

Filtrage collaboratif et Système de recommandation

Ricco Rakotomalala

Les systèmes de recommandations sont au cœur de notre activité web. L'objectif est de produire auprès d'un « client » (visiteur d'un site, acheteur potentiel d'un produit, ...) une liste personnalisée de suggestions (produits, liens à cliquer, ...) en rapport avec ses préoccupations et attentes.

Exemples : Amazon, livres similaires, produits complémentaires (ex. proposer sacoches pour achat d'un portable) ; Netflix pour les films et séries ; suggestions d'articles à lire ; suggestions de vidéos sur YouTube ([à voir](#)),...



Intérêt pour le « **client** » : orientation (rapide) vers les éléments les plus pertinents, réduire le temps de recherche, améliorer l'expérience utilisateur...



Intérêt pour le « **vendeur** » : plus de ventes, plus de clics, en maintenant l'intérêt du client, le faire rester plus longtemps, l'orienter vers certains produits, le fidéliser...

Retour aux résultats de la recherche pour « gil jourdan »



Feuilleter

Album
EUR 24,00

9 d'occasion à partir de
9 neufs à partir de EUR 24,00

Voulez-vous le faire livrer en 15 h et 5 mins et économiser sur le prix de votre commande ?

Note: Cet article est noté 4,5 sur 5 par 9 clients

Détails



Voir les 7 images

Gil Jourdan : L'Intégrale 1 Album – 5 juin 2009
de **Maurice Tillieux** (Auteur)
★★★★★ 9 commentaires client

Partager    

EUR 24,00
Tous les prix incluent la TVA.
Livraison à EUR 0,01 en France métropolitaine.

Produits fréquemment achetés ensemble



Prix total: EUR 72,00
[Ajouter ces trois articles au panier](#)

- Certains de ces articles seront expédiés plus tôt que les autres. [Afficher l'information](#)
- ☒ Cet article : Gil Jourdan : L'Intégrale 1 par Maurice Tillieux Album EUR 24,00
 - ☒ Gil Jourdan - L'Intégrale - tome 2 - Gil Jourdan 2 (intégrale) 1960 - 1963 par Maurice Tillieux Album EUR 24,00
 - ☒ Gil Jourdan : L'Intégrale 3 par Maurice Tillieux Relié EUR 24,00

Les clients ayant acheté cet article ont également acheté

Page 1 sur 19



Gil Jourdan - L'Intégrale - tome 2 - Gil Jourdan 2 (intégrale) 1960 - 1963
Maurice Tillieux
★★★★★ 8
Relié
EUR 24,00 



Gil Jourdan : L'Intégrale 3
Maurice Tillieux
★★★★★ 8
Relié
EUR 24,00 



Gil Jourdan - L'Intégrale - tome 4 - Gil Jourdan 4 (intégrale) 1970 - 1979
Tillieux
★★★★★ 5
Album
EUR 24,00 



Johan et Pirlouit - L'Intégrale - tome 1 - Johan et Pirlouit intégrale 1...
Peyo
★★★★★ 4
Album
EUR 20,50 



Johan et Pirlouit - L'Intégrale - tome 2 - Johan et Pirlouit intégrale 2 réédition
Peyo
Album
EUR 24,00 

Recommandation basée sur les transactions.

Recommandation basée sur les utilisateurs (clients).

4

personnalisation absolue, mais pas de calculs...

(01.fév.2017 – 8h12)

Affaire Penelope Fillon

Ce que l'on sait de l'affaire Fillon

Une semaine après la publication des premières informations concernant de supposés emplois fictifs de Penelope Fillon, « Le Canard enchaîné » a fait de nouvelles révélations, mercredi.

Le Monde.fr avec AFP | 31.01.2017 à 20h47 • Mis à jour le 01.02.2017 à 07h26

Abonnez-vous à partir de 1 € Réagir Ajouter Partager (562) Tweeter

Les plus partagés

- 1 Penelope Fillon aurait reçu 900 000 euros au total, selon « Le Canard enchaîné » 12747
- 2 Pedro Almodovar présidera le jury de Cannes 8243
- 3 Il y a 75 ans, les Etats-Unis renvoyaient 254 migrants vers la mort dans les camps nazis 7384
- 4 Marine Le Pen refuse de restituer 300 000 euros au Parlement européen 7381
- 5 Vogue, Allure, Paradiso... Plusieurs noms de marques de cigarettes et cigares vont être interdits 2688

« Partage » = Filtrage actif. Un utilisateur décide de recommander un document à certains membres de sa communauté.

impersonnel, auto-entretenu

La recommandation peut s'appuyer sur la popularité auprès des internautes.

La même idée de popularité est derrière les recommandations du type : « Les plus lus » ou « Les plus commentés ».

voir vidéo "Benjamin Tardy -- SISE 2016" -- Lyon Data Science

1. Filtrage collaboratif - Principe
2. Approche centrée utilisateur
3. Approche centrée item
4. Evaluation des prédictions
5. Plus loin avec le filtrage collaboratif
6. Bibliographique

Principe du filtrage collaboratif

FILTRAGE COLLABORATIF

Dans son sens récent, le **filtrage collaboratif** est sous-jacent aux systèmes de recommandation. Il regroupe des techniques qui visent à opérer **une sélection sur les éléments à présenter aux utilisateurs** (**filtrage**) **en se basant sur le comportement et les goûts exprimés de très nombreux autres utilisateurs** (**collaboration**) ([Wikipédia](#)).

Exemple

Pierre, Paul, Jacques, et les autres aiment les produits A et B.
René, qui aime le produit A, va sûrement aimer le produit B...

Le **recueil d'information** joue un rôle crucial dans le processus, il peut être :

- **Explicite**. L'utilisateur attribue des **notes** aux produits ou indique son appréciation (*like*).
Avantages : pas d'ambiguïtés sur les goûts et les centres d'intérêts de l'utilisateur.
Inconvénients : biais de déclaration, exagération souvent.
- **Implicite**. Recueil basé sur le **comportement** (achats, clics, durée sur une page). **Avantages** : objectivité. **Inconvénients** : grande volumétrie, aucune indication sur l'appréciation.

Nous sommes dans le cadre du data mining. On cherche à identifier – à partir d'une base de données – les **régularités de comportement ou les associations** (**patterns**) entre les **utilisateurs** et les **items** (produits, thèmes, etc.) en se basant sur leurs évaluations (**notes**) et leurs attitudes passées (**clics**, **achats**).

Mais, les données présentant des caractéristiques spécifiques (dimensionnalité, mises à jour fréquentes, données manquantes à profusion, matrices creuses, ...), nous utiliserons préférentiellement certaines techniques.

simplistes même



Plutôt que les techniques sophistiquées, on privilégiera souvent les approches simples rapides, pouvant appréhender les **fortes dimensionnalités / volumétries** et résistants au **sur** apprentissage.

robustes



Le tableau de données pour le filtrage collaboratif se présente *souvent* comme suit :

	Item 1	Item 2	Item 3	Item 4
Pierre	?	2	7	8
Paul	4	1	?	7
Jacques	3	8	?	4



Les valeurs correspondent à une note (par ex. allant de 0 à 10). A la place d'une note, nous pouvons avoir une pondération binaire (achat / non achat).



« ? » veut dire qu'on n'a pas l'avis de l'utilisateur sur l'item et **qu'on souhaite l'estimer justement.**



A la place de l'utilisateur, nous pouvons avoir une transaction (un panier d'achat). Nous nous rapprochons du cadre de l'analyse des associations.

Principe : les meilleures recommandations proviennent des individus qui présentent des goûts ou comportements **similaires**.

idée : qui se ressemblent (profil, comportements) aiment ou achètent les mêmes produits

Point de départ. Nous devons disposer d'une base où les préférences d'un grand nombre d'utilisateurs sont disponibles.

Prédiction pour un individu.

1. Un individu **u** exprime ses préférences par un système de notation par ex.
2. Le système recherche les utilisateurs qui présentent un « profil » le plus proche (en termes de préférences)
3. A partir de ce voisinage, les notes pour chaque item, non évalué par **u**, sont calculées. Lui sont alors recommandés les items pour lesquels une note élevée est prédite.

Technique des plus proches voisins (Memory based)

APPROCHE CENTRÉE UTILISATEUR

Quelle note attribuer à René pour l'item n°4 ?

	Item 1	Item 2	Item 3	Item 4
Pierre	?	2	7	8
Paul	4	1	?	7
Jacques	3	8	?	4
René	4	1	6	?



Identifier les utilisateurs dont le profil de notes est le plus proche de René. Se servir des notes de ces individus pour l'item n°4 pour estimer la note de René.

La recommandation ne tient absolument pas compte de la nature ou du contenu de l'item (chaussure, boîte de lessive, etc..., c'est pareil).

Les éléments clés de l'algorithme sont :

- Disposer d'une mesure de similarité ;
- Décider du nombre de voisins ;
- Calcul de la note agrégée, avec possiblement une pondération tenant compte de la proximité.

	Item 1	Item 2	Item 3	Item 4
Pierre	?	2	7	8
Paul	4	1	?	7
Jacques	3	8	?	4
René	4	1	6	?

$r_{u,i}$ Note (rating) de l'utilisateur u pour l'item i

\bar{r}_u Moyenne des notes attribuées par l'utilisateur u (ex. $\bar{r}_{Paul} = \frac{1}{3}(4 + 1 + 7) = 4$)


Similarité corrélation entre deux individus (x,y)

$$cor(x,y) = \frac{\sum_{i \in I_{xy}} (r_{x,i} - \bar{r}_x)(r_{y,i} - \bar{r}_y)}{\sqrt{\sum_{i \in I_{xy}} (r_{x,i} - \bar{r}_x)^2 \times \sum_{i \in I_{xy}} (r_{y,i} - \bar{r}_y)^2}}$$

I_{xy} représente l'ensemble des items où on dispose des valeurs à la fois pour x et y

Similarité cosinus entre deux individus (x,y)

$$cos(x,y) = \frac{\sum_{i \in I_{xy}} r_{x,i} r_{y,i}}{\sqrt{\sum_{i \in I_x} r_{x,i}^2 \times \sum_{i \in I_y} r_{y,i}^2}}$$



	René	
	corrélacion	cosinus
Pierre	1.000	0.993
Paul	1.000	1.000
Jacques	-1.000	0.567

Avec deux observations exploitables seulement, la **corrélacion** ne semble pas très efficace ici. Elle **présente quand même l'avantage de centrer les notes, neutralisant ainsi l'influence des utilisateurs qui notent systématiquement trop haut ou trop bas.**

N est la taille du voisinage **U** des plus proches utilisateurs, c'est un paramètre de l'algorithme.

Plusieurs approches possibles, avec des pondérations et des corrections plus ou moins sophistiquées.

$$1 \quad r_{u^*,i} = \frac{1}{N} \sum_{u \in U} r_{u,i}$$

$$2 \quad r_{u^*,i} = \frac{\sum_{u \in U} \text{simil}(u^*, u) \times r_{u,i}}{\sum_{u \in U} |\text{simil}(u^*, u)|}$$

$$3 \quad r_{u^*,i} = \bar{r}_{u^*} + \frac{\sum_{u \in U} \text{simil}(u^*, u) \times (r_{u,i} - \bar{r}_u)}{\sum_{u \in U} |\text{simil}(u^*, u)|}$$

	Item 1	Item 2	Item 3	Item 4
Pierre	?	2	7	8
Paul	4	1	?	7
Jacques	3	8	?	4
René	4	1	6	?

René	
	cosinus
Pierre	0.993
Paul	1.000
Jacques	-1.000

Pierre et Paul sont les **N = 2** utilisateurs présentant la similarité la plus élevée avec René (au sens du cosinus).

$$\hat{r}_{René,4} = \frac{1}{2} (8 + 7) = 7.5$$

$$\hat{r}_{René,4} = \frac{0.993 \times 8 + 1.000 \times 7}{|0.993| + |1.000|} = 7.498$$

$$\hat{r}_{René,4} = 3.66 + \frac{0.993 \times (8 - 5.67) + 1.000 \times (7 - 4)}{|0.993| + |1.000|} = 6.33$$

Avantages :

- Les calculs sont simples. Les résultats sont faciles à expliquer.
- On peut travailler toujours à partir d'une base constamment mise à jour

Inconvénients :

- Parcourir la base à chaque recommandation à effectuer pour identifier le voisinage est coûteux. Heureusement des heuristiques permet de réduire les temps de calcul (ex. [Locality-sensitive hashing](#)).
- Quand il y a trop de données manquantes (au démarrage du système notamment), problème d'estimation. *Cold start problem*.
- L'absence d'une note peut signifier une opinion. Ca ne m'intéresse pas donc je ne note pas.

Remarque :

- La transposition de la démarche à l'approche centrée item est possible. Définir une mesure de similarité entre items, définir les items les plus proches de l'item 4, proposer une prédiction pour René via une note agrégée.

Utilisation d'une méthode de data mining (model based)

APPROCHE CENTRÉE ITEM

Idée : prédire les valeurs d'un item à partir des autres. Problème de régression classique avec prise en compte des données manquantes.

	Item 1	Item 2	Item 3	Item 4
Pierre	?	2	7	8
Paul	4	1	?	7
Jacques	3	8	?	4
René	4	1	6	?



Régression multiple. Dimensionnalité et volumétrie rendent l'approche impossible. Nombre de paramètres à estimer trop élevé.



Régression simple. On pourrait tester les 3 régressions simples pour prédire « Item 1 » et ne conserver que la meilleure. Idem pour les autres. Même là, le volume de calcul est considérable. Et on se heurte de plus au problème du **sur apprentissage**.

Construire une régression simple de type

$$y = x + b$$

- ➔ Un seul paramètre à estimer. Simplicité des calculs. La constante b est estimée par la moyenne des écarts entre les notes des items y et x .
- ➔ La contrainte sur la pente introduit une forme de régularisation, prévenant les problèmes de sur apprentissage.
- ➔ Une augmentation d'une manière significative du nombre d'utilisateurs ne pénalise pas le système.

Prédire item 4

	Item 1	Item 2	Item 3	Item 4
Pierre	?	2	7	8
Paul	4	1	?	7
Jacques	3	8	?	4
René	4	1	6	?

Valeur de « item 1 »
pour René.



A partir de Item 1 : $b_1 = \frac{(7 - 4) + (4 - 3)}{2} = 2$

$$\hat{r}_{René,4}^1 = 4 + 2 = 6$$

A partir de Item 2 : $b_2 = \frac{(8 - 2) + (7 - 1) + (4 - 8)}{3} = 2.67$

$$\hat{r}_{René,4}^2 = 1 + 2.67 = 3.67$$

A partir de Item 3 : $b_3 = \frac{(8 - 7)}{1} = 1$

$$\hat{r}_{René,4}^3 = 6 + 1 = 7$$

On combine ces prédictions via une
moyenne pondérée par le nombre
d'utilisateurs ayant noté
simultanément les items concernés.



$$\hat{r}_{René,4} = \frac{2 \times 6 + 3 \times 3.67 + 1 \times 7}{2 + 3 + 1} = 5.0$$



Deux utilisateurs ont noté simultanément les
items 4 et 1 (si l'on ne compte pas René).

Avantages :

- Les calculs sont simples. Les résultats sont faciles à expliquer.
- On peut travailler toujours à partir d'une base constamment mise à jour en déploiement.
- La mise à jour des constantes des modèles peut être lissée dans le temps

Inconvénients :

- On doit parcourir toute la base et consolider les prédictions en déploiement
- Il y a quand même $\frac{p \times (p-1)}{2}$ constantes à estimer à partir des données.
- Quand il y a trop de données manquantes (au démarrage du système notamment), problème d'estimation toujours. *Cold start problem*.

Schémas et critères d'évaluation des prédictions

EVALUATION DES PRÉDICTIONS

Nous sommes dans une démarche d'analyse prédictive, en utilisant des techniques – un peu particulières certes – de machine learning. Un dispositif d'évaluation est nécessaire.



Nous pouvons utiliser un schéma holdout (apprentissage-test), mais aussi des techniques de ré-échantillonnage de type validation croisée ou leave-one-out.



Le critère d'évaluation doit tenir compte de 2 aspects importants :

- On prédit les valeurs d'une note ;
- L'individu statistique est le couple « utilisateur – item »

Base d'apprentissage

User	pry	(1995)	ye	(1995)	ms	(1995)	rtv	(1995)	cat	(1995)	so	(1995)	rys	(1995)	be	(1995)	ng	(1995)	lll	(1995)
1	5	3	4	3	3	5	4	1	5	3										
2	4																			2
3																				
4																				
5	4	3																		
6	4							2	4	4										
7						5				5	5	5	4							
8								3												
9								5	4											
10	4				4															
11										4										
12						5														
13	3	3			5	1			2	4	3									
14									5											
15	1							1			4									
16	5					5			5	5	5									
17	4							4			3									
18	5					3			5		5	5								
19					4															
20	3																			
21	5					2			5											
22		2			5															
23	5								4	4										
24								4	5	5										
25	5							4	4											
26	3							3			4									
27																				
28						3			5											
29																				
30		3							4											
31																				
32								4		3										
33																				
34																				
35																				
36																				
37								4												
38	5																			
39																				
40																				

User	pry	(1995)	ye	(1995)	ms	(1995)	rtv	(1995)	cat	(1995)	so	(1995)	rys	(1995)	be	(1995)	ng	(1995)	lll	(1995)
1	5	3	4	3	3	5	4	1	5	3										
2	4																			2
3																				
4																				
5	4	3																		
6	4									2	4	4								
7																				
8																				
9																				
10	4																			
11																				
12																				
13	3	3				5	1			2	4	3								
14																				
15	1																			
16	5																			
17	4																			
18	5																			
19																				
20	3																			
21	5																			
22		2																		
23	5																			
24																				
25	5																			
26	3																			
27																				
28																				
29																				
30		3																		

Construction d'un modèle ou stockage mémoire simplement

Base de test

User	pry	(1995)	ye	(1995)	ms	(1995)	rtv	(1995)	cat	(1995)	so	(1995)	rys	(1995)	be	(1995)	ng	(1995)	lll	(1995)
1	5	3	4	3	3	5	4	1	5	3										
2	4																			2
3																				
4																				
5	4	3																		
6	4																			
7																				
8																				
9																				
10	4																			

Base initiale

Prédiction $\hat{r}_{u,i}$

Root Mean Squared Error

$$RMSE = \sqrt{\frac{\sum_{(u,i) \in K} (r_{u,i} - \hat{r}_{u,i})^2}{|K|}}$$

Mean Absolute Error

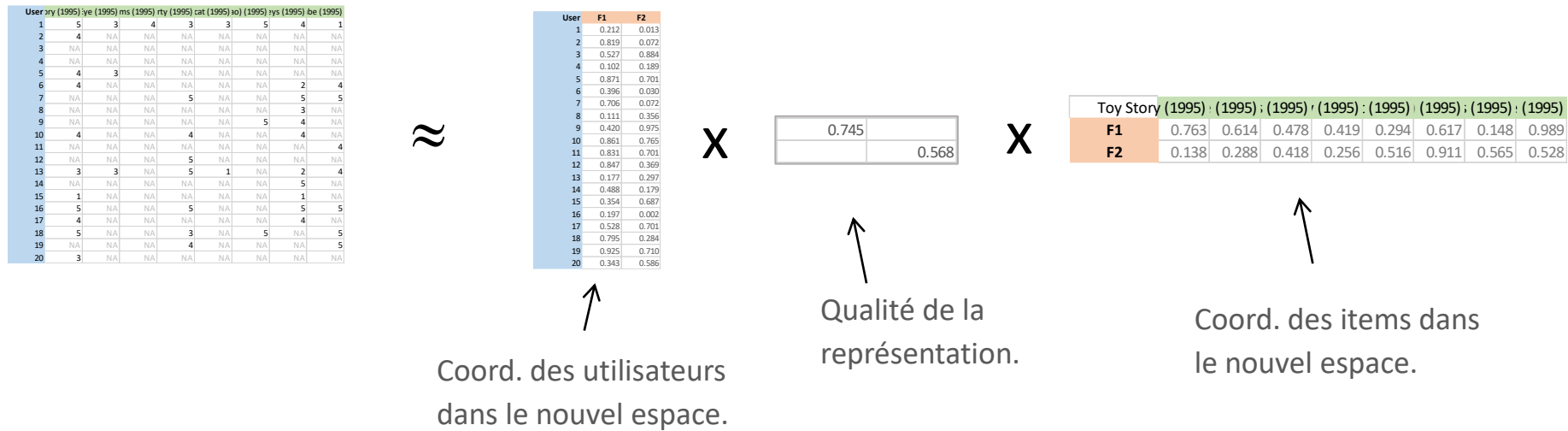
$$MAE = \frac{\sum_{(u,i) \in K} |r_{u,i} - \hat{r}_{u,i}|}{|K|}$$

On prend chaque individu **u** de l'échantillon test.
On essaie prédire sa note pour l'item **i** à partir :
des notes des individus de la base d'apprentissage
et de ses notes disponibles pour ses autres items.
K est l'ensemble des couples (utilisateur, item)
pour lesquels la confrontation est effectuée.

Test : n'intègre que les couples (u, i) renseignés !

PLUS LOIN AVEC LE FILTRAGE COLLABORATIF

Par ex. se projeter dans un espace de dimension réduite avec la **décomposition en valeur singulière (SVD)**.



- ➡ On peut reconstituer approximativement le tableau de notes à partir des coordonnées dans l'espace réduit. Intéressant si $p' \ll p$ (p' taille de la nouvelle dimension ; p nombre d'items initialement), stockage, déploiement, etc. (Voir [Topic Modeling](#))
- ➡ On peut traiter les utilisateurs / items supplémentaires.
- ➡ Double enjeux forts : dimensionnalité et, surtout, gestion des données manquantes (un algorithme EM est utilisé). autres approches aussi, ex. NIPALS

Considérer les notes comme des classes (NA peut être traitée comme une valeur à part entière parmi les prédictives) et s'appuyer sur la méthode Naive Bayes (modèle d'indépendance conditionnelle).

Par ex.
prédire les
valeurs de
l'item n°1

$$\ln P(item_1 = v / item_2, item_3, \dots) \propto \ln P(item_1 = v) + \sum_{j=2}^p \ln P(item_j / item_1 = v)$$
$$\hat{r}_{u,1} = \arg \max_v \ln P(item_1 = v / item_2, item_3, \dots)$$



Connue, reconnue. Facile à implémenter. Màj facile également.



Stockage des paramètres compliqué avec la forte dimensionnalité.

(en nombre d'items..., surtout qu'ils évoluent constamment....)

Plutôt qu'aux préférences, on peut s'intéresser directement aux achats. Soit sous forme de transaction (panier d'achat) soit en les reliant aux utilisateurs.

	Item 1	Item 2	Item 3	Item 4
Pierre	0	0	1	1
Paul	0	1	1	1
Jacques	1	1	1	0

➡ Les fréquences d'achats des items peuvent déjà constituer des recommandations.



➡ Les techniques de recherche des règles d'association s'appliquent pleinement. (objet de la séance 4 de notre cours....)

➡ Si on s'intéresse aux transactions, on peut y ajouter la temporalité avec l'extraction des motifs séquentiels. sequential pattern mining

La recommandation par le contenu s'appuie sur les caractéristiques des items et leur correspondance avec le profil des utilisateurs.

19 catégories

ID	title	unknown	Action	Adventure	Animation	Children's	Comedy	Crime	Documenta
1	Toy Story (1995)	0	0	0	1	1	1	0	0
2	GoldenEye (1995)	0	1	1	0	0	0	0	0
3	Four Rooms (1995)	0	0	0	0	0	0	0	0
4	Get Shorty (1995)	0	1	0	0	0	1	0	0
5	Copycat (1995)	0	0	0	0	0	0	1	0
6	Shanghai Triad (Yao a yao dao waipo qiao) (1995)	0	0	0	0	0	0	0	0
7	Twelve Monkeys (1995)	0	0	0	0	0	0	0	0
8	Babe (1995)	0	0	0	0	1	1	0	0

1664 films

L'idée est de se projeter dans un espace de dimension réduite défini par les propriétés des items.

1664 films en tout

Ex. Pierre a aimé « Toy Story » et « Babe » $\rightarrow x_{\text{Pierre}} = (1, 0, 0, 0, 0, 0, 0, 1, 0, \dots)$

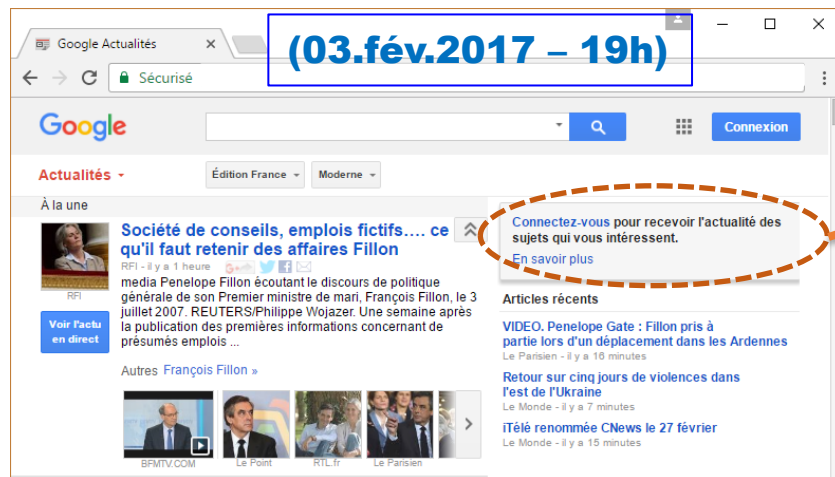
19 catégories en tout

Pierre est projeté dans un nouvel espace $\rightarrow y_{\text{Pierre}} = (0, 0, 0, 1, 2, 2, 0, 0, \dots)$

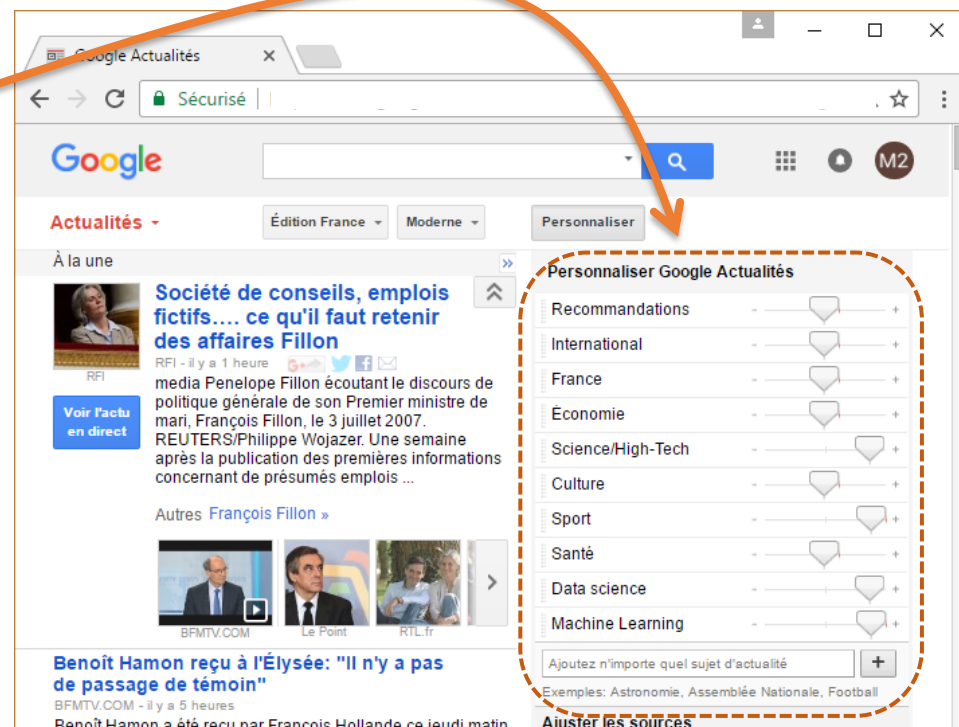
On est passé d'une dim. $(n \times 1664)$ à $(n \times 19)$, plus condensée, les méthodes de machine learning peuvent en tirer parti.

ex. Synopsis des films....

La recommandation par le contenu peut prendre comme source **le texte associé à l'item**, à partir duquel nous pouvons extraire des thèmes (ex. via la représentation bag-of-words). On peut alors les mettre en relation avec les préférences de l'utilisateur et/ou les sujets qui l'intéressent. **Nous sommes dans un cadre typique de text mining** que nous connaissons.



On peut choisir des **thématiques prédéfinies** ou **ajouter nos sujets via des mots-clé** (ex. data science). L'importance qui leur est accordée peut être modulée.



Les systèmes de recommandations basés sur le filtrage collaboratif est un thème d'actualité incontournable avec la profusion des données.

Les challenges techniques sont nombreux, en termes de volumétrie,...

Souvent, les méthodes simples et rapides, pouvant appréhender des grandes bases et facilement parallélisables, sont à privilégier.

Aimer et acheter ne relève pas toujours des mêmes mécanismes psychologiques (acheter un cadeau pour autrui sans aimer ; aimer mais ne jamais acheter parce que trop cher, etc.).

Le danger est d'enfermer un utilisateur dans une certaine thématique, sans possibilité de l'ouvrir vers d'autres perspectives (mixer les préférences avec la popularité, ou encore des propositions aléatoires,...)

Il faut nécessairement une certaine connaissance de l'utilisateur (profil, préférences,...) avant de pouvoir lui faire des propositions.

(et on nous demande tout le temps de nous connecter, de renseigner des infos sur nous, etc...)

Ouvrages et articles

- Coelho L.P., Richer W., « Building Machine Learning Systems with Python », 2nd Edition, Packt Publishing, 2015. [Chapitre 8](#).
- Conway D., White J.M., « Machine Learning for Hackers », O'Reilly, 2012. [Chapitre 10](#).
- Ekstrand M.D., Riedl J.T., Konstan J.A., « Collaborative Filtering Recommender Systems », in Foundations and Trends in Human-Computer Interaction, vol. 4, n°2, p. 81-173, 2011.
- Leskovec J., Rajaraman A., Ullman J., « Mining of Massive Datasets », 2nd Edition, 2014. [Chapitre 9](#).
- Zhao Y., Cen Y., « Data Mining Applications with R », Academic Press, 2014. [Chapitre 5](#).
- Wikipédia, « [Collaborative filtering](#) ».