Mais dans le cas du jeu de données MovieLens, qui provient de l'utilisation réelle d'un système de filtrage collaboratif, rien ne garantit que  $E_{R\acute{e}elles}$  couvre l'ensemble des informations que les utilisateurs auraient évaluées positivement si l'intégralité des documents disponibles leur avait été présentée. Nous complétons donc cette mesure « absolue » par la mesure [2], qui normalise le nombre d'erreurs par le nombre de recommandations produites. Cette mesure diminue l'impact d'un mauvais rappel possible de  $E_{R\acute{e}elles}$ , et offre un bon complément, se prêtant bien à une évaluation comparative entre les différentes approches.

– *Corrélation de Pearson* (Breese *et al.*, 1998) : cette mesure est souvent utilisée dans le domaine du filtrage collaboratif pour le processus de prédiction de recommandations, mais on peut aussi l'utiliser pour estimer la corrélation entre les recommandations produites et les évaluations réelles :

$$M_3 = \frac{\sum_j (p_j - \bar{p})(e_j - \bar{e})}{\sqrt{\sum_j (p_j - \bar{p})^2 \sum_j (e_j - \bar{e})^2}} \quad [3]$$

où  $\bar{e}$  : moyenne des scores des évaluations faites par l'utilisateur,  
 $\bar{p}$  : moyenne des scores prédits dans les recommandations.

Plus les taux d'erreurs  $M_1$  et  $M_2$  diminuent, plus la qualité de l'ensemble de recommandations  $R$  augmente. Au contraire, plus  $M_3$  est grand, meilleure est la qualité des recommandations.

### 4.3. Résultats

Les résultats des mesures  $M_1$ ,  $M_2$  et  $M_3$  en fonction du nombre de meilleures recommandations (Top\_N) à présenter aux utilisateurs sont illustrés dans **Tableau 1**, **Tableau 2** et **Tableau 3**, où nous comparons chaque liste de recommandations aux données de référence  $E_{Réelles}$ . L'approche servant de base de comparaison ( $R_{Pearson}$ ) est sur la première ligne, et la meilleure approche est en italique.

Approche \ IRI	Top_5	Top_10	Top_15	Top_20	Moyenne
$R_{Pearson}$	1,76%	2,84%	4,09%	5,30%	<b>3,50%</b>
<i><math>R_{Genre}</math></i>	<i>1,38%</i>	<i>2,48%</i>	<i>4,09%</i>	<i>5,91%</i>	<b><i>3,47%</i></b>
$R_{Eval}$	3,32%	4,30%	4,95%	4,99%	<b>4,39%</b>
$R_{Demog}$	2,59%	4,15%	4,79%	6,51%	<b>4,51%</b>

**Tableau 1.** Résultats de la mesure d'erreur moyenne absolue ( $M_1$ ).

Approche \ IRI	Top_5	Top_10	Top_15	Top_20	Moyenne
$R_{Pearson}$	27,97%	24,02%	22,83%	22,45%	<b>24,32%</b>
<i><math>R_{Genre}</math></i>	<i>26,59%</i>	<i>24,96%</i>	<i>23,18%</i>	<i>22,44%</i>	<b><i>24,29%</i></b>
$R_{Eval}$	43,45%	29,54%	27,59%	27,45%	<b>32,01%</b>
$R_{Demog}$	83,32%	83,44%	79,74%	79,83%	<b>81,58%</b>

**Tableau 2.** Résultats de la mesure d'erreur moyenne absolue ( $M_2$ ).

Approche \ IRI	Top_5	Top_10	Top_15	Top_20	Moyenne
$R_{Pearson}$	28,95%	25,90%	24,02%	24,97%	<b>25,96%</b>
<i><math>R_{Genre}</math></i>	<i>29,92%</i>	<i>30,99%</i>	<i>31,53%</i>	<i>32,23%</i>	<b><i>31,17%</i></b>
$R_{Eval}$	31,39%	30,40%	32,13%	30,52%	<b>31,11%</b>
$R_{Demog}$	18,08%	19,45%	20,34%	19,20%	<b>19,27%</b>

**Tableau 3.** Résultats de la mesure de corrélation de Pearson ( $M_3$ ).

#### 4.4. Analyse

De façon générale, la liste de recommandations  $R_{\text{Genre}}$  domine les autres.

Logiquement, le **Tableau 1** et le **Tableau 2** donnent les mêmes résultats de comparaison entre les ensembles de recommandations à un facteur de  $|E_{R_{\text{réelles}}}| / |R|$  près selon l'ensemble de recommandations  $R$  en question.

La comparaison des taux d'erreurs entre  $R_{\text{Genre}}$  et les autres ensembles de recommandations montre l'utilité de notre approche comme une solution personnalisée pour le problème du démarrage à froid dans un système de filtrage. Néanmoins, ces bons résultats ne permettent pas de déterminer quelle part de ces performances sont dues à la production des premières recommandations, puisqu'elles dépendent de plusieurs facteurs comme la performance de la méthode de classification, la qualité du calibrage des paramètres utilisés dans le filtrage collaboratif par niveau d'accord, et l'hybridation du filtrage par des attributs démographiques.

Les résultats du **Tableau 3** confirment la qualité de l'ensemble  $R_{\text{Genre}}$  par rapport à l'ensemble  $R_{\text{Pearson}}$  et montrent toujours la mauvaise qualité de  $R_{\text{Démog}}$ . Ils rendent également compte d'une qualité relativement stable en fonction de la taille des ensembles de recommandations.

Par ailleurs, nous avons montré dans (Nguyen *et al.*, 2006), en analysant ces mêmes données, que l'attribut Genre peut être mieux prédit par les autres attributs, que l'attribut Evaluation. Ce résultat est ici confirmé en pratique.

Pour conclure, nous obtenons des résultats meilleurs que la technique classique prise pour base de comparaison, et du fait de l'absence d'intervention de l'utilisateur dans notre approche, nous concluons à un rapport coût/bénéfice bien meilleur.

#### 5. Conclusion

Nous avons défini une méthode qui s'appuie sur les données disponibles à froid pour offrir à tout nouvel utilisateur un premier vecteur de positionnement complet sans qu'il ait d'effort à fournir.

Nous avons montré que ce vecteur permet de générer des recommandations de meilleure qualité que celles produites suite à un processus de démarrage à froid classique, où l'utilisateur doit fournir un effort important (en l'occurrence des évaluations). Nous obtenons donc un rapport coût/bénéfice bien meilleur que celui de l'approche classique prise pour base de comparaison. Nous suggérons donc d'utiliser cette méthode comme préalable à un processus de démarrage à froid classique, afin de rendre ce dernier plus performant car mieux ciblé.

Cette première approche a validé expérimentalement le principe de correction automatique de vecteurs de positionnement. Bien évidemment, les résultats obtenus

nous encourage à approfondir la méthode, et notamment d'envisager une approche théorique approfondie.

Nous envisageons également de compléter ces évaluations en comparant et combinant notre approche à d'autres techniques de démarrage à froid plus élaborées. Nous aimerions aussi étudier l'impact de la taille de l'ensemble des données utilisées pour la phase d'apprentissage sur les performances, de façon à étudier la sensibilité de cette approche au problème de démarrage à froid de type « nouveau système ».

Par ailleurs, la méthode proposée peut s'appliquer dans d'autres cadres que celui du démarrage à froid. En effet, au cours de l'utilisation d'un système de recommandation, les utilisateurs rencontrent souvent des situations où leurs communautés sont périmées ou leur profil ne reflète plus correctement leur besoin, par exemple lors du changement d'intérêt (« concept drift »). On peut alors considérer que le profil, ou « vecteur de positionnement », n'est que partiellement fiable. L'approche présentée ici peut alors être appliquée pour corriger automatiquement les valeurs douteuses de ce vecteur afin de servir de point de départ à une évolution plus radicale du profil de l'utilisateur.

Enfin, cette approche, en s'appuyant sur une classification par règles, offre une capacité d'explication de ses processus qui pourrait être exploitée dans le cadre de l'interaction avec l'utilisateur, pour l'aider à se repositionner au sein des communautés lorsqu'il n'est pas satisfait des recommandations qu'il reçoit. Nous comptons explorer cette voie dans nos travaux futurs.

## 6. Bibliographie

- Bouzhoub M., Kostadinov D., « Personnalisation de l'information : Aperçu de l'état de l'art et définition d'un modèle flexible de profils », *Actes de la 2<sup>ème</sup> Conférence en Recherche d'Information et Applications (CORIA'05)*, France, 2005, p.201-218.
- Breese J. S., Heckerman D., Kadie C., « Empirical Analysis of Predictive Algorithms for Collaborative Filtering », *The 14<sup>th</sup> Conference on Uncertainty In Artificial Intelligence (UAI'98)*, Madison, Wisconsin, USA, 1998, p.43-52.
- Burke R., « Hybrid recommender systems: Survey and experiments », *User Modeling and User Adapted Interaction*, vol. 12, n° 4, 2002, p. 331-370.
- C5.0, Release 2.02, September 2005, <http://www.rulequest.com/see5-info.html>.
- CBA v2.1, June 2001, <http://www.comp.nus.edu.sg/~dm2/>.
- Claypool M., Gokhale A., Miranda T., « Combining Content-Based and Collaborative Filters », *Proceedings of the 22<sup>nd</sup> International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'99*, Berkeley, CA, USA, 1999.
- Good N., Schafer J. B., Konstan J. A., Borchers A., Sarwar B., Herlocker J., Riedl J., « Combining collaborative filtering with personal agents for better recommendations »,

*Proceedings of the Sixteenth National Conference on Artificial Intelligence*, Orlando, USA, 1999, p. 439- 446.

Krulwich B., « Lifestyle Finder: Intelligent user profiling using large-scale demographic data », *AI Magazine*, vol. 18, n° 2, 1997, p. 37-45.

Liu B., Ma Y., Wong C-K., « Improving an Association Rule Based Classifier », *Proceedings of the 4<sup>th</sup> European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2000)*, Lyon, France, September 13-16, 2000, p. 504-509.

Maltz D., Ehrlich E., « Pointing the way: Active collaborative filtering », *Proceedings of the SIGCHI conference on Human factors in computing systems, CHI'95*, Denver, Colorado, USA, 1995, p. 202-209.

Melville P., Mooney R. J., Nagarajan R., « Content-Boosted Collaborative Filtering for Improved Recommendations », *Proceedings of the Eighteenth National Conference on Artificial Intelligence, AAAI'02*, Edmonton, Canada, 2002, p. 187-192.

Middleton S. E., Alani H., Shadbolt N. R., De Roure D. C., « Exploiting Synergy Between Ontologies and Recommender Systems », *Proceedings of the 11<sup>th</sup> International World Wide Web Conference WWW-2002, International Workshop on the Semantic Web*, Hawaii, USA, 2002.

Middleton S. E., Shadbolt N. R., De Roure D. C., « Ontological user profiling in recommender systems », *ACM Transactions on Information Systems (TOIS)*, vol. 22, n° 1, ACM Press, 2004, p. 54-88.

Nguyen A.-T., Denos N., Berrut C., « Cartes de communautés pour l'adaptation interactive de profils dans un système de filtrage d'information », *Actes du 23<sup>ème</sup> Congrès annuel de l'Informatique des Organisations et Systèmes d'Information et de Décision, INFORSID'05*, Grenoble, 24-27 mai, 2005, p. 253-268.

Nguyen A.-T., Denos N., Berrut C., « Modèle d'espaces de communautés basé sur la théorie des ensembles d'approximation dans un système de filtrage hybride », *Actes de la 3<sup>ème</sup> Conférence en Recherche Information et Applications, CORIA'06*, Lyon, 2006.

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

Quinlan J. R., « C4.5 : Programs for Machine Learning », *Morgan Kaufmann*, San Mateo, USA, 1993.

Rashid A., Albert I., Cosley, D., Lam S. K., Mcnee S. M., Konstan J. A., Riedl J., « Getting to Know You: Learning New User Preferences in Recommender Systems », *Proceedings of the 7<sup>th</sup> International Conference on Intelligent User Interfaces, IUI'02*, San Francisco, California, USA, 2002, p. 127-134.

Schein A. I., Popescul A., Ungar L. H., « Generative Models for Cold-Start Recommendations », *Proceedings of the 2001 SIGIR Workshop on Recommender Systems*, USA, 2001.