

Filtrage Collaboratif avec un Algorithme d'Ordonnement

JF. Pessiot, V. Truong, N. Usunier, M. Amini et P. Gallinari
LIP6 - Université Pierre et Marie Curie

Petit Problème

JF, Vinh et Nico ont donné leur avis sur 4 films :

	La Même	Les Bronzés	Inland Empire	Taxi 4
JF	4	?	5	1
Vinh	?	?	4	2
Nico	1	3	?	?

Plan

Filtrage Collaboratif

Présentation

Prédiction de Notes

Approche Alternative

Ordonnancement pour le Filtrage Collaboratif

Un Modèle d'Ordonnancement

Algorithme et Implémentation

Expériences et Résultats

Conclusion et Perspectives

Plan

Filtrage Collaboratif

Présentation

Prédiction de Notes

Approche Alternative

Ordonnancement pour le Filtrage Collaboratif

Un Modèle d'Ordonnancement

Algorithme et Implémentation

Expériences et Résultats

Conclusion et Perspectives

Définition

Definition

Filtrage Collaboratif :

- permet d'automatiser la sélection et la recommandation d'articles (**filtrage**)
- sert à prédire les préférences d'un utilisateur à partir des préférences d'autres utilisateurs (**collaboratif**)

Applications et Enjeux

Les systèmes de FC sont utilisés pour recommander des produits culturels :

- des films (MovieLens, Ymdb)
- de la musique (Lastfm, Indy)
- des pages web (Del.icio.us, StumbleUpon)

Le FC permet de générer des recommandations **personnalisées** à l'utilisateur. Pour les sites commerciaux (Amazon, iTunes, CDnow...), il présente donc des enjeux financiers importants.

Approche Standard

Approche standard repose sur la prédiction de notes :

- considérer la matrice de notes (utilisateurs \times articles)
- matrice à «trous»
- tâche = prédiction des notes manquantes

Exemple

	La Même	Les Bronzés	Inland Empire	Taxi 4
JF	4	?	5	1
Vinh	?	?	4	2
Nico	1	3	?	?



$$\begin{pmatrix} 4 & 2 & 5 & 1 \\ 5 & 1 & 4 & 2 \\ 1 & 3 & 2 & 4 \end{pmatrix}$$

Etat de l'Art

De nombreuses approches pour la prédiction de notes :

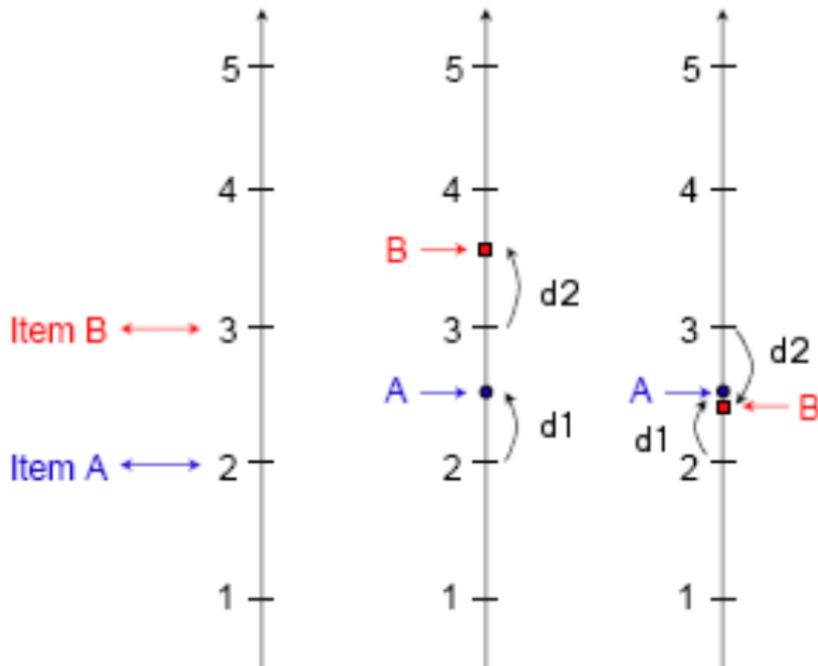
- modèles probabilistes (Hoffman 2004, Marlin 2004)
- modèles factoriels linéaires (Srebro 2003, Rennie 2005, Pessiot 2006)
- ordonnancement d'instances (Crammer 2002, Yu 2006)

Remarque sur la Prédiction de Notes

- but d'un système de recommandation = recommander N articles à chaque utilisateur
- l'ordre des articles est plus important que la prédiction des notes

Prédire ordre des articles \neq prédire les notes

Exemple



Prédiction de Préférences

Pour chaque utilisateur, prédire directement l'ordre des articles :

- prédire un score réel pour chaque article
- ordonner les articles en fonction des scores
- présenter les N articles correspondant aux meilleurs scores

Exemple

	La Même	Les Bronzés	Inland Empire	Taxi 4
JF	4	?	5	1
Vinh	?	?	4	2
Nico	1	3	?	?



$$\begin{pmatrix} 0.6 & -0.1 & 0.8 & -0.4 \\ 0.9 & -0.7 & 0.5 & -0.2 \\ -0.8 & 0.7 & -0.4 & 0.8 \end{pmatrix}$$

Plan

Filtrage Collaboratif

Présentation

Prédiction de Notes

Approche Alternative

Ordonnancement pour le Filtrage Collaboratif

Un Modèle d'Ordonnancement

Algorithme et Implémentation

Expériences et Résultats

Conclusion et Perspectives

Définitions

Notations :

- utilisateur x représenté par $u_x \in \mathbb{R}^k$
- article a représenté par $i_a \in \mathbb{R}^k$
- intérêt de a pour x : $\langle i_a, u_x \rangle$

Formalisation matricielle :

- U matrice des n utilisateurs ($n \times k$)
- I matrice des p articles ($k \times p$)
- $(UI)_{xa} = \langle i_a, u_x \rangle$

Préférences par Paires

- utilisateur x , préférence par paire (a, b) (a est préféré à b)
- modèle (U, I)
- si (a, b) et $(UI)_{xa} \leq (UI)_{xb}$, alors la prédiction est fausse

$[[(UI)_{xa} \leq (UI)_{xb}]]$ est l'erreur de prédiction du modèle sur (a, b)

$$D(U, I) = \sum_x \sum_{(a,b) \in y_x} [[(UI)_{xa} \leq (UI)_{xb}]]$$

Fonction de Coût

Nombre d'erreurs sur les préférences par paires :

$$D(U, I) = \sum_x \sum_{(a,b) \in y_x} [|(UI)_{xa} \leq (UI)_{xb}|]$$

Borne supérieure :

$$D(U, I) \leq \sum_x \sum_{(a,b) \in y_x} e^{(UI)_{xb} - (UI)_{xa}}$$

Fonction de coût régularisée :

$$\mathcal{R}(U, I) = \sum_x \sum_{(a,b) \in y_x} e^{(UI)_{xb} - (UI)_{xa}} + \mu_U \|U\|^2 + \mu_I \|I\|^2$$

Algorithme

Problème d'optimisation

$$(U^*, I^*) = \underset{U, I}{\operatorname{argmin}} \mathcal{R}(U, I)$$

Algorithme

- Initialiser U, I
- Répéter jusqu'à convergence :
 - 1 $U \leftarrow \operatorname{argmin}_U \mathcal{R}(U, I)$
 - 2 $I \leftarrow \operatorname{argmin}_I \mathcal{R}(U, I)$

Implémentation

Problème :

- considérer toutes les paires d'articles possibles
- calculs en $O(nvkp^2)$
- n utilisateurs, notes de 1 à v , k variables latentes et p articles
- inapplicable sur des bases réelles

Stratégie :

- parcours «intelligent» de l'ensemble des paires d'articles
- complexité en $O(nvkp)$
- linéaire en p

Base de Test

Base de films MovieLens :

- 6,040 utilisateurs et 3,706 films
- 1,000,209 notes de 1 à 5
- 95.5% de notes manquantes

Génération des bases :

- pour chaque utilisateur, choisir 2 notes de test
- utiliser le reste pour l'apprentissage
- total de 10 bases pour l'apprentissage / test

Evaluation

Mesure d'évaluation :

$$MRE = \frac{1}{n} \sum_x [|(I^T u_x)_{a_x} \leq (I^T u_x)_{b_x}|]$$

Deux approches pour la prédiction de notes :

- Factorisation Matricielle Non-négative (FMN)
- Décomposition en Valeurs Singulières (DVS)

Résultats

	FMN gén.	DVS pond.	Ordo.
k	9	9	8
MRE	0.2658	0.2770	0.2737

Améliorations :

- Normalisation par utilisateur
- Critères d'arrêt
- Coefficients de régularisation

Plan

Filtrage Collaboratif

Présentation

Prédiction de Notes

Approche Alternative

Ordonnancement pour le Filtrage Collaboratif

Un Modèle d'Ordonnancement

Algorithme et Implémentation

Expériences et Résultats

Conclusion et Perspectives

Conclusion

- Nouvelle approche basée sur la prédiction des préférences
- Algorithme optimisant erreur d'ordonnancement
- Résultats encore à interpréter et à améliorer

Perspectives

- Liens avec l'apprentissage multi-tâches
- Extension à d'autres cadres (exemple : apprentissage semi-supervisé)
- Démo en ligne !

`http://connex.lip6.fr/~pessiot`