Système de Recommandation et Matrix Factorization: Rapport du Projet RecSys2022

Nelson VICEL–FARAH, Antoine ZELLMEYER,
Karen KASPAR, Romain BRAND

Juillet 2022

1 Les datasets utilisés

Le challenge RecSys 2022 consiste en l'élaboration d'un système de recommandation permettant, grâce à plusieurs datasets, de prédire l'article acheté lors d'une session. Les données sont réparties sur quatre fichiers csv:

- `train_sessions.csv`: Une séquence d'articles vues lors d'une session
- `train_purchases.csv`: L'article acheté lors de la session
- `test_leaderboard_sessions.csv`: Les sessions pour lesquelles il faut générer des recommandations dans le cadre du concours RecSys2022.
- `item_features.csv`: Description de chaque article avec une valeur par catégorie
- `candidate_items.csv`: Liste de tous les items recommandables.

Nous avons dans un premier temps calculé un score de similarité basé sur le nombre de caractéristiques en commun entre deux articles, et avons donc principalement exploité le fichier `item_features.csv`. Puis nous avons ajouté à celle-ci la similarité calculé en se basant sur l'ordre de visite des articles dans la session et avons donc entraîné notre modèle avec les données des fichiers `train_sessions.csv` et `train_purchases.csv`, tout en calculant la similarité des features avec `item_features.csv` et avons testé le modèle avec les données issues du dataset `test_leaderboard_sessions.csv`.

2 Fonction de similarité

2.1 Similarité basé sur la ressemblance entre deux items

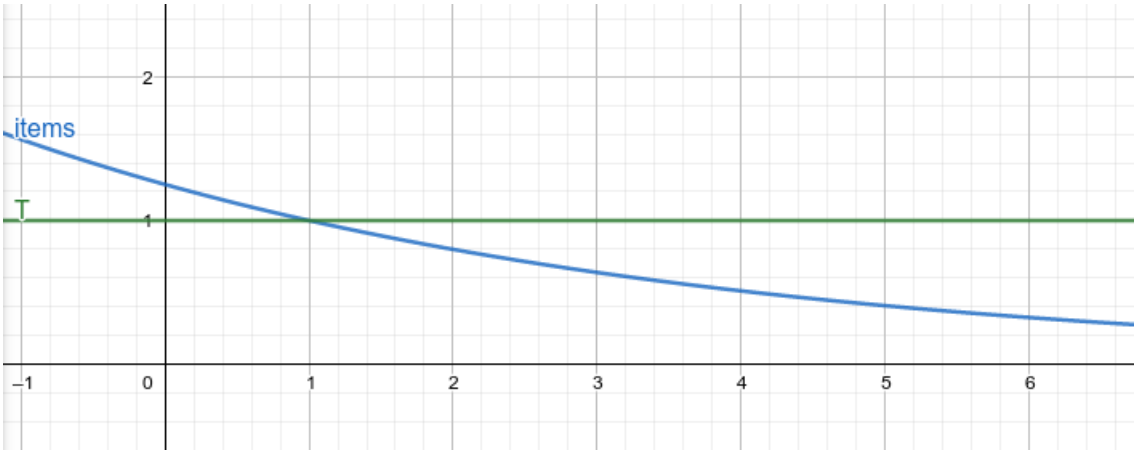
Pour l'élaboration de notre modèle de recommandation, nous nous sommes d'abord basé sur un score calculé à partir de la ressemblance entre un item de la session et l'article acheté. Ainsi, pour chaque catégorie en commun, le score de similarité augmentait légèrement et si la valeur de la catégorie est identique, le score augmentait davantage.

2.2 Similarité basé sur l'ordre de visite des items dans une session

L'idée générale derrière cette partie est d'appliquer des fonctions qui décroissent en fonction de la distance entre deux items au sein d'une session. Les fonctions idéales pour cela sont $e^{-distance}$ avec $distance$ que l'on peut recentrer ou réduire selon nos l'ordre de grandeur que nous souhaitons avoir. Ainsi nous obtenons

$$score_{ij} = \left(\sum_{s} \frac{\exp(-(\varphi_1(\Delta T) + \varphi_2(\Delta items)))}{|s|} \right) * feature\_sim(i,j)$$

Avec φ_1 et φ_2 des applications linéaires arbitraires définies manuellement pour pondérer l'influence de ΔT (distance temporelle) et de $\Delta items$(distance en quantité d'items qui les séparent dans la session s) sur le résultat. $|s|$ étant la taille de la session et $feature\_sim$ étant le score de similarité entre les caractéristiques des items basé sur le dataset `item_features`.



Nous pouvons constater sur les courbes des $\exp(\varphi(x))$ en fonction des Δ respectifs que la distance temporelle a moins d'influence sur le résultat que la distance en nombre d'items. Cela a été établi car nous estimons que le temps écoulé entre deux visites d'items est moins corrélé à la similarité que le nombre d'items visités entre-temps.

Cela appliqué à chaque item nous donne un tableau de similarité que l'on pourra parcourir pour générer des recommandations par session.

3 Recommandations

Pour recommander des items à une session nous calculons la somme des scores de similarité des items de la session avec tous les autres items du dataset. Les 100 items ayant la meilleure similarité totale avec les items de la session seront conservés. Si nous obtenons moins que 100 items, nous piochons parmi les 100 items les plus populaires (les plus achetés).

4 Évaluation des recommandations

Le système de score défini par RecSys2022 pour évaluer les recommandations est le *Mean Reciprocal Rank* qui calcule le score à partir de plusieurs recommandations classées par rang de la plus plausible à la moins plausible de la manière suivante

$$MRR(Q) = \frac{1}{|Q|} \sum_{i}^{|Q|} \frac{1}{rank_i}$$

Sachant Q la liste des recommandations par session.

5 Résultats

Nous observons nos résultats en générant des recommandations sur le dataset `train_sessions` et en appliquant la MRR sur `train_purchases` afin d'évaluer la capacité de notre modèle à prédire les achats des sessions d'entraînement.

Nous obtenons ainsi un score maximal d'environ **0.17** ce qui peut être satisfaisant compte tenu du fait que le meilleur score au concours est de **0.23**.