Algorithmes de recommandation : comment ça marche ?




Jill-Jênn Vie[13] Solène Pichereau[3] Ryan Lahfa[3] Hisashi Kashima[2]
Basile Clement[3] Kévin Cocchi[3] Thomas Chalumeau[3] Florian Yger[4]

[1] Inria

[2] Université de Kyoto & RIKEN AIP (Tokyo)

[3] Mangaki (Paris, France)

[4] Université Paris-Dauphine

MANGAKI
LA RECOMMANDATION FRANCOPHONE D'ANIME ET MANGA

Recommandation d'articles

Mangaki, recommandations d'anime/manga

Notez des anime/manga et recevez des recommandations

380 000 notes de 8 000 utilisateurs sur 29 000 œuvres

Codé en Python

- Tout le code est libre et ouvert : github.com/mangaki

Prix : Microsoft Prize (2014) Japan Foundation (2016)

Financé par NLnet (Pays-Bas), Commission européenne

Systèmes de recommandation

Problem

- Chaque personne note peu d'œuvres (1 %)
- Comment inférer les notes manquantes ?

Example

Sacha	?	5	2	?
Ondine	4	1	?	5
Pierre	3	3	1	4
Joëlle	5	?	2	?

Systèmes de recommandation

Problem

- Chaque personne note peu d'œuvres (1 %)
- Comment inférer les notes manquantes ?

Example

				
Sacha	3	5	2	2
Ondine	4	1	4	5
Pierre	3	3	1	4
Joëlle	5	2	2	5

Systèmes de recommandation

Problem

- Chaque personne note peu d'œuvres (1 %)
- Comment inférer les notes manquantes ?

Example

Sacha	?	5	2	?
Ondine	4	1	?	5
Pierre	3	3	1	4
Joëlle	5	?	2	?

Systèmes de recommandation

Problem

- Chaque personne note peu d'œuvres (1 %)
- Comment inférer les notes manquantes ?

Example

Sacha	3	5	2	2
Ondine	4	1	4	5
Pierre	3	3	1	4
Joëlle	5	2	2	5

Qu'est-ce qu'un algorithme de machine learning ?

Fit (entraîner)

Ondine	aime	*Zootopie*
Ondine	adore	*Porco Rosso*
Sacha	adore	*La Traversée du temps*
Sacha	n'aime pas	*Seul sur Mars*

Predict (prédire)

Ondine	?	*Seul sur Mars*
Sacha	?	*Zootopie*

Qu'est-ce qu'un algorithme de machine learning ?

Fit (entraîner)

Ondine	aime	*Zootopie*
Ondine	adore	*Porco Rosso*
Sacha	adore	*La Traversée du temps*
Sacha	n'aime pas	*Seul sur Mars*

Predict (prédire)

Ondine	adore	*Seul sur Mars*
Sacha	aime	*Zootopie*

Qu'est-ce qu'un mauvais algorithme de machine learning ?

Fit

Ondine	like	*Zootopie*
Ondine	favorite	*Porco Rosso*
Sacha	favorite	*La Traversée du temps*
Sacha	dislike	*Seul sur Mars*

100 % correct

Predict

Ondine	n'aime pas	*Seul sur Mars* (en fait : adore)
Sacha	neutre	*Zootopie* (en fait : aime)

20 % correct

N'arrive pas à généraliser

Qu'est-ce qu'un bon algorithme de machine learning ?

Fit

Ondine	adore	*Zootopie* (en fait : aime)
Ondine	adore	*Porco Rosso*
Sacha	adore	*La Traversée du temps*
Sacha	n'aime pas	*Seul sur Mars*

90 % correct

Predict

Ondine	aime	*Seul sur Mars* (en fait : adore)
Sacha	adore	*Zootopie* (en fait : aime)

90 % correct

Comment comparer des méthodes ?

n'aime pas	ne verra pas	neutre	veut voir	aime	adore
-2	-0.5	0.1	0.5	2	4

Pénalités

Si je prédis : adore pour adore $\rightarrow$ erreur de 0

n'aime pas pour adore $\rightarrow$ erreur de $(4-(-2))^2 = 36$

aime for adore $\rightarrow$ erreur de 4

Erreur : moyenne des (différences)^2

RMSE : racine carrée de la valeur ci-dessus

Validation croisée

- On entraîne sur 80 % des données (entraînement)
- On cache les 20 % de données restantes (validation)
- On s'en sert pour évaluer la performance des modèles

Plus l'erreur de validation est basse, mieux c'est

Algorithme des plus proches voisins

Pour recommander des films à quelqu'un :

- On introduit un score de similarité entre personnes
- On détermine les 10 personnes les plus proches de quelqu'un
- On lui recommande ce qu'elles ont aimé qu'il n'a pas vu

Nos données

	007	Batman 1	Shrek 2	Toy Story 3	Star Wars 4	Twilight 5
Alice	+	−	0	+	0	−
Bob	−	0	+	−	+	+
Charles	+	+	+	+	−	−
Daisy	+	+	0	0	+	−
Everett	+	−	+	+	−	0

Quel score de similarité entre utilisateurs choisir ?

Calcul du score

	007	Batman 1	Shrek 2	Toy Story 3	Star Wars 4	Twilight 5
Alice	$+$	$-$	0	$+$	0	$-$
Charles	$+$	$+$	$+$	$+$	$-$	$-$

À quel point Alice est proche de Charles ?

Calcul du score

	007	Batman 1	Shrek 2	Toy Story 3	Star Wars 4	Twilight 5
Alice	$+$	$-$	0	$+$	0	$-$
Charles	$+$	$+$	$+$	$+$	$-$	$-$
Score	$+1$	-1		$+1$		$+1$

$$score(\text{Alice}, \text{Charles}) = 3 + (-1) = 2$$

	007	Batman 1	Shrek 2	Toy Story 3	Star Wars 4	Twilight 5
Alice	$+$	$-$	0	$+$	0	$-$
Bob	$-$	0	$+$	$-$	$+$	$+$
Score	-1			-1		-1

$$score(\text{Alice}, \text{Bob}) = -3$$

Alice est plus proche de Charles que de Bob

Score de similarité entre personnes

	Alice	Bob	Charles	Daisy	Everett
Alice	4	−3	2	1	3
Bob	−3	5	−3	−1	−2
Charles	2	−3	6	2	3
Daisy	1	−1	2	4	−1
Everett	3	−2	3	−1	5

Qui sont les 2 plus proches voisins d'Alice ?

Calcul des prédictions

	007	Batman 1	Shrek 2	Toy Story 3	Star Wars 4	Twilight 5
Alice	+	−	?	+	?	−
Charles	+	+	+	+	−	−
Daisy	+	+	0	0	+	−
Everett	+	−	+	+	−	0

Connaissant ses voisin·es, quelles sont les chances d'Alice d'apprécier ces films ?

Calcul des prédictions

	007	Batman 1	Shrek 2	Toy Story 3	Star Wars 4	Twilight 5
Alice	$+$	$-$	$+$	$+$	$-$	$-$
Charles	$+$	$+$	$+$	$+$	$-$	$-$
Daisy	$+$	$+$	0	0	$+$	$-$
Everett	$+$	$-$	$+$	$+$	$-$	0

On peut calculer la moyenne :

$$\textit{prediction}(\text{Alice}, \text{Star Wars 4}) = -0{,}333\ldots$$

Factorisation de matrice $\rightarrow$ réduire la dimension pour généraliser

Idée : Apprendre une représentation des utilisateurs et œuvres de sorte que les gens aiment des œuvres proches d'eux

Fit

- R notes, U vecteurs des utilisateurs, W vecteurs des œuvres.

$$R = UW^T \qquad \hat{r}_{ij}^{ALS} = \langle U_i, W_j \rangle$$

Factorisation de matrice → réduire la dimension pour généraliser

Idée : Apprendre une représentation des utilisateurs et œuvres de sorte que les gens aiment des œuvres proches d'eux

Fit

- R notes, U vecteurs des utilisateurs, W vecteurs des œuvres.

$$R = UW^T \qquad \hat{r}_{ij}^{ALS} = \langle U_i, W_j \rangle$$

Predict : Est-ce que l'utilisateur i va aimer l'œuvre j?

- Il suffit de calculer $\langle U_i, W_j \rangle$ pour le savoir

Algorithme ALS : moindres carrés alternés (Zhou, 2008)

- Jusqu'à convergence ($\sim$ 20 itérations) :
  - Fixer U (utilisateurs) améliorer W (œuvres) de façon à minimiser l'erreur
  - Fixer W améliorer U

Illustration de la minimisation alternée

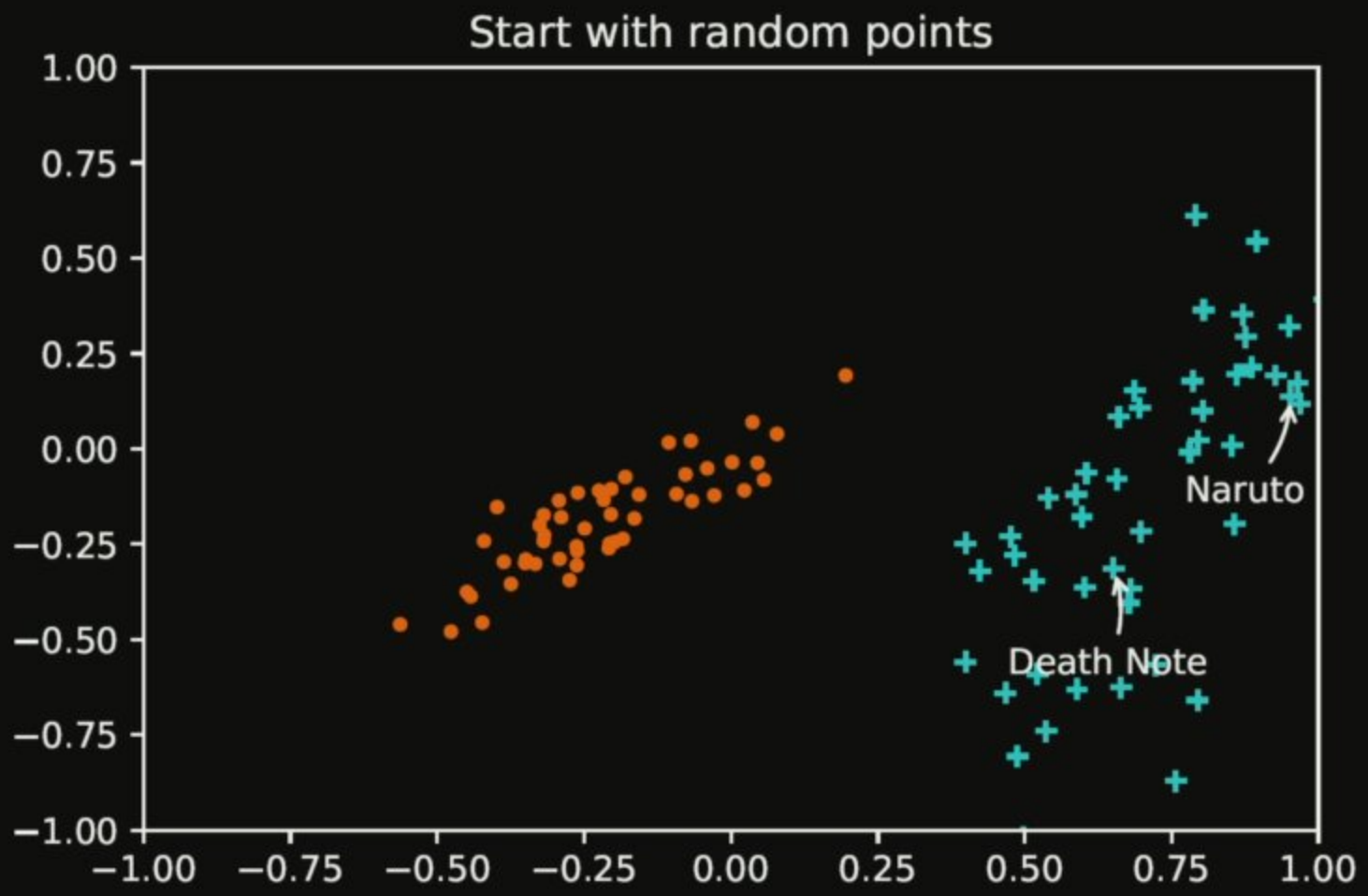


Illustration de la minimisation alternée

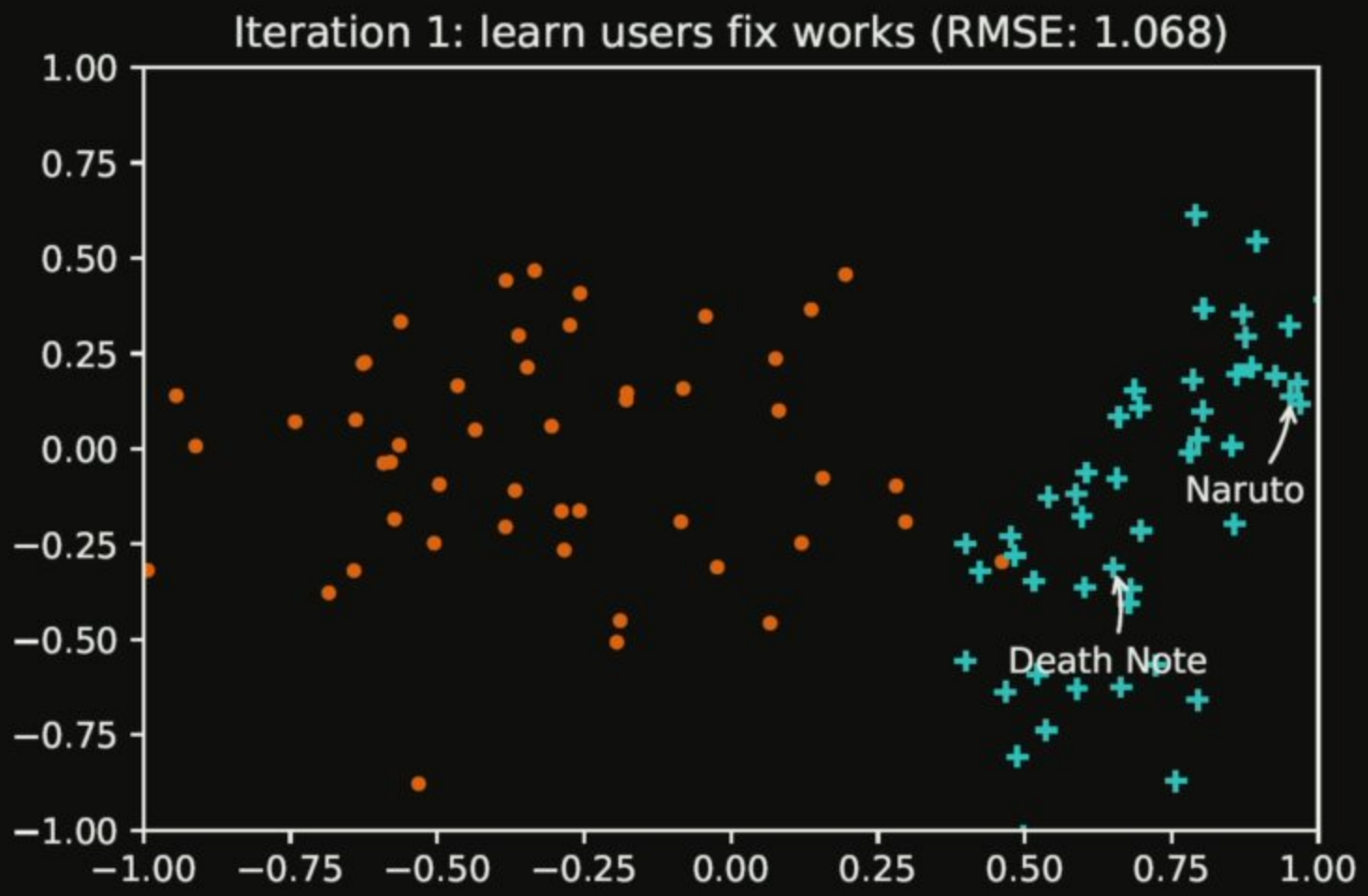


Illustration de la minimisation alternée

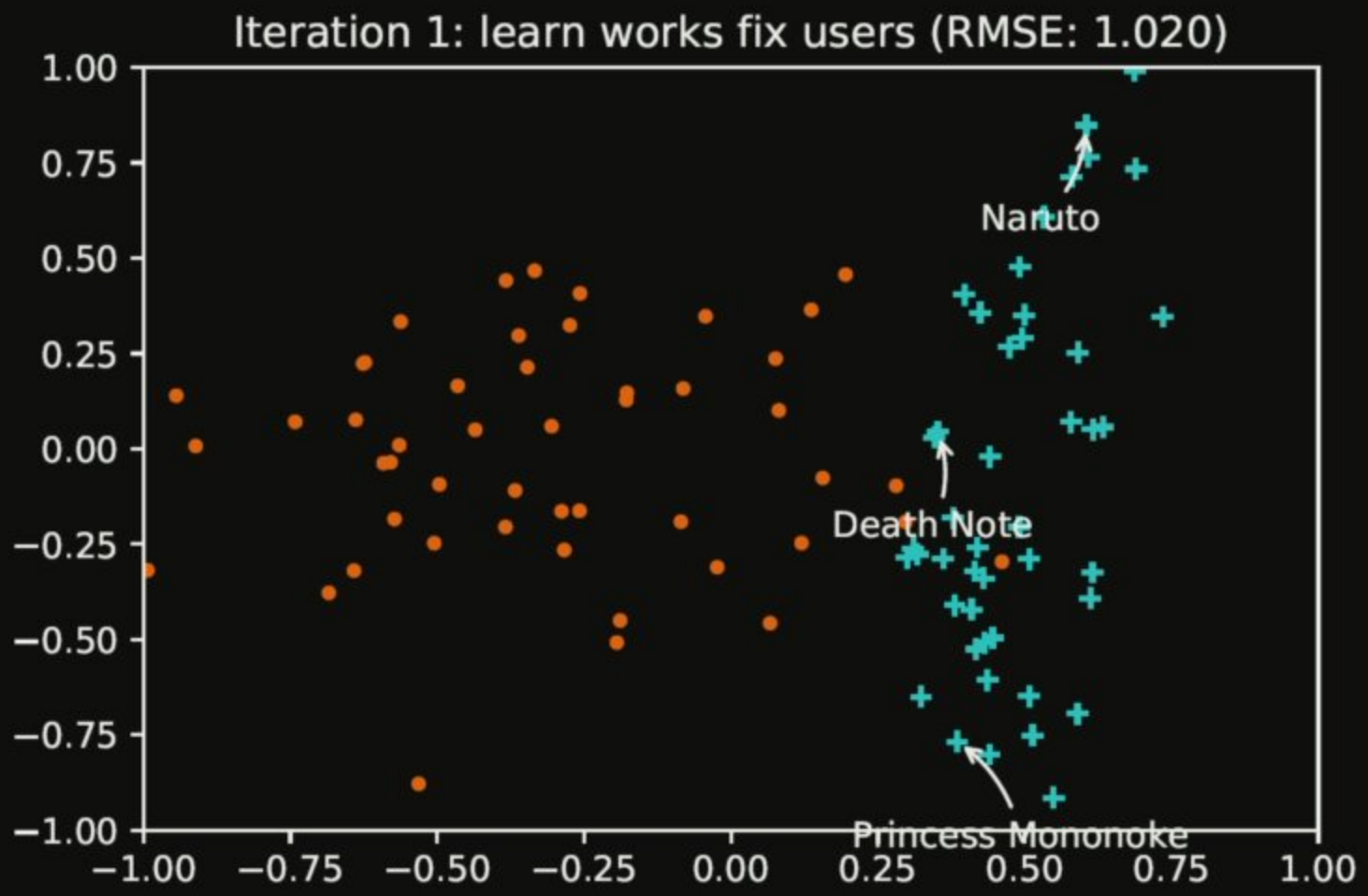


Illustration de la minimisation alternée

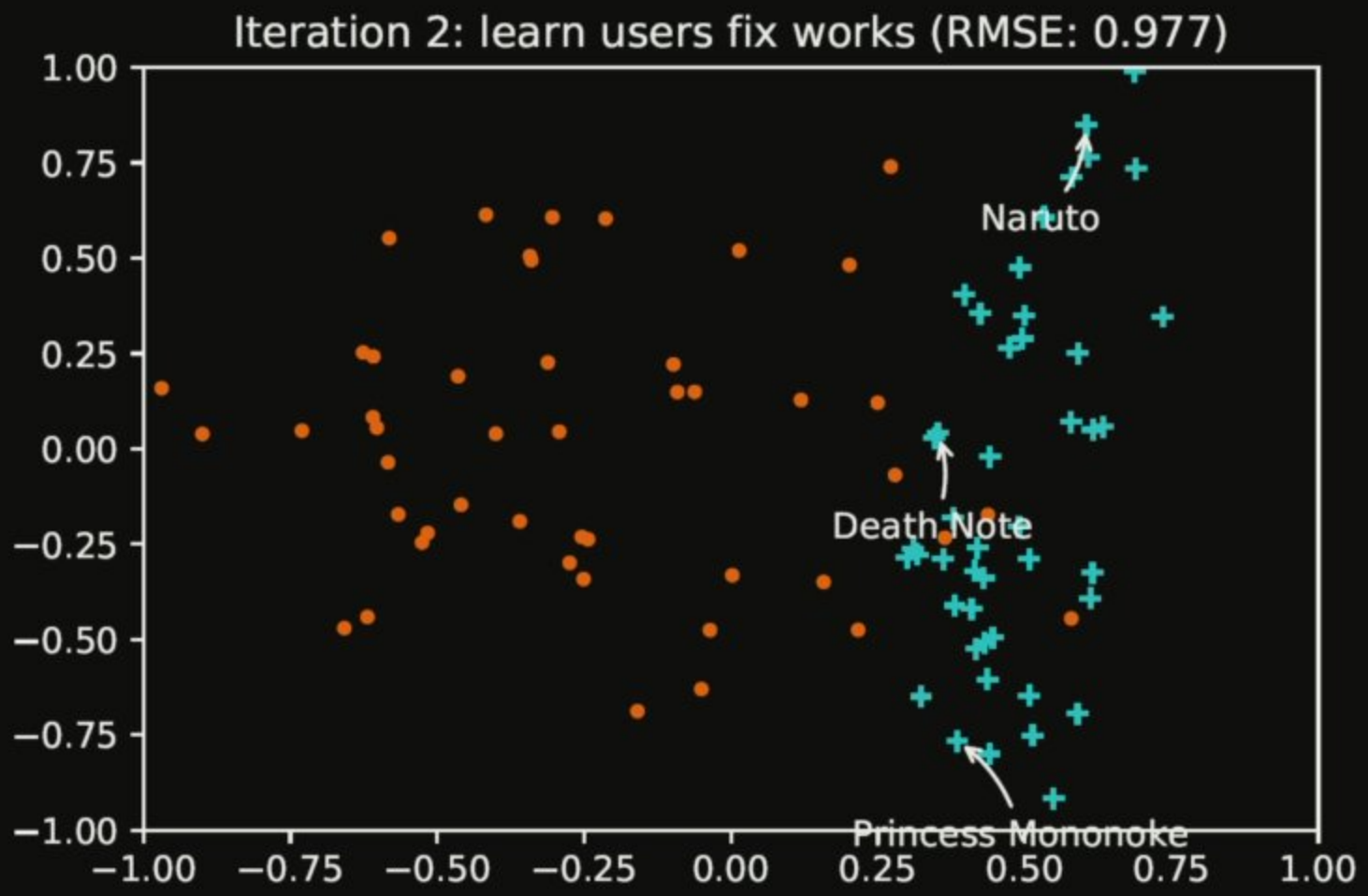


Illustration de la minimisation alternée

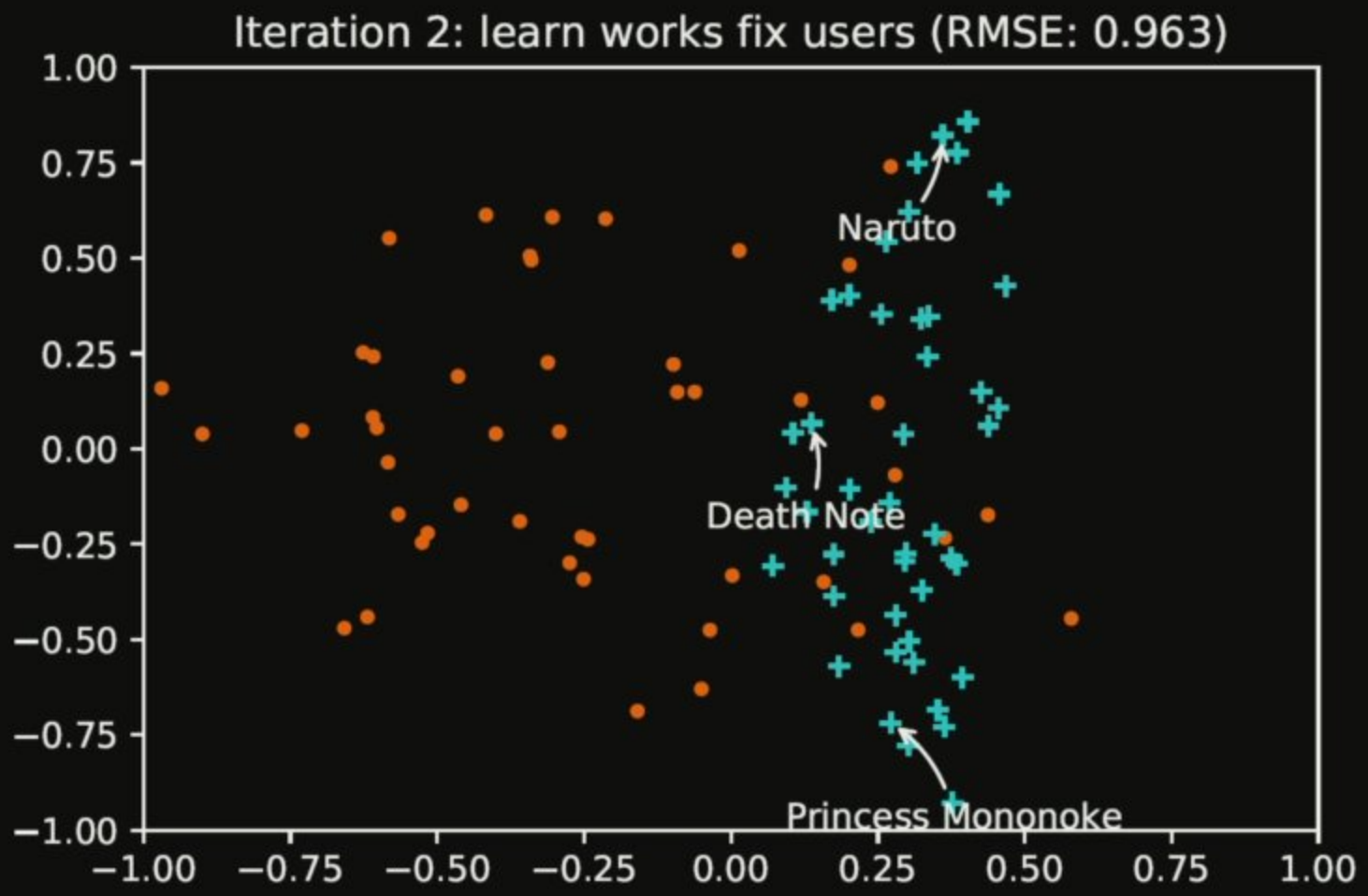


Illustration de la minimisation alternée

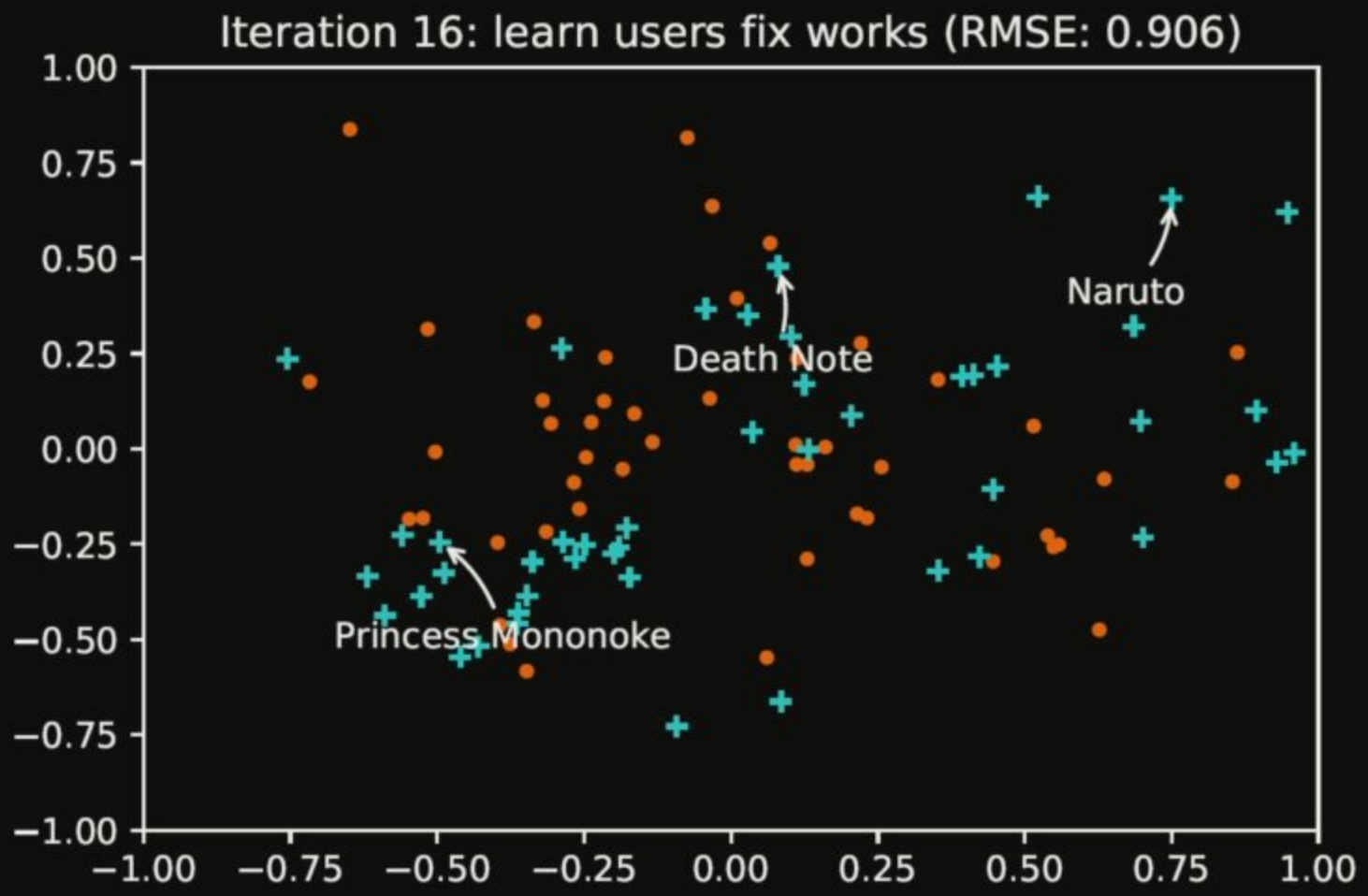
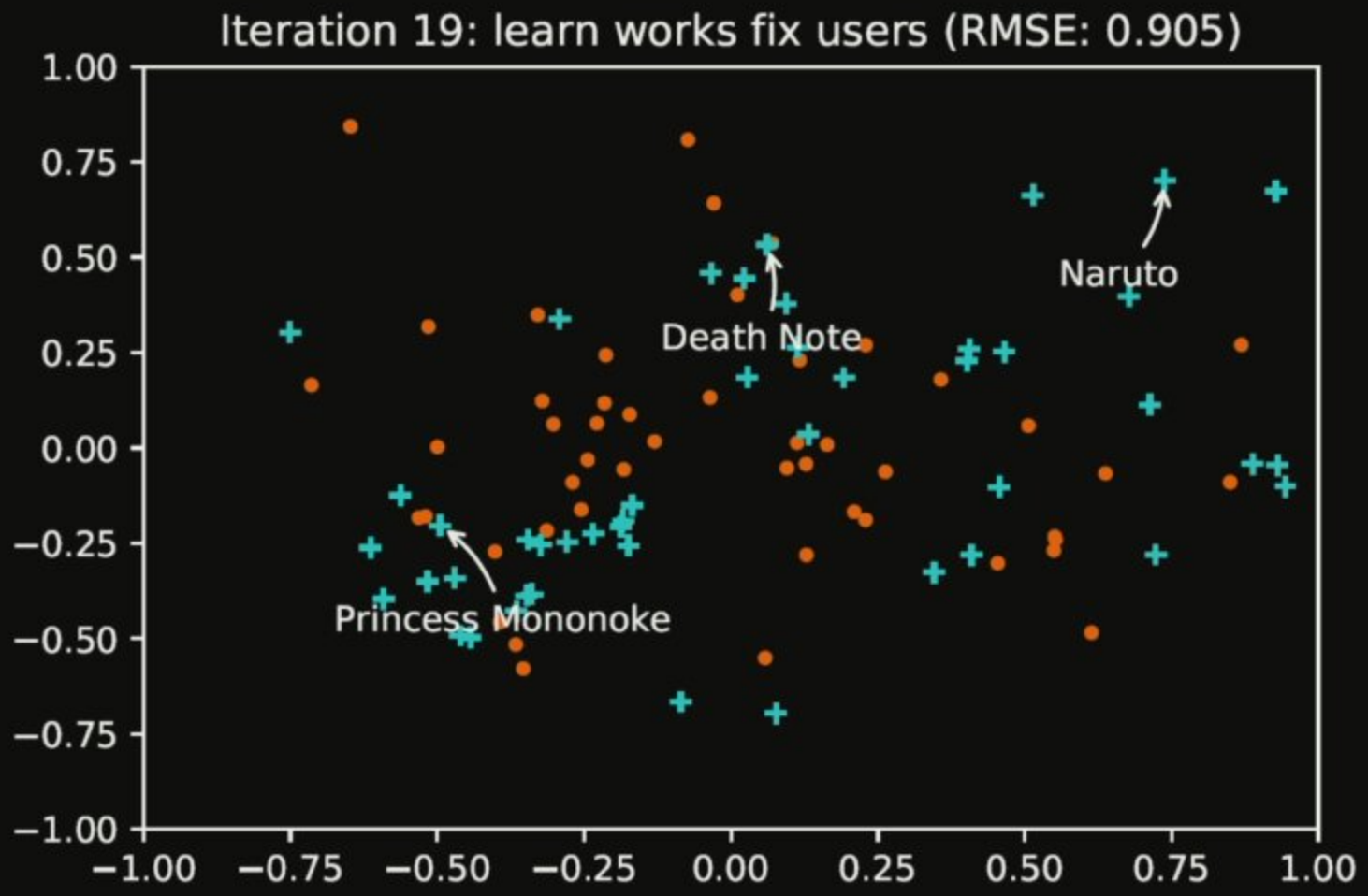
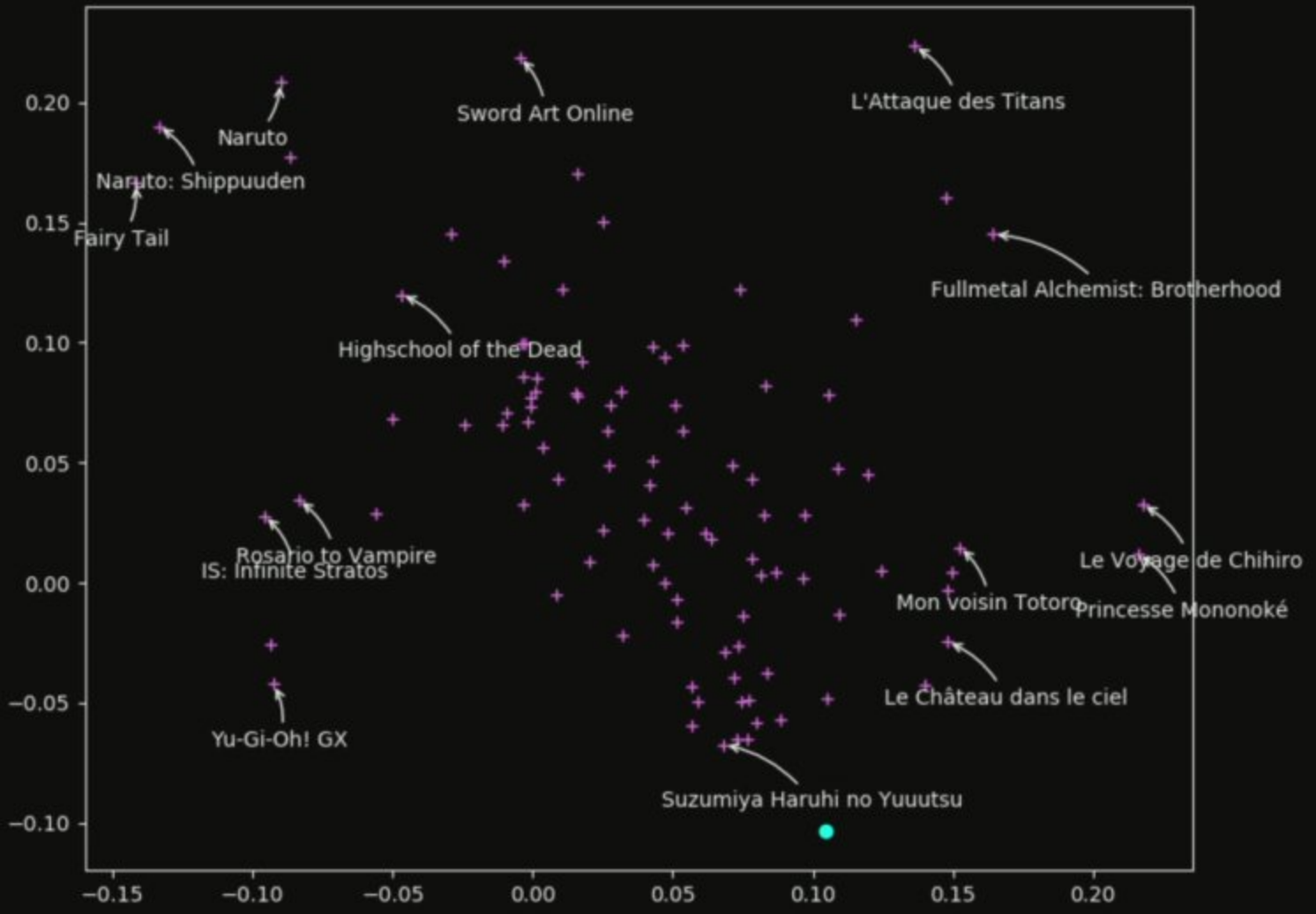


Illustration de la minimisation alternée



Visualisation des œuvres : points proches $\iff$ goûts similaires



Où êtes-vous sur la carte ?

Mangaki Map: mangaki.fr/map

Est-ce la méthode parfaite ?

À votre avis ?

Est-ce la méthode parfaite ?

À votre avis ?

Problème : démarrage à froid

- Si on n'a pas de notes pour une œuvre
  on ne sait pas où elle est sur la carte :-(

Aucun moyen de distinguer entre œuvres non notées

Mais on a énormément de posters !

Comment utiliser cette information pour améliorer les recommandations ?

Illustration2Vec (Saito and Matsui, 2015)

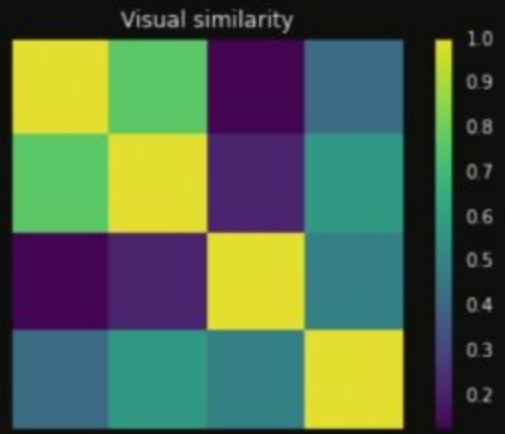
Prediction results

#	General Tag	Confidence
1.	1girl	86.1%
2.	thighhighs	84.0%
3.	solo	79.2%
4.	red hair	73.1%
5.	long hair	66.4%
6.	breasts	53.7%
7.	gloves	38.0%
8.	weapon	34.0%
9.	elbow gloves	28.3%
10.	high heels	14.0%
11.	tattoo	10.9%
#	Character Tag	
#	Copyright Tag	
#	Rating	
1.	safe	68.4%
2.	questionable	29.3%
3.	explicit	1.92%

(Solène Pichereau, sedeto.fr) (Saito and Matsui, 2015)

- Modèle CNN entraîné sur des photos (ImageNet, Danbooru)
- Reconnaît 502 tags, pondérés par confiance

Représentation des œuvres et des posters



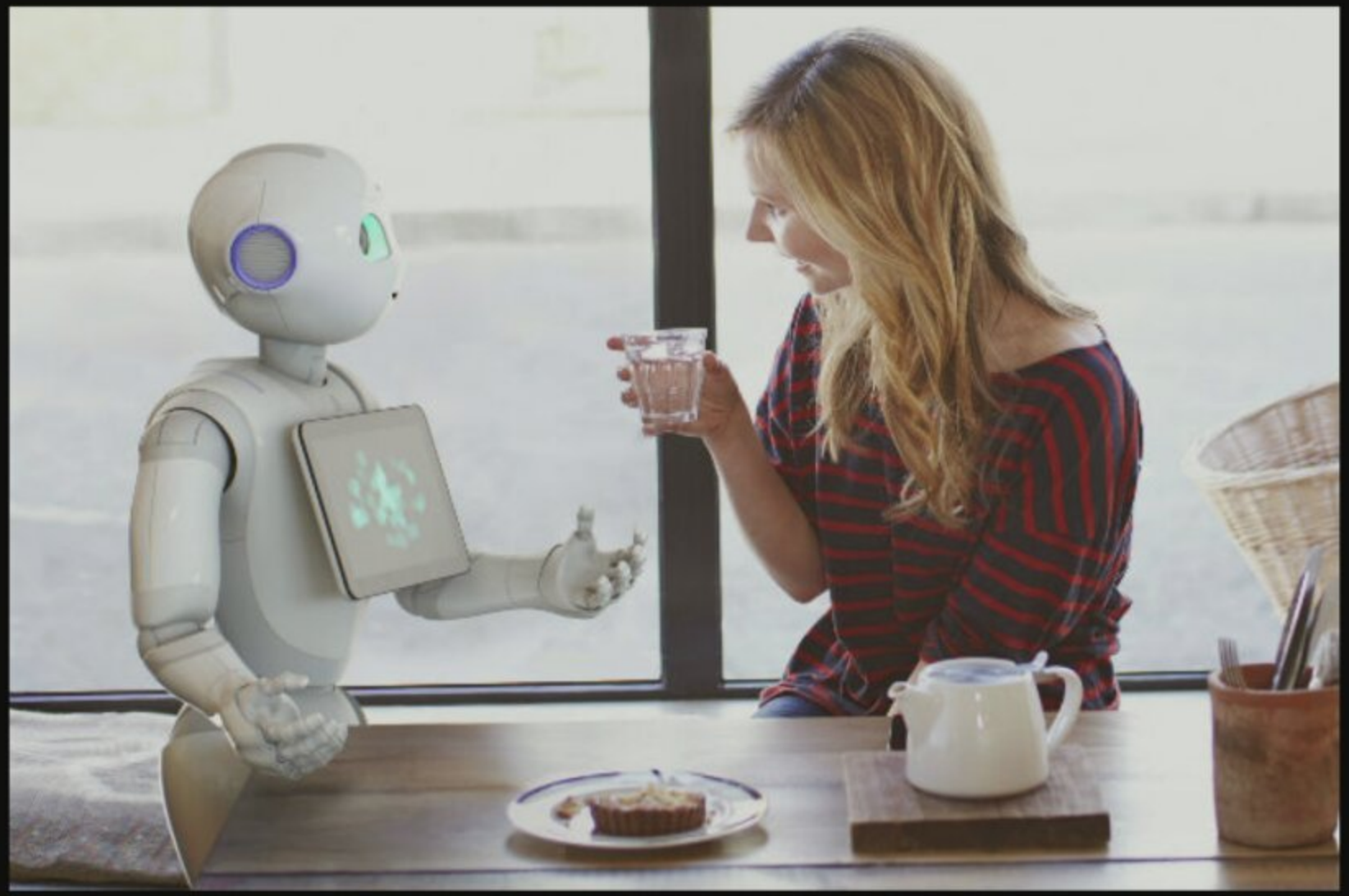
Posters proches

Étape suivante ?

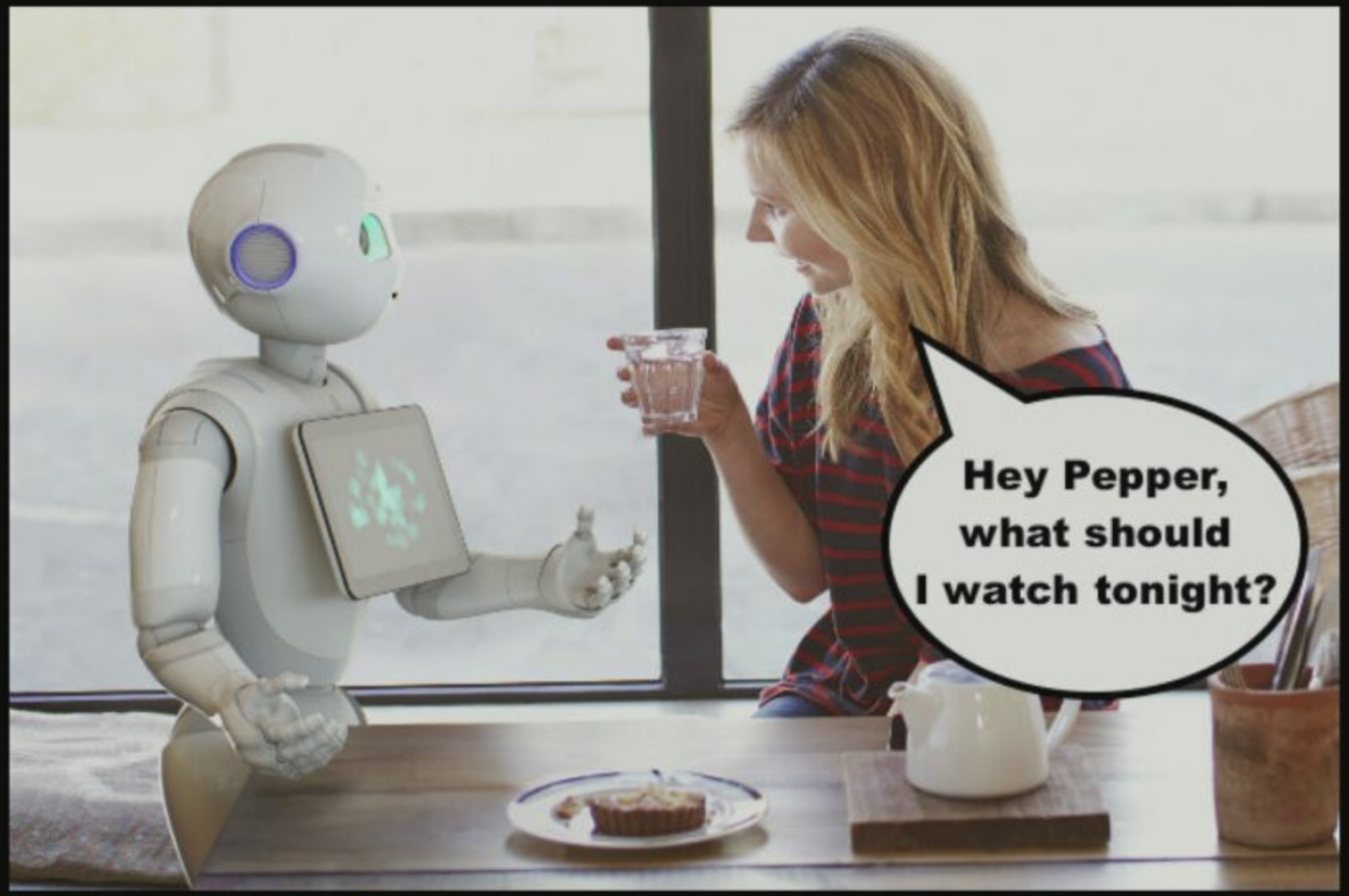
Extraire les frames des épisodes

Cowboy Bebop EP 23 “Brain Scratch”, Sunrise

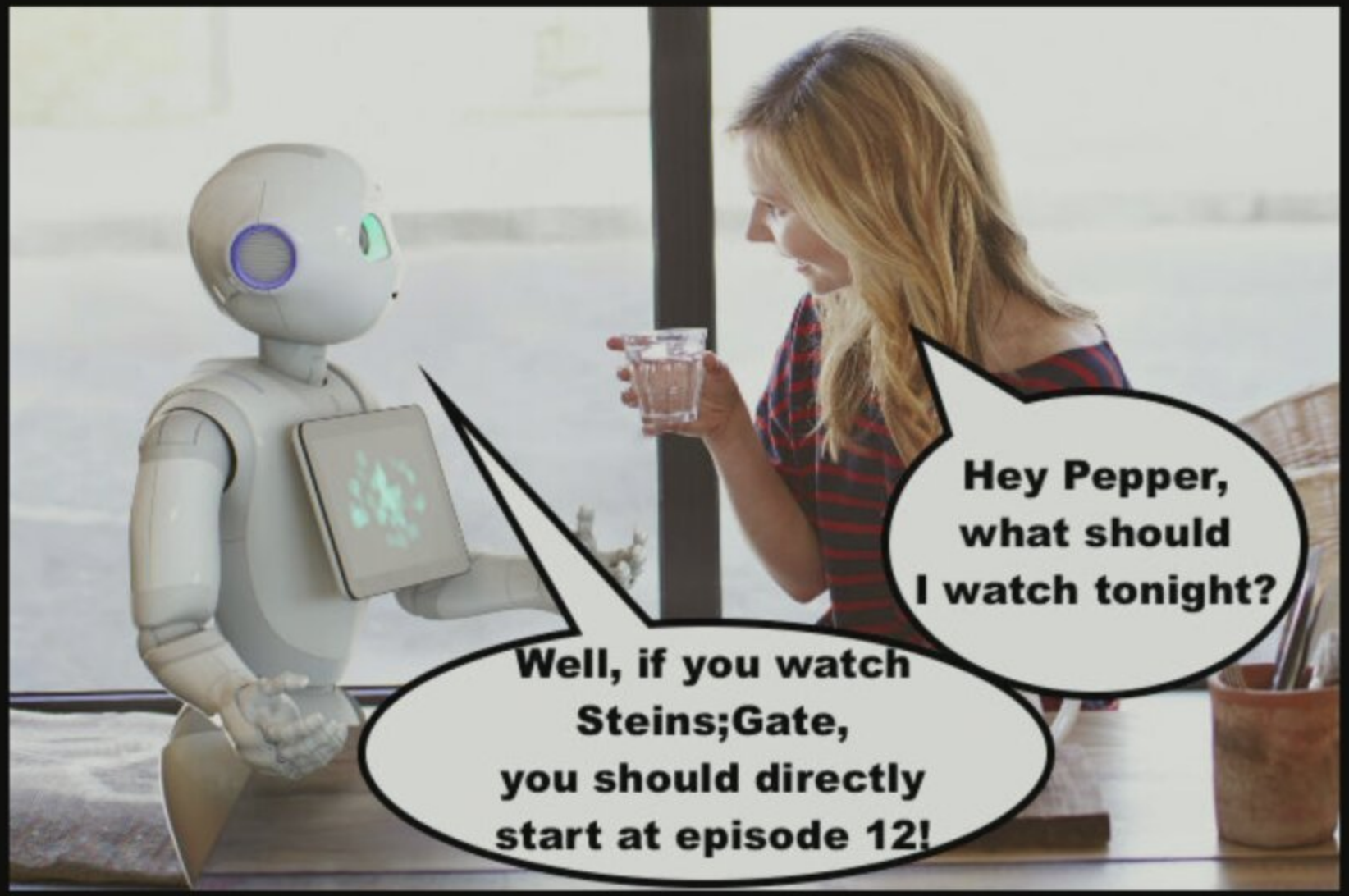
Bientôt : assistant de visionnage



Bientôt : assistant de visionnage



Bientôt : assistant de visionnage



Quels sont les risques ?

Une petite anecdote

- Le 2 octobre 2006, Netflix a lancé un concours :
  Le premier qui bat notre algorithme de plus de 10 % remportera 1 million de dollars.
  et ont filé des données pseudononymisées

Une petite anecdote

- Le 2 octobre 2006, Netflix a lancé un concours :
  Le premier qui bat notre algorithme de plus de 10 % remportera 1 million de dollars.
  et ont filé des données pseudononymisées
- La moitié de la communauté en IA s'est jetée sur le problème

Une petite anecdote

- Le 2 octobre 2006, Netflix a lancé un concours :
  Le premier qui bat notre algorithme de plus de 10 % remportera 1 million de dollars.
  et ont filé des données pseudononymisées
- La moitié de la communauté en IA s'est jetée sur le problème
- Le 8 octobre, quelqu'un a battu Cinematch

Une petite anecdote

- Le 2 octobre 2006, Netflix a lancé un concours :
  Le premier qui bat notre algorithme de plus de 10 % remportera 1 million de dollars.
  et ont filé des données pseudononymisées
- La moitié de la communauté en IA s'est jetée sur le problème
- Le 8 octobre, quelqu'un a battu Cinematch
- Le 15 octobre, 3 équipes l'avaient battu, dont 1 de 1,06 %

Une petite anecdote

- Le 2 octobre 2006, Netflix a lancé un concours :
  Le premier qui bat notre algorithme de plus de 10 % remportera 1 million de dollars.
  et ont filé des données pseudononymisées
- La moitié de la communauté en IA s'est jetée sur le problème
- Le 8 octobre, quelqu'un a battu Cinematch
- Le 15 octobre, 3 équipes l'avaient battu, dont 1 de 1,06 %
- Le 26 juin 2009, une équipe n°1 bat Cinematch de 10,05 %
  $\rightarrow$ last call : plus qu'un mois pour gagner

Une petite anecdote

- Le 2 octobre 2006, Netflix a lancé un concours :
  Le premier qui bat notre algorithme de plus de 10 % remportera 1 million de dollars.
  et ont filé des données pseudononymisées
- La moitié de la communauté en IA s'est jetée sur le problème
- Le 8 octobre, quelqu'un a battu Cinematch
- Le 15 octobre, 3 équipes l'avaient battu, dont 1 de 1,06 %
- Le 26 juin 2009, une équipe n°1 bat Cinematch de 10,05 %
  $\rightarrow$ last call : plus qu'un mois pour gagner
- Le 25 juillet 2009, une équipe n°2 bat Cinematch de 10,09 %

Une petite anecdote

- Le 2 octobre 2006, Netflix a lancé un concours :
  Le premier qui bat notre algorithme de plus de 10 % remportera 1 million de dollars.
  et ont filé des données pseudononymisées
- La moitié de la communauté en IA s'est jetée sur le problème
- Le 8 octobre, quelqu'un a battu Cinematch
- Le 15 octobre, 3 équipes l'avaient battu, dont 1 de 1,06 %
- Le 26 juin 2009, une équipe n°1 bat Cinematch de 10,05 %
  $\rightarrow$ last call : plus qu'un mois pour gagner
- Le 25 juillet 2009, une équipe n°2 bat Cinematch de 10,09 %
- L'équipe n°1 fait 10,09 % aussi

Une petite anecdote

- Le 2 octobre 2006, Netflix a lancé un concours :
  Le premier qui bat notre algorithme de plus de 10 % remportera 1 million de dollars.
  et ont filé des données pseudononymisées
- La moitié de la communauté en IA s'est jetée sur le problème
- Le 8 octobre, quelqu'un a battu Cinematch
- Le 15 octobre, 3 équipes l'avaient battu, dont 1 de 1,06 %
- Le 26 juin 2009, une équipe n°1 bat Cinematch de 10,05 %
  $\rightarrow$ last call : plus qu'un mois pour gagner
- Le 25 juillet 2009, une équipe n°2 bat Cinematch de 10,09 %
- L'équipe n°1 fait 10,09 % aussi
- 20 minutes plus tard l'équipe n°2 fait 10,10 %

Une petite anecdote

- Le 2 octobre 2006, Netflix a lancé un concours :
  Le premier qui bat notre algorithme de plus de 10 % remportera 1 million de dollars.
  et ont filé des données pseudononymisées
- La moitié de la communauté en IA s'est jetée sur le problème
- Le 8 octobre, quelqu'un a battu Cinematch
- Le 15 octobre, 3 équipes l'avaient battu, dont 1 de 1,06 %
- Le 26 juin 2009, une équipe n°1 bat Cinematch de 10,05 %
  $\rightarrow$ last call : plus qu'un mois pour gagner
- Le 25 juillet 2009, une équipe n°2 bat Cinematch de 10,09 %
- L'équipe n°1 fait 10,09 % aussi
- 20 minutes plus tard l'équipe n°2 fait 10,10 %
- ... En fait, les deux équipes étaient ex æquo sur la validation

Une petite anecdote

- Le 2 octobre 2006, Netflix a lancé un concours :
  Le premier qui bat notre algorithme de plus de 10 % remportera 1 million de dollars.
  et ont filé des données pseudononymisées
- La moitié de la communauté en IA s'est jetée sur le problème
- Le 8 octobre, quelqu'un a battu Cinematch
- Le 15 octobre, 3 équipes l'avaient battu, dont 1 de 1,06 %
- Le 26 juin 2009, une équipe n°1 bat Cinematch de 10,05 %
  $\rightarrow$ last call : plus qu'un mois pour gagner
- Le 25 juillet 2009, une équipe n°2 bat Cinematch de 10,09 %
- L'équipe n°1 fait 10,09 % aussi
- 20 minutes plus tard l'équipe n°2 fait 10,10 %
- ... En fait, les deux équipes étaient ex æquo sur la validation
- ... Du coup c'est la première équipe à envoyer ses résultats qui a gagné (équipe 1, 10,09 %)

Confidentialité des utilisateurs

- Août 2009, Netflix annonce une saison 2

Confidentialité des utilisateurs

- Août 2009, Netflix annonce une saison 2
- Entre-temps, en 2007 deux chercheurs de l'université du Texas ont été capables d'identifier les utilisateurs du jeu de données anonymisées en croisant les données avec IMDb

Confidentialité des utilisateurs

- Août 2009, Netflix annonce une saison 2
- Entre-temps, en 2007 deux chercheurs de l'université du Texas ont été capables d'identifier les utilisateurs du jeu de données anonymisées en croisant les données avec IMDb
- (année approximative de naissance, code postal, films vus)

Confidentialité des utilisateurs

- Août 2009, Netflix annonce une saison 2
- Entre-temps, en 2007 deux chercheurs de l'université du Texas ont été capables d'identifier les utilisateurs du jeu de données anonymisées en croisant les données avec IMDb
- (année approximative de naissance, code postal, films vus)
- En décembre 2009, 4 utilisateurs de Netflix ont attaqué Netflix en justice

Confidentialité des utilisateurs

- Août 2009, Netflix annonce une saison 2
- Entre-temps, en 2007 deux chercheurs de l'université du Texas ont été capables d'identifier les utilisateurs du jeu de données anonymisées en croisant les données avec IMDb
- (année approximative de naissance, code postal, films vus)
- En décembre 2009, 4 utilisateurs de Netflix ont attaqué Netflix en justice
- Mars 2010, arrangement à l'amiable, la plainte est close

Informations démographiques

Savoir qui est un garçon ou une fille sur le site : pour ou contre ?

Informations démographiques

Savoir qui est un garçon ou une fille sur le site : pour ou contre ?

Fairness : s'assurer que l'erreur du modèle ne varie pas trop sur des catégories de population

Il faut mesurer les discriminations pour réduire les inégalités

Allez, on récapitule

Comment ça marche ?

- recommander des films ⇒ inférer des notes qui n'existent pas
- validation croisée pour comparer les méthodes

1. k plus proches voisins
2. factorisation de matrice ⇒ apprendre une représentation
3. utiliser l'information présente dans les posters

Quels sont les risques ?

- Confidentialité : réidentifier des personnes
- Est-ce que le modèle est ouvert ?
- Est-ce qu'il aggrave des biais ?

Merci pour votre attention !

jj@mangaki.fr

MANGAKI

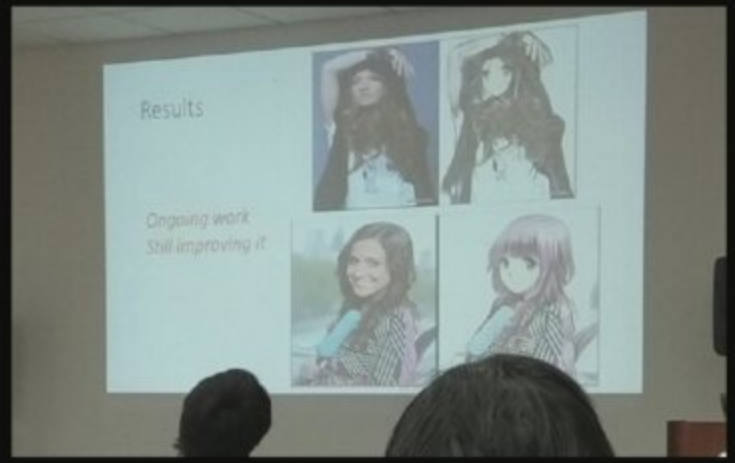
LA RECOMMANDATION FRANCOPHONE D'ANIME ET MANGA

mangaki.fr

Twitter : @MangakiFR, @jjvie

Pour en savoir plus

- AI for Manga & Anime: research.mangaki.fr
- Notamment un article sur LinuxMag qui explique comment programmer un algorithme de recommandation en Python



Minimisation alternée

Trouver U_k qui minimise

$$f(U_k) = \sum_{i,j} (\underbrace{U_i \cdot W_j}_{pred} - \underbrace{r_{ij}}_{real})^2 + \underbrace{\lambda ||U_i||_2^2 + \lambda ||W_j||_2^2}_{regularization}$$

(au fait : la dérivée de $u \cdot v$ par rapport à u est v)

Minimisation alternée

Trouver U_k qui minimise

$$f(U_k) = \sum_{i,j} (\underbrace{U_i \cdot W_j}_{pred} - \underbrace{r_{ij}}_{real})^2 + \underbrace{\lambda ||U_i||_2^2 + \lambda ||W_j||_2^2}_{regularization}$$

(au fait : la dérivée de $u \cdot v$ par rapport à u est v)

Trouver les zéros de

$$f'(U_k) = \sum_{j \text{ rated by } k} 2(U_k \cdot W_j - r_{kj}) W_j + 2\lambda U_k = 0$$

peut s'écrire $AU_k = B$ so $U_k = A^{-1}B$ (facile)

Complexité : $O(n^3)$ où n est le nombre d'œuvres notées par U_k