

Recommandation Hybride basée sur l'Apprentissage Profond

Lamia Berkani*, Imene Kerboua**
Sofiane Zeghoud***

*Laboratoire LRIA, Département Informatique., USTHB, Bab Ezzouar,
Alger, Algérie

`lberkani@usthb.dz`

**Institut de la Communication, Université Lumière Lyon 2,
Lyon, France

`imene.kerboua@univ-lyon2.fr`

*** Department of Computer Science and Engineering,
School of Electronic Information and Electrical Engineering,
Shanghai Jiao Tong University,
Shanghai, Chine

`sozeghoud@outlook.com`

Résumé. Cet article explore l'utilisation des réseaux de neurones profonds pour l'apprentissage de la fonction d'interaction à partir des données. Nous proposons une approche de recommandation hybride, qui combine le filtrage collaboratif (FC) et le filtrage à base de contenu (FBC) selon une architecture basée sur les deux modèles: la factorisation matricielle généralisée et le perceptron multicouches. Des expérimentations approfondies sur la base MovieLens-1M montrent des améliorations significatives de notre approche par rapport aux méthodes existantes en particulier pour la situation de démarrage à froid.

1 Introduction

Parmi les différentes techniques de filtrage collaboratif (FC) (Resnick et al., 1994), la factorisation matricielle (MF) (He et al., 2016; Koren, 2008) est la plus populaire. Elle permet de projeter les utilisateurs et les items dans un espace latent partagé, en utilisant un vecteur de caractéristiques latentes pour représenter un utilisateur ou un item. Par la suite, l'interaction d'un utilisateur sur un élément est modélisée comme le produit interne de ses vecteurs latents. Cependant, malgré l'efficacité de MF pour le FC, il est bien connu que ses performances peuvent être entravées par le simple choix du produit interne de la fonction d'interaction. Plusieurs efforts de recherche ont été consacrés à son amélioration, comme son intégration avec des modèles basés sur le voisinage (Koren, 2008). Récemment, certains travaux ont appliqué des techniques d'apprentissage profond. He et al. (2017) ont exploré l'utilisation des réseaux de neurones profonds pour l'apprentissage de la fonction d'interaction à partir des données.

S'inspirant de cette approche de FC neuronal (NCF : Neural CF) basée sur les deux modèles (He et al., 2017) : (1) la Factorisation Matricielle Généralisée (Generalized Matrix Factorization - GMF) et le Perceptron Multicouches (Multi Layer Perceptron - MLP), nous avons proposé une approche hybride appelée NHybF (Neural Hybrid Filtering), qui combine le FC et le filtrage à base de contenu (FBC) selon la même architecture basée sur les modèles GMF et HybMLP (Hybrid MLP).

La suite de cet article sera structurée en quatre sections : la Section 2 présente quelques travaux liés à l'utilisation de l'apprentissage profond dans la recommandation. La Section 3 propose une approche de recommandation hybride combinant le FC et FBC selon une architecture neuronale multicouches qui utilise le modèle linéaire GMF et le modèle non linéaire HybMLP. Les résultats des évaluations, en utilisant la base MovieLens-1M, seront présentés dans la Section 4. Finalement, la section 5 met en évidence les résultats les plus importants et trace quelques perspectives futures.

2 État de l'art

Récemment, avec le développement du deep learning, les réseaux de neurones ont été utilisés dans divers domaines, tels que le traitement du langage naturel et la vision par ordinateur. Dans les systèmes de recommandation, certaines méthodes basées sur les réseaux de neurones sont proposées pour apprendre les représentations des utilisateurs et des items à partir des données de notation et de commentaires de texte (Chen et al., 2018; Lu et al., 2018). He et al. (2017) ont modélisé la matrice d'évaluation utilisateur-item en utilisant un réseau neuronal à rétroaction multicouches. D'un autre côté, He et al. (2018) ont proposé une approche basée sur le FC basé et les réseaux de neurones convolutifs.

Nous nous sommes particulièrement intéressés à l'approche NCF (He et al., 2017), qui propose de créer un MLP qui permet l'approximation de la fonction d'interaction sans faire de produit entre les facteurs latents. Afin d'améliorer la précision de l'algorithme, le modèle MF a été associé au MLP. De cette concaténation des deux modèles relèvera un troisième modèle de MF neuronale (NeuMF). En entrée sont donnés les identifiants des utilisateurs et des items. La couche suivante constitue les vecteurs des facteurs latents des utilisateurs et des items pour chaque modèle, sous forme de couche d'inclusion (embedding layer). Puis deux parties, l'une où est repris le modèle général de factorisation matricielle et une deuxième qui apprend la fonction d'interaction. Enfin, les résultats de ces deux méthodes sont regroupées afin d'améliorer la prédiction. En sortie, une approximation de l'interaction "1" ou "0" qui peut être interprétée comme l'intérêt que portera l'utilisateur à l'item. Néanmoins, cette approche ne prend pas en considération le problème de démarrage à froid, c'est pour cela que nous allons proposer, dans cet article, une nouvelle approche hybride appelée NHybF (Neural Hybrid Filtering), qui va combiner le FC et le FBC selon la même architecture basée sur les modèles GMF et HybMLP.

3 Recommandation hybride neuronale (NHybF)

Nous proposons une hybridation du FC et FBC basé sur les réseaux de neurones. Les couches de notre modèle sont décrites comme suit (voir figure 1) :

Couche d'entrée (Input Layer). Nous considérons en entrée les identifiants des utilisateurs et des items, en plus des informations et caractéristiques recueillies relatives à chacun d'entre eux. Ces informations seront par la suite ajoutées aux facteurs appris par le modèle (i.e. seront concaténées à l'Embedding).

Couches d'inclusion (Embedding Layer). L'embedding est la représentation de variables discrètes en un vecteur de nombres continus. En termes de réseaux de neurones, il s'agit de vecteurs, de représentation de variables discrètes, appris. Nous prenons comme exemple le Word Embedding qui s'est avéré très efficace dans l'interprétation du langage naturel. Les mots sont préalablement codés en One-Hot Encoding. Le vecteur sera donc très long et surtout très creux. Le but étant de concevoir un algorithme capable de former des vecteurs moins creux et qui posséderaient une relation logique entre eux.

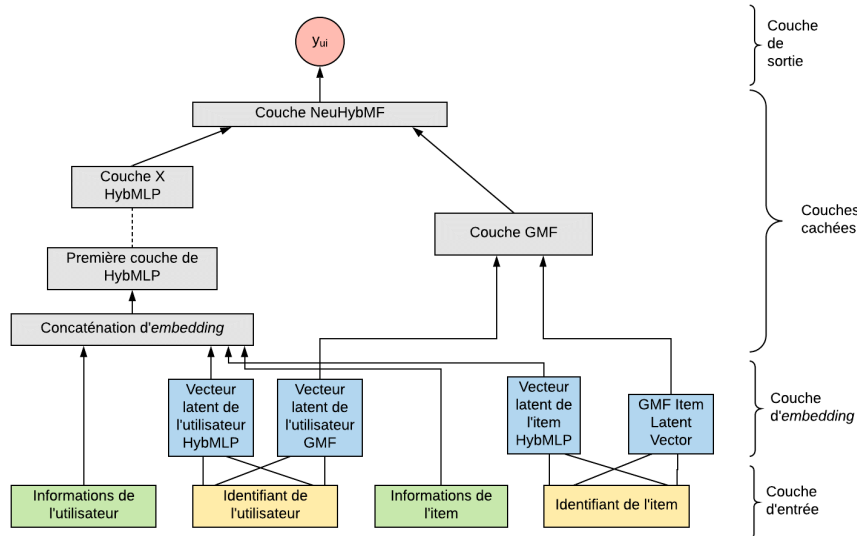


FIG. 1 – Schéma global de l'approche de recommandation NHybF.

Factorisation Matricielle Généralisée (GMF) - Modèle linéaire. Cette partie représente l'opération faite par la MF pour calculer les prédictions.

La description textuelle des couches du modèle est donnée comme suit :

- *Embedding d'utilisateurs GMF* : Vecteurs de facteurs latents des utilisateurs ;
- *Embedding d'items GMF* : Vecteurs de facteurs latents des items.

Recommandation Hybride basée sur l'Apprentissage Profond

- *Couche de multiplication* : assure la multiplication élément par élément de l'embedding (facteurs) des utilisateurs et celui des items.

Perceptron Multicouches Hybride (HybMLP) - Modèle non-linéaire. Cette partie se charge de l'apprentissage de la fonction d'interaction (He et al., 2017) :

$$y_{ui} = \begin{cases} 1 & \text{s'il y a une interaction entre l'utilisateur } u \text{ et l'item } i \\ 0 & \text{sinon} \end{cases} \quad (1)$$

La description textuelle des différentes couches du modèle HybMLP est donnée comme suit (voir figure 2) :

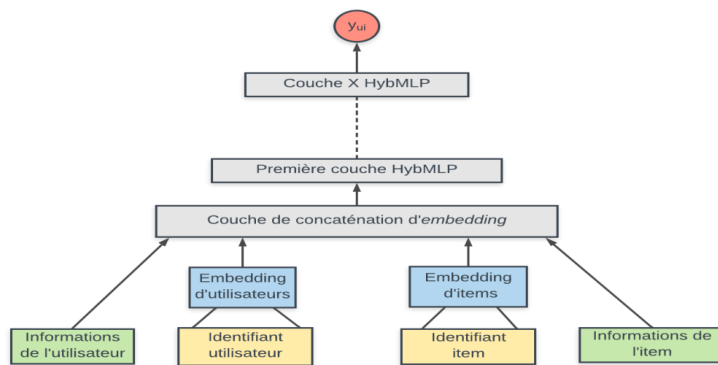


FIG. 2 – Architecture du modèle HybMLP.

- *Embedding d'utilisateurs HybMLP* : Vecteurs de facteurs latents des utilisateurs.
- *Embedding d'items HybMLP* : Vecteurs de facteurs latents des items.
- *Couche de concaténation* : dans cette couche s'effectue la concaténation des embeddings avec les caractéristiques non-apprises données en entrée. Cette couche constitue la première couche cachée du modèle HybMLP. Etant donné, que la concaténation ne suffit pas pour apprendre l'interaction entre les caractéristiques des utilisateurs et des items, ces données seront passées par d'autres couches cachées.
- Autre(s) couche(s) cachée(s) (HybMLP Hidden Layer(s)) : Ces couches sont ajoutées pour permettre l'apprentissage des interactions comme expliqué précédemment. La fonction d'activation utilisée dans chaque couche est ReLU (Rectified Linear Unit : $ReLU(x) = \max(0, x)$) car elle permet de réduire les risques de sur-apprentissage et la saturation des neurones.

Pour décider du nombre de nœuds optimal de chaque couche nous en fixons un pour la dernière que nous appellerons PF (Facteurs Prédicatifs), la couche précédente aura le double de ce nombre de nœuds et ainsi de suite jusqu'à la première.

Factorisation Matricielle Neuronale Hybride (Neural Hybrid Matrix Factorization - NeuHybMF). Nous concaténons les dernières couches des deux modèles précédents (voir figure 1). Pour la couche de concaténation, les résultats des deux modèles précédents sont combinés pour permettre d’avoir de meilleurs résultats. La dernière couche de HybMLP est concaténée à celle de GMF. L’entraînement de ce modèle peut se faire de deux manières : avec ou sans le pré-entraînement des modèles le composant (GMF et HybMLP).

Couche de sortie (Output Layer). Cette couche prend en entrée les vecteurs des dernières couches de HybMLP et GMF concaténés précédemment et les fait passer par une fonction d’activation. Le résultat devant être compris entre 0 et 1.

4 Expérimentations

Nous avons utilisé la base MovieLens 1M (ML-1M), avec un pré-traitement pour transformer les données, en vecteurs exploitables. Nous considérons deux types de données, explicites et implicites. Les données explicites représentent le degré de préférence de l’utilisateur, illustré par une appréciation sur un item (exemple : "j’aime" ou "je n’aime pas"). Tandis que les données implicites sont des données floues qui représentent le comportement d’un utilisateur vis-à-vis d’un item (exemple : cliquer, visionner un film). Pour les métriques d’évaluation, nous avons utilisé HR et NDCG.

4.1 Évaluations préliminaires

Nous avons évalué les modèles HybMLP et NeuHybMF (noté NHybF). Nous avons effectué l’entraînement du modèle de deux manières différentes : (1) entraîner GMF et HybMLP en même temps ; et (2) entraîner GMF et HybMLP séparément, puis entraîner la dernière couche de concaténation après avoir récupéré les poids des deux modèles. Le tableau 1 présente les meilleurs paramètres obtenus, qui seront utilisés dans la suite des évaluations.

| Algorithme | Taille Embedding | Facteurs prédictifs | Nb. Couches | Nb. Instances négatives |
|------------|------------------|---------------------|-------------|-------------------------|
| GMF | 128 | / | / | 3 |
| MLP | 64 | 128 | 5 | 3 |
| HybMLP | 64 | 128 | 5 | 3 |

TAB. 1 – *Meilleurs paramètres des modèles.*

4.2 Évaluation du Top K de la recommandation d'items

La figure ci-dessous (figure 3) illustre l'évaluation des Top-K listes d'items recommandés avec une variation de K.

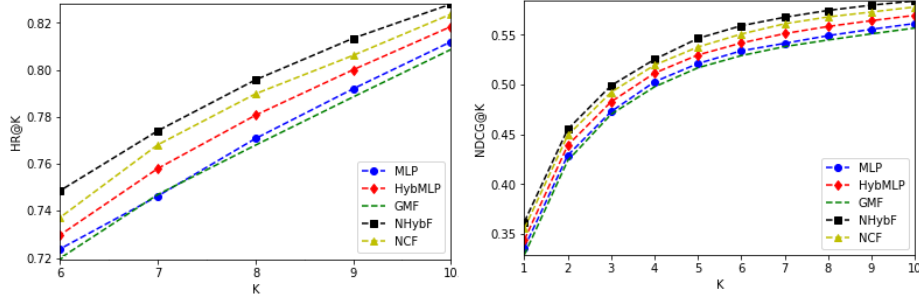


FIG. 3 – Top-K recommandation d'items (avec variation de K).

Nous pouvons constater d'après ces résultats que le modèle NHybMLP surpasse les approches de l'état de l'art (NCF (He et al., 2017), MLP et GMF) et que HybMLP a donné de meilleurs résultats que GMF et MLP).

Évaluation avec des utilisateurs inconnus. Nous prenons les meilleures structures des modèles entraînés précédemment et les entraînons de nouveau sur le jeu de données modifié où les utilisateurs des données d'entraînement ne sont pas les mêmes que ceux des données de test. Les figures suivantes (figure 4, figure 5, figure 6 et figure 7) illustrent les résultats obtenus.

Le modèle GMF affiche des résultats inférieurs à ceux d'autres modèles vu qu'il re-

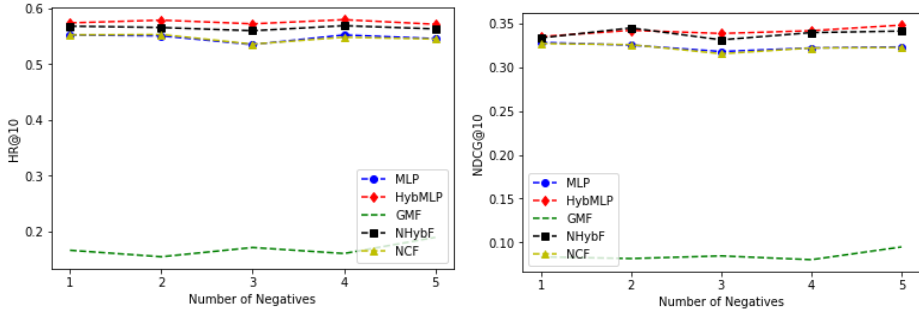


FIG. 4 – Performance selon les instances négatives par instance positive - avec GMF.

quiert les vecteurs latents des utilisateurs ainsi que ceux des items pour fonctionner, à l'inverse de MLP qui devrait apprendre à faire un FC basé items.

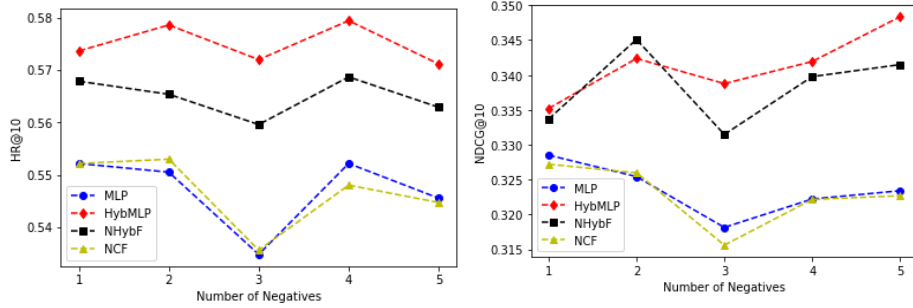


FIG. 5 – Performance selon les instances négatives par instance positive - sans GMF.

Nous pouvons remarquer que HybMLP a une meilleure performance que NHybF et ceci car GMF n’arrive pas à faire les prédictions comme expliqué précédemment.

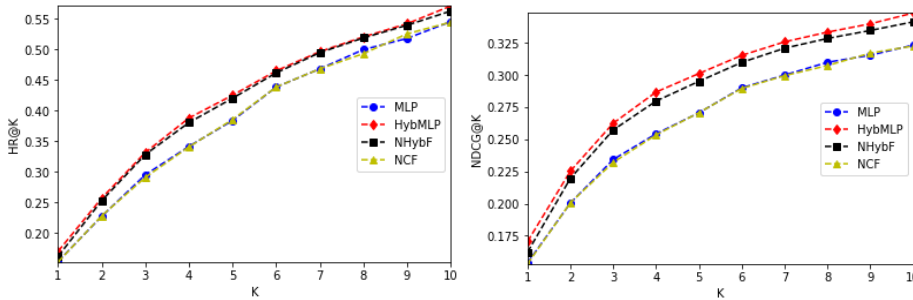


FIG. 6 – Top-K recommandation d’items (K variant de 1 à 10) - sans GMF.

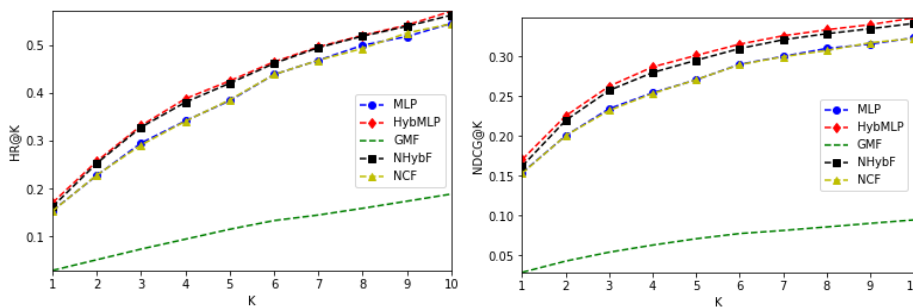


FIG. 7 – Top-K recommandation d’items (K variant de 1 à 10) - avec GMF.

5 Conclusion

Dans ce travail, nous avons proposé un algorithme de recommandation hybride qui combine le FC et FBC selon une architecture basée sur les modèles GMF et HybMLP. Les évaluations effectuées sur la base MovieLens-1M ont montré une amélioration des performances de l'algorithme hybride NHybF (basé sur GMF et HybMLP) comparé à l'algorithme de FC neuronal sous une architecture similaire (He et al., 2017) (basé sur GMF et MLP). Cette performance permet ainsi d'atténuer le problème de démarrage à froid.

Comme perspectives, nous envisageons d'enrichir notre modèle en considérant d'autres données plus complexes comme les commentaires que pourrait donner un utilisateur sur un item, approfondir nos expérimentations avec d'autres bases de données et explorer d'autres architectures comme les réseaux de neurones convolutifs.

Références

- Chen, C., M. Zhang, Y. Liu, et S. Ma (2018). Neural intentional rating regression with review-level explanations. *In Proceedings of the WWW*, 1583–1592.
- He, X., X. Du, X. Wang, F. Tian, J. Tang, et T. Chua (2018). Outer product-based neural collaborative filtering.
- He, X., L. Liao, H. Zhang, L. Nie, X. Hu, et T.-S. Chua (2017). Neural collaborative filtering. *International World Wide Web Conference Committee (IW3C2), WWW 2017, April 3–7, Perth, Australia*.
- He, X., Zhang, H., M.-Y. Kan, et T.-S. Chua (2016). Fast matrix factorization for online recommendation with implicit feedback. *In SIGIR*, 549–558.
- Koren, Y. (2008). Factorization meets the neighborhood: A multifaceted collaborative filtering model. *In KDD*, 426–434.
- Lu, Y., R. Dong, et B. Smyth (2018). Co-evolutionary recommendation model: Mutual learning between ratings and reviews. *In Proceedings of the WWW*, 773–782.
- Resnick, P., N. Iakovou, M. Sushak, P. Bergstrom, et J. Riedl (1994). Grouplens: An open architecture for collaborative filtering of netnews. *Proceedings of the Computer Supported Cooperative Work Conference*, 175–186.

Summary

This article explores the use of deep neural networks for learning the interaction function from data. We propose a hybrid recommendation approach that combines collaborative filtering (CF) and content-based filtering (CBF) in an architecture based on both models : generalized matrix factorization and multilayer perceptron. Extensive experiments on the MovieLens-1M database show significant improvements in our approach compared to existing methods, especially for the cold start situation.