

MLWEEK - Formation Machine Learning

Cours 9 : Complements

Systèmes de recommandation

Jeff Abrahamson

François-Marie Giraud

12 Juin 2019



<https://www.ml-week.com/>

Machine Learning

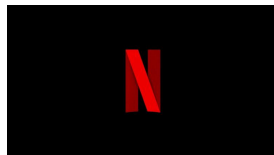
Cours 9 : Complements
Systèmes de recommandation

Machine Learning

Système de recommandation

Filtrage de l'information :

Produire une information personnalisé pour un "client"



Intérêt mutuel :

- Facilitation des usages pour le “client”
- Des bénéfices accrus pour le “vendeur”

3 types de recommandation :

- centrée sur l'utilisateur (user-based)
- centrée sur le contenu (item-based)
- collaborative (ou sociale)

2 types de collecte des données :

- active
- passive

Machine Learning

Filtrage Collaboratif

≈ Système de recommandation

Méthode hybride :

- user-based
- item-based
- collaboratif

Données explicites et implicites :

Explicite : donnée de qualité mais contraignante pour l'utilisateur (donc potentiellement biaisé)

Implicite : donnée objective mais pauvre du point de vue de la qualité

Grosse volumétrie de données \Rightarrow algorithmes simples et rapides

Machine Learning

Filtrage Collaboratif user-based

Exemple de données collectées :

	film 1	film 2	film 3	film 4
u1	?	2	5	?
u2	5	1	4	3
u3	?	?	1	5
u4	3	3	?	4

Principe : Une bonne recommandation est faite à partir d'individus similaires

Condition : Avoir un échantillon important d'individus représentatifs



Filtrage Collaboratif user-based

	film1	film2	film3	film4
u1	?	2	5	?
u2	5	1	4	3
u3	?	?	1	5
u4	3	3	?	4

Comment prédire les notes manquantes ? (et donc faire une recommandation)

Algorithme pour déterminer $u1(\text{film1})$:

- Identifier les K plus proches utilisateurs de u1
- Agréger leurs notes pour le film1

On a donc besoin :

- d'une mesure de similarité entre les utilisateur
- de trouver un K optimal
- d'une fonction d'agrégation

Similarité de corrélation (Pearson) :

$$\text{cor}(a, b) = \frac{\sum_{i \in I} (a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i \in I} (a_i - \bar{a})^2 * \sum_{i \in I} (b_i - \bar{b})^2}}$$

Similarité cosinus :

$$\text{cos}(a, b) = \frac{a * b^t}{|a| * |b|}$$

Et plein d'autres dans la littérature...

Agrégation des notes des K utilisateurs retenus :

- Moyenne
- Moyenne pondéré par la similarité
- ...

- Des calculs simples mais il faut parcourir toute la base dès qu'on a une nouvelle entrée ou modification, heureusement des heuristiques existent (Locality-sensitive hashing)
- “Cold start problem”

Machine Learning

Filtrage Collaboratif item-based

Filtrage Collaboratif item-based

	item 1	item 2	item 3	item 4
u1	1	2	5	?
u2	5	1	4	3
u3	?	?	1	5
u4	3	3	?	4

Pour remplir les cellules manquantes, au lieu de traiter les lignes on traite les colonnes.

Une première solution est de transposer la matrice et d'appliquer l'algorithme décrit pour le user-based.

Filtrage Collaboratif item-based

Méthode couramment utilisée \Rightarrow régression simplifiée.

	item 1	item 2	item 3	item 4
u1	1	2	5	?
u2	5	1	4	3
u3	?	?	1	5
u4	3	3	?	4

Filtrage Collaboratif item-based

Sachant la colonne item1, on cherche uniquement une valeur de biais telle que :

$$item4 = item1 + b_1$$

Pour ce faire, on prend le biais moyen entre ces deux colonnes :

$$b_1 = \frac{\sum_{i \in I} (item4_i - item1_i)}{\#I}, \text{ dans notre exemple : } b_1 = \frac{(3-5)+(4-3)}{2} = -0.5$$

	item 1	item 2	item 3	item 4
u1	1	2	5	?
u2	5	1	4	3
u3	?	?	1	5
u4	3	3	?	4

Filtrage Collaboratif item-based

On calcul alors les biais pour chaque colonne, nous donnant une prediction de la valeur $u1(item4)$ sachant $u1(item1)$, une autre sachant $u1(item2)$ et une dernière sachant $u1(item3)$.

La prédiction finale pour $u1(item4)$ est la moyenne pondérée de toutes ces prédictions.

$$b_1 = -0.5 \Rightarrow u1(item4)_1 = 0.5$$

$$b_2 = 1.5 \Rightarrow u1(item4)_2 = 2.5$$

$$b_3 = 1.5 \Rightarrow u1(item4)_3 = 2.5$$

$$u1(item4) = \frac{\#l_1 * 0.5 + \#l_2 * 2.5 + \#l_3 * 2.5}{\#l_1 + \#l_2 + \#l_3} = \frac{11}{6} \approx 1.83$$

	item 1	item 2	item 3	item 4
u1	1	2	5	1.83
u2	5	1	4	3
u3	?	?	1	5
u4	3	3	?	4

- Calculs simples à mettre en oeuvre et à tenir à jour
- Beaucoup de paramètres à maintenir quand on a beaucoup d'item :
 $\left(\frac{\#item(\#item-1)}{2} \right)$
- "Cold start problem"

Machine Learning

Filtrage Colaboratif - Évaluation

On utilise des techniques d'évaluation du machine learning impliquant des bases d'apprentissage et de test.

$$RMSE = \sqrt{\frac{\sum_{u,i} (u(i)_{pred} - u(i))^2}{\#Q}}$$

où $\#Q$ est le nombre de prédictions que l'on fait dans la base.

Machine Learning

Filtrage Collaboratif - Aujourd'hui

Filtrage Collaboratif - SVD

Utilisation de SVD pour réduire le nombre de dimensions du problème.

$M = U^P * V^P * I^P$, où :

M est la matrice user/item de \mathbf{R}^{n*m}

U^P la matrice des vecteurs de base orthonormée de \mathbf{R}^n (input)

V^P la matrice diagonale des valeurs singulières de U

I^P la matrice des vecteurs de base orthonormée de \mathbf{R}^m (output)

$$M = \begin{array}{|c|c|c|} \hline 2 & 5 & ? \\ \hline 1 & 4 & 3 \\ \hline ? & 1 & 5 \\ \hline 3 & ? & 4 \\ \hline \end{array} =$$

$$\begin{array}{|c|c|} \hline U^P_{(1,1)} & U^P_{(1,2)} \\ \hline U^P_{(2,1)} & U^P_{(2,2)} \\ \hline U^P_{(3,1)} & U^P_{(3,2)} \\ \hline U^P_{(4,1)} & U^P_{(4,2)} \\ \hline \end{array} * \begin{array}{|c|c|} \hline V^P_{(1)} & 0 \\ \hline 0 & V^P_{(2)} \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline I^P_{(1,1)} & I^P_{(1,2)} & I^P_{(1,3)} \\ \hline I^P_{(2,1)} & I^P_{(2,2)} & I^P_{(2,3)} \\ \hline \end{array}$$

Avantages :

On choisit le nombre de valeurs singulières de notre transformation (PCA).

Filtrage collaboratif user-based $\Rightarrow U^P$

Filtrage collaboratif item-based $\Rightarrow I^P$

La SVD est extrêmement coûteuse sur des grandes matrices :

$$O(\min(m * n^2, m^2 * n))$$

Heureusement des algorithmes approchés existent. Ces approches tirent parti du fait que les matrices sont généralement sparses et que l'on ne s'intéresse qu'à un nombre limité de valeurs singulières.

Par exemple, on peut calculer les 20 premières valeurs singulières d'une matrice 100k x 100k avec 1M valeurs non-nulles en moins d'une seconde (redsvd).

Utilisation d'Embeddings sur les utilisateurs et les items afin de pouvoir exprimer la matrice des ratings ainsi :

$$M = U^{embed} * I^{embed}$$

Pour Faire une recommandation :

$$\operatorname{argmax}(u_{query} * I^t)$$

Jusqu'à présent, la nature des items était sans importance.

Que les items soient des livres ou des voitures, l'algorithme ne faisait aucune différence.

On peut utiliser les détails textuels, les images (, etc...) sur les items afin d'améliorer les recommandations. (Embeddings, Catégorisation des items)

Machine Learning

Filtrage Collaboratif - Conclusions

- Défis technologique :
 - Grande volumétrie \Rightarrow algorithmes simples et rapides
 - Quel est le minimum nécessaire d'information sur un utilisateur ?
- Défis sociologique :
 - aimer \neq acheter
 - prédiction par agrégation \Rightarrow peu propice à l'exploration

Avez-vous des questions ?

TP 5

Filtrage Colaboratif

TP 5 : Filtrage Colaboratif

www.filtrage-colaboratif.ipynb