Algorithmes de recommandation

Gilbert Saporta

Chapitre 11

ALGORITHMES DE RECOMMANDATION

Gilbert Saporta

11.1 Introduction

Nés dans les années 90, les systèmes de recommandation se sont développés avec l'essor du commerce électronique. Pour faire simple, le problème consiste à estimer la note qu'un utilisateur donnerait à un produit (livre, film, musique, restaurant, etc.) qu'il ne connaît pas, à partir des notes données par lui-même et d'autres utilisateurs à des produits similaires. Le système recommande alors le produit qui a le plus de chance de plaire à l'utilisateur, ou qui maximise une certaine fonction de profit pour le vendeur.

En reprenant les notations de Adomavicius et Tuzhilin [2005], C désigne l'ensemble des utilisateurs et S l'ensemble des items pouvant être recommandés, u désigne une fonction de $C \times S$ dans $\mathbb{R}$ qui mesure l'utilité de l'item s pour l'utilisateur c. Cette utilité est souvent une note, par exemple un nombre d'étoiles de 1 à 5 pour un film.

Comme l'écrivent Adomavicius et Tuzhilin [2005], « Le problème central des systèmes de recommandation réside dans le fait que l'utilité u n'est généralement pas définie sur l'ensemble de l'espace $C \times S$ mais seulement sur un sous-ensemble de celui-ci. Cela signifie que u doit être extrapolé à l'ensemble de l'espace $C \times S$ ». Il s'agit donc d'un problème de complétion d'une matrice de très grande taille car le nombre d'utilisateurs et le nombre de produits atteignent souvent plusieurs centaines de milliers. En plus de la matrice de notes, on peut disposer de descriptions des utilisateurs (âge, genre, etc.) et de descriptions des items (genre du film, acteurs, etc.) comme schématisé dans la Figure 11.1.

Les systèmes de recommandation ont donné lieu à de très nombreuses publications utilisant une terminologie spécifique comme *filtrage*. Le lecteur intéressé pourra se reporter à ces deux ouvrages monumentaux : Aggarwal [2016] et Ricci *et al.* [2015] ainsi qu'au chapitre 9 de Leskovec *et al.* [2020] ou à la synthèse plus ancienne de Bobadilla *et al.* [2013]. Nous ne décrirons ici que les *algorithmes*, qui ne constituent qu'une partie des systèmes de recommandation.

11.1.1 Domaines d'utilisation

En plus du commerce électronique (Amazon, Alibaba etc.), de la recommandation de films, de vidéos, de musique ou d'informations, on trouve des applications moins classiques dans le domaine de la santé (recommandation de régimes, d'aides à la thérapeutique). La série d'ateliers *HealthRecSys* au sein des conférences de l'ACM sur les systèmes de recommandation fourmille d'exemples.

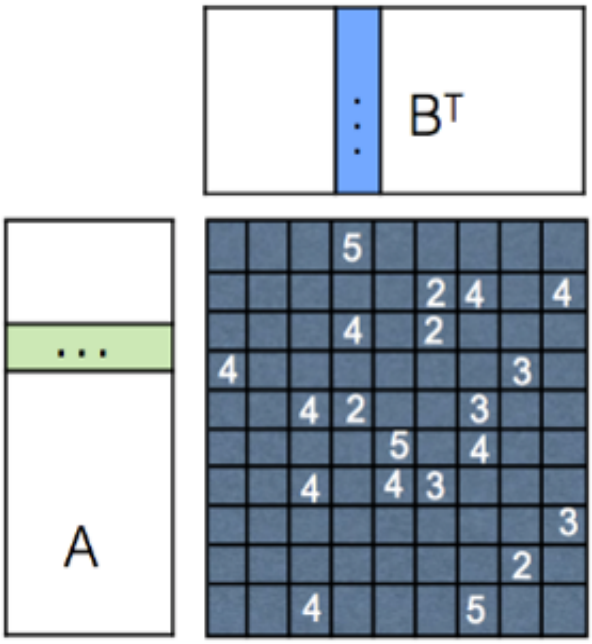

Figure 11.1 : *Données de recommandation*

On notera également des utilisations pour recommander des sites internet d'offres emploi à partir des données textuelles d'annonces et des performances d'annonces passées (Séguéla et Saporta [2011]).

11.1.2 Types de données

Les données les plus riches sont celles où l'utilité u est une note sur une échelle de préférence. Ce sont des notes explicites y compris le cas binaire où on a recueilli des appréciations comme « j'aime » et « je n'aime pas ».

Un type particulier de données appelées données « unaires » s'y ajoute. Il s'agit de données implicites où seul le « 1 » a un sens, comme dans un acte d'achat. Comme le dit Aggarwal [2016] : *When a customer buys an item, it can be viewed as a preference for the item. However, the act of not buying an item from a large universe of possibilities does not always indicate a dislike.*

11.2 Les méthodes de « filtrage » (imputation par voisinage)

Typiques des premiers développements des algorithmes de recommandation, ces méthodes sont en fait des imputations par plus proches voisins pondérés. Prenons l'exemple suivant (figure 11.2) adapté de Khoury [2021] : quelle note Annie donnerait-elle au film Yor ?

Selon l'information utilisée sur les utilisateurs et les items, on distingue de nombreux types de « filtrage ».

	Vampyres	Wild Wild Planet	Xanadu	Yor	Zardoz
Annie	4	5	1	?	2
Bob	2	4	2	5	4
Cédric	2	1	4	2	4

Age	Sexe	Province	Genre
23	F	Québec	SF
24	M	Québec	SF
18	M	Alberta	Fantaisie

Genre	Horreur	SF	Fantaisie	SF	Fantaisie
Année	1974	1965	1980	1983	1973
Acteurs connus?	Non	Non	Oui	Non	Oui

Figure 11.2 : *Quelle note Annie donnerait-elle au film Yor ?*

11.2.1 Filtrage par contenu

Le principe est qu'un utilisateur c qui a aimé certains items aimera des items similaires. L'algorithme initial d'Amazon est de ce type : Linden *et al.* [2003]. Dans l'exemple précédent, on va donc estimer la note qu'aurait donnée Annie à Yor par la moyenne des notes qu'elle a données à d'autres films, pondérées par la similarité entre le film Yor et ceux qu'elle a notés. Si toutes les caractéristiques des films étaient des variables catégorielles, on pourrait utiliser la distance de Jaccard. Dans d'autres contextes, où les données de la matrice $\mathbf{B}$ de la Figure 11.1 sont des données textuelles de fréquences de termes, on utilisera par exemple la similarité TF-IDF (Salton [1988]) .

11.2.2 Filtrage collaboratif et démographique

Dans le filtrage collaboratif on suppose que des utilisateurs ayant les mêmes goûts aimeront les mêmes items. On va donc estimer la note qu'aurait donnée Annie par la moyenne des notes données par les autres utilisateurs pondérées par la similarité entre Annie et les autres utilisateurs. Les mesures de similarité couramment utilisées sont le coefficient de corrélation entre lignes de la matrice des utilités ou bien sa version non-centrée ou cosinus.

On notera que cette méthode n'utilise que les notes des utilisateurs et ne tient pas compte des données que l'on peut connaitre sur eux.

Le filtrage démographique fait lui l'hypothèse que des utilisateurs avec les mêmes profils démographiques (âge, genre, etc.) aimeront les mêmes items. On pondère alors les notes par les similarités entre profils démographiques.

11.2.3 Filtrage social ou par communauté

On suppose ici que des utilisateurs de la même communauté dans un réseau social aimeront les mêmes items. Ce type de filtrage tient compte des liaisons entre utilisateurs. On en trouvera des illustrations dans Fogelman-Soulié *et al.* [2020].

11.2.4 Filtrage hybride

Les méthodes hybrides consistent à combiner deux ou plusieurs méthodes de filtrage. Une des plus simples revient à faire la moyenne des notes obtenues par filtrage par contenu et par filtrage

collaboratif.

11.3 Les méthodes utilisant des modèles

11.3.1 Les méthodes d'apprentissage supervisé

Puisqu'il s'agit d'estimer des valeurs inconnues, alors que l'on dispose d'informations sur les utilisateurs et les items, toutes les méthodes de type régression ou classification (avec données manquantes) peuvent être mises en oeuvre et même combinées comme on le verra avec la compétition Netflix. Elles peuvent être très efficaces mais ne sont pas spécifiques.

11.3.2 Réduction de dimension : *SVD* et factorisation non négative

Les modèles à facteurs latents sont considérés comme l'état de l'art dans les systèmes de recommandation. Ils s'appuient sur des méthodes bien connues de réduction de dimension pour remplir les cases manquantes. Cette approche tente d'expliquer les évaluations en supposant que à la fois les items et les utilisateurs sont caractérisés par un petit nombre de facteurs (Koren *et al.* [2009]). Autrement dit (Leskovec *et al.* [2020]) on conjecture que la matrice d'utilité est le produit de deux matrices longues et fines de même rang k faible comme l'illustre la Figure 11.3

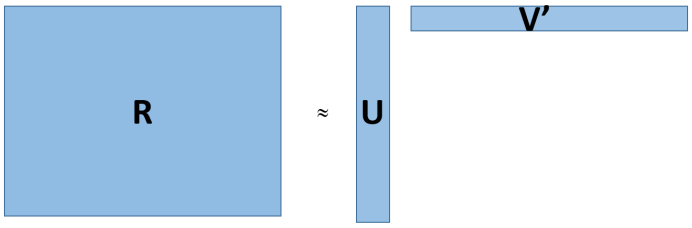

Figure 11.3 : *Factorisation de matrices*

La connaissance de $\mathbf{U}$ et $\mathbf{V}$ permet d'estimer en une seule fois toutes les données manquantes. Selon les contraintes mises sur $\mathbf{U}$ et $\mathbf{V}$ on dispose de plusieurs approches (Aggarwal [2016]) : la factorisation $\mathbf{UV}$ non-contrainte consiste à minimiser le carré de la norme de Frobenius $\left\|\mathbf{R} - \mathbf{UV}'\right\|^2$ sur les cases non-manquantes. La décomposition en valeurs singulières (*SVD*) tronquée à l'ordre k exige de plus l'orthogonalité des colonnes de $\mathbf{U}$ et $\mathbf{V}$. Une implémentation possible consiste à initialiser les valeurs manquantes par les moyennes, à effectuer ensuite une *SVD* régularisée, réestimer les valeurs manquantes et itérer jusqu'à convergence (voir le chapitre 5). La factorisation non-négative introduite par Paatero et Tapper [1994], même si elle est moins performante, est souvent préconisée dans le cas de données unaires car elle permet des interprétations simples des interactions entre utilisateurs et items dans les cas où les utilisateurs disposent d'un mécanisme pour spécifier leur préférence pour un élément, mais pas pour spécifier leur aversion. La factorisation non-négative s'applique pour des matrices $\mathbf{R}$ à termes non-négatifs et consiste à imposer que les termes de $\mathbf{U}$ et $\mathbf{V}$ soient positifs ou nuls. Elle ne permet pas de visualisation, car les composantes ne sont pas orthogonales, mais des classifications et des prévisions. Elle est couramment utilisée en fouille de textes.

11.4 La compétition Netflix

En 2006 la société Netflix, qui n'était à l'époque qu'un loueur de DVD en ligne, promit un million de dollars à qui améliorerait de 10 % la racine carrée de l'erreur quadratique moyenne (*RMSE* =0,9525) de son algorithme *Cinematch* de prévision des notations de films. Cette compétition suscita 44014 soumissions provenant de 5169 équipes et attira durablement l'attention sur les algorithmes de recommandation.

L'ensemble des données mises à disposition par Netflix portait sur plus de 100 millions d'évaluations de films effectuées par des clients anonymes entre le 31 décembre 1999 et le 31 décembre 2005. 17 770 films avaient été évaluées par 480 189 utilisateurs. Les données présentaient une grande variabilité : si en moyenne les utilisateurs avaient évalué plus de 200 films et que chaque film avait été évalué en moyenne par plus de 5 000 utilisateurs, certains films n'avaient été évalués que par trois personnes, tandis qu'un utilisateur particulier avait même évalué plus de 17 000 films. La matrice était très creuse avec seulement 1,18 % de notes attribuées.

Deux ensembles de 1,4 millions de paires films×utilisateurs où les notes étaient dissimulées servaient de tests. L'un des jeux tests servait pour communiquer publiquement les performances, le deuxième pour classer les compétiteurs. Le reste était l'ensemble d'apprentissage.

La compétition fut remportée par l'équipe *BellKor* avec un meta-modèle combinant linéairement 107 modèles de base selon une technique de *blend* (également dénommée *stacking*). Le *RMSE* était abaissé à 0,8567. Les 107 modèles de base se répartissant en : 9 *asymmetric factor models*, 15 modèles de régression, 16 machines de Boltzmann, 30 méthodes de factorisation, 18 méthodes de plus proches voisins, 9 combinaisons partielles des précédentes, 7 méthodes d'imputation utilisant l'ensemble de qualification et 3 méthodes spéciales ! On trouvera le détail dans le rapport de l'équipe victorieuse : Bell *et al.* [2007].

L'équipe concurrente *The Ensemble Team* était arrivée à la même erreur quadratique moyenne, avec une combinaison de seulement 24 modèles de base. Elle ne gagna pourtant pas la compétition car sa solution fut soumise 20 minutes après celle de *BellKor*.

Il est à noter que les meilleures solutions utilisaient essentiellement des méthodes d'apprentissage (*Machine Learning*) plutôt que les méthodes de filtrage évoquées plus haut dans ce chapitre. Aucune information concernant les utilisateurs ni les films n'était utilisée : il s'agissait donc d'une pure complétion de matrice. Comme le remarquent Leskovec *et al.* [2020] page 349, Cinematch n'était d'ailleurs pas un algorithme très performant. Il ne surpassait que de 3% une méthode hybride élémentaire consistant à prendre la moyenne du score moyen d'un film sur l'ensemble des utilisateurs l'ayant noté et de la moyenne des notes données par l'utilisateur sur l'ensemble des films qu'il avait noté.

Netflix ne mit cependant pas en production la solution primée qui aurait demandé un trop grand effort en termes d'industrialisation, d'autant que l'entreprise se tournait vers la diffusion en ligne sur Internet et la production de contenus. L'article de Gomez-Uribe et Hunt [2015] dévoile partiellement les algorithmes de recommandation utilisés actuellement par Netflix.

11.5 Conclusion

Sur le plan opérationnel, les algorithmes de recommandation basés sur les méthodes d'apprentissage statistique semblent actuellement les plus performants et l'emporter sur les méthodes historiques de filtrage.

L'attention s'est portée récemment sur des algorithmes neuronaux, plus particulièrement de *Deep Learning* comme en témoigne Covington *et al.* [2016] pour YouTube, mais la preuve de leur supériorité est encore loin d'être acquise, voir Dacrema *et al.* [2019].

Les applications des systèmes de recommandation sont de plus en plus nombreuses et envahissent notre vie quotidienne sans que l'on s'en aperçoive, comportant des risques éthiques :

- Conformisme des recommandations qui ne proposent en général que des items voisins de ceux que l'on a appréciés.
- Risque d'enfermement dans des « bulles de filtre » dénoncé pour les réseaux sociaux et les moteurs de recherches par Pariser [2011].

Les algorithmes de recommandation peuvent également contribuer à propager des biais racistes et sexistes quand l'apprentissage s'effectue sur des données reflétant des stéréotypes.