# Apport des données thématiques dans les systèmes de recommandation : hybridation et démarrage à froid

Frank Meyer*, Eric Gaussier**
Fabrice Clerot*, Julien Schluth***

* Orange Labs, 2, Av. Pierre Marzin 22307 Lannion cedex
franck.meyer@orange-ftgroup.com, fabrice.clerot@orange-ftgroup.com
** Laboratoire d'Informatique de Grenoble BP 53 - 38041 Grenoble cedex 9
Eric.Gaussier@imag.fr
*** julien.schluth@gmail.com

**Résumé**. Des travaux récents (Pilaszy et al., 2009) suggèrent que les métadonnées sont quasiment inutiles pour les systèmes de recommandation, y compris en situation de cold-start : les données de logs de notation sont beaucoup plus informatives. Nous étudions, sur une base de référence de logs d'usages pour la recommandation automatique de DVD (Netflix), les performances de systèmes de recommandation basés sur des sources de données collaboratives, thématiques et hybrides en situation de démarrage à froid (cold-start). Nous exhibons des cas expérimentaux où les métadonnées apportent plus que les données de logs d'usage (collaboratives) pour la performance prédictive. Pour gérer le cold-start d'un système de recommandation, nous montrons que des approches "en cascade", thématiques puis hybrides, puis collaboratives, seraient plus appropriées.

## 1 Introduction

Le but des systèmes de recommandation automatique est d'aider des utilisateurs à trouver des produits (appelés items) qui les intéressent dans de très grands catalogues. On distingue en général deux types de technique, celle dite de filtrage collaboratif, et celle dite de filtrage thématique (Adomavicius & Tuzhilin, 2005). Lorsqu'un système de recommandation utilise à la fois des données d'usages des "utilisateurs" et des métadonnées des "items", on parle de système de recommandation hybride (Burke, 2007). Les systèmes de recommandation ont un point faible connu, celui dit du cold start : en l'absence de données d'usages, leurs performances sont détériorées (Adomavicius & Tuzhilin, 2005). Dès lors, utiliser des approches dites thématiques, c'est-à-dire basées sur des métadonnées descriptives des items, à la place des méthodes collaboratives, ou en renfort, est une idée récurrente. Les méthodes thématiques et hybrides ont été à notre connaissance peu étudiées sur les données de référence largement accessibles que constitue la base Netflix (Netflix Prize 2007). (Pilazy et al., 2009) abordent en partie ce sujet en utilisant des méthodes de factorisation de matrices appliquées à des données de Netflix aussi bien collaboratives (logs du Netflix Prize), que thématiques (site www.netflix.com). Mais leur étude aboutit à la conclusion que les métadonnées présentent peu d'intérêt. Nous étudions pour notre part l'apport des méthodes thématiques et hybrides selon une technique de type k-plus-proches voisins, qui est plus utilisée en contexte opérationnel (Koren, 2010), et selon un protocole dédié à l'analyse du cold start..

# 2 Notations et formules utilisées

## 2.1 Formules générales

Nous utiliserons les notations suivantes : u, v désignent des utilisateurs, i, j des items, $T_i$ l'ensemble des utilisateurs ayant noté l'item i, $S_u$ l'ensemble des items notés par u, $r_{ui}$ la note de l'utilisateur u pour l'item i, $\overline{r_i}$ la note moyenne de l'item i sur l'ensemble des logs de notation donnés, $\hat{r}_{ui}$ la note prédite, pour un utilisateur u et un item i. Il faut distinguer 2 types de matrice sous-jacentes : les logs de notations d'une part, représentables par une matrice item x utilisateur avec des notes, et les logs d'achats ou les métadonnées d'autres part, représentables par une matrice booléenne, qu'elle soit item x utilisateur ou item x caractéristique. Pour le cas où l'information est binaire, la formule de similarité de Jaccard peut être utilisée.

$$Jaccard(i,j) = \frac{\left|\{T_i \cap T_j\}\right|}{\left|\{T_i \cup T_j\}\right|}$$

$$\text{Pearson}(i,j) = \frac{\sum_{u \in T_i \cap T_j}\left(r_{ui} - \overline{r_i}\right)\left(r_{uj} - \overline{r_j}\right)}{\sqrt{\sum_{u \in T_i \cap T_j}\left(r_{ui} - \overline{r_i}\right)^2 \sum_{u \in T_i \cap T_j}\left(r_{uj} - \overline{r_j}\right)^2}}$$

$$\text{ExtendedPearson}(i,j) = \frac{\sum_{u \in T_i \cap T_j}\left(r_{ui} - \overline{r_i}\right)\left(r_{uj} - \overline{r_j}\right)}{\sqrt{\sum_{u \in T_i \cup T_j}\left(r_{ui} - \overline{r_i}\right)^2 \sum_{u \in T_i \cup T_j}\left(r_{uj} - \overline{r_j}\right)^2}}$$

$$\text{ExtendedMix}(i,j) = \text{jaccard}(i,j) \times \left(1 + \textit{ExtendedPearson}(i,j)\right)\big) / 2$$

$$\hat{r}_{ui} = \overline{r_i} + \frac{\sum_{\{j \in S_u\}} sim(i,j) \times \left(r_{uj} - \overline{r_j}\right)}{\sum_{\{j \in S_u\}} \left|sim(i,j)\right|}$$

*scoring « Mean-based »*

$$\hat{r}_{ui} = \frac{\sum_{\{j \in S_u\}} sim(i,j) \times \left(r_{uj}\right)}{\sum_{\{j \in S_u\}} \left|sim(i,j)\right|}$$

*scoring « Not Mean-Based »*

*Tab. 1 Formules utilisées*

Pour le cas des logs de notations, la formule de similarité entre 2 items la plus courante est la formule de Pearson (Adomavicius, 2005). Cette formule souffre d'un grand biais : 2 items qui n'ont qu'un utilisateur en commun, qui a noté ces 2 items de manière identiques, vont avoir une similarité maximale. Pour corriger ce biais nous introduisons la similarité « ExtentedPearson ». Son effet est de prendre en compte, dans le dénominateur, l'intégralité des notes sur les items. Notons qu'une combinaison des similarités de Pearson et de Jaccard permet de traiter à la fois des logs de notations et des logs d'achats et a été proposée et évaluée dans (Candillier et al. 2008) : la similarité « ExtendedMix ». est adaptée à des contextes opérationnels où on ne saurait pas à l'avance la nature des logs à traiter.

Nous utilisons de plus deux formules standard de prédiction de notes, dites de « scoring », fondées sur les éléments introduits précédemment, sim(i,j) désignant une fonction de similarité. La formule « Mean-Based » suppose la connaissance des notes moyennes sur les items des autres utilisateurs, ce qui n'est pas forcément le cas, par exemple pour un moteur de recommandation personnel embarqué sur un terminal mobile sans accès aux autres utilisateurs. Dans le cas d'un système mono-utilisateur on utilise la formule « Not-Mean-Based » : seules les données de l'utilisateur courant sont utilisées.

### 2.2 Mesures effectuées

La Root Mean Squared Error est une mesure bien connue pour évaluer un modèle de prédiction de notes. Si on considère un ensemble de notes à prédire en Test, noté $R_T$, que l'on veut comparer avec un ensemble de notes prédites $R_P$, alors la RMSE est définie par :

$$rmse(R_T, R_P) = \sqrt{\frac{1}{|R_T|} \sum_{r_{u,j} \in R_T} \left( r_{u,i} - \hat{r}_{u,i} \right)^2}$$

avec

| | |
|---|---|
| $r_{u,i}$ | note de l'utilisateur u pour l'item i connue en test |
| $\hat{r}_{u,i}$ | note de l'utilisateur u pour l'item i, prédite par le modèle |
| $\lvert R_T \rvert$ | taille de l'ensemble de test (nombre de notes à prédire). |

## 3 Protocole

### 3.1 Modélisation utilisée

Une méthode très utilisée dans le domaine du filtrage collaboratif est le calcul de la matrice de similarité item-item (appelé aussi matrice item-item) : (Sarwar et al., 2001). Les modèles de recommandation à base de matrice item-item sont actuellement ceux qui procurent le maximum d'avantages en terme de couverture fonctionnelle, de tenue de charge, de transparence, de réactivité aux changements de profils. Ils restent par ailleurs, dans le domaine opérationnel, très compétitifs en performance prédictive par rapport aux modèles de factorisation de matrices. (Sarwar et al., 2001), (Koren, 2010). Nos modèles utilisent donc cette approche K-plus-proches voisins de type item-item appliquée soit à des vecteurs de notations (profils utilisateurs) soit à des vecteurs booléens (items représentés par des métadonnées). K est égal à 200 dans tous nos tests, ce qui constitue un bon compromis performance / tenue de charge (Bell & al, 2007), (Koren, 2010). En conséquence pour chaque item i, sim(i,j) ne sera définie que pour les k=200 plus-proches-voisins de l'item i, et donnera 0 dans les autres cas.

### 3.2 Données utilisées : Netflix + IMDB

Pour la partie collaborative nous avons utilisé les logs de Netflix (NetFlix Prize 2007). Pour la partie filtrage thématique nous avons utilisé des données structurées de la base Internet IMDB (www.imdb.com). Chaque couple (attribut, valeur) d'un film de la base IMDB est considéré comme une information booléenne, appelée descripteur qui peut être lié à un item de Netflix. Par exemple l'item (Star Wars I) aura les descripteurs (Genre, Science-Fiction), (Director, Georges Lucas), (Actor, Harrison Ford), (Actor, Carrie Fisher), etc. Les données d'IMDB ont été jointes automatiquement avec celles de Netflix en utilisant le croisement sur le titre et l'année. Le taux de succès de la jointure automatique a été estimé par sondage à 94%. Les problèmes viennent majoritairement des documentaires sportifs ou animaliers qui n'ont pas été référencés dans IMDB mais qui étaient disponibles à la location dans la base Netflix.

## 3.3 Processus général

Le protocole de test à chaque étape de mesure est le suivant :

1. Sélectionner N utilisateurs au hasard dans Netfix
2. Sélectionner tous les logs $r_{u,i}$ de ces N utilisateurs (tous les items notés par ces utilisateurs)
3. Séparer les logs obtenus en 2 parties, Apprentissage et Test (un utilisateur u aura donc un profil $u_A$ en apprentissage un profil $u_T$ en test)
4. Pour chaque type de méthode construire un modèle de matrice de similarité item-item sur la base des items connus en apprentissage uniquement
    a) Collaboratif : utiliser les logs en apprentissage pour calculer les similarités, puis utiliser une méthode de scoring Mean-Based
    b) Thématique : utiliser les métadonnées IMDB pour calculer les similarités, puis utiliser une méthode de scoring Not-Mean-Based
    c) Hybride : utiliser les métadonnées IMDB pour calculer les similarités, puis utiliser une méthode de scoring Mean-Based
5. La RMSE est enfin calculée sur les parties en ensemble de Test de chaque profil utilisateur.

## 3.4 Simulation du Cold-Start

Nous faisons varier le nombre d'utilisateur de manière logarithmique. Nous nous plaçons dans 2 contextes :

1. profils "longs", où nous utilisons 90% de chaque profil utilisateur en apprentissage, ce qui correspond en moyenne à des profils utilisateur de 180 items notés (le reste du profil étant en Test et servant à évaluer la RMSE).
2. profils "courts", où nous utilisons 10% de chaque profil utilisateurs en Apprentissage, soit en moyenne que 20 items notés (les 90% restant étant en Test).

Les résultats sont donnés dans les 2 graphiques suivants FIG. 1 et FIG. 2.

Notons que dans tous les cas de figure, le système fait parfois appel en cascade à des prédicteurs par défaut (moyenne item ou moyenne utilisateur ou (moyenne item + moyenne utilisateur)/2 si l'information est disponible), dont la performance dépend de la taille des logs. Ceci explique la légère variation de la performance du système thématique.

Sur des profils longs, en cold-start, un recommandeur thématique semble intéressant jusqu'à environ 100 utilisateurs différents. Au delà, le système collaboratif est plus avantageux. Le système hybride ne parait pas se différencier réellement du système collaboratif. Dès lors, sur les profils longs, nous pouvons rejoindre les conclusions de (Pilaszy et al., 2009) : sur des grosses masses de données, la technique purement collaborative domine les autres. Dans le cas de profils courts en revanche, les choses ne sont pas si simples. Le filtrage thématique est globalement le plus intéressant jusqu'à environ 1000 utilisateurs. Au delà l'hybridation légère devient plus intéressante, jusqu'à environ 100000 utilisateurs où le filtrage collaboratif redevient compétitif.

En conclusion, si les méthodes collaboratives semblent indépassables sur des grands volumes de données et dans le cas où les profils des utilisateurs sont de grande taille (autour de 200), elles sont par contre en difficulté lorsque les utilisateurs sont moins bien représentés, par des profils de petite taille, de l'ordre de 20 notes. Or les situations de cold-start les plus réalistes sont justement celles où les utilisateurs n'ont pas encore beaucoup

rempli leur profil. Dans ce cas, on note qu'un moteur thématique est intéressant au démarrage d'un service, jusqu'à environ 1000 utilisateurs. Au delà, à partir de 1000 à 2000 utilisateurs, l'hybridation devient intéressante.

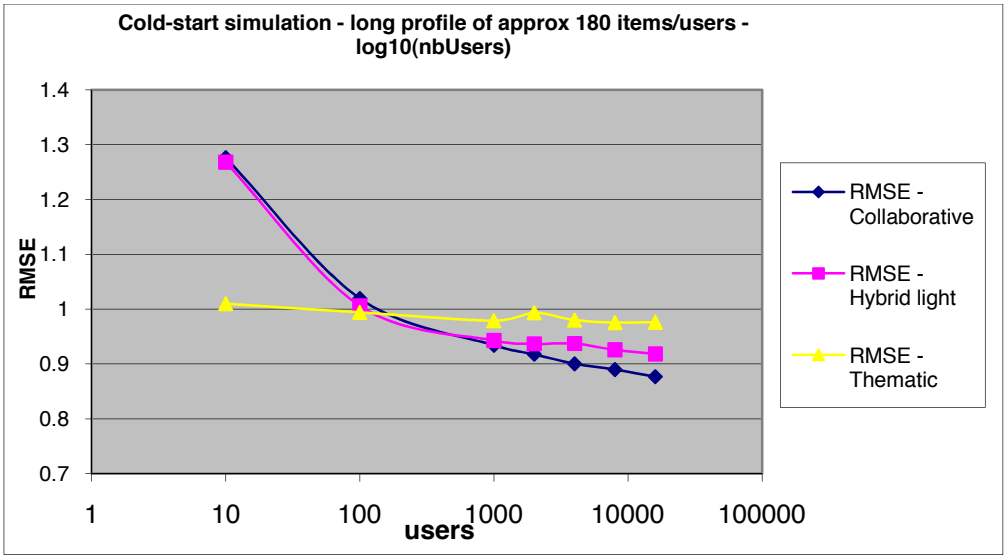


*FIG. 1 -Simulation d'un cold-start sur des profils longs*

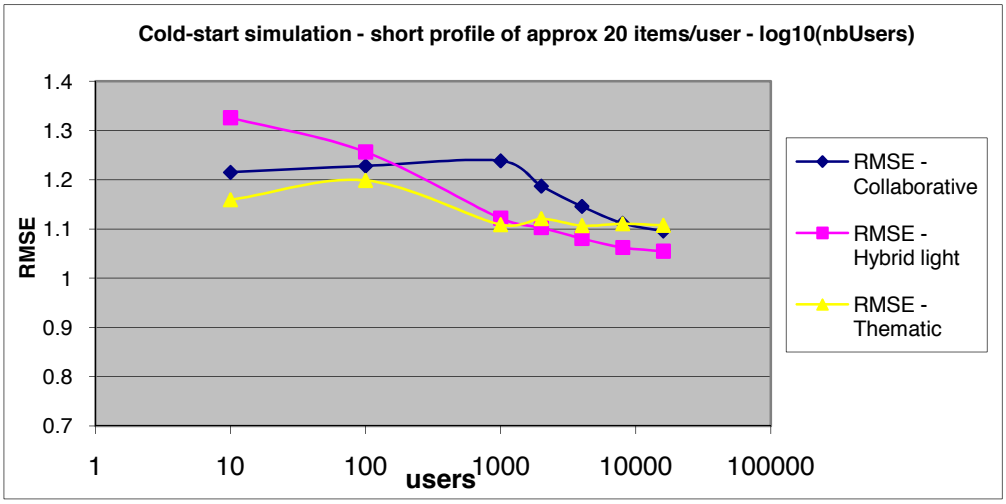


*FIG. 2 - Simulation d'un cold-start sur des profils courts*

# 4 Conclusion et perspectives

Nous avons étudié les performances des systèmes de recommandation basés sur des techniques de matrice de similarité item-item sur des données de référence (Netflix) et selon différents protocoles de démarrage à froid (cold-start).

Les systèmes thématiques et hybrides sur des techniques de recommandation basées sur ces matrices item-item présentent une utilité pour le cold-start. Ces systèmes sont intéressants quand les données d'usages sur les utilisateurs sont peu nombreuses. Les méthodes thématiques et hybrides sont d'autant plus utiles que les profils de l'utilisateur sont courts. Il serait maintenant intéressant de travailler, en reprenant la classification de (Burke, 2007), sur un système d'hybridation par "Switch" dynamique, qui utiliserait un système thématique, hybride ou collaboratif selon la taille ou d'autres caractéristiques particulières des profils.

# Références

(Adomavicius & Tuzhilin, 2005) Adomavicius, G., and 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*, 17(6), 734–749.

(Bell et al, 2007) Bell, R. and Koren, Y. Improved Neighborhood-based Collaborative Filtering, *KDD-Cup and Workshop*, ACM press, 2007.

(Burke 2007) Burke, R. Hybrid Web Recommender Systems. The Adaptive Web 2007.

(Candillier et al. 2008) Candillier, L. , Meyer, F. , Fessant, F. (2008). Designing Specific Weighted Similarity Measures to Improve Collaborative Filtering Systems. ICDM 2008: 242-255

(Koren, 2010) Koren, Y. (2010). Factors in the neighbors: Scalable and accurate collaborative filtering. TKDD, 4 (1), 2010.

(Netflix Prize 2007) Bennet, J.; and Lanning, S., “The Netflix Prize”, KDD Cup and Workshop, 2007. www.netflixprize.com.

(Pilaszy et al., 2009) Pilászy, I. Tikk, D (2009). Recommending new movies: even a few ratings are more valuable than metadata. RecSys 2009: 93-100

(Sarwar et al., 2001) Sarwar, B. M., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In 10th International W3 Conference.

# Summary

We study, with the reference database “Netflix”, the performance of a recommender system during cold-starts, using different data sources and methods: collaborative, thematic and hybrid. To manage the cold-start problem of a recommender system, we show that a "cascade" approach, using a thematic method then a hybrid method and at last a collaborative one, would be more appropriate.