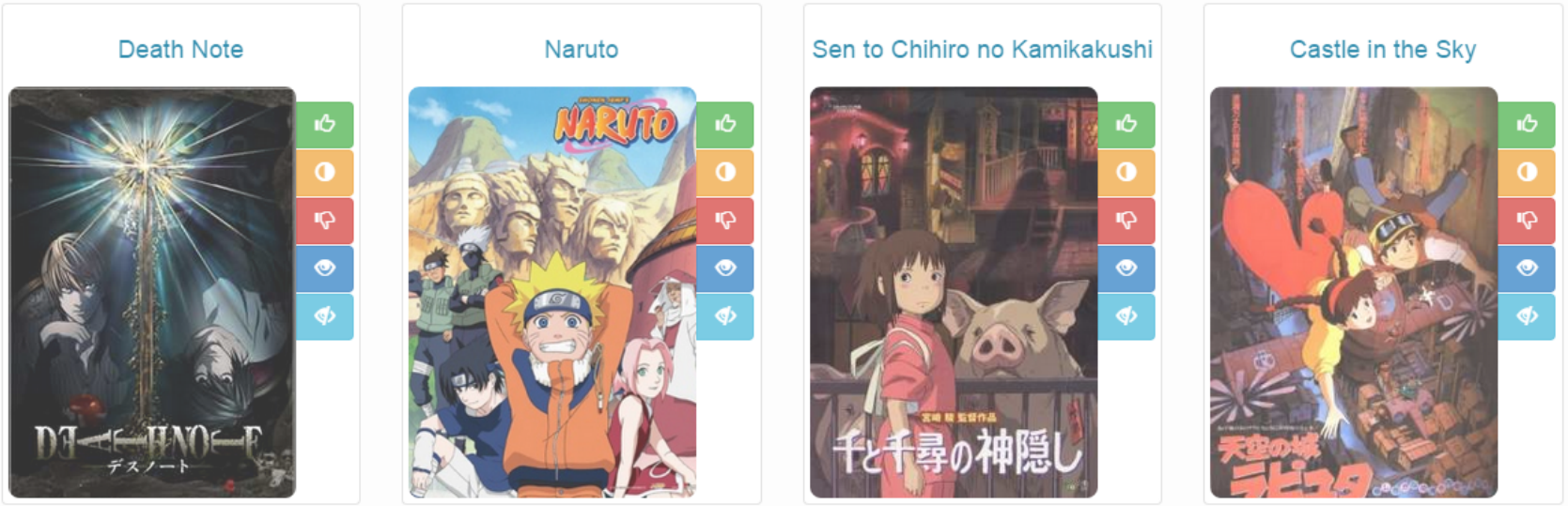
Mangaki.fr, système de recommandation de mangas et d'anime

Jill-Jênn Vie

Lycée Thiers

16 octobre 2015

Un système de recommandation



Principe

- Un utilisateur s'inscrit et rentre ses préférences
- Le système lui recommande des films susceptibles de lui plaire

Objectifs

- Elles doivent être pertinentes (sinon l'utilisateur s'en va)
- Rapides à calculer (sinon l'utilisateur s'en va)

Filtrage collaboratif

Problème

- On dispose d'utilisateurs $u = 1, \ldots, n$ et d'items à noter $i = 1, \ldots, m$
- Chaque utilisateur u attribue une note à une partie des items (r_{ui} : note de l'utilisateur u sur l'item i)

$\Rightarrow$ Quels nouveaux items recommander à chaque utilisateur ?

Exemple (notes sur 5)

	Death Note	*L'Attaque des titans*	*Naruto*	*Bleach*
Sacha	*****	****	?	?
Ondine	**	?	*	?
Pierre	*	?	****	?

Objets : n vecteurs à m dimensions, éléments de $\{-1, 0, 1\}^m$

k plus proches voisins

Intuition

- On introduit un score de similarité entre utilisateurs
- On détermine les k utilisateurs les plus proches d'un utilisateur u
- On lui recommande ce qu'ils ont aimé qu'il n'a pas vu

Produit scalaire

Soit n un entier, u et v deux vecteurs de $\mathbb{R}^n$:

- $u = (u_1, \ldots, u_n)$
- $v = (v_1, \ldots, v_n)$

Le produit scalaire de u et v est donné par :

$$\begin{aligned} u \cdot v &= u_1 v_1 + \ldots + u_n v_n \\ &= ||u|| \cdot ||v|| \cdot cos(u, v). \end{aligned}$$

Similarité

$\mathcal{R}_u$: le vecteur de notes $(r_{u1}, r_{u2}, \ldots, r_{um})$ $\quad u = 1, \ldots, n$
Le score de similarité entre 2 utilisateurs u et v est donné par :

$$score(u, v) = \mathcal{R}_u \cdot \mathcal{R}_v.$$

Intuition

Les points communs augmentent le score :

	Paprika	*Oldboy*	*Gattaca*	*12 Monkeys*
Alice	1	-1	0	0
Bob	1	1	-1	0
Charles	1	-1	1	-1

$$score(Alice, Bob) = 1 + (-1) = 0$$
$$score(Alice, Charles) = 1 + 1 = 2$$

Alice est plus proche de Charles que de Bob.

Estimation des notes inconnues

$N(u)$: les k plus proches voisins de u, $\quad u = 1, \ldots, n$
notés $\{v_1, \ldots, v_k\}$

$$\widehat{r_{ui}} = \frac{r_{v_1 i} + \ldots + r_{v_k i}}{k}$$

On calcule $\widehat{r_{ui}}$ pour chaque film i non noté $\Rightarrow$ les 10 meilleurs.

Version pondérée : les plus proches ont plus de poids

$$\widehat{r_{ui}} = \frac{\sum_{v \in N(u)} w_v \times r_{vi}}{\sum_{v \in N(u)} w_v} \quad \text{où } w_v = score(u, v)$$

Questions :

- Comment choisir la bonne valeur de k dans « k plus proches voisins » ?

Variantes :

- Faire la similarité non sur les utilisateurs sur mais sur les films.

Évaluation du modèle par validation croisée

À quel point j'ai bien recommandé ?

- Je suppose que je connais 80 % des utilisateurs (*train*)
- Je teste les recommandations sur les 20 % restants (*test*)

Pénalité : les moindres carrés

$$RMSE = \sum_{u,i} (\widehat{r_{ui}} - r_{ui})^2.$$

Une autre méthode : complétion de matrice

Supposons que la matrice M de notes soit de faible rang r :

$$M = \begin{pmatrix} \mathcal{R}_1 \\ \mathcal{R}_2 \\ \vdots \\ \mathcal{R}_n \end{pmatrix} = \boxed{} = \boxed{C} \boxed{P}$$

Chaque ligne $\mathcal{R}_u$ est une combinaison linéaire des lignes de P.

$$M : n \times m \qquad C : n \times r \qquad P : r \times m.$$

$$\mathcal{R}_1 = c_{11} P_1 + c_{12} P_2 + \ldots + c_{1r} P_r \qquad C_1 = (c_{11}, c_{12}, \ldots, c_{1r})$$

Exemple

Si P	P_1 : « aventure »	P_2 : « romance »	P_3 : « plot twist »
Si C_u	0,2	$-0{,}5$	0,6

ça veut dire :

j'aime un peu l'aventure, je n'aime pas la romance,
j'aime beaucoup les plot twists.

Mon sujet de thèse

Sur quels items vaut-il mieux sonder un nouveau venu pour le profiler efficacement ?

- populaires, pour que l'utilisateur puisse les noter
- controversés, pour que ce soit informatif

(Arbres de décision, tests adaptatifs.)

Quelques liens

Merci de votre attention !

- mangaki.fr
- jjv@lri.fr

(P. S. – C'est la Code Week, apprenez à coder et tentez le concours Prologin.org, c'est gratuit et sans obligation d'achat !)