https://www.overleaf.com/9277705dvpcbsnqnmsn#/33462787/

Rappel des notations utilisées

Pour un problème donné, nous avons les variables suivantes :

- N_U le nombre d'utilisateurs ;
- N_M le nombre de films ;
- N le nombre de notes ;
- $U = \{u_1, u_2, ..., u_{N_U}\}$ l'ensemble des utilisateurs ;
- $M = \{m_1, m_2, ..., m_{N_M}\}$ l'ensemble des films ;
- $Y = (y_{u,m})_{(u,m)\in U\times M}$ telle que : $y_{u,m}$ = note donnée par l'utilisateur u pour le film m ;
- $R = (r_{u,m})_{(u,m)\in U\times M}$ telles que : $r_{u,m} = \begin{cases} 1 & \text{si l'utilisateur } u \text{ a noté le film } m \\ 0 & \text{sinon} \end{cases}$
- $\Omega_0 = \{(u,m) \in U \times M | r_{u,m} = 0\}$, ensemble des couples (u,m) où l'utilisateur u n'a pas noté le film m ;
- $\Omega_1 = \{(u,m) \in U \times M | r_{u,m} = 1\}$, ensemble des couples (u,m) où l'utilisateur u a noté le film m ;
- $\forall u_0 \in U, \mathcal{M}(u_0) = \{m | r_{u_0,m} = 1\}$, l'ensemble des films qu'a vu l'utilisateur u_0 ;
- $\forall m_0 \in M, \mathcal{U}(m_0) = \{u | r_{u,m_0} = 1\}$, l'ensemble des utilisateurs qui ont vu le film m_0 ;
- $\forall u_0 \in U, \overline{\mathcal{M}(u_0)} = M \setminus \mathcal{M}(u_0)$, l'ensemble des films que n'a pas vu l'utilisateur u_0

Note des auteurs : Ce rapport va de pair avec le code implémenté sous R qui est disponible sur https://github.com/kkmm001/StatApp-Collaborative-Filtering.

1 Introduction

1.1 Présentation de la base ml-100k

Nous considérons les bases de données suivantes : data.Ratings, data.Movies et data.Users. Nous noterons dans la suite :

- N_U le nombre d'utilisateurs ;
- N_M le nombre de films ;
- N le nombre de notes.

Pour la base ml-100k, après suppression des doublons : $N_U = 943$, $N_M = 1663$ et $N = 99737$.

La base des notes : data.Ratings Cette base comprend pour chaque couple (utilisateur, film) un entier comprise en 1 et 5 représentant la note attribuée par l'utilsateur au film.

La base des films : data.Movies Cette base comprend toutes les données pour caractériser un film :
◇ l'identifiant du film ;
◇ le titre du film ;
◇ l'année de sortie ;
◇ la variable `IMDbURL` qui indique le lien url du film sur le site `http://imdb.com` ;
◇ 19 variables booléennes qui caractérisent le genre cinématographique : `unknown`, `action`, `adventure`, `animation`, `children.s`, `comedy`, `crime`, `documentary`, `drama`, `fantasy`, `film.noir`, `horror`, `musical`, `mystery`, `romance`, `sci.fi`, `thriller`, `war`, `western`.
Ainsi, chaque film est caractérisé par 23 variables, dont 19 booléennes.

La base des utilisateurs : data.Users Cette base comprend toutes les données pour caractériser un utilisateur :
◇ l'identifiant de l'utilisateur ;
◇ l'âge ;
◇ le sexe ;
◇ l'activité professionnelle ;
◇ le code postal.

Un extrait de chaque base du problème ml-100k est disponible en annexe (tableaux **??**, **??** et **??**).

1.2 Notation pour la prédiction

Soit $U = \{u_1, u_2, ..., u_{N_U}\}$ l'ensemble des utilisateurs et $M = \{m_1, m_2, ..., m_{N_M}\}$ l'ensemble des films.
Considérons la matrice $Y = (y_{u,m})_{(u,m)\in U\times M}$ telle que :

$$y_{u,m} = \text{note donnée par l'utilisateur } u \text{ pour le film } m \, (y_{u,m} \in [\![1, 5]\!])$$

Y est donc une matrice $N_U \times N_M$. A quoi ressemble une telle matrice ?
Pour le cas du probleme ml-100k, Y est une matrice de taille 943×1663 comprenant 1 568 209 éléments dont exactement 99 737 valeurs non nulles ; donc le taux de complétion est de 6.4%. C'est donc une matrice creuse.
Pour recommander des films, nous allons tenter de prédire les notes de tous les films non notés de la base par l'utilisateur. Introduisons pour cela les variables booléennes $r_{u,m}$ pour $(u, m) \in U \times M$ telles que :

$$r_{u,m} = \begin{cases} 1 & \text{si l'utilisateur } u \text{ a noté le film } m \\ 0 & \text{sinon} \end{cases}$$

Ainsi $y_{u,m}$ (la note) n'a de sens que si $r_{u,m} = 1$. Le but est donc de déterminer les éléments de $Y = (y_{u,m)}$ tels que $r_{u,m} = 0$.

Remarque : les prédictions utiliseront uniquement les données présentes dans la base des notes data.Ratings. Aucune information issue de la base des utilisateurs ou de la base des films ne sera exploitée par le programme.

Nous noterons les ensembles suivants :
- $\Omega_0 = \{(u,m) \in U \times M | r_{u,m} = 0\}$;
- $\Omega_1 = \{(u,m) \in U \times M | r_{u,m} = 1\}$;
- $\forall u_0 \in U, \mathcal{M}(u_0) = \{m | r_{u_0,m} = 1\}$, l'ensemble des films qu'a vu l'utilisateur u_0 ;
- $\forall m_0 \in M, \mathcal{U}(m_0) = \{u | r_{u,m_0} = 1\}$, l'ensemble des utilisateurs qui ont vu le film m_0

2 Prédiction

2.1 Algorithmes naïfs

Considérons $u_0 \in U, m_0 \in M$ tels que $(u_0, m_0) \in \Omega_0$.

Aléatoire (`random-unif`) On affecte de manière aléatoire suivant une distribution uniforme sur $[\![1, 5]\!]$ une note à l'élément y_{u_0,m_0}.

Aléatoire (`random-samp`) On affecte de manière aléatoire suivant la distribution observée des notes, une note à l'élément y_{u_0,m_0}.

Note unique : la moyenne de toutes les notes (`mean`) Cette seconde approche donne à tous les éléments la valeur :

$$y_{u_0,m_0} := \overline{y} \triangleq \frac{1}{N} \sum_{(u,m)\in\Omega_1} y_{u,m}$$

Note unique : la moyenne des moyennes des films (`meanOfMovies`) Ici, ce sera une autre valeur qui sera attribuée à l'ensemble Ω_0 : on affecte à y_{u_0,m_0} la moyenne des moyennes des films.

Note unique : la moyenne des moyennes par utilisateur (`meanOfUsers`) Par analogie, on affecte ici la valeur la moyenne des moyennes des utilisateurs.

Prédiction par la moyenne des notes du film (`meanByMovie`) On affecte dans ce cas la même note à l'ensemble $\{y_{u,m_0} | (u, m_0) \in \Omega_0\}$ la valeur :

$$y_{u_0,m_0} := \overline{y_{\cdot,m_0}} \triangleq \frac{1}{|\mathcal{U}(m_0)|} \sum_{u\in\mathcal{U}(m_0)} y_{u,m_0}$$

$\overline{y_{\cdot,m_0}}$ représente la note moyenne donnée au film m_0 par les utilisateurs.

Prédiction par la moyenne des notes de l'utilisateur (`meanByUser`) De manière duale, on affecte ici la même note à l'ensemble $\{y_{u_0,m} | (u_0, m) \in \Omega_0\}$ la valeur :

$$y_{u_0,m_0} := \overline{y_{u_0,\cdot}} \triangleq \frac{1}{|\mathcal{M}(u_0)|} \sum_{m \in \mathcal{M}(u_0)} y_{u_0,m}$$

$\overline{y_{u_0,\cdot}}$ représente la note moyenne donnée par l'utilisateur u_0.

Certaines de ces méthodes sont totalement inefficaces pour du filtrage collaboratif (toutes hormis la prédiction par la moyenne des notes des films) car certaines prédictions ne permettent pas d'établir un classement de films, empéchant de fait la recommandation ...

2.1.1 Résultats des algorithmes naïfs

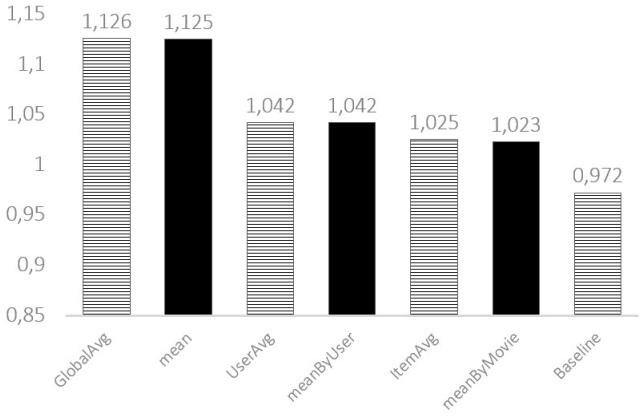
Pour les algorithmes naïfs, une fois établie la méthode de prédiction, nous nous sommes confrontés à une nouvelle difficulté. En effet, à travers la prédiction de notes comme étant la moyenne des films, il arrivait que celle-ci vaille 5 (la note maximale, donc très recommandable) car le film n'avait été noté qu'une seule fois. Pour remédier à ce soucis d'objectivité, nous avons dû introduire un seuil de visionnage. Il suffit alors de recommander parmi les films non-visionnés par l'utilisateur qui ont dépassés ce seuil de visionnage ceux qui ont récolté les meilleures moyennes. Il ressort du tableau **??** en annexe que la méthode de prédiction par la moyenne des notes des films est la meilleure parmi les méthodes naïves car elle minimise l'erreur RMSE comme le montre le tableau récapitulatif suivant :

Table 1 – Moyenne des erreurs des méthodes naïves

	Moyenne des erreurs RMSE
`random-unif`	**1.699**
`random-samp`	**1.594**
`meanOfMovies`	**1.207**
`meanOfUsers`	**1.127**
`mean`	**1.125**
`meanByUser`	**1.042**
`meanByMovie`	**1.023**

La figure ci-dessous compare les erreurs RMSE de nos résultats (en noir) avec ceux des benchmarks (en motifs rayés).

Figure 1 – Benchmark pour les méthodes naïves



Détails des méthodes
- mean, meanByUser et meanByMovie sont les méthodes implémentées dans ce rapport ;
- GlobalAbg, UserAvg et ItemAvg sont les mêmes méthodes que celles vues dans ce rapport, les résultats sont disponibles sur le site `http://www.librec.net` ;
- Baseline est une méthode développée dans [**?**] qui utilise à la fois la moyenne de l'utilisateur et la moyenne du film.

2.2 Méthode des plus proches voisins

Dans cette partie, nous développerons la méthode User-User Collaborative Filtering. Cette technique consiste, pour un individu, à rechercher un échantillon d'utilisateurs qui lui sont le plus proches, au sens d'une certaine similarité. Ainsi, toujours pour cet individu, et pour un film donné, il devient possible de prédire la note qu'il aurait donné à ce film à partir des notes de ses plus proches voisins.

Cette méthode repose sur deux hypothèses :
— les goûts cinématographiques ne changent pas au cours du temps ;
— les individus qui ont eu les mêmes goûts cinématographiques dans le passé sont plus susceptibles de partager les mêmes goûts dans le futur.

Techniquement, cette méthode fera intervenir différents paramètres, ce qui multipliera le nombre d'approches possibles. Une prédiction dépendra en effet :
— de la notion de similarité utilisée ;
— du nombre de plus proches voisins considéré ;
— du prédicteur (la fonction d'évalution de la note)

2.2.1 Notion de similarité

Pour calculer la similarité entre deux utilisateurs, on se restreint aux seules composantes qu'ils ont en commun. Ainsi pour définir les plus proches voisins, il faut se donner une mesure de similarité entre utilisateurs. Comme le coefficient de corrélation de Pearson est l'une des similarités les plus présentes dans la littérature, comme le précise l'article [**?**] d'une part. Et que d'autre part la corrélation de Pearson est la seule métrique recommandée dans l'article [**?**] de Herlocker et al, qui ont essayé différentes approches pour la méthode des plus proches voisins, c'est pourquoi nous l'avons donc choisi. Notons toutefois que d'autres métriques apparaissent également dans la littérature comme la corrélation de Spearman ou la similarité cosinus.

La corrélation de Pearson Considérons y_{u_1} et y_{u_2} les vecteurs des notes attribuées aux films visionnés par les utilisateur u_1 et u_2. Le coefficient de corrélation de Pearson entre l'individu u_1 et u_2 est défini par :

$$s_{u_1,u_2} = \frac{\sum_{m \in \mathcal{M}(u_1) \cap \mathcal{M}(u_2)} (y_{u_1,m} - \overline{y_{u_1,\cdot}})(y_{u_2,m} - \overline{y_{u_2,\cdot}})}{\sqrt{\sum_{m \in \mathcal{M}(u_1) \cap \mathcal{M}(u_2)} (y_{u_1,m} - \overline{y_{u_1,\cdot}})^2 \sum_{m \in \mathcal{M}(u_1) \cap \mathcal{M}(u_2)} (y_{u_2,m} - \overline{y_{u_2,\cdot}})^2}}$$

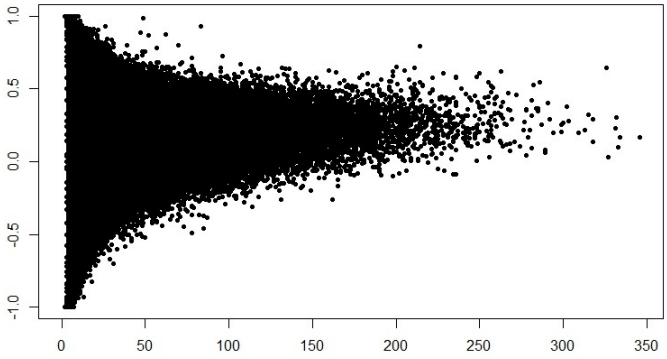
Ainsi seules les notes des films vu à la fois par l'utilisateur u_1 et l'utilisateur u_2 sont prises en compte pour le calcul du coefficient de corrélation de Pearson.

La valeur du coefficient de corrélation est comprise entre – 1 et 1 : une valeur proche de 1 signifie que les deux individus notent les films de la même manière (ils ont donc des goûts cinématographiques similaires), tandis qu'une corrélation proche de -1 signifie que leur notation est opposée (leurs goûts sont antagonistes).

Comme révélé par l'article [**?**] et les travaux de [**?**], un défaut de corrélation apparaît lorsque deux individus ont peu de films notés en commun : *"In our experience with collaborative filtering systems, we have found that it was common for the active user to have highly correlated neighbors that were based on a very small number of co-rated items. These neighbors that were based on tiny samples (often three to five co-rated items) frequently proved to be terrible predictors for the active user."* (Herlocker and al, 2002).

En effet, d'après la figure 2, il y a lien entre une similarité élevée et le nombre de films en commun, qui a lui-même un lien avec le nombre de film noté par individu. Il apparaît que les individus ayant une similarité forte partagent peu de films en commun. Compte tenu du fait que la prédiction sera obtenue à partir de ces "voisins", les résultats seront potentiellement médiocres.

FIGURE 2 – Lien entre corrélation de Pearson et nombre de films en commun



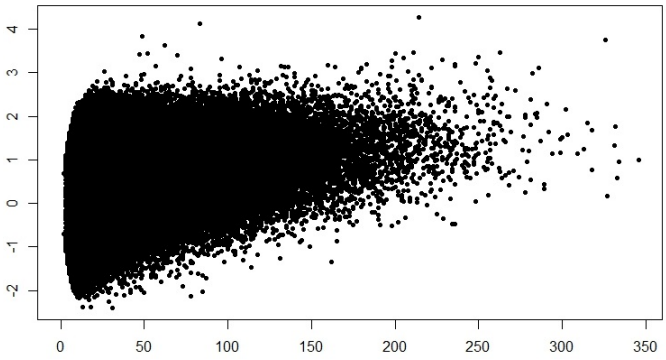
Pour résoudre ce problème, nous avions dans un premier temps implémenté une version basée sur "un seuil de voisinage" : seuls les voisins ayant un nombre de films en commun supérieur à un seuil défini pouvait se prétendre voisin de l'utilisateur actif. Cependant, les résultats ne semblent pas être significatifs pour les prédicteurs considérés dans l'étude (voir les sous-parties 2.2.2 et **??**). L'implémentation sous R comprend toujours cette nuance mais aucun résultat ne sera abordé dans le rapport.

La corrélation RFP Pour palier ce problème, nous avons innové dans un second temps avec une nouvelle similarité : RFP (pour ratings-frequency Pearson), définie par le produit de la corrélation de Pearson et du logarithme du nombre de films vus en commun :

$$s_{u_1,u_2} = \frac{\sum\limits_{m\in\mathcal{M}(u_1)\cap\mathcal{M}(u_2)} (y_{u_1,m} - \overline{y_{u_1,\cdot}})(y_{u_2,m} - \overline{y_{u_2,\cdot}})}{\sqrt{\sum\limits_{m\in\mathcal{M}(u_1)\cap\mathcal{M}(u_2)} (y_{u_1,m} - \overline{y_{u_1,\cdot}})^2 \sum\limits_{m\in\mathcal{M}(u_1)\cap\mathcal{M}(u_2)} (y_{u_2,m} - \overline{y_{u_2,\cdot}})^2}} \times \ln(|\mathcal{M}(u_1)\cap\mathcal{M}(u_2)|)$$

Ainsi, le nombre de films en commun intervient directement dans cette formule : pour une même corrélation de Pearson, un voisin ayant plus de films en commun sera privilégié dans le cas où la similarité de Pearson était positive. Cette nouvelle définition nous est venue grâce à la méthode TF-IDF utilisée dans l'extraction de mots-clés. Cette fois-ci, les voisins avec une similarité élevée (et donc choisi par l'algorithme lors du calcul de la prédiction) seront de meilleurs qualités (c'est-à-dire qu'ils partageront davantage de films en commun avec l'utilisateur actif), comme le suggère la figure 3.

FIGURE 3 – Lien entre corrélation RFP et nombre de films en commun



Remarque : lors de l'implémentation du programme calculant les similarités en individus, la similarité avec soi-même n'a pas été calculée : la valeur NA était retournée. En effet, un utilisateur ne sera jamais considéré comme son propre voisin.
Deux autres similarités basées sur des notions de distance ont été implémentées dans le programme mais ne seront pas abordées dans le rapport.

2.2.2 Prédicteurs

Un prédicteur est une fonction qui retourne une note à partir de la matrice de similarité, du nombre de plus proches voisins et des notes des utilisateurs. Nous en avons explicité et implémenté cinq (liste non exhaustive).

Soient un utilisateur u_0 et un film m_0 tels que $(u_0, m_0) \in \Omega_0$. On notera dans la suite s_{u_1,u_2} la similarité entre les utilisateurs u_1 et u_2.
L'ensemble $\mathcal{N}_{K,s}(u_0, m_0) = \{u \in U | (u, m_0) \in \Omega_1$ et $s_{u_0,u}$ parmi les K plus grandes valeurs de $s_{u_0,\cdot}\}$ désigne les voisins considérés pour u_0 dans la prédiction de la note du film m_0 avec la similarité s, où $s_{u_0,\cdot}$ désigne l'ensemble des similarités de u_0 avec les autres utilisateurs. Nous pouvons remarquer que le cardinal de $\mathcal{N}_{K,s}(u_0, m_0)$ n'est pas fixe : pour un film m_0 vu moins de K, le cardinal vaut $|\mathcal{U}(m_0)|$, et K sinon.

Prédicteur mean Le premier consiste, pour un nombre de voisins K fixé à prédire :

$$\text{prédicteur } \texttt{mean} \qquad y_{u_0,m_0} := \frac{1}{|\mathcal{N}_{K,s}(u_0, m_0)|} \sum_{u \in \mathcal{N}_{K,s}(u_0,m_0)} y_{u,m_0},$$

où $|\mathcal{U}(m_0)|$ désigne le nombre d'utilisateurs ayant vu le film m_0. Dans cette première approche, la somme est effectuée sur les voisins de u_0 ayant vu le film considéré : la somme contient donc au maximum K termes.

Prédicteurs weighted et weighted&a Les deux suivants consistent à pondérer la note par la similarité : plus un voisin est proche de l'utilisateur considéré, plus sa note aura de l'importance dans la prédiction.
Ainsi les deux prédicteurs sont `weighted` et `weighted&a` (`&a` désigne la valeur absolue au dénominateur) :

$$\text{prédicteur } \texttt{weighted} \qquad y_{u_0,m_0} := \frac{\sum_{u \in \mathcal{N}_{K,s}(u_0,m_0)} y_{u,m_0} \times s_{u,u_0}}{\sum_{u \in \mathcal{N}_{K,s}(u_0,m_0)} s_{u,u_0}},$$

et

$$\text{prédicteur } \texttt{weighted\&a} \qquad y_{u_0,m_0} := \frac{\sum\limits_{u \in \mathcal{N}_{K,s}(u_0,m_0)} y_{u,m_0} \times s_{u,u_0}}{\sum\limits_{u \in \mathcal{N}_{K,s}(u_0,m_0)} |s_{u,u_0}|}$$

L'utilisation de la valeur absolue nous a été suggérée par l'article [**?**].

Prédicteurs weighted-centered et weighted-centered&a Les deux derniers prédicteurs reposent sur la pondération-centrée des notes : la note prédite dépend directement des moyennes des utilisateurs.
Ainsi, ces deux prédicteurs sont `weighted-centered` et `weighted-centered&a` :

$$\text{prédicteur } \texttt{weighted-centered} \qquad y_{u_0,m_0} := \overline{y_{u_0,\cdot}} + \frac{\sum\limits_{u \in \mathcal{N}_{K,s}(u_0,m_0)} (y_{u,m_0} - \overline{y_{u,\cdot}}) \times s_{u,u_0}}{\sum\limits_{u \in \mathcal{N}_{K,s}(u_0,m_0)} s_{u,u_0}},$$

et

$$\text{prédicteur } \texttt{weighted-centered\&a} \qquad y_{u_0,m_0} := \overline{y_{u_0,\cdot}} + \frac{\sum\limits_{u \in \mathcal{N}_{K,s}(u_0,m_0)} (y_{u,m_0} - \overline{y_{u,\cdot}}) \times s_{u,u_0}}{\sum\limits_{u \in \mathcal{N}_{K,s}(u_0,m_0)} |s_{u,u_0}|},$$

où $\overline{y_{u_0,\cdot}}$ désigne la note moyenne de l'utilisateur u_0.

D'autres prédicteurs font intervenir la variance des utilisateurs et des films, comme dans l'article de Herlocker and al (1999) [**?**]. Toutefois, cette approche ne semble pas donner de meilleurs résultats d'après les auteurs de cet article.

Remarque : le fait de considérer les valeurs absolues au dénominateur évite "l'explosion" des prédictions dans certains cas. En effet, si l'on considérons un K proche du nombre de visionnage du film en question, toutes les personnes ayant visionnées interviendront dans la prédiction. Nous considérerons donc dans la formule de prédiction de la note des individus tantôt proches (coefficient de corrélation positive) et tantôt éloignés (négative). La somme des similarités peut ainsi être proche de la valeur nulle, entraînant de fait l'"explosion" des prédictions. Pour pallier ce problème, nous avons considéré la somme des valeurs absolues (d'où le rajout de `&a`). De plus, lors de chacune des prédictions, une majoration à 5 et une minoration à 1 sera faite.