

# Machine Learning

Filtrage Collaboratif user-based

---

# 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

# Filtrage Collaboratif user-based

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)



- 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”