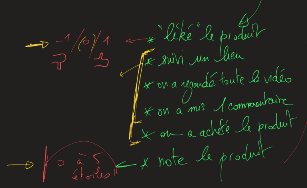


Systèmes de recommandation

But : recommander des "produits" à des utilisateurs au bon goût

Exemple : l'utilisateur a déjà visité le site
→ on connaît quelques-uns de ses goûts



Idée de base : recommander les produits appréciés par les utilisateurs qui ont les mêmes goûts que vous

1^{er} algo. basique : "collaborative filtering"

note sur 5	Inception	OSS 117	Shrek	similitude avec Beke
Antoine	4	3	1	0,46
Beke	2	5	1	
Camille	5		3	0,4
Daniel	1	4		0,48

Etape 1 : on calcule le vecteur normalisé de chaque utilisateur
Δ on normalise seulement sur les coefficients des communes

→ on calcule le produit scalaire de Beke avec tous les autres utilisateurs

→ un coef de similitude

Etape 2 : on calcule la moyenne des notes par le film considéré pondérée par le coef de similitude.

note entre 0 et 1

	Inception	OSS 117	Shrek
Antoine	0,8	0,6	0,2
Beke	0,4	1	
Camille	1		0,6
Daniel	0,2	0,8	

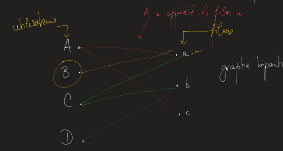
$$\frac{0,4 \times 1}{1} = 0,4$$

$$\frac{0,6 \times 0,2 + 1 \times 0,8}{2} = 0,48$$

Algo 2

Idée : classer les films (par note) dans une liste par ordre "croissant" de préférence

Exemple : l'utilisateur peut avoir une liste de films



1^{re} tentative : proposer "basique" → classer les films en fonction de la note de l'utilisateur

2^{de} tentative : Idée : se recentrer sur l'utilisateur et sélectionner une "variante" "personnalisée" dans la même alternative

Rem : 1 utilisateur qui aime un film donne une information plus précieuse qu'un utilisateur qui aime beaucoup de films

2 films aimés par peu de monde donne plus d'information à l'algo

→ simulation alternative

$$\frac{1 \times 0,46 + 3 \times 0,4}{0,4 + 0,46} = \frac{1,66}{0,86} \approx 2$$

Equations du p. n. basique

$$\pi_A = \frac{\pi_A}{2} + \frac{\pi_B}{1} + \frac{\pi_C}{2}$$

$$\pi_B = \frac{\pi_C}{2} + \frac{\pi_D}{1}$$

$$\pi_C = \frac{\pi_A}{2}$$

$$\pi_A = \frac{\pi_A}{3} + \frac{\pi_C}{1}$$

$$\pi_B = \frac{\pi_A}{3}$$

$$\pi_C = \frac{\pi_A}{3} + \frac{\pi_D}{2}$$

$$\pi_D = \frac{\pi_B}{2}$$

on résout ce système (on utilise la mult. de matrice)

→ on obtient

les scores π_A, π_B, π_C

Pb : les scores ne sont pas spécifiques à un utilisateur.

2^{de} tentative

Idée : se recentrer sur l'utilisateur et sélectionner une "variante" "personnalisée" dans la même alternative



on utilise une matrice de films (ex. 2 matrices) grand (n x m) et donc pour E on résout au moins de l'utilisateur B

m utilisateurs m films

U matrice m x m

π_U vecteur taille n pour les utilisateurs

π_F matrice m x m

vecteur taille m pour les films

$$U \pi_U = \pi_F$$

$$U_{ij} = \begin{cases} \frac{1}{d} & \text{si l'utilisateur } i \text{ aime le film } j \\ 0 & \text{sinon} \end{cases}$$

$$F = (1 - \epsilon) \times \begin{cases} \frac{1}{d} & \text{si } j \text{ aime le film } i \\ 0 & \text{sinon} \end{cases} + \begin{cases} \epsilon & \text{si } j = B \\ 0 & \text{sinon} \end{cases}$$

Algo

$$\pi_U \leftarrow \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \leftarrow B$$

$$\pi_F \leftarrow \begin{pmatrix} 1/m \\ \vdots \\ 1/m \end{pmatrix}$$

Répéter k fois :

$$\tilde{\pi}_U \leftarrow \pi_U$$

$$\pi_U \leftarrow F \cdot \pi_F$$

$$\pi_F \leftarrow U \cdot \tilde{\pi}_U$$

Classer π_F par score décroissant
Proposer à B les films de score élevé.