Projet Java DUT 1 Université Paris-Est Marne-la-Vallée
Système de recommandation

1 Modalités

Le projet est à rendre pour le 31 mai 2015. Il est à faire en binôme de deux étudiants d'un même groupe de TP. Le projet sera envoyé par email à vos enseignants de tp et cours sous la forme d'une archive `Nom1Nom2.zip` où `Nom1` et `Nom2` sont les noms des deux étudiants. Il y aura une petite soutenance de projet pendant la dernière séance de TP ou à une autre date.

L'archive doit contenir les fichiers et répertoires suivant :

- un fichier README au format texte qui indique les noms et prénoms des membres du binôme, et comment compiler et lancer le programme;
- un répertoire source (src) contenant l'ensemble des fichiers sources;
- un répertoire docs contenant la documentation;
- un manuel d'utilisation au format PDF (le fichier docs/user.pdf) indiquant l'ensemble des fonctionnalités et comment les faire fonctionner;
- une documentation pour développeur au format PDF (le fichier docs/dev.pdf) indiquant les choix pour la structure du programme, les algorithmes et méthodes utilisés, ainsi que les API et classes utilisées pour les implémenter.
- un répertoire (docs/api) contenant la documentation générée par javadoc.

2 Sujet du projet

Les systèmes de recommandation sont des systèmes permettant de recommander à des clients des produits susceptibles de les intéresser, par exemple des produits qui sont similaires à d'autres produits qu'ils ont déjà achetés et appréciés, ou des produits qu'ont déjà appréciés d'autres clients qui ont des goûts similaires.

Ils sont basés sur des algorithmes que l'on appelle algorithmes de recommandation. Ils sont utilisés par les fournisseurs de service en ligne, par exemple par ceux qui proposent des catalogues de vidéo à la demande comme Netflix.

Le projet consiste à implémenter un algorithme de recommandation très simple qui est décrit dessous.

3 Détail du sujet

3.1 Données

On suppose que le système de recommandation est réalisé par un fournisseur de films à la demande.

Le fournisseur dispose de données concernant ses clients. On suppose pour décrire le problème que l'on a n clients et m films disponibles dans le catalogue. Le fournisseur dispose pour chaque client de l'ensemble des films qu'il a déjà

loués avec une note d'évaluation de 0 à 10 (0 s'il n'a pas aimé du tout, 10 s'il a adoré le film).

Par exemple, le fournisseur dispose des informations suivantes

```
<client 0, film 1, note= 2>
<client 0, film 2, note= 7>
<client 0, film 3, note= 8>
<client 0, film 4, note= 6>
<client 1, film 0, note= 4>
<client 1, film 1, note= 1>
<client 1, film 3, note= 7>
<client 2, film 0, note= 3>
<client 2, film 1, note= 8>
<client 2, film 3, note= 4>
```

Chaque ligne indique qu'un client i a attribué une note à un film j.

Ces données peuvent se voir comme une matrice r appelée *matrice d'évaluation* (*rating matrix*). Ainsi

- r_{ij} est la note attribuée par le client i au film j si cette note existe
- $r_{ij} = -$ si la note du client i pour le film j n'est pas connue (soit il n'a pas loué le film, soit il ne l'a pas noté).

r	film 0	film 1	film 2	film 3	film 4
client 0	-	2	7	8	6
client 1	4	1	-	7	-
client 2	3	8	-	4	-

En pratique, les clients ne vont noter qu'un très petit nombre de films parmi tous les films disponibles et cette matrice a beaucoup de tirets. On dit qu'elle est creuse.

3.2 Similarité de deux clients

On estime que deux clients ont des comportements d'achat similaires si leur score de similarité est grand. Ce score de similarité dépend des notes attribuées par les deux clients et tient compte du fait que certains clients ont tendance à mettre des notes plus élevées ou plus basses que la moyenne.

Le *score de similarité* $\rho(i, i')$ entre deux clients i et i' est donné par la formule suivante. Tout d'abord on définit l'ensemble $K(i, i')$ des films j pour lesquels r_{ij} et $r_{i'j}$ sont connus (ensemble de films communs achetés et notés par i et i'). On calcule ensuite une correlation entre les notes de i et i' pour ces films communs.

$$\rho(i,i') = \frac{\sum_{j\in K(i,i')}(r_{ij} - E_K(r_i))(r_{i'j} - E_{K(i,i')}(r_{i'}))}{\sqrt{\sum_{j\in K(i,i')}(r_{ij} - E_{K(i,i')}(i))^2}\sqrt{\sum_{j\in K(i,i')}(r_{i'j} - E_{K(i,i')}(i'))^2}},$$

$$E_{K(i,i')}(i) = \frac{\sum_{j\in K(i,i')} r_{ij}}{|K(i,i')|}, \quad E_{K(i,i')}(i') = \frac{\sum_{j\in K(i,i')} r_{i'j}}{|K(i,i')|}.$$

où $E_{K(i,i')}(i)$ est la moyenne des notes du client i sur les films de $K(i, i')$. Les moyennes ne prennent en compte que les films notés en commun.

Par exemple, si $i = 0$ est le client 0 et $i' = 1$ le client 1, on a

$$K(\text{client 0, client 1}) = \{\text{film 1}, \text{film 3}\}.$$

Cet ensemble est de cardinal 2.

$$\rho(\text{client 0, client 1}) = \frac{(2-5)(1-4)+(8-5)(7-4)}{\sqrt{(2-5)^2+(8-5)^2}\sqrt{(1-4)^2+(7-4)^2}} = 1,$$

$$E_{K(\text{client 0, client 1})}(\text{client 0}) = \frac{(2+8)}{2} = 5,$$

$$E_{K(\text{client 0, client 1})}(\text{client 1}) = \frac{(1+7)}{2} = 4.$$

On calcule aussi

$$\rho(\text{client 0, client 2}) = -1,$$
$$\rho(\text{client 2, client 3}) \sim -0,756.$$

Les valeurs du score de similarité sont des réels entre -1 et 1. On considère que plus le score de similarité entre deux clients est élevé, plus ces clients ont des chances de souhaiter acheter les mêmes choses.

Remarquez que le score de similarité entre un client et lui-même est 1 (le score maximal). Si on multiplie toutes les notes d'un client par 2 par exemple, ou si on augmente toutes ses notes de 2 points, le score de similarité de ce client avec un autre client de change pas.

3.3 Recommandation

Pour un client i donné, le système de recommandation va choisir un autre client j qui a le plus fort score de similarité avec lui. Il va ensuite proposer au client i un article que j a noté et que i n'a pas acheté et qui a la plus forte note mise par i parmi ces articles.

Sur l'exemple, imaginons que l'on souhaite faire une proposition d'achat au client 1. On lui proposera le film 2. En effet, le client 0 est considéré comme le plus similaire par ses goûts au client 1 et parmi les films que le client 0 a vus (et notés) et que le client n'a pas encore achetés, il y a le film 2 noté 7 et le film 4 noté 6. Le système de recommandation proposera celui qui a été le plus apprécié par le client 0, donc le film 2.

3.4 Travail à réaliser

On demande de

- réfléchir à l'organisation du projet (répartition du travail dans le binôme, structure du projet, structures de données et classes choisies, hiérarchie des classes, etc). On décrira les structures de données et classes Java choisies dans la partie documentation développeur;
- construire un petit jeu de données : clients , films et notes. On devra pouvoir lire les données dans un fichier texte. Chaque ligne du fichier contient une "data" [1] du type suivant

[1] Les "training data" proposées par Netflix pour son concours jusqu'en 2009 (voir Netflix prize sur wikipedia) étaient de type <user, movie, date of grade, grade> avec la date de notation en plus.

```
<user, movie, rate>
```

Exemple de fichier texte de données

```
<Pierre Dupont, Harry Potter 1, 5>
<Pierre Dupont, Harry Potter 2, 6>
<Pierre Dupont, Blade Runner, 5>
<David Dupond, Le Bon la brute et le truand, 2>
```

- pouvoir afficher la liste des films achetés par un client avec les notes associées;
- programmer le calcul des scores de similarité entre deux clients;
- pouvoir fournir une recommandation de film à un client;
- écrire les documentations demandées dans la section Modalités;
- prévoir d'arriver à la soutenance de projet avec un projet compilé et exécutable;
- prévoir une démonstration pour la soutenance avec quelques exemples pertinents.

Prévoyez des améliorations uniquement si vous avez le temps. Commencez par quelque chose de minimal et qui "marche".