Modélisation mathématique
Dossier : Apprentissage machine et systèmes de recommandation

L'apprentissage machine (*machine learning* en anglais) est une branche en plein essor de la science des données. Nous proposons dans ce sujet d'aborder le fonctionnement des systèmes de recommandation, qui en connaissant les goûts d'un utilisateur pour un certain nombre d'items (films, musique, livres, etc...) peuvent lui proposer de nouveaux items qui sont susceptibles d'être à son goût. Les systèmes de recommandation sont utilisés par de nombreuses plateformes commerciales (Netflix, Deezer, etc...).
L'objectif de ce dossier est de comprendre les principes sous-jacents, et de les implémenter.

Données utilisées pour l'apprentissage
Un jeu de données est composé :

- d'utilisateurs
- d'items
- de notations : chaque utilisateur a donné des notes à certains des items.

L'objectif est de suggérer à un utilisateur donné des items qu'il est susceptible d'apprécier. Une façon de faire est de prévoir la note que l'utilisateur va donner à chaque item, et de lui suggérer les items obtenant la meilleure prédiction.

Filtrage collaboratif
Nous nous intéresserons aux algorithmes de type *filtrage collaboratif*.
Les algorithmes de filtrage collaboratif peuvent être classifiés en deux catégories :

- filtrage basé sur l'utilisateur : le système recommande des items que des utilisateurs similaires ont aimé.
- filtrage basé sur l'item : le système identifie la similarité des items, et recommande les items similaires à ceux que l'utilisateur aime.

Pour un filtrage basé sur l'utilisateur, il faut définir une mesure de la similarité entre utilisateurs, et une façon d'agréger les notes attribuées par les utilisateurs similaires.
Pour un filtrage basé sur l'item, il faut définir une mesure de la similarité entre items.
Plusieurs options sont possibles pour les mesures de similarité.

Performance de l'algorithme
Comment évaluer la performance d'un système de recommandation ?
On dispose d'un jeu de données constituée de notes. On efface p.ex. 10% des données (aléatoirement), et on essaye de reconstruire les notes qui ont été effacées. On compare ensuite la prédiction de notre modèle de recommandation avec la vérité des notes qui n'ont pas été prises en compte.

Travail demandé

- Comprendre les méthodes des systèmes de recommandation, et notamment du filtrage collaboratif (cf par exemple lien wikipedia ci-dessous, ou toute autre ressource que vous trouverez).
- Télécharger les jeux de données proposés sur moodle : un jeu correspond à des notes simulées, dans l'autre les données sont incomplètes (les valeurs -1 dans le fichier sont considérées comme de l'absence de note).
- Déterminer la similarité entre le premier utilisateur (ligne d'indice 0) et les 10 suivants. On donnera les résultats avec 3 chiffres après la virgule, au moins pour les mesures de similarité suivantes : similarité de Pearson, similarité cosinus.

- Implémenter au moins un filtre basé sur l'utilisateur, et un filtre basé sur l'item. Comparer leurs performances sur les jeux de données jouet. Envisager et implémenter des améliorations possibles.
- Si le temps le permet : télécharger le jeu de données MovieLens (lien ci-dessous) et effectuer de la prédiction de notes.

Liens
Wikipedia : `https://en.wikipedia.org/wiki/Collaborative_filtering`
Un blog scientifique :
`https://towardsdatascience.com/intro-to-recommender-system-collaborative-filtering-64a238194a26`
MovieLens dataset : `https://grouplens.org/datasets/movielens/` (regarder celui *recommended for education and development*)
Des notions sur les systèmes de recommandation (mais pas sur le filtrage collaboratif !)
`https://www.datacamp.com/community/tutorials/recommender-systems-python`