Université Paul Sabatier

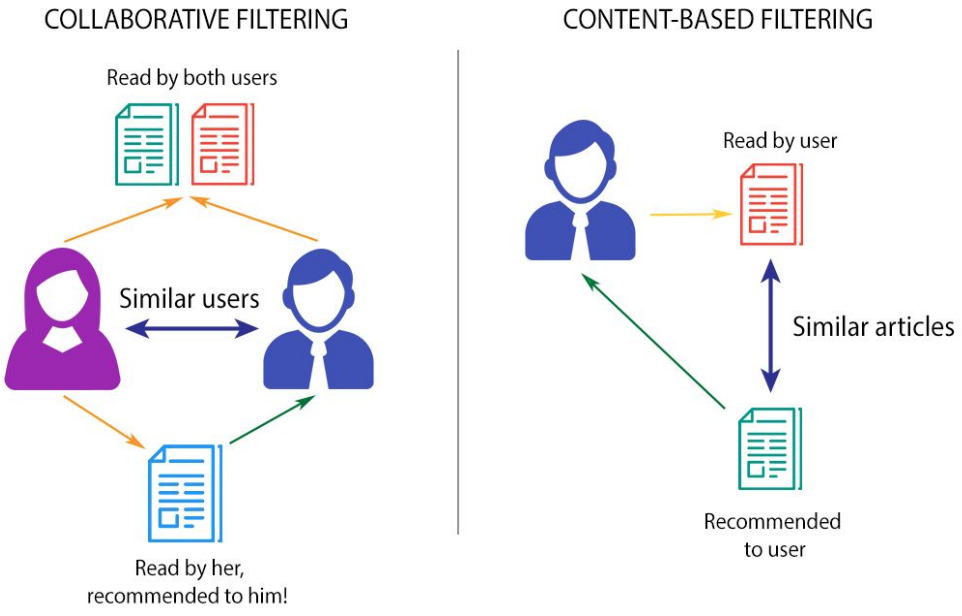
Département Informatique

GINESTET Clément
RODRIGUEZ Diego

20 novembre 2019

Apprentissage machine et systèmes de recommandation

Projet de modélisation M3202



TOULOUSE
AUCH - CASTRES
iut
Midi-Pyrénées
UNIVERSITÉ TOULOUSE III

Table des matières

Introduction générale

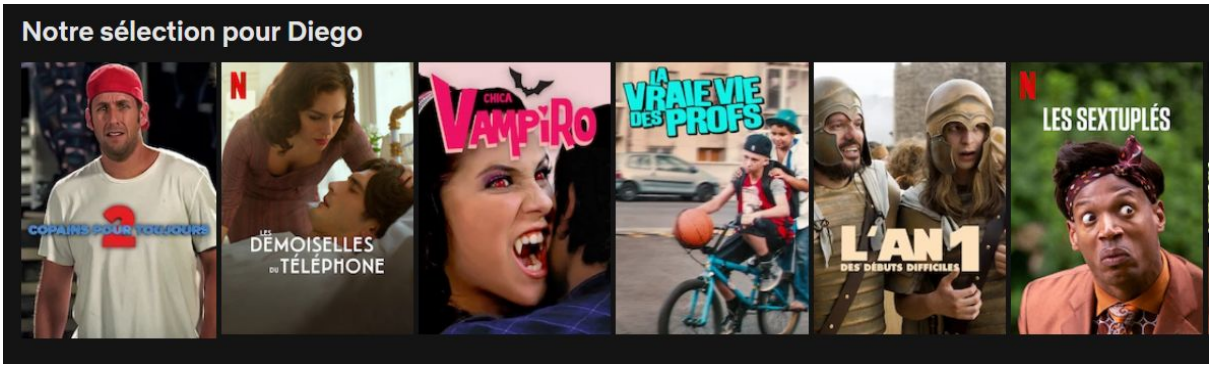
Dans ce dossier, nous abordons le fonctionnement des systèmes de recommandation, qui, en connaissant les goûts d'un utilisateur pour un certain nombre d'items (films, musique, livres, etc…) peuvent lui proposer de nouveaux items qui sont susceptibles de lui plaire. Les systèmes de recommandation sont utilisés par de nombreuses plateformes commerciales (Netflix, Deezer, etc…). L'objectif de ce dossier est d'expliquer et de comprendre les principes des systèmes de recommandation et comment ils fonctionnent afin de les implémenter, de les tester et de les comparer.

Les systèmes de recommandation auxquels nous nous intéressons ici sont de type "filtrage collaboratif". Ce terme fut proposé pour la première fois par David Golberg et ses collaborateurs chez Xerox en 1992.

Pour pouvoir manipuler des données d'utilisateur et appliquer le filtrage collaboratif, nous avons besoin de jeux de données qui contiennent des utilisateurs, des items et des notations que chaque utilisateur attribue à certains des items. Ainsi, la problématique consiste à prédire quelles notes pourraient attribuer des utilisateurs à des items qu'ils n'ont pas encore notés afin de leur recommander des items qui pourraient leur plaire. Ces items-là seront ceux qui ont obtenus les notes prédites les plus élevées.

En général, le filtrage collaboratif se fonde sur les évaluations déjà exprimées par un utilisateur à propos d'autres items.

Dans un contexte réel, suggérer aux utilisateurs des items qui pourraient leur plaire représente un outils commercial important pour garder les consommateurs sur les plateformes pour les pousser à consommer un service comme sur Netflix par exemple.



Le filtrage collaboratif est composé généralement de deux étapes :

- Rechercher des utilisateurs qui ont les mêmes comportements avec l'utilisateur à qui l'on souhaite faire des recommandations.
- Utiliser les notes des utilisateurs similaires pour calculer une liste de recommandations pour cet utilisateur.

C'est donc pour appliquer un système de recommandation que les services comme Netflix demandent aux utilisateurs de noter les différents produits ou services dont ils ont profités.

L'ensemble des notations des utilisateurs permet d'établir leurs profils et de les comparer aux autres clients. Ainsi, si Monsieur Dupont a aimé les films Matrix, Superman et Harry Potter, on va chercher dans la Base de Données les clients qui semblent avoir les mêmes goûts. Pour illustrer cet exemple, supposons que ces clients ont également adoré le film Titanic. Dans ce cas, le logiciel est en droit de supposer que M. Dupont va lui aussi apprécier Titanic et va donc le lui proposer.

Les algorithmes de filtrage collaboratif peuvent être classifiés en deux catégories :

1) Le 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. Ensuite, il faut rechercher des utilisateurs qui ont les mêmes comportements avec l'utilisateur à qui l'on souhaite faire des recommandations puis utiliser les notes des utilisateurs similaires pour calculer une liste de recommandations pour cet utilisateur.

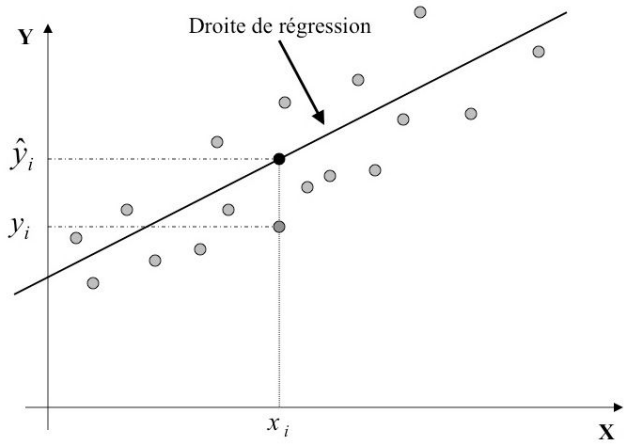
2) Le filtrage basé sur l'item : il faut définir une mesure de la similarité entre items. Plusieurs options sont possibles pour les mesures de similarité. Cette méthode est similaire à celle basée sur les utilisateurs, il suffit d'inverser la matrice de l'ancienne méthode pour obtenir la matrice item-item déterminant des relations entre des objets "pairs". Ensuite, grâce à cette matrice, il sera possible de proposer des items susceptibles d'être aimés par les utilisateurs.

I - Exposé mathématique du problème

Pour trouver un moyen de prédire une note que mettrait un utilisateur sur un item. Il y a de multiples méthodes.

Tout d'abord, dans le cadre de notre projet, nous possédons deux matrices : une qui contient un jeu de données complet et une autre qui contient un jeu de données incomplet. L'objectif est donc de prédire les données qu'il manque sur la matrice incomplète puis de la comparer avec la matrice complète afin de savoir quelle méthode est la plus efficace.

La plus classique consiste à calculer la similarité présente entre tous les utilisateurs pour trouver les plus proches voisins de chaque utilisateur. En effet, cette similarité renvoie une valeur située entre -1 et 1. Une valeur positive indique que les variables varient dans le même sens, tandis qu'une valeur négative signifie que les individus qui ont des notes élevés pour la première variable auront tendance à avoir des notes faibles pour la deuxième variable et inversement.



Chaque utilisateur peut être considéré comme un vecteur incomplet dont nous ne connaissons que quelques composantes. Il est cependant possible de calculer une similarité entre de tels vecteurs en se restreignant aux seules composantes qu'ils ont en commun. Nous supposons dans tout ce sujet que les notes attribuées par les utilisateurs aux items sont des variables aléatoires X et Y qui suivent une distribution conjointe inconnue. Nous considérons donc dans ce sujet que la relation entre les variables est linéaire.

Nous pouvons donc définir le coefficient de corrélation entre X et Y par la formule de Bravais-Pearson centrée utilisateur avec $r_{a,i}$ qui correspond à la note 'r' donnée par un utilisateur 'a' pour un item 'i' :

$$w_{a,u} = \frac{\sum_{i \in I} (r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i \in I} (r_{a,i} - \bar{r}_a)^2 \sum_{i \in I} (r_{u,i} - \bar{r}_u)^2}}$$

Cette formule cherche la similarité entre les utilisateurs ‘a’ et ‘u’. On peut aussi utiliser la formule de Bravais-Pearson pour une version centrée item. Pour cela, nous pouvons soit inverser la matrice et utiliser la formule ci-dessus, soit, en gardant la même matrice utiliser la formule suivante:

$$w_{i,j} = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_i)^2} \sqrt{\sum_{u \in U} (r_{u,j} - \bar{r}_j)^2}}$$

La formule de la pondération calcule la covariance entre les deux variables en la divisant par la racine du produit des variances, comme on peut le voir avec les espérances au numérateur. Mais cette formule est une approximation d’une autre formule plus théorique :

$$p = \frac{Cov(X,Y)}{\sqrt{Var(X) * Var(Y)}}$$

Nous n'utilisons pas la formule théorique dans ce sujet car nous supposons que nos variables X et Y sont linéaires. De plus, les formules que nous allons détailler par la suite sont seulement centrées utilisateur.

Il existe d’autres manières de calculer la similarité, comme cette variante de Bravais-Pearson qui se base sur un calcul de cosinus.

$$\text{simil}(x,y) = \cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{||\vec{x}|| \times ||\vec{y}||} = \frac{\sum_{i \in I_{xy}} r_{x,i} r_{y,i}}{\sqrt{\sum_{i \in I_x} r_{x,i}^2} \sqrt{\sum_{i \in I_y} r_{y,i}^2}}$$

Nous pouvons aussi utiliser le Tau de Kendall qui mesure la corrélation de rang entre deux variables.

$$\tau = \frac{(\text{nombre de paires concordantes}) - (\text{nombre de paires discordantes})}{\frac{1}{2} \cdot n \cdot (n-1)}$$

Enfin, pour pouvoir arriver à l’étape finale de nos algorithmes de recommandation, il faut retenir les utilisateurs ayant les similarités les plus proches (que l’on nommera par la suite les voisins) obtenus grâce aux formules précédentes et ensuite calculer une moyenne pondérée de leurs évaluations pour fournir une prédiction de note.

Il y a tout d’abord la moyenne classique qui ne prend pas en compte les valeurs de similarité avec ‘n‘ le nombre total d’items que l’utilisateur a déjà noté :

$$\frac{\sum_u r}{\sum_u n}$$

Il y a ensuite la formule de pondération qui prend compte les moyennes des notes des plus proches voisins :

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u \in K} (r_{u,i} - \bar{r}_u) \times w_{a,u}}{\sum_{u \in K} w_{a,u}}$$

Pour évaluer la qualité de nos systèmes de recommandation, il faut comparer le tableau rempli que nous avons à disposition avec les autres tableaux que nous avons créés. On utilise donc des indicateurs d'écarts afin de comparer les tableaux case par case puis on peut établir en moyenne l'écart qu'il y a entre les notes que nous avons prédites et les notes réelles. Nous utilisons trois méthodes de comparaison :

Premièrement, l'Erreur Moyenne Absolue qui calcule la valeur absolue de la différence entre la note prédite et la note réelle pour chaque case :

$$MAE = \frac{\sum_{\{u,i\}} |p_{u,i} - r_{u,i}|}{N}$$

Deuxièmement, l'écart quadratique moyen qui est plus sensible aux grosses erreurs, aussi appelé la dispersion, qui est égal à la somme du biais au carré et de la variance :

$$RMSE = \sqrt{\frac{\sum_{\{u,i\}} (p_{u,i} - r_{u,i})^2}{N}}$$

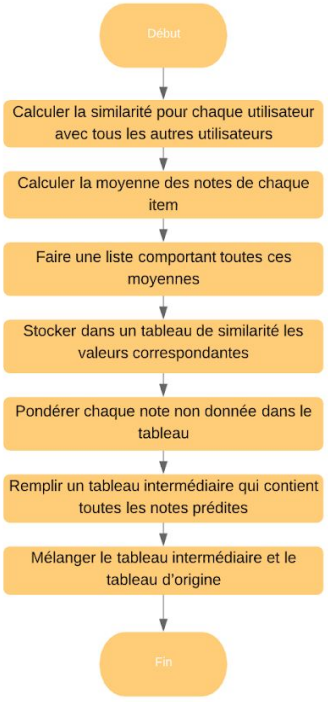
Troisièmement, l'Erreur Moyenne du biais qui ne prend pas en compte la valeur absolue et qui estime une valeur qui est plus ou moins grande de la valeur exacte. En effet, le biais est la moyenne de l'erreur :

$$MBE = \frac{\sum_{u,i} p_{u,i} - r_{u,i}}{N}$$

II - Algorithmes pour résoudre le problème

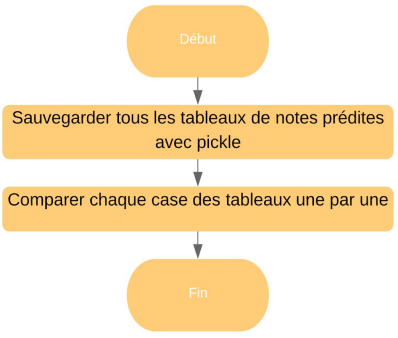
Tout d'abord, le programme utilisé pour résoudre le problème a entièrement été codé avec la version 2.7 de Python. Nous ne décrivons pas les méthodes pour appliquer exactement les formules car les algorithmes sont basés sur des parcours de tableaux, des conditions et des boucles. De plus, nous avons produit de nombreux algorithmes qui se ressemblent. Ainsi, nous avons décidé de ne pas tous les détailler.

Prédiction des notes



Pour calculer les similarités, on utilise les différentes méthodes : la corrélation de Bravais-Pearson, la méthode du cosinus et le tau de Kendall. Pour cela, il suffit de placer la méthode que l'on souhaite utiliser dans la séquence "Calculer la similarité pour chaque utilisateur avec tous les autres utilisateurs" de l'algorigramme ci-dessus. Toutes ces méthodes de calcul sont équivalentes avec le filtrage centré item : inverser les matrices suffit pour obtenir les résultats version item.

Comparaisons



Grâce à la méthode pickle, on conserve les tableaux que l'on crée. Cela permet ensuite de comparer le tableau contenant les notes prédites avec celui qui contient les notes réelles.

III - Résultats

Nous avons choisi d'utiliser les résultats des erreurs absolues moyennes (Mean absolute error) qui sont les résultats les plus intéressants à exploiter et nous les présenterons sous forme de tableaux.

Exemple d'affichage de la console pour la corrélation de Bravais-Pearson centrée utilisateur :

```
Mean absolute error : 0.2796116
Root Mean Squared Error : 0.483434038934
Mean Bias Error : -0.0018153999999999976
```

Pour les calculs, nous utilisons le tableau donné incomplet avec 50% de notes manquantes ainsi que la formule de pondération.

	Méthode centrée utilisateur	Méthode centrée item
Corrélation de Bravais-Pearson	0.28	0.52
Méthode du cosinus	0.66	0.74
Tau de Kendall	0.27	0.53

Nous remarquons grâce à la formule MAE (Mean absolute error), deux résultats. Effectivement, nous observons que le tau de Kendall est la méthode la plus efficace pour calculer la similarité. En parallèle, les calculs centrés utilisateurs semblent plus efficaces. Cependant, les résultats de cette dernière observation s'expliquent par la taille de la matrice. En effet, celle-ci n'est pas carré, elle est composée de 100 utilisateurs pour 1000 items. S'il y avait eu plus d'utilisateurs que d'items, les méthodes centrées item seraient probablement plus efficaces que celles centrées utilisateurs.

Nous présenterons le reste des résultats avec des méthodes centrées utilisateurs et en utilisant le tau de Kendall car, bien que très proche de la méthode de Bravais-Pearson, il s'agit de la méthode la plus efficace.

.

Nous testons maintenant le Tau de Kendall en utilisant la moyenne normale pour prédire les notes :

	Moyenne normale	Moyenne pondérée
Tau de Kendall	0.58	0.27

Nous observons ici que la moyenne normale est moins efficace que la pondération car l'écart moyen est environ deux fois plus élevé. Nous continuerons donc les calculs en utilisant la méthode de pondération.

Maintenant, voici le tableau des calculs en utilisant les utilisateurs les plus proches. Cette méthode permet de limiter les calculs des pondérations, ce qui réduit la dimensionnalité du problème.

	10 plus proches voisins	4 plus proches voisins	Tous les utilisateurs voisins
Tau de Kendall	0.25	0.30	0.27

Nous observons que les notes prédites, en ne prenant en compte que les 10 plus proches voisins, sont les plus précises. Cependant, celles concernant les 4 plus proches voisins sont moins précises que celles prenant en compte tout le tableau. Nous pouvons en déduire que réaliser les calculs avec un nombre de voisins proches limité donne de meilleurs résultats, mais qu'un nombre trop limité peut avoir l'effet contraire. Il faut donc choisir un juste milieu pour obtenir des notes proches de la réalité.

Testons maintenant avec un tableau qui a été effacé à 75% :

	Tableau effacé à 75%	Tableau effacé à 50%
Tau de Kendall	0.49	0.27

Nous constatons que le nombre de notes effacés et la précision des prédictions sont proportionnels. En effet, moins le tableau est rempli, moins les prédictions sont précises. Nous n'avons pas pu tester nos programmes avec des tableaux encore moins remplis car nous nous retrouvions avec des items qui n'étaient notés par aucun utilisateur et des utilisateurs qui ne notaient aucun item. Nous ne pouvons donc pas appliquer nos méthodes lorsque ce cas s'applique.

Au sujet de la précision des prédictions pour chaque calcul, l'erreur moyenne du biais (Mean Bias Error) se situe entre -0.005 et 0.005, ainsi, nous pouvons en déduire que les méthodes utilisées prédisent la bonne note à des taux variants de 99.5% à 99.9%. Dans notre cas, les méthodes centrées utilisateurs sont donc très précises pour un jeu de données contenant 100 utilisateurs et 1000 items.

Conclusion

Grâce aux résultats ci-dessus, dans le cas de notre sujet, nous pouvons estimer que la meilleure méthode pour calculer la similarité est le Tau de Kendall, bien que la corrélation de Bravais-Pearson soit aussi très intéressante. Les méthodes centrées utilisateurs sont aussi plus précises que les méthodes centrées items. Pour obtenir la note la plus précise possible, il faut pondérer les similarités plutôt que de calculer simplement une moyenne. De plus, il faut estimer les plus proches voisins d'un utilisateur et d'un item afin de les classer par catégories pour avoir des résultats plus précis. Pour finir, plus la matrice utilisateurs-items contient des notes attribuées par les utilisateurs, plus les recommandations vont être rapides et précises.

La difficulté de ce sujet réside dans l'application des algorithmes de filtrage collaboratif sur de réels jeux de données, cette mise en application nécessite d'énormes quantités de données (généralement des matrices utilisateurs-items de l'ordre de 10^4 lignes * 10^5 colonnes), ce qui implique qu'il faut trouver les algorithmes les plus optimisés sinon les calculs des données prendraient trop de temps à l'échelle humaine. En réalité, les utilisateurs notent rarement les items. Les matrices sont donc très souvent creuses et comportent environ moins de 1% de notes. Les prédictions sont donc moins proches de la note réelle et des erreurs peuvent s'immiscer dans le résultat final. Par ailleurs, il est difficile de mesurer la performance d'un système de filtrage collaboratif, ce qui est nécessaire pour identifier la technique la mieux adaptée aux données que l'on a collectées.

Lorsque l'on ajoute de nouveaux utilisateurs ou de nouveaux items, cela pose problème car il est difficile de les assimiler avec les données déjà existantes, il faut trouver la meilleure façon d'intégrer un nouvel utilisateur

rapidement sans lui demander de renseigner une fiche de préférences. D'autres méthodes et d'autres technologies sont donc utilisées comme par exemple le temps que l'utilisateur passe sur la page d'un site afin de deviner ses préférences.

Pour conclure, il est difficile de comparer plusieurs algorithmes de recommandation entre eux, en particulier lorsque l'évaluation a lieu en ligne, c'est-à-dire alors même que le système est employé par les utilisateurs. Il existe en effet des boucles de rétroaction entre les recommandations produites par le système et les préférences exprimées par les utilisateurs. Par exemple, un item populaire sera souvent recommandé, donc souvent évalué par les utilisateurs. Son poids, dans les mesures de similarité et le calcul des recommandations ultérieures, sera de ce fait augmenté, ce qui faussera le système.

Les enjeux de ces performances d'algorithmes sont donc très importants de nos jours et principalement utilisés dans un aspect commercial pour pouvoir inciter les utilisateurs à consommer du contenu en les amenant vers ce qu'ils préfèrent. L'intérêt économique derrière ces algorithmes est élevé et des sommes d'argent majeures sont mises en jeu pour trouver des solutions toujours plus efficaces.

Pour aller plus loin, il existe aussi d'autres méthodes plus complexes que nous n'avons pas abordées telles que la corrélation du rang de Spearman ou le principe d'entropie car ces méthodes ne sont pas adaptées pour notre sujet.

Merci d'avoir prit le temps de lire notre dossier.

Diego RODRIGUEZ et Clément GINESTET

Bibliographie

Ci-dessous sont mentionnés les sites qui nous ont permis de mieux appréhender le sujet et comprendre comment fonctionne le filtrage collaboratif.

- https://en.wikipedia.org/wiki/Collaborative_filtering
- https://fr.wikipedia.org/wiki/Tau_de_Kendall
- https://www.datacamp.com/community/tutorials/recommender-systems-python
- Le document que nous a donné Monsieur Lahcen
- Le document technique disponible sur moodle