Inspiration des sondages d'opinion pour réduire la latence en filtrage collaboratif.

Armelle Brun, Anne Boyer

Inspiration des sondages d'opinion pour réduire la latence en filtrage collaboratif

Armelle Brun et Anne Boyer

LORIA - Nancy Université
615, rue du jardin botanique - 54506 Vandœuvre lès Nancy
{Armelle.Brun, Anne.Boyer}@loria.fr

RÉSUMÉ. *Le filtrage collaboratif est l'une des approches les plus populaires des systèmes de recommandation. En filtrage collaboratif, le système cherche à estimer les préférences de l'utilisateur actif en exploitant les préférences (les notes) des utilisateurs similaires à cet utilisateur actif : ses voisins. Le filtrage collaboratif fait face au problème de latence : il ne peut recommander un nouvel item à des utilisateurs tant que cet item n'a pas été noté un nombre suffisant de fois. Pour diminuer ce problème de latence, nous proposons une approche à base de mentors s'inspirant des sondages d'opinion. Un mentor est un utilisateur fiable et représentatif, sur qui l'on peut compter. La connaissance des notes (opinions) des mentors sur les nouveaux items permet d'estimer les notes de la population entière. Nous montrons que, sur le corpus que nous avons utilisé, certaines méthodes de sélection des mentors requièrent les notes de seulement 6 mentors pour estimer les notes de la population entière, avec une erreur faible. Ainsi, lorsqu'un nouvel item est intégré dans le système, seules 6 notes sont requises pour faire des recommandations de bonne qualité à l'ensemble de la population.*

ABSTRACT. *Collaborative filtering is one of the most popular approaches in recommender systems. In collaborative filtering, the system exploits the ratings of the active user's like-minded users: his neighbors, to estimate his ratings on the items he has not rated yet and recommend him the items with the highest estimated ratings. Collaborative filtering faces a latency problem: a new item cannot be recommended to any user while this item has not been rated a sufficiently high number of times. To alleviate this latency problem, we propose a new approach based on mentors, inspired from opinion polls. A mentor is a reliable, representative and trusted user. The knowledge of the ratings of the mentors on new items allows to estimate the ratings of the whole population. We show that, on the corpus tested, when using some of the mentor selection methods, only 6 mentors are sufficient, to reliably estimate the ratings of the whole population.*

MOTS-CLÉS : *Systèmes de recommandation, Filtrage collaboratif, Sondages d'opinion.*

KEYWORDS: *Recommender systems, Collaborative filtering, Opinion polls.*

1. Introduction

La démocratisation d'Internet s'est accompagnée d'une croissance de la quantité d'informations accessibles par les utilisateurs, qui sont désormais incapables de gérer cette énorme masse d'informations et sont souvent insatisfaits. Les systèmes de recommandation (Goldberg *et al.,* 1992, Adomavicius *et al.,* 2005) ont pour but de pallier ce problème de surcharge d'informations en suggérant aux utilisateurs des items en lien avec leurs attentes et leurs goûts (Adomavicius *et al.,* 2005). L'approche la plus populaire des sytèmes de recommandation est le filtrage collaboratif (FC) (Goldberg *et al.,* 1992). En FC aucune information *a priori* sur les items et sur les utilisateurs n'est connue ; seuls leurs identifiants et les appréciations des utilisateurs sur les items, sous forme de note, sont connus. Pour déterminer quel(s) item(s) recommander à un utilisateur, dit utilisateur actif, le FC utilise les préférences des utilisateurs ayant des préférences similaires à lui : ses voisins. Malgré son succès dû à la qualité des recommandations qu'il fournit, le FC a l'inconvénient de ne pouvoir recommander de nouveaux items. En effet, pour qu'un nouvel item puisse être recommandé à l'utilisateur actif, il doit avoir été noté par un certain nombre de voisins de cet utilisateur. On parle de problème de latence.

Dans cet article, nous nous intéressons à la réduction du temps de latence d'un système de FC : nous cherchons à diminuer le nombre de notes requises pour permettre la recommandation de nouveaux items. Nous supposons que si un sous-ensemble fiable et représentatif d'utilisateurs a noté un nouvel item, alors ils pourront recommander cet item à l'ensemble des autres utilisateurs. Le problème de latence sera alors réduit. Nous cherchons donc à repérer l'ensemble des utilisateurs dont il faut connaître la note sur un nouvel item donné : les mentors. Un mentor doit donc non seulement être représentatif d'un sous-ensemble d'utilisateurs, il doit également fournir des recommandations fiables. La recherche des mentors en FC et les sondages d'opinion ont le même objectif : chercher le sous-ensemble de personnes à sonder dans une population, afin de déduire l'opinion de la population entière. Dans cet article, nous cherchons donc à déterminer de façon adéquate les mentors de la population : les personnes qu'il faut sonder afin de déduire de façon la plus fiable possible l'avis de la population entière. Comparativement à l'approche classique du FC, les mentors seront des "recommendeurs" plus fiables que les voisins. De plus, chaque mentor recommandera son opinion à de nombreux utilisateurs alors qu'un voisin peut être voisin d'un petit nombre d'utilisateurs. Ainsi, un utilisateur actif ne recevra des recommandations que d'un nombre restreint d'utilisateurs, mais ces utilisateurs seront fiables.

Dans la seconde section nous présentons le FC et tout particulièrement la façon dont la sélection de voisins est classiquement effectuée. La troisième section décrit les techniques standard d'échantillonnage et de sondage d'opinion. La quatrième section présente la façon dont nous proposons de nous inspirer des techniques de sondage d'opinion pour rechercher les mentors. La section 5 valide expérimentalement l'approche proposée. Enfin, nous concluons et présentons des perspectives à ce travail.

2. Le filtrage collaboratif

Un système de filtrage collaboratif (FC) (Goldberg *et al.*, 1992), exploite les notes $r(u, i)$ des utilisateurs u sur des items i. Soit a l'utilisateur actif à qui le système va recommander des items. Pour estimer quels items doivent être présentés à a, un système de FC estime la note $r^*(a, i)$ que a affecterait à chaque item i qu'il n'a pas noté, puis le système lui recommande les items ayant les notes estimées les plus élevées. Pour estimer $r^*(a, i)$, le système exploite les notes des utilisateurs u' ayant noté l'item i et ayant des préférences similaires à a : les voisins de a (Candillier *et al.*, 2007). Plus un utilisateur est similaire à a, plus son poids dans l'estimation de $r^*(a, i)$ est élevé.

Il existe deux approches principales de la sélection de voisins : la sélection directe de voisins sélectionne les K utilisateurs ayant la similarité la plus élevée avec a (Herlocker *et al.*, 1999). Bien que ne passant pas à l'échelle, dans cette approche l'ensemble des voisins est centré sur l'utilisateur a et les recommandations sont de qualité. La sélection de voisins basée sur une classification effectue un regroupement des utilisateurs (Castagnos *et al.*, 2006) sur leur similarité aves a. Les voisins de a sont les utilisateurs appartenant au même groupe que a. Cette approche a l'avantage de passer à l'échelle mais les communautés ne sont pas centrées sur l'utilisateur actif et les performances peuvent en être diminuées.

3. Sondages d'opinion et échantillonnage

Les sondages d'opinion on pour but de connaître l'opinion d'une population entière ; un sous-ensemble de la population : un échantillon, est interrogé et l'objectif est d'obtenir un échantillon de personnes dont l'opinion est exactement la même que celle que l'on aurait obtenue si l'on avait interrogé l'ensemble de la population. Plusieurs méthodes d'échantillonnage sont classiquement utilisées : le sondage aléatoire simple, le sondage par strate, le sondage par cluster, le sondage proportionnel à la taille, etc. (Groves *et al.*, 2009). Nous présentons ici la méthode la plus populaire d'échantillonnage : le sondage par strate. Le sondage par strate segmente la population en groupes mutuellement exclusifs, également appelés strates. Le critère de regroupement est une informaton sur les membres de la population : âge, sexe, profession, etc. Un échantillon est extrait de chaque strate : des personnes sont sélectionnées aléatoirement, chaque strate est alors réprésentée. L'échantillonnage par strate mène à des estimations de qualité. Cependant, cette approche nécessite de disposer d'informations sur les membres de la population, ce qui peut s'avérer difficile dans certains cas.

4. Inspiration des sondages d'opinion pour la réduction de la latence

Pour atténuer le problème de la latence du FC, nous proposons d'exploiter un sous-ensemble d'utilisateurs, appelés mentors. Un mentor étant souvent défini comme un conseiller expérimenté, alors si le système dispose de l'opinion de mentors sur un nouvel item, alors ces mentors pourront le "conseiller" à un ensemble d'utilisateurs.

Dans notre approche, nous voyons les mentors comme des utilisateurs à qui le système demandera de noter les nouveaux items. Le problème du choix des mentors, de façon à réduire au maximum le temps de latence n'a, à notre connaissance, jamais été étudié dans la littérature. Pour résoudre le problème de la recherche de mentors nous proposons de nous inspirer des sondages d'opinion. En FC, ces personnes sondées seront les mentors. Nous proposons d'utiliser le sondage par strate. Un regroupement des utilisateurs en strates est effectué en utilisant un algorithme de clustering.
Nous recherchons le mentor au sein de chaque strate. Lorsque l'opinion d'un mentor est connue (dans notre cas la note qu'il donne à un item) alors les notes sur cet item, des utilisateurs appartenant à la même strate que ce mentor seront déduites.
En FC, l'opinion des voisins est pondérée par leur similarité avec l'utilisateur actif. Nous proposons que l'opinion d'un utilisateur soit égale à celle de son mentor.

En sondage d'opinion, le choix des personnes à interroger dans chaque groupe est aléatoire. En FC, le système dispose des notes des utilisateurs sur les items et nous proposons d'exploiter cette information pour aider à la sélection des utilisateurs à sonder au sein des strates : les utilisateurs représentatifs de leur groupe.

L'ensemble des utilisateurs peut être vu sous la forme d'un réseau. Dans le cadre de l'analyse des réseaux et tout particulièrement des réseaux sociaux (Wasserman *et al.*, 1994), de nombreuses mesures comme la centralité et l'influence ont été proposées. Nous choisissons de les exploiter dans le but de sélectionner les mentors. La centralité est une mesure générale qui représente la position d'un nœud dans un graphe. Deux mesures de centralité sont classiquement utilisées dans la litérature (Freeman, 1978, Wasserman *et al.*, 1994) : la centralité de degré et la centralité de proximité ou de distance. **La centralité de degré** considère les utilisateurs connectés au plus grand nombre d'utilisateurs au sein d'un groupe comme des utilisateurs représentatifs du groupe. **La centralité de proximité ou de distance** considère les nœuds avec une faible distance avec la plupart des autres nœuds comme des nœuds de centralité de distance élevée. Précisons ici que la notion de centralité a été rarement utilisée en FC. Nous pouvons par exemple citer (Cantador *et al.*, 2009) qui utilise la centralité des utilisateurs dans le but de recommander des annotations.

L'influence d'un nœud dans un réseau peut également être utilisée pour déterminer la représentativité de ce nœud. (Agarwal *et al.*, 2008) considère que l'influence d'un utilisateur peut être reflétée par son **activité**. Nous proposons donc de définir les utilisateurs les plus actifs d'un groupe comme étant les mentors du groupe.

5. Expérimentations

5.1. *Corpus et Evaluation*

Nous avons travaillé sur la base de l'état de l'art MovieLens[1], composée de préférences utilisateurs sur des films. Ces préférences sont des notes entre 1 et 5. La

1. http ://www.movielens.org

base recense 1682 utilisateurs, 943 items et 100k préférences. Nous avons divisé l'ensemble des items en deux ensembles : les items d'apprentissage, représentant 80% des items et les 20% restant sont les items de test : les nouveaux items.

L'apprentissage des classes et le choix des mentors se fait sur les items et les notes du corpus d'apprentissage. Une fois les mentors choisis, les notes de ces derniers sont extraites du corpus de test (pour simuler le fait qu'ils ont noté de nouveaux items) et exploitées afin de déduire les notes des utilisateurs sur ces mêmes items de test. La qualité des déductions, et donc des notes estimées, est évaluée à l'aide de la mesure MAE (Mean Absolute Error) qui calcule l'erreur moyenne sur les notes estimées par le système de recommandation, on cherche donc à minimiser la MAE.

Nous avons testé deux algorithmes de partitionnement des utilisateurs. Le premier algorithme est l'algorithme k-means (ou k-moyennes) (McQueen, 1967) qui construit des classes minimisant la distance euclidienne entre le centre de gravité (centroïde) et les utilisateurs des classes. Il est très souvent utilisé en FC pour construire des classes d'utilisateurs. Le second algorithme est l'algorithme PAM (ou k-médoïdes) (Kaufman *et al.*, 1987) qui exploite les médoïdes des classes d'utilisateurs : le centre de chaque classe est représenté par un utilisateur réel.

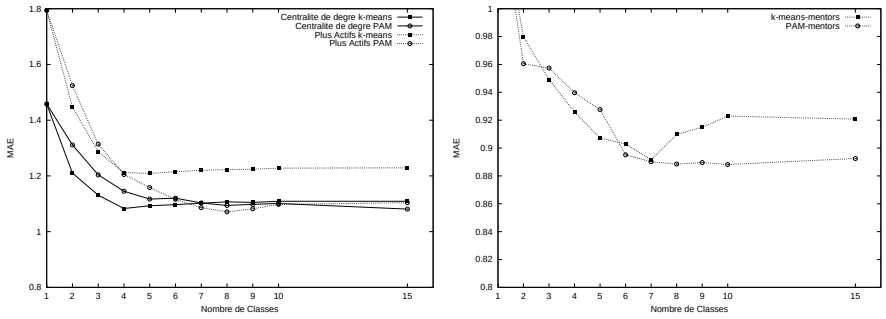
Ces deux algorithmes requièrent le nombre de classes en entrée, nous ne connaissons pas *a priori* ce nombre, alors nous conduirons nos expérimentations sur un nombre de classes variant de 1 à 15. Notons que les algorithmes k-means et PAM sont sensibles aux valeurs d'initialisation (choisies aléatoirement), les valeurs présentées dans les expérimentations ci-dessous sont des valeurs moyennes sur 20 exécutions.

5.2. *Mentors de plus grande centralité de degré*

Ici le mentor d'une classe est l'utilisateur avec la centralité de degré la plus élevée : celui connecté au plus grand nombre d'utilisateurs dans la classe. 2 utilisateurs sont connectés s'ils ont covoté un nombre minimum de films, ici 5. Le graphique gauche de la Figure 1 présente les valeurs de MAE correspondantes. Les valeurs de MAE sont très proches pour k-means et PAM, et l'optimum est atteint avec un nombre de respectivement 5 et 6 classes, la MAE est de $1,07$ et $1,02$.

Notons dès maintenant que nous ne pouvons pas comparer directement ces MAE avec des valeurs de l'état de l'art. En effet, bien que les expérimentations soient conduites sur un corpus de l'état de l'art, notre approche est, à notre connaissance, le premier travail qui s'intéresse à la maîtrise de la latence sur ce corpus. Cependant, une MAE de référence peut être obtenue en affectant pour chaque couple (utilisateur, item) du corpus de test, la note moyenne de cet utilisateur sur le corpus d'apprentissage (Candillier *et al.*, 2007). Sur ce corpus, la MAE de référence obtenue est $0,94$. Nous pouvons donc conclure que l'exploitation de la centralité de degré ne permet pas d'obtenir des estimations de grande qualité.

Figure 1. *MAE avec mentors sélectionnés selon la centralité de degré ou l'activité.*



5.3. *Mentor de plus grande centralité de distance*

Dans cette section, l'utilisateur ayant la centralité de distance la plus élevée (distance moyenne la plus faible) avec les autres utilisateurs de sa classe est considéré comme un mentor. Nous pouvons noter que des utilisateurs très proches d'un petit nombre d'utilisateurs pourront être choisis comme mentors de leur classe car leur centralité de distance sera très élevée, bien qu'ils soient connectés à peu d'autres utilisateurs. Le graphique droit de la Figure 1 présente les MAE correspondantes en fonction de l'algorithme de partitionnement.

La MAE la plus faible pour les algorithmes k-means et PAM est obtenue avec respectivement 7 et 8 classes. Le nombre optimal de 7 classes est supérieur à celui obtenu avec la sélection par connectivité, mais la MAE obtenue est de meilleure qualité : $0,88$, ce qui correspond à une diminution de 14%. De plus, cette MAE est significativement inférieure à la MAE de référence : une baisse de plus de 6% est obtenue.

K-means mène à des performances moins régulières que l'algorithme PAM. Ceci est dû au fait que k-means construit des classes non pas à partir d'un utilisateur réel mais à partir de l'iso-barycentre des classes qui peut être très différent de l'utilisateur central ; les performances du système peuvent donc en être impactées.

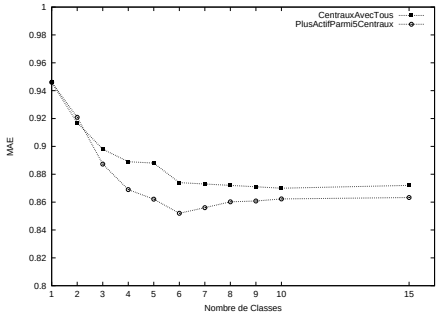
Au vu des résultats des 3 expériences précédentes, nous pouvons dire que PAM mène à une MAE soit plus faible, soit plus stable. Par conséquent, dans l'expérience qui suit, nous exploitons uniquement l'algorithme PAM.

5.4. *Utilisateurs centraux actifs*

Ici nous choisissons les mentors comme étant les utilisateurs à la fois centraux mais également actifs (Chang, 2010). Pour cela, nous exploitons 2 mesures. Dans la première mesure nous considérons, pour chaque classe, les n utilisateurs les plus au centre. Le mentor choisi est le plus actif parmi ces n utilisateurs. Dans les expéri-

mentations qui suivent, n sera fixé à 5. La seconde mesure est une adaptation de la mesure de centralité de la section 5.3. La centralité d'un utilisateur est sa distance moyenne avec tous les utilisateurs de la classe. Lorsque deux utilisateurs n'ont pas covoté suffisamment d'items, une valeur maximale est affectée. Un utilisateur très proche d'un petit nombre d'utilisateurs, mais avec une grande distance avec les autres aura une centralité de distance faible (contrairement à la première mesure de centralité). À l'opposé, un utilisateur ayant une distance moins faible, mais avec un grand nombre d'utilisateurs aura une centralité de distance plus élevée, et sera donc considéré comme mentor. La Figure 2 présente les MAE correspondant à ces deux mesures.

Figure 2. *MAE avec mentors utilisateurs centraux actifs*



Les MAE obtenues avec les deux mesures sont légèrement inférieures comparativement à l'exploitation de l'utilisateur central, qui avait permis d'obtenir les meilleures MAE jusqu'à présent. C'est la première mesure qui mène à la MAE la plus faible : $0,85$ pour 6 classes, nombre légèrement plus faible que le nombre optimal de classes de l'expérience précédente.

En conclusion, sur ce corpus, la création de 6 classes et l'exploitation de l'utilisateur le plus actif parmi les utilisateurs les plus centraux mène à une diminution de près de 10% par rapport à la MAE de référence. Ainsi, lors de l'ajout d'un nouvel item dans le système, l'opinion de 6 utilisateurs permet d'obtenir des estimations relativement fiables de l'opinion de l'ensemble des utilisateurs du système.

6. Conclusion

Dans cet article nous nous sommes intéressés à la réduction du temps de latence du filtrage collaboratif. Nous adoptons une approche à base de mentors et nous nous inspirons des sondages d'opinion. Un ensemble d'utilisateurs, appelés mentors, est considéré comme l'ensemble des utilisateurs dont l'opinion est représentative d'un sous-ensemble de la population. Lorsqu'un nouvel item est intégré au système, celui-ci demande en priorité aux mentors de noter cet item, ainsi l'opinion de l'ensemble de la population sur cet item peut être déduite. Nous avons montré que, sur le corpus

choisi, l'exploitation de 6 mentors permet d'obtenir la MAE la plus faible. Ces 6 mentors étant choisis comme étant les utilisateurs centraux, tout en étant actifs.
Nous envisageons d'implanter cette approche sur d'autres corpus de données et d'étudier l'évolution des mentors au cours du temps afin de déterminer si un utilisateur conserve son statut de mentor au au fur et à mesure de l'évolution de la base de notes.

7. Bibliographie

Adomavicius G., Tuzhilin A., « Toward the Next Generation of Recommender Systems : A Survey of the State-of-the-Art », *IEEE Tran. on Knowledge and Data Engineering*, vol. 17, n° 6, p. 734-749, 2005.

Agarwal N., Liu H., Tang L., Yu P., « Identifying the Influential Bloggers in a Community », *Proc. of the Inter. Conf. on Web Search and Web Data Mining*, p. 207-218, 2008.

Brun A., Castagnos S., Boyer A., « A positively directed mutual information measure for collaborative filtering », *2nd Inter. Conf. on Information Systems and Economic Intelligence*, p. 943-958, 2009.

Burke R., « Hybrid Recommender Systems : Survey and Experiments », *User Modeling and User-Adapted Interaction*, vol. 12, n° 4, p. 331-370, 2002.

Candillier L., Meyer F., Boullé M., « Comparing State-of-the-Art Collaborative Filtering Systems », *Proc. of 5th Inter. Conf. on Machine Learning and Data Mining in Pattern Recognition*, p. 548-562, 2007.

Cantador I., Vallet D., Jose J., « Measuring Vertex Centrality in Co-occurrence Graphs for Online Social Tag Recommendation », *9th Europ. Conf. on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2009.

Castagnos S., Boyer A., « A Client/Server User-Based Collaborative Filtering Algorithm : Model and Implementation », *Proc. of the 17th Europ. Conf. on Artificial Intelligence (ECAI 2006)*, p. 617-621, 2006.

Chang E., « AdHEat - A New Influence-based Social Ads Model and its Tera-Scale Algorithms », *Proc. of the Workshop on Algorithms for Modern Massive Data Sets*, 2010.

Freeman L., « Centrality in Social Networks. Conceptual Clarification », *Social Networks*p. 215-239, 1978.

Goldberg D., Nichols D., Oki B., Terry D., « Using collaborative filtering to weave an information tapestry », *Communications of the ACM*, vol. 35, n° 12, p. 61-70, 1992.

Groves R. M., Fowler F., Couper M., Lepkowski J., Singer E., Tourangeau R., *Survey Methodology*, vol. 2nd edition, Wiley, 2009.

Herlocker J., Konstan J., Borchers A., Riedl J., « An algorithmic framework for performing collaborative filtering », *Proc. of the SIGIR conference*, p. 230-237, 1999.

Kaufman L., Rousseeuw P. J., *Statistical Data Analysis Based on the L1Norm*, North Holland/Elsevier, chapter Clustering by means of medoids, p. 405-416, 1987.

McQueen J., « Some methods for classification and analysis of multivariate observations », *Proc. of the 5th Symposium on Math, Statistics and Probability*, p. 281-297, 1967.

Wasserman S., Faust K., *Social Network Analysis : Methods and Applications*, Cambridge University Press, 1994.