Introduction aux systèmes de recommandation sociaux

Emmanuel Viennet, Daniel Bernardes, Mamadou Diaby, Raphaël Fournier et Françoise Fogelman-Soulié

L2TI - Université Paris 13

7/10/2016

1. Introduction
2. Critères de performance
3. Bref état de l'art
4. Expériences
   - Jeux de données
   - Reproduction des systèmes classiques
5. Applications

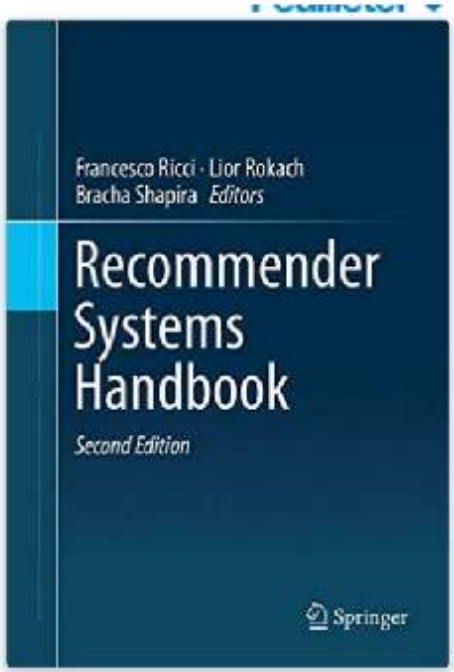
Systèmes de recommandation

- «Les clients ayant acheté ceci achètent aussi cela»
- «Recommandé pour vous, Emmanuel »
- «Ces emplois pourraient vous intéresser»
- «Connaissez vous ces personnes ?»

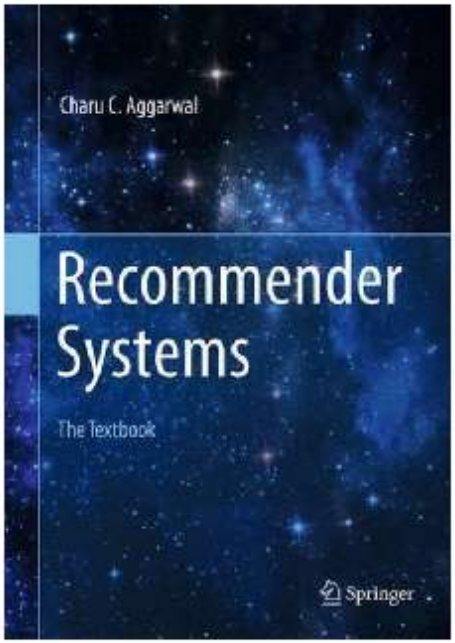
Mais aussi

- Recherche d'information personnalisée
- Ciblage publicitaire (bannières)

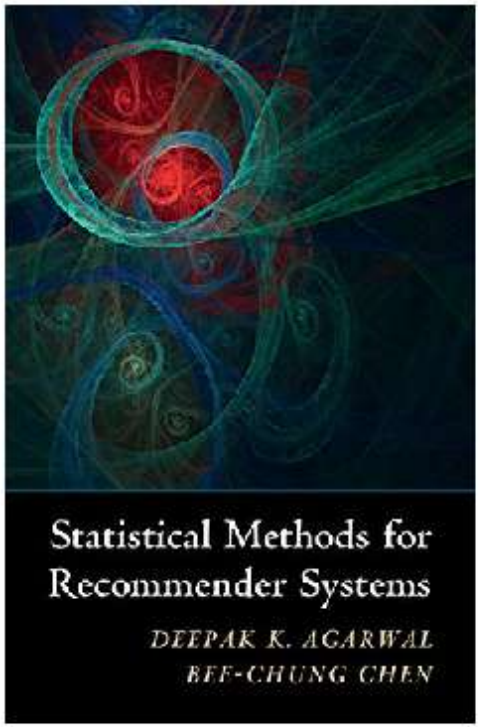
Quelques ouvrages



Ricci et al.
2015



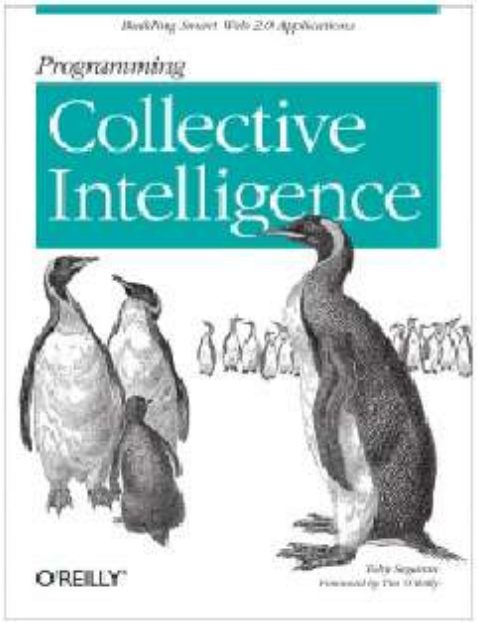
Aggarwal
3/2016



Aggarwal
2/2016

Quelques ouvrages

Pour tester des idées simples ou des projets étudiants:



(Toby Segaran, 2007)

Présentation issue d'un article

A Social Formalism and Survey for Recommender Systems

Daniel Bernardes, Mamadou Diaby*, Raphaël Fournier,
Françoise Fogelman-Soulié, Emmanuel Viennet
Université Paris 13, Sorbonne Paris Cité, L2TI, 93430, Villetaneuse, France.
* Work4 Labs, 3 Rue Moncey, 75009, Paris, France
{daniel.bernardes, mamadou.diaby, raphael.fournier, soulie,
emmanuel.viennet}@univ-paris13.fr

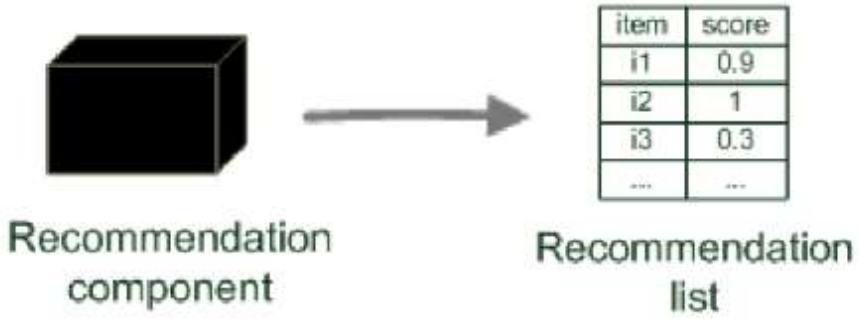
ABSTRACT

This paper presents a general formalism for Recommender Systems based on Social Network Analysis. After introducing the classical categories of recommender systems, we present our Social Filtering formalism and show that it ex-

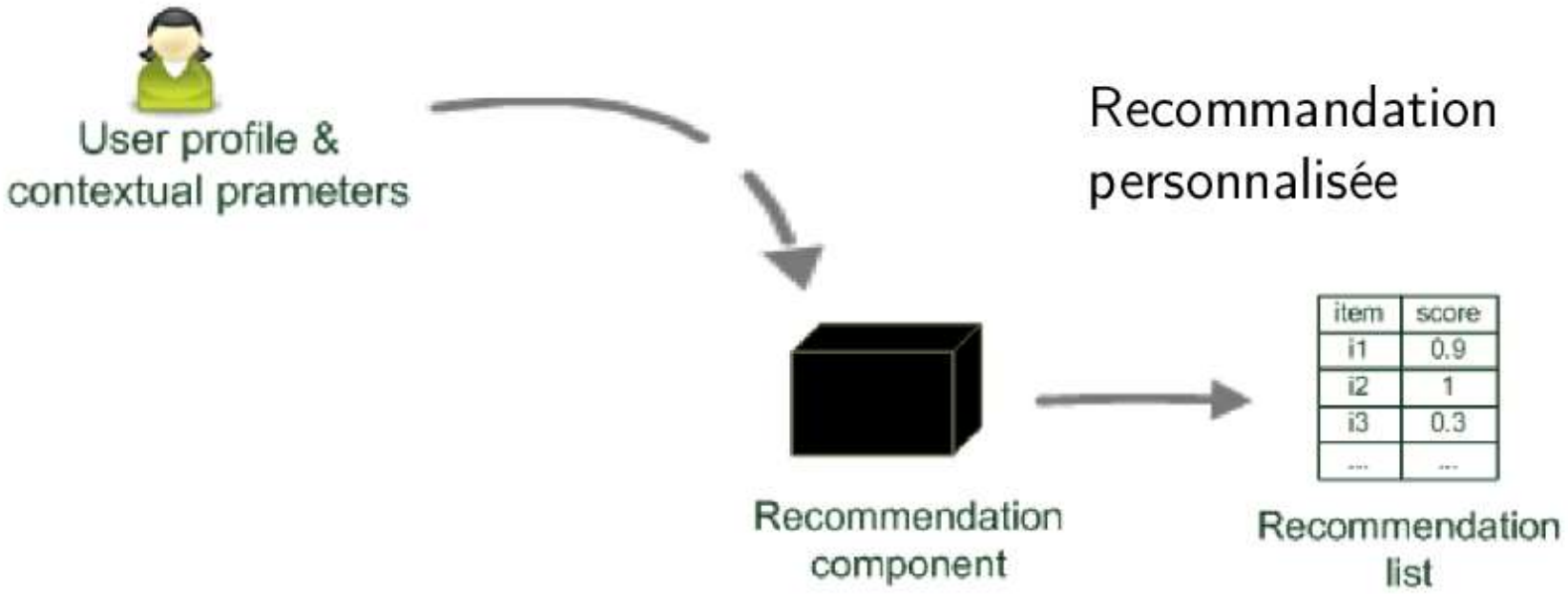
demographics), about products (their features) and about user interactions with the products [8; 25], either *explicit* (rating, satisfaction) or *implicit* (product purchased, book read, song heard, content clicked etc.) More recently, Social networks and social media (blogs, social tagging sites,

SIGKDD Explorations, Vol. 16, Issue 2, December 2014

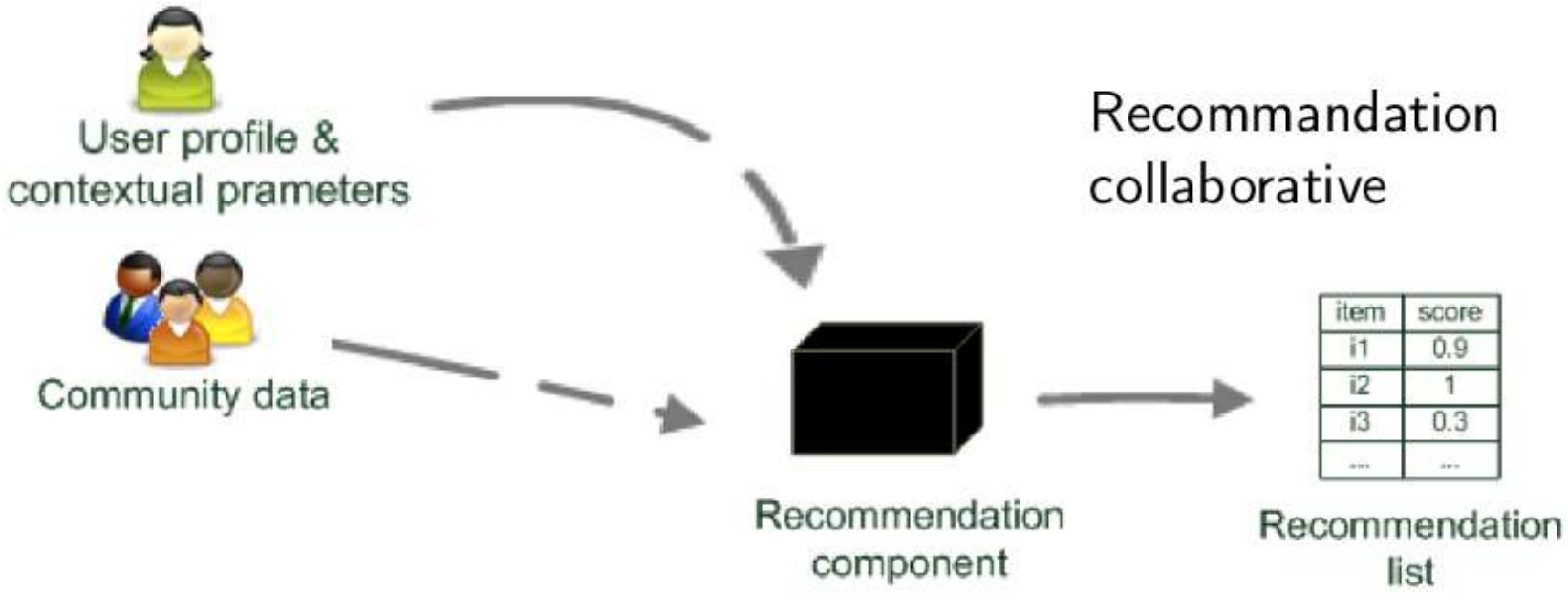
Systèmes de recommandation



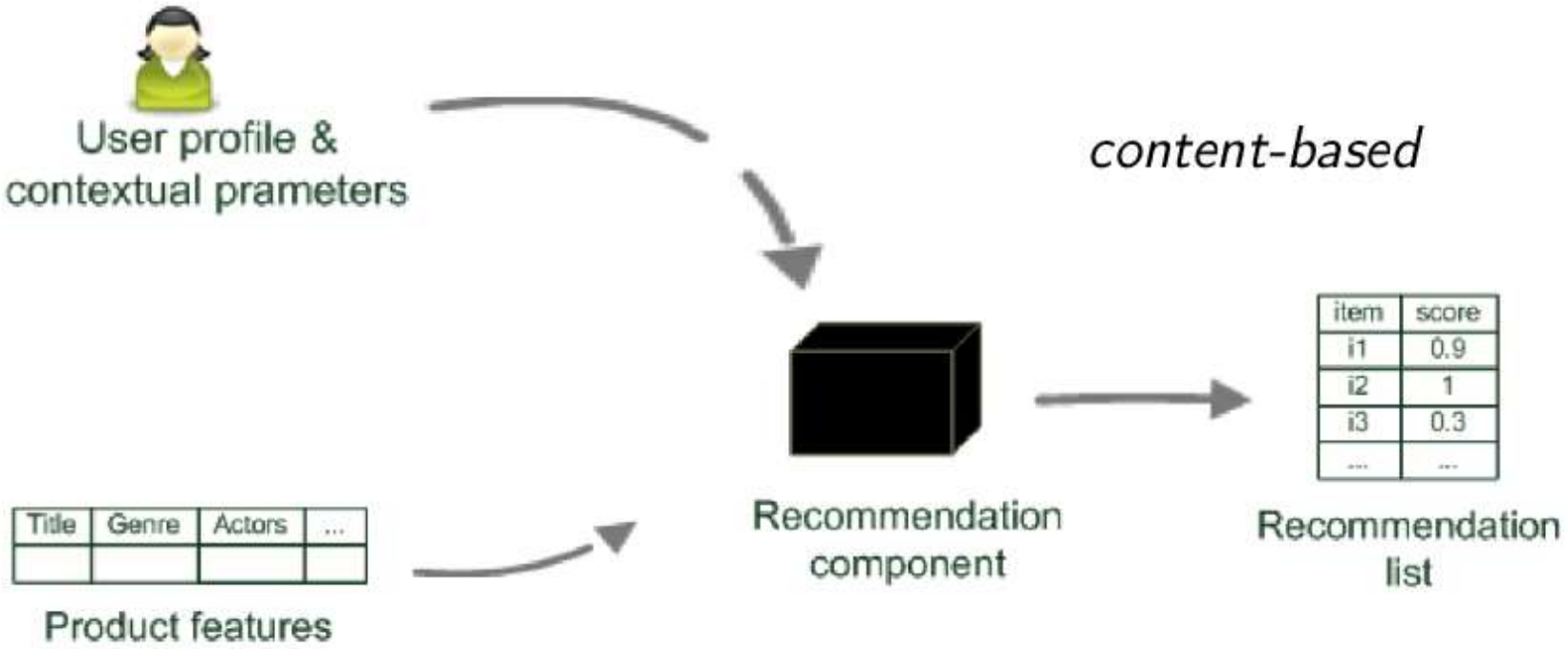
Systèmes de recommandation



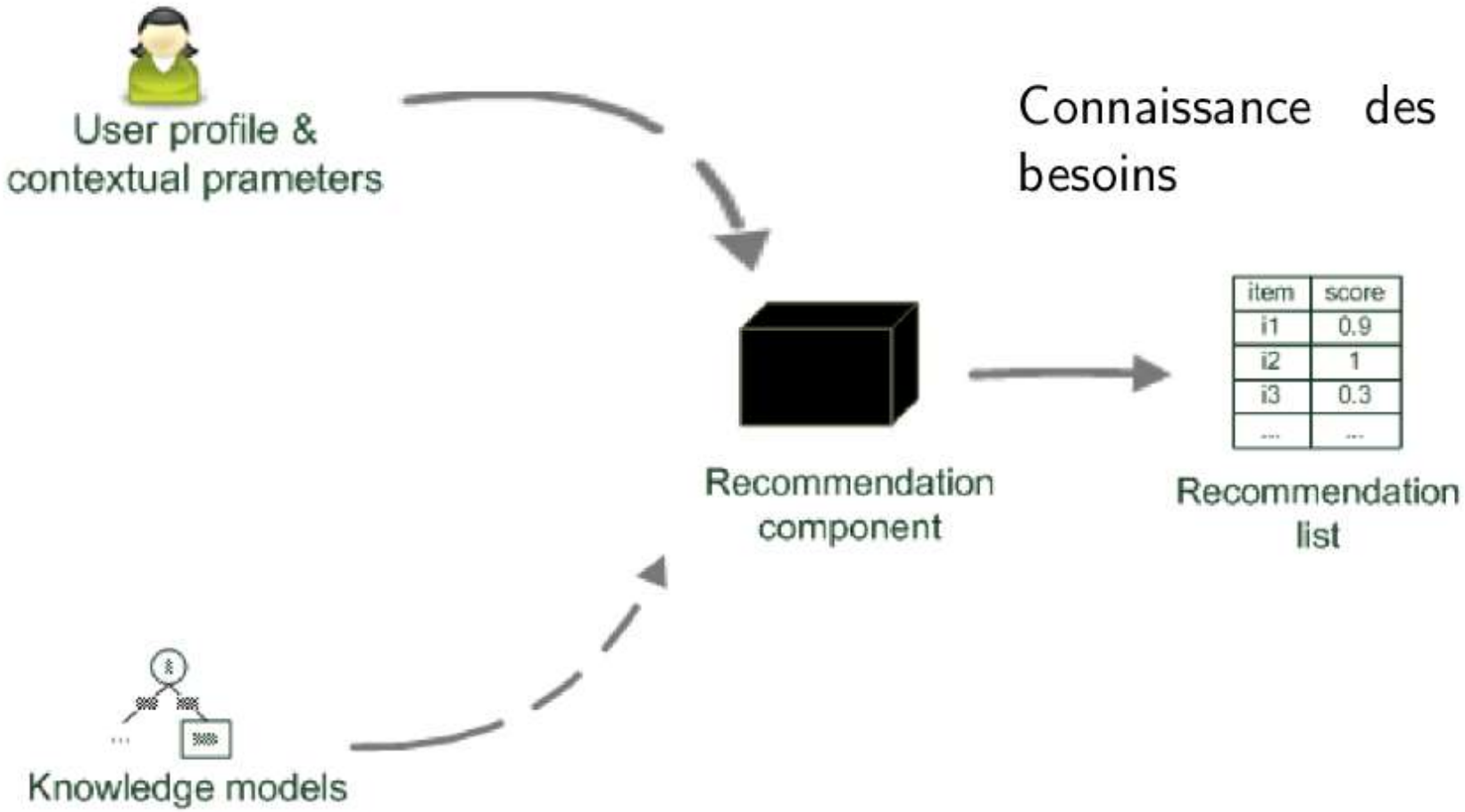
Systèmes de recommandation



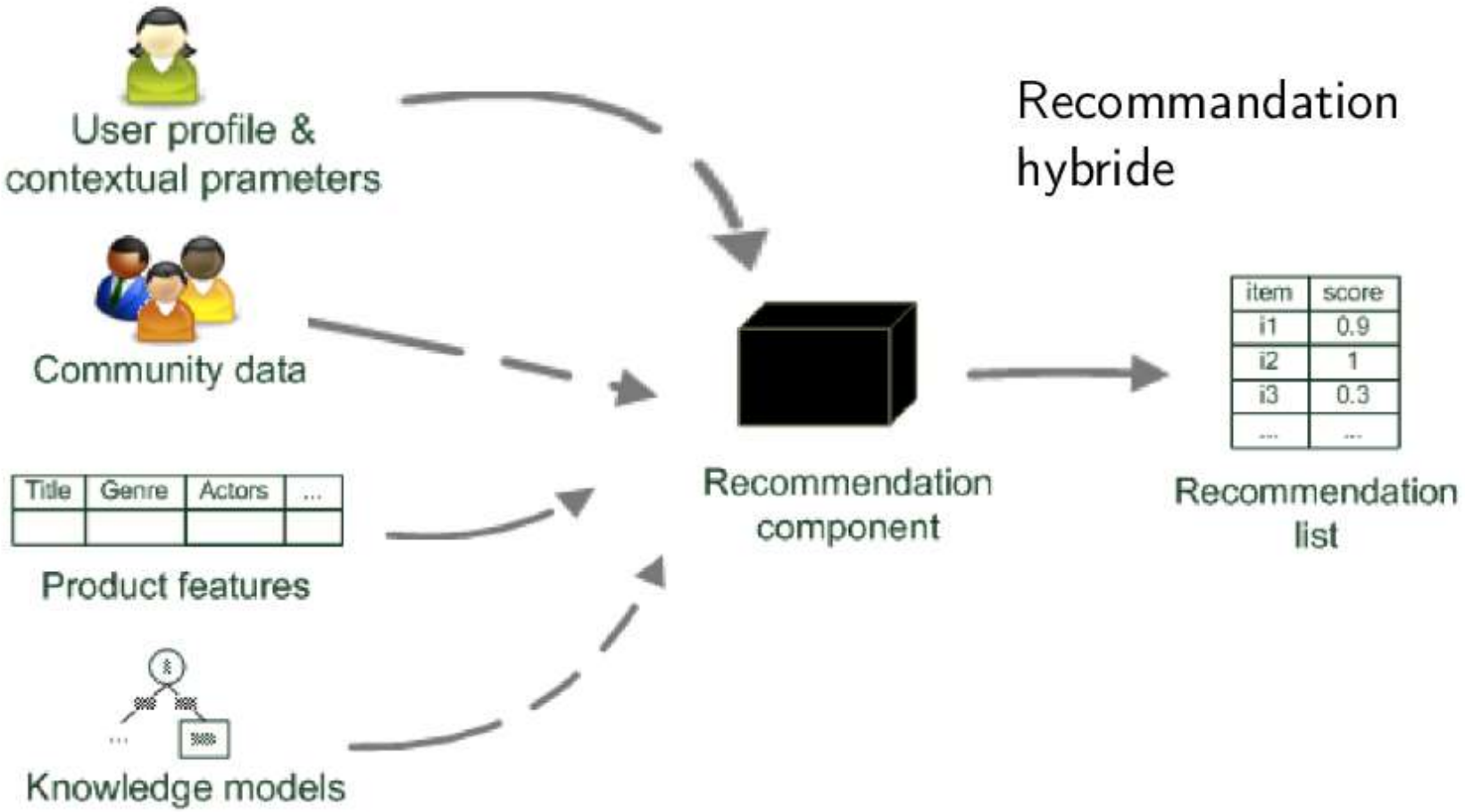
Systèmes de recommandation



Systèmes de recommandation



Systèmes de recommandation



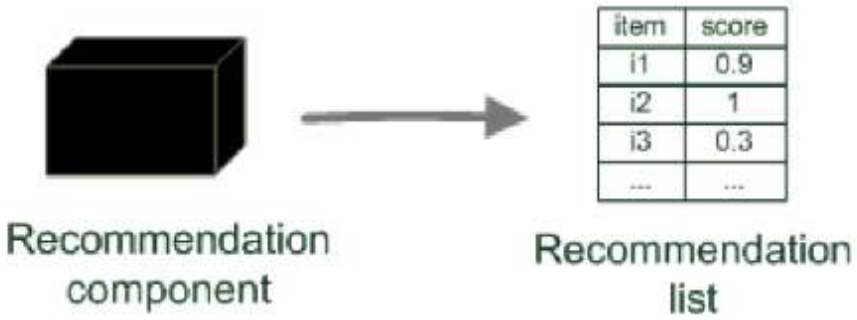
Approches pour les recommandations

	Avantages	Inconvénients
Collaboratives	Pas besoin de contenu, apprend le «marché», «sérendipité»	Nécessite interactions (notes, achats) démarrage à froid
Contenu	Pas besoin d'interactions, comparaison entre items	Nécessite descriptions, pas de surprises, démarrage à froid pour les nouveaux utilisateurs
Connaissances	Déterministe, ok à froid	statique, mise au point coûteuse

Critères d'évaluation:

Le résultat d'un système de recommandation, pour un utilisateur u est:

- soit un score r_{ui} associé à chaque item i;
- soit une liste ordonnée de k produits (ou *items*).



Critères d'évaluation de la recommandation

On veut évaluer si les items recommandés ont été appréciés par l'utilisateur : ont-ils été achetés, consultés, cliqués ?
$\rightsquigarrow$ méthodologies d'évaluation

- Qu'est-ce qu'une bonne recommandation ?
- Quelle stratégie de recommandation ? Lien nécessaire avec le modèle économique

Indicateurs possibles:

- chiffre d'affaire
- promotion de certains produits
- taux de clics, durée des visites
- fidélisation des clients
- satisfaction des utilisateurs

Critères d'évaluation de la recommandation

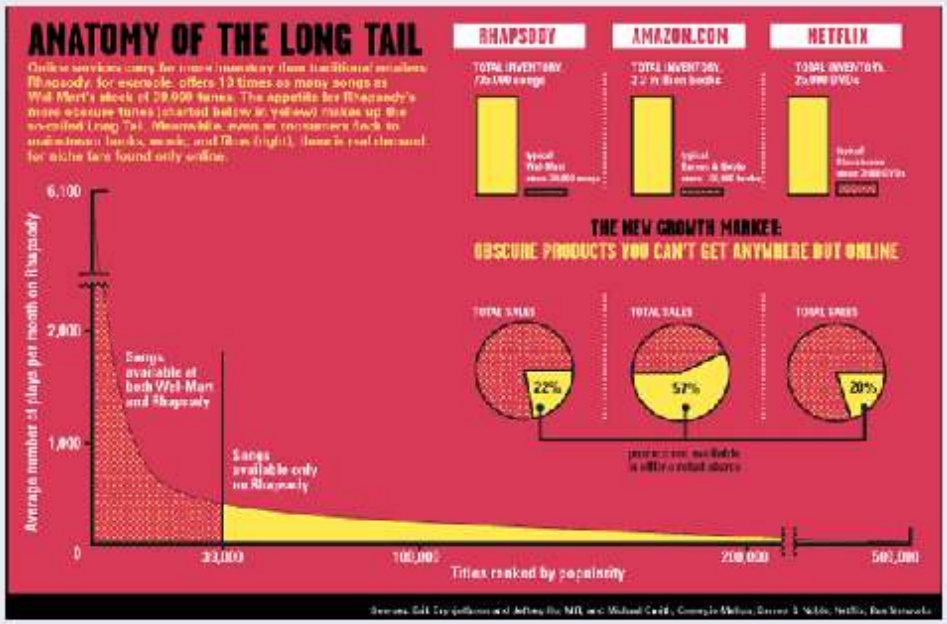
Pour l'aide à la recherche d'information:

- réduire le temps de recherche
- fournir des suggestions correctes

(suppose que l'utilisateur sait ce qu'il veut)

Pour la recommandation proprement dite:

- suggérer des items auxquels l'utilisateur n'a pas pensé (le surprendre, *sérendipité*)
- recommander des items rares, dans la queue de la distribution *(long tail)*



Critères d'évaluation de la recommandation

En pratique:

- Tests A/B
- Enquêtes de satisfaction
- Académiques: données passées, ensembles de test (précautions: nouveaux utilisateurs, nouveaux items)

Critères d'évaluation: Prévision de score

On évalue la proximité entre le **score** prédit $\hat{r}_{ij}$ et celui qui sera observé r_{ij} (en fait rarement observé !)

- AUC *Area Under the Curve*
- RMSE (*Root Mean Square Error*) et MAE (*Mean Absolute Error*) définies comme:

$$RMSE = \sqrt{\frac{1}{I}\sum_{i,j}(r_{ij} - \hat{r}_{ij})^2}$$
$$MAE = \frac{1}{I}\sum_{i,j}|r_{ij} - \hat{r}_{ij}| \quad (1)$$

où I est le nombre de scores de l'ensemble d'évaluation.

Critères d'évaluation: recherche d'information

- *Rappel@k* et *Precision@k* définis comme :

$$\begin{aligned} \text{Rappel@}k &= \frac{1}{L}\sum_a \frac{Card(R_a \cap T_a)}{Card(T_a)} \\ \text{Précision@}k &= \frac{1}{L}\sum_a \frac{Card(R_a \cap T_a)}{k} \end{aligned} \tag{2}$$

où $R_a = (i_1^a, i_2^a, ..., i_k^a)$ est l'ensemble des k items recommandés à a et T_a l'ensemble cible pour a. On peut aussi calculer l'AUC de la courbe de *Rappel@k* en fonction de k.

- *Rappel@k* et *Precision@k* définis comme :

$$\text{Rappel@}k = \frac{1}{L}\sum_{a}\frac{Card(R_a \cap T_a)}{Card(T_a)}$$
$$\text{Précision@}k = \frac{1}{L}\sum_{a}\frac{Card(R_a \cap T_a)}{k} \quad (2)$$

  où $R_a = (i_1^a, i_2^a, ..., i_k^a)$ est l'ensemble des k items recommandés à a et T_a l'ensemble cible pour a. On peut aussi calculer l'AUC de la courbe de *Rappel@k* en fonction de k.

- F_β-mesure : F_β prend en compte à la fois le rappel et la précision. La mesure F_1 est la plus courante. F_β est définie comme :

$$F_\beta@k = \frac{(1+\beta^2) \times prec@k \times rappel@k}{(\beta^2 \times prec@k) + rappel@k} \quad (3)$$

- MAP (Kaggle), ...

Critères d'évaluation: qualitatifs

- **couverture des utilisateurs** : proportion de ceux qui reçoivent des recommandations

$$\text{UsersCoverage@}k = \frac{\#\text{ Utilisateurs dans Test avec } k \text{ recos}}{L_{\text{test}}}$$

où L_{test} est le nombre d'utilisateurs de l'ensemble de test.

- **nombre moyen de recommandations** :

$$\text{AvNbRec@}k = \sum_{K=0}^{k-1} K \frac{\#\text{ Utilisateurs de Test avec } k \text{ recos}}{L_{\text{test}} - \#\text{ Utilisateurs avec } k \text{ recos}}$$

- **couverture des items** :

$$\text{ItemsCoverage@}k = \frac{\#\text{ Items dans les listes de recos}}{C}$$

Critères d'évaluation: qualitatifs (suite)

Couverture de la tête et de la traîne : Si on ordonne les items par popularité décroissante (nombre d'utilisateur ayant consommé l'item), on peut définir la *tête* comme les 20% items les plus populaires, et la *traîne* comme 80% restant, et on mesure le taux d'items recommandés appartenant à ces deux sous-ensembles :

$$\text{TauxTête@}k = \frac{1}{L_{\text{test}}} \sum_{u \in \text{Test}} \frac{\#\text{ Reco pour } u \in \text{ Tête}}{\#\text{ Reco pour } u}$$

$$\text{TauxTraîne@}k = \frac{1}{L_{\text{test}}} \sum_{u \in \text{Test}} \frac{\#\text{ Reco pour } u \in \text{ Traîne}}{\#\text{ Reco pour } u}$$

où l'on somme sur les L_{test} utilisateurs de l'ensemble de test.

Matrice d'interaction

Matrice d'interaction $\mathbf{R}$ dans le cas de notes (ici des nombres de 1 à 5).

	Monuments Men	Django Unchained	Forrest Gump	Gran Torino	Pulp Fiction
Amy	3			2	5
Paul		4			2
Rob	5	4	1		
Liz	2		3		

L utilisateurs et C produits

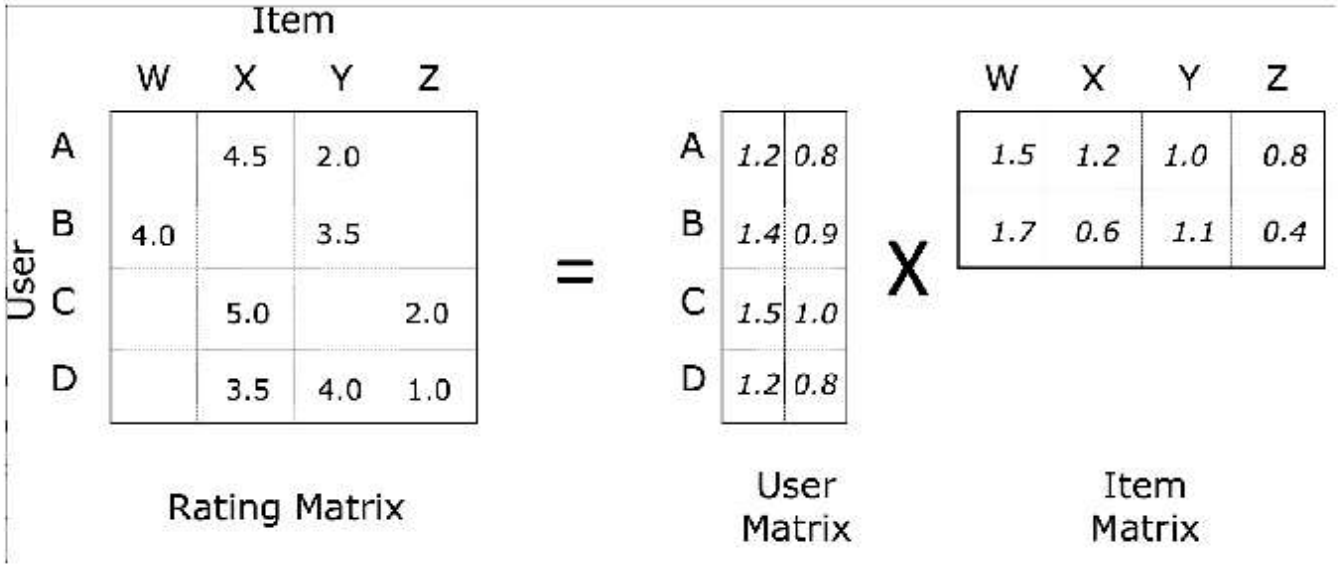
- $\mathbf{R}$ est extrêmement creuse

Familles d'approches pour la recommandation

- Basées sur le contenu
  $\rightsquigarrow$ définir similarité entre items (TF-IDF, ...)

- Collaboratives
  - Modèles de score:
    - modèles probabilistes (réseaux bayesiens)
    - SVM, NN, ...
    - factorisation de matrices
  - Méthodes à mémoire
    - règles d'associations
    - filtrage collaboratif (CF)

Modèles basés sur la factorisation de matrices



- réduction de dimension: ACP
- SVD décomposition en valeurs singulières
- NMF *Non-negative matrix factorization*

Coûteux mais implémentations distribuées (*Big Data*)
+ versions *online*

Règles d'association

Transactions

id	items
1	Pain
2	Lait, Beurre
3	Pain, Vin, Camembert
4	Vin, Camembert
...	...

Règles

$\{ \text{Vin} \} \Longrightarrow \{ \text{Camembert} \}$

$\{ \text{Pain, Vin} \} \Longrightarrow \{ \text{Camembert} \}$

On calcule le *support* et la *confiance* de chaque règle.

Principe général des méthodes à mémoire

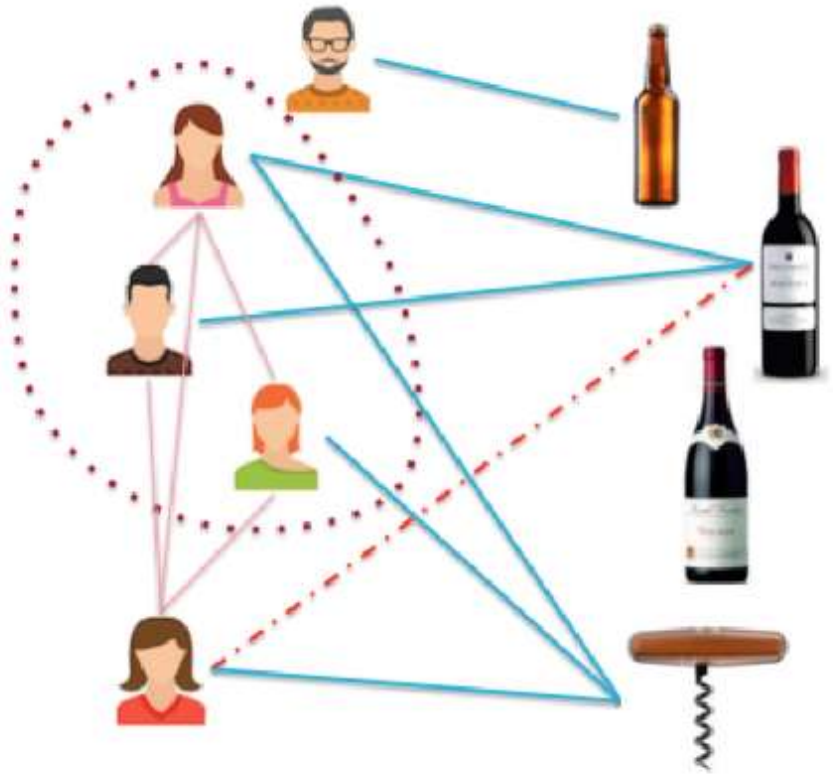
Similarité
Entre utilisateurs ou
entre items

Score pour trier les items

recommandations

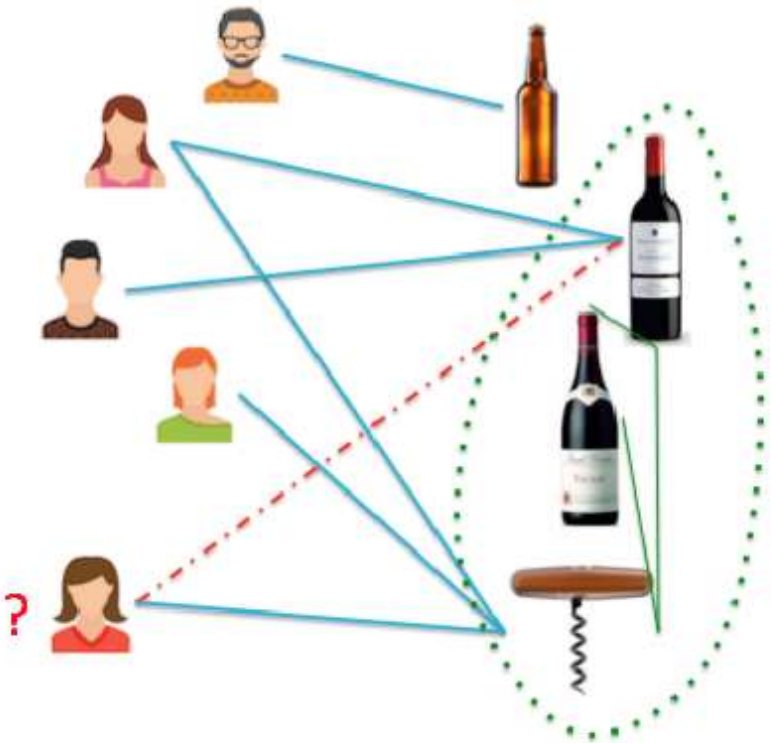
Principe général des méthodes à mémoire

user-based: comparaisons entre utilisateurs



Principe général des méthodes à mémoire

item-based: comparaisons entre items (produits)



Similarité entre utilisateurs

	Monuments Men	Django Unchained	Forrest Gump	Gran Torino	Pulp Fiction
Amy	3			2	5
Paul		4			2
Rob	5	4	1		
Liz	2		3		

$\overrightarrow{c}(i)$

$\} \overrightarrow{l}(a)$

Similarité cosinus:

$$\text{Sim}(a,u) = \cos\left[\overrightarrow{l}(a), \overrightarrow{l}(u)\right]$$

$$= \frac{\sum_{i=1}^{C} r_{ai} \times r_{ui}}{\sqrt{\sum_{i=1}^{C} (r_{ai})^2} \sqrt{\sum_{i=1}^{C} (r_{ui})^2}}$$

Similarité entre utilisateurs

	Monuments Men	Django Unchained	Forrest Gump	Gran Torino	Pulp Fiction
Amy	3			2	5
Paul		4			2
Rob	5	4	1		
Liz	2		3		

$\overrightarrow{c}(i)$

$\} \overrightarrow{l}(a)$

Similarité «Pearson» (PCC):

$$\text{Sim}(a,u) = \text{PCC}\left[\overrightarrow{l}(a), \overrightarrow{l}(u)\right]$$

$$= \frac{\sum_{i \in I(a) \cap I(u)} \left[r_{ai} - \bar{l}(a)\right]\left[r_{ui} - \bar{l}(u)\right]}{\sqrt{\sum_{i \in I(a) \cap I(u)} \left[r_{ai} - \bar{l}(a)\right]^2} \sqrt{\sum_{i \in I(a) \cap I(u)} \left[r_{ui} - \bar{l}(u)\right]^2}}$$

Similarité entre utilisateurs

	Monuments Men	Django Unchained	Forrest Gump	Gran Torino	Pulp Fiction
Amy	3			2	5
Paul		4			2
Rob	5	4	1		
Liz	2		3		

$\overrightarrow{c}(i)$

$\} \overrightarrow{l}(a)$

Cosinus asymétrique:

$$\text{Sim}(a,u) = \text{asym-cos}_{\alpha}\left[\overrightarrow{l}(a), \overrightarrow{l}(u)\right]$$

$$= \frac{\sum_{i=1}^{C} r_{ai} \times r_{ui}}{\left[\sum_{i=1}^{C} r_{ai}^2\right]^{\alpha} \times \left[\sum_{i=1}^{C} r_{ui}^2\right]^{1-\alpha}}$$

Similarités «sociales»

Si $\mathbf{R}$ est binaire, le lien entre deux utilisateurs a et u (ou deux items i et j) représente une *règle d'association*

$$\text{Supp}(a \to u) = \frac{\#\text{ Items cons. by } a \text{ and } u}{\#\text{ Items}} = \frac{1}{C}\sum_{i=1}^{C} r_{ai} r_{ui}$$

$$\text{Supp}(i \to j) = \frac{\#\text{ Users who cons. } i \text{ and } j}{\#\text{ Users}} = \frac{1}{L}\sum_{u=1}^{L} r_{ui} r_{uj}$$

$$\text{Supp}(a \to u) = \frac{\#\text{ Items cons. by } a \text{ and } u}{\#\text{ Items}} = \frac{1}{C}\sum_{i=1}^{C} r_{ai} r_{ui}$$

se généralise au cas où $\mathbf{R}$ n'est pas binaire:

$$\text{Supp}(a \to u) = \frac{1}{C}\sum_{i=1}^{C} r_{ai} r_{ui}$$

qui a la même forme que la similarité cosinus:

$$\cos(a, u) = \frac{\sum_{i=1}^{C} r_{ai} \times r_{ui}}{\sqrt{\sum_{i=1}^{C} (r_{ai})^2}\sqrt{\sum_{i=1}^{C} (r_{ui})^2}}$$

$\rightsquigarrow$ utiliser le support comme mesure de similarité

Similarités «sociales»: Jaccard

L'indice de Jaccard mesure la similarité d'ensembles d'objets en comptant le nombre d'éléments qu'ils ont en commun. Dans le cas binaire, *user-based*:

$$\text{Jaccard}(a,u) = \frac{Card\left[\overrightarrow{l}(a) \cap \overrightarrow{l}(u)\right]}{Card\left[\overrightarrow{l}(a) \cup \overrightarrow{l}(u)\right]}$$

$$= \frac{\#\text{Items Cons. par } a \text{ et } u}{(\#\text{Items Cons. par } a) + (\#\text{Items Cons. par } u) - (\#\text{Items Cons. par } a \text{ et } u)}$$

$$= \frac{\sum_{i=1}^{C} r_{ai} \times r_{ui}}{\sum_{i=1}^{C} r_{ai} + \sum_{i=1}^{C} r_{ui} - \sum_{i=1}^{C} r_{ai} \times r_{ui}}$$

Voisinage $K(a)$

A partir de la similarité:

- seuil sur la distance
- k plus proches voisins

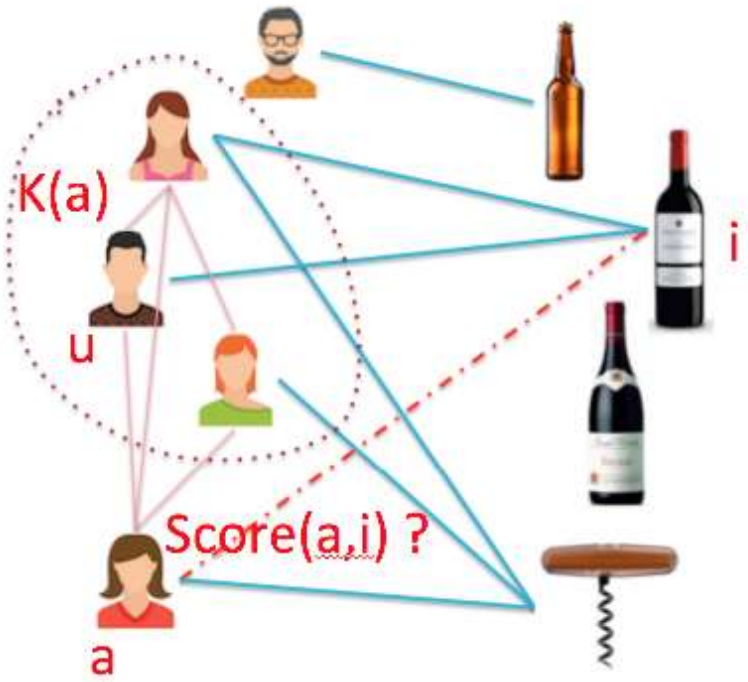
A partir des graphes:

liens implicites ou *explicites* (sociaux)

- voisins (premier cercle)
- communauté locale
- communauté globale (issue d'une partition du graphe maximisant la *modularité*)

Voisinage et score

Score rendant compte du «lien» entre l'utilisateur a et l'item i:

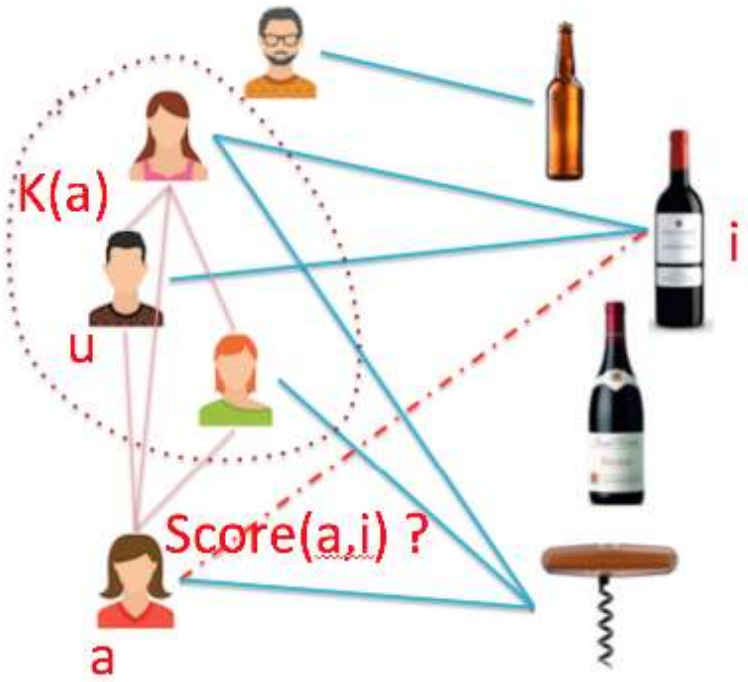


$$\text{Score}(a,i) = \sum_{u \in K(a)} r_{ui} \times f\left[\text{Sim}(a,u)\right]$$

$K(a)$ est un voisinage de a:
utilisateurs «proches»

Voisinage et score

Score rendant compte du «lien» entre l'utilisateur a et l'item i:



$K(a)$ est un voisinage de a: utilisateurs «proches»

Variantes utilisées:

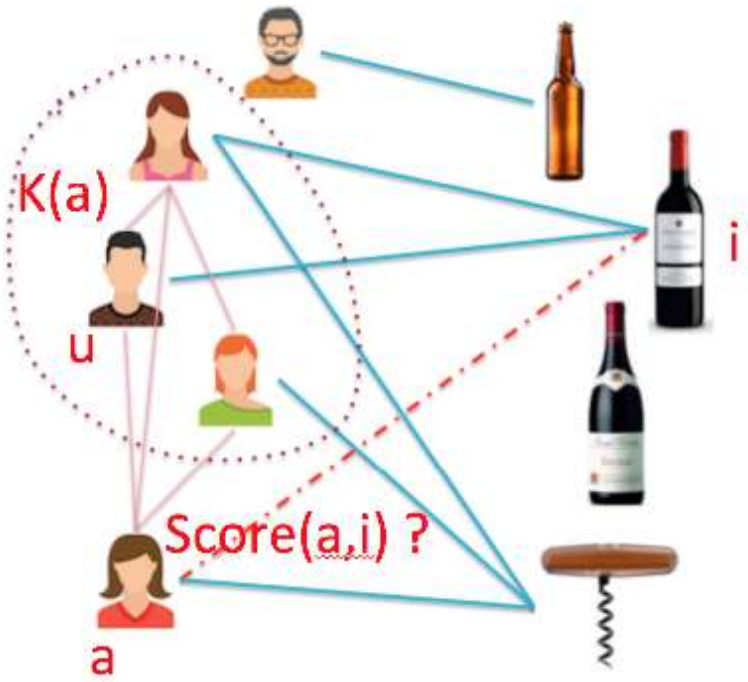
$$\text{Score}(a,i) = \frac{1}{\text{card}[K(a)]} \sum_{u \in K(a)} r_{ui}$$

$$\text{Score}(a,i) = \frac{\sum_{u \in K(a)} r_{ui} \times \text{Sim}(a,u)}{\sum_{u \in K(a) \cap U(i)} |\text{Sim}(a,u)|}$$

$$\text{Score}(a,i) = \bar{l}(a) + \frac{\sum_{u \in K(a) \cap U(i)} (r_{ui} - \bar{l}(u)) \times \text{Sim}(a,u)}{\sum_{u \in K(a) \cap U(i)} |\text{Sim}(a,u)|}$$

Voisinage et score

Score rendant compte du «lien» entre l'utilisateur a et l'item i:



$K(a)$ est un voisinage de a:
utilisateurs «proches»

Autre variante (Aiolli 2013):

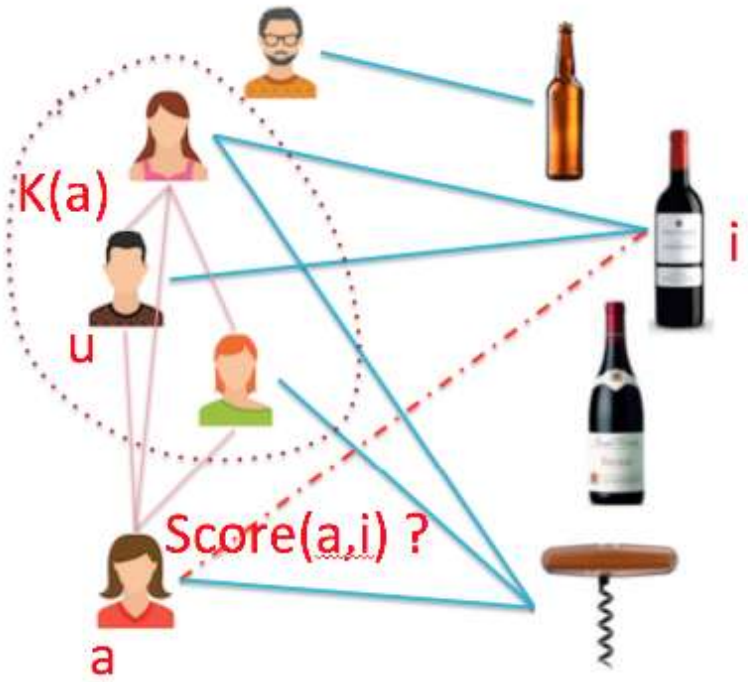
$$\text{Score}(a,i) = \sum_{u \in K(a)} r_{ui} \times \left[\text{Sim}(a,u)\right]^q$$

ou en *user-based*:

$$\text{Score}(a,i) = \sum_{j \in V(i)} r_{aj} \times \left[\text{Sim}(i,j)\right]^{q'}$$

Voisinage et score

Score rendant compte du «lien» entre l'utilisateur a et l'item i:



$K(a)$ est un voisinage de a:
utilisateurs «proches»

Puis on recommande k items:

$$\text{Score}(a, i_1^a) \geq \text{Score}(a, i_2^a) \geq ... \geq \text{Score}(a, i_k^a)$$

Comparaisons empiriques

Données	MovieLens1M		Flixster	
Méthodes	CF IB (cosine, average)	ACF IB (α=0, q=5)	CF IB (cosine, average)	ACF IB (α=0, q=5)
Couverture des items	-	-	-	-
- Proportion dans tête	22.16%	12.44%	15.24%	5.40%
- Proportion dans traîne	66.97%	87.75%	92.00%	99.00%
Temps de calcul	33.03%	12.25%	8.00%	1.00%
	0 :05 :00	0 :05 :00	2 :30 :00	2 :30 :00

Données	MSD	
Méthodes	CF IB (cosine, average)	ACF IB (α=0, q=5)
Couverture des items	-	-
- Proportion dans tête	0.00%	0.00%
- Proportion dans traîne	-	-
Temps de calcul	2 :30 :00	2 :30 :00

Comparaisons empiriques

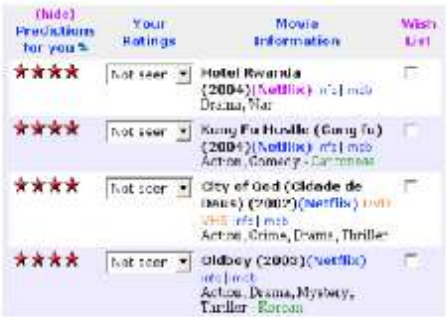
Objectifs

- se forger une intuition sur les propriétés des méthodes
- tester les implémentations (*benchmarks* reproductibles)

Voir `https://bitbucket.org/danielbernardes/socialfiltering`

Jeux de données

Données	#Evals	Type d'éval.	Util.	Items	Réseau Social
LastFM	92 834	comptes	1892	17 632	25 434
MovieLens1M	1 M	notes	6 040	3 883	–
Flixster	8,2 M	notes	1 M	49 000	26,7 M
MSD	48 M	comptes	1,2 M	380 000	–

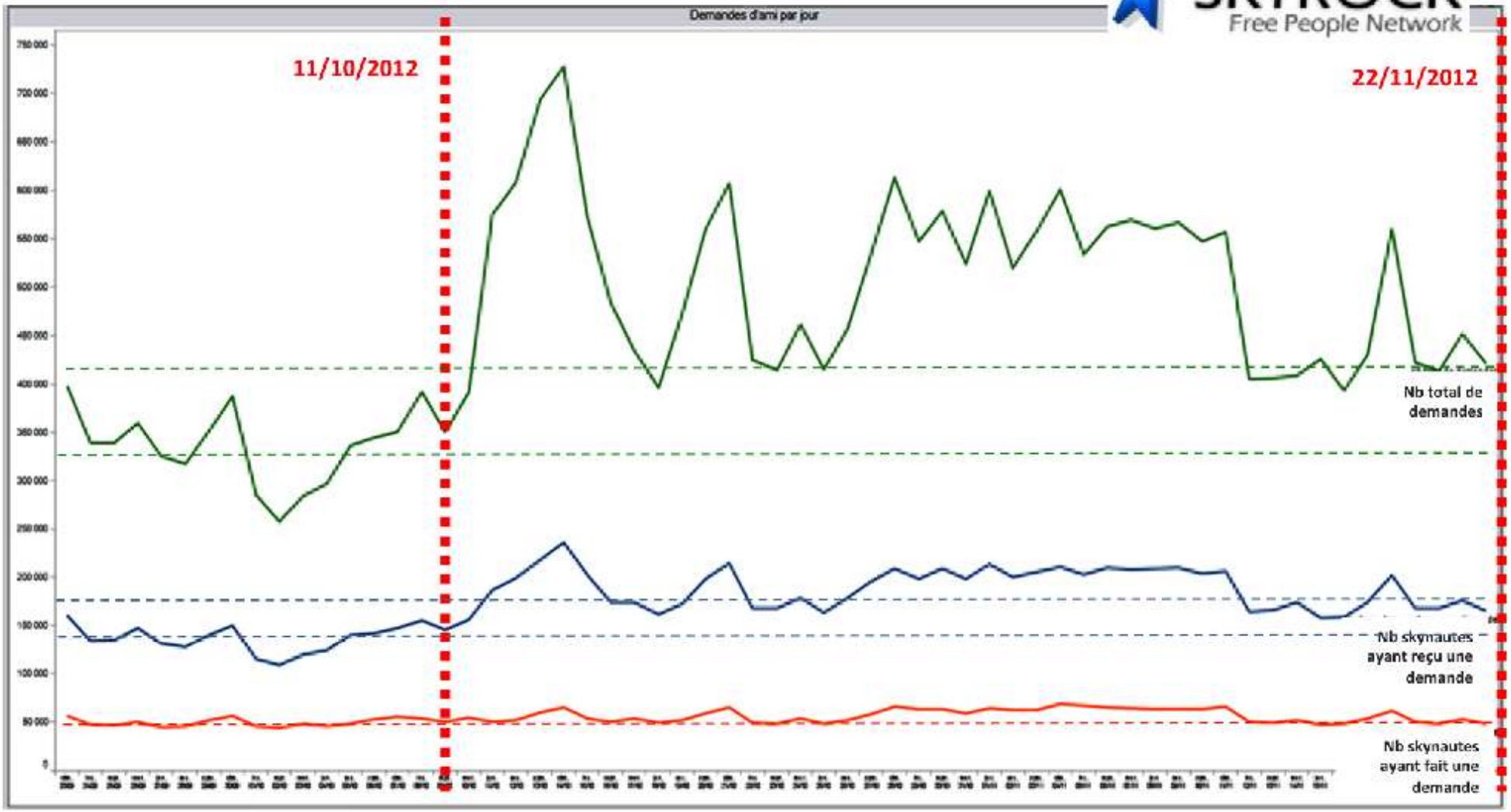


Systèmes de référence

Données / Méthodes	LastFM			Flixster		
	Popularité	Bigrammes	NMF	Popularité	Bigrammes	NMF
MAP @ 10	0.058	**0.144**	0.005	0.038	**0.098**	0.005
Prec @ 10	0.053	**0.082**	0.048	0.061	**0.120**	0.075
Rappel @ 10	0.165	**0.260**	0.161	0.092	**0.126**	0.116
Couverture des utilisateurs	**100%**	52.86%	**100%**	**100%**	74.00%	79.65%
Utilisateurs partiellement couverts	0%	47%	0%	0%	26.00%	20.35%
– Nb moyen de recos	-	3.4	-	-	2.3	2.8
Couverture des items	0.46%	1.13%	**2.87%**	0.06%	0.33%	**0.60%**
– Proportion dans traîne	0%	0%	**1.95%**	**0%**	**0%**	**0%**
– Proportion dans tête	100%	100%	**98.05%**	**100%**	**100%**	**100%**
Temps de calcul	**0:01:00**	0:05:00	1:00:00	**0:01:00**	0:05:00	4:00:00

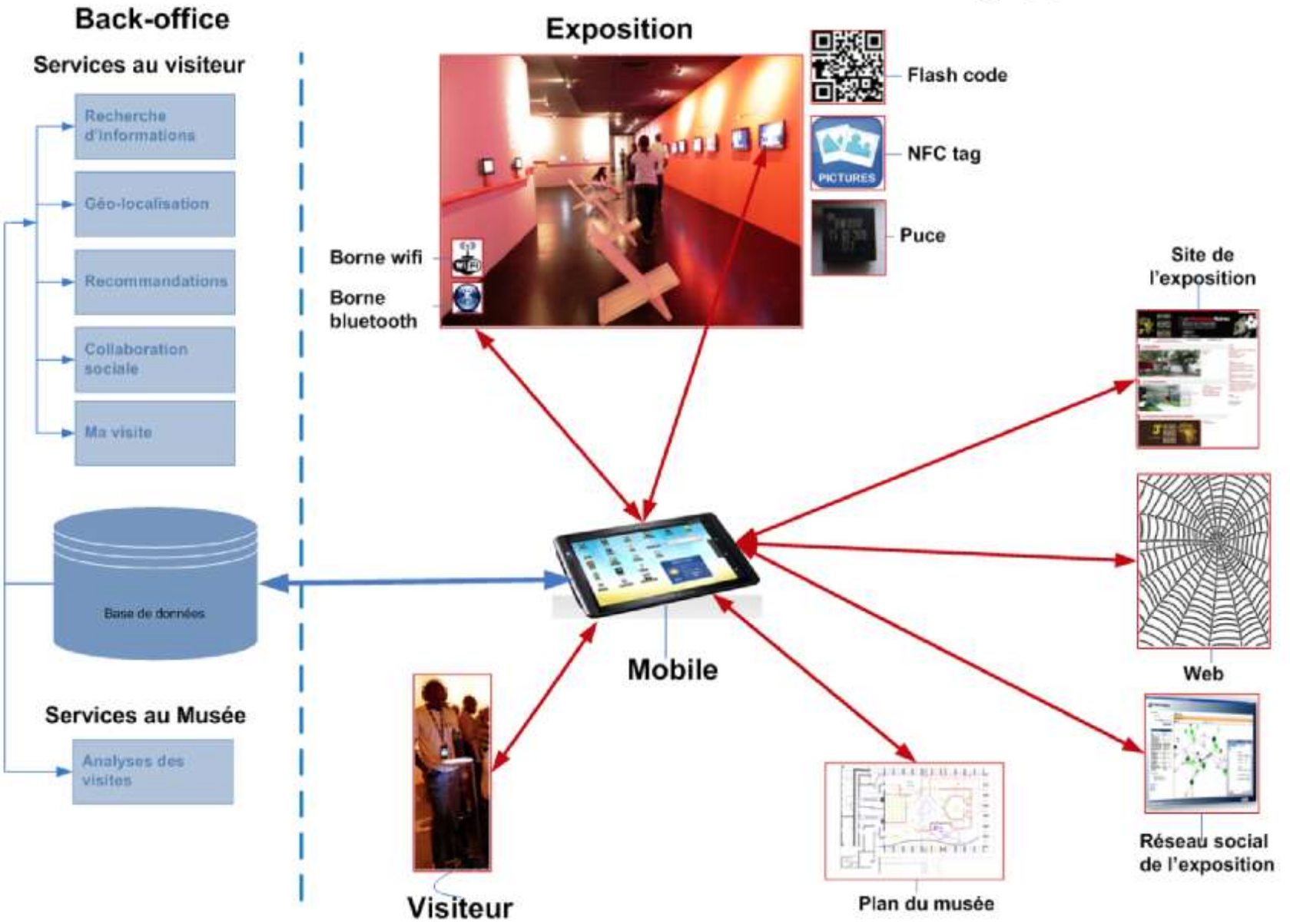
Données / Méthodes	MovieLens1M			MSD		
	Popularité	Bigrammes	NMF	Popularité	Bigrammes	NMF
MAP @ 10	0.097	**0.149**	0.008	0.022	**0.135**	*
Prec @ 10	0.150	**0.206**	0.141	0.017	**0.042**	*
Rappel @ 10	0.120	**0.155**	0.113	0.055	**0.175**	*
Couverture des utilisateurs	**100%**	99.50%	**100%**	**100%**	0%	*
Utilisateurs partiellement couverts	0%	0.50%	0%	0%	100%	*
– Nb moyen de recos	-	5.0	-	-	1.9	*
Couverture des items	0.62%	**3.47%**	**3.47%**	**0.008%**	0.001%	*
– Proportion dans traîne	0%	**0.30%**	0.01%	**0.00%**	**0.00%**	*
– Proportion dans tête	**100%**	**99.70%**	99.99%	**100.00%**	**100.00%**	*
Temps de calcul	**0:01:00**	0:05:00	2:00:00	**0:14:52**	0:30:00	$>$ 4 days

Recommandation d'amis sur Skyrock

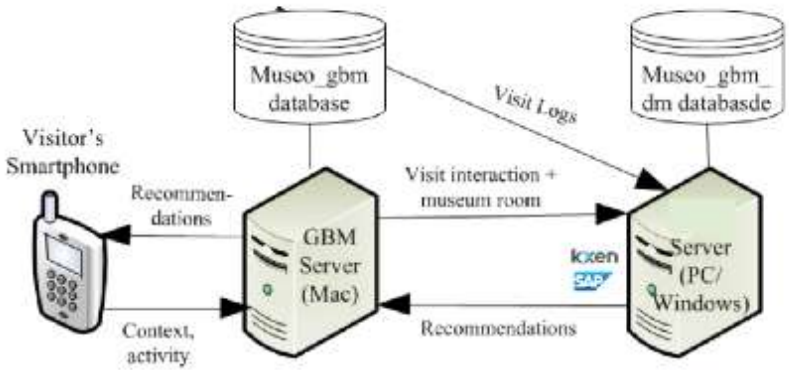


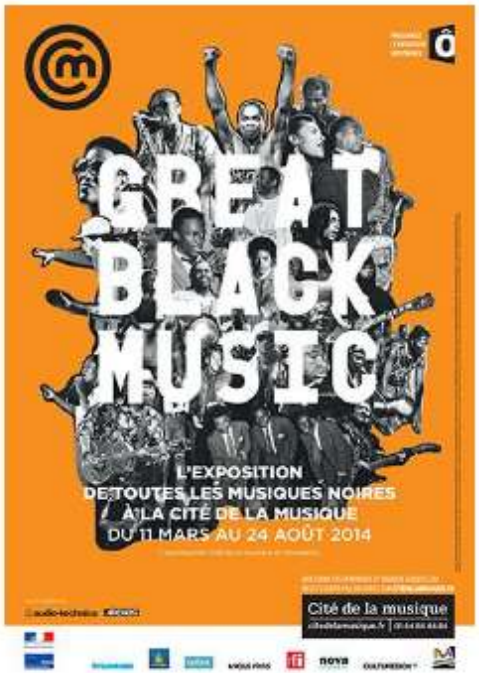
Recommandation d'œuvres dans des musées

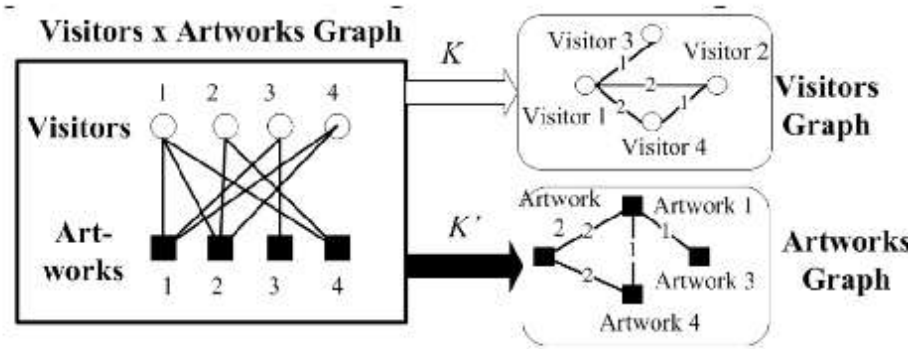
http://ammico.fr



Recommandation d'œuvres dans des musées







Données anonymisées disponibles:
67000 visiteurs, 600 POI, 1,6 millions d'interactions

`http://www-l2ti.univ-paris13.fr/~viennet/GBM.shtml`

Thèse CIFRE de Mamadou Diaby, juin 2015

Recommending jobs to social network users

WORK4

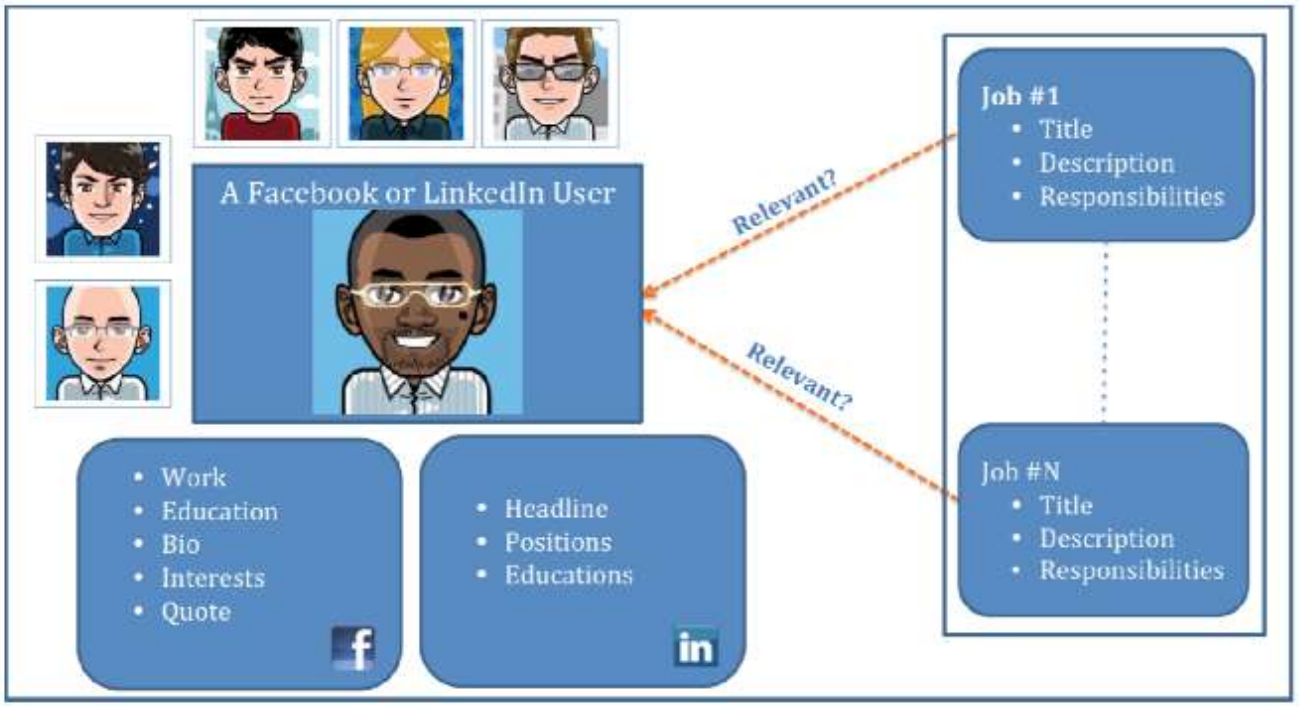
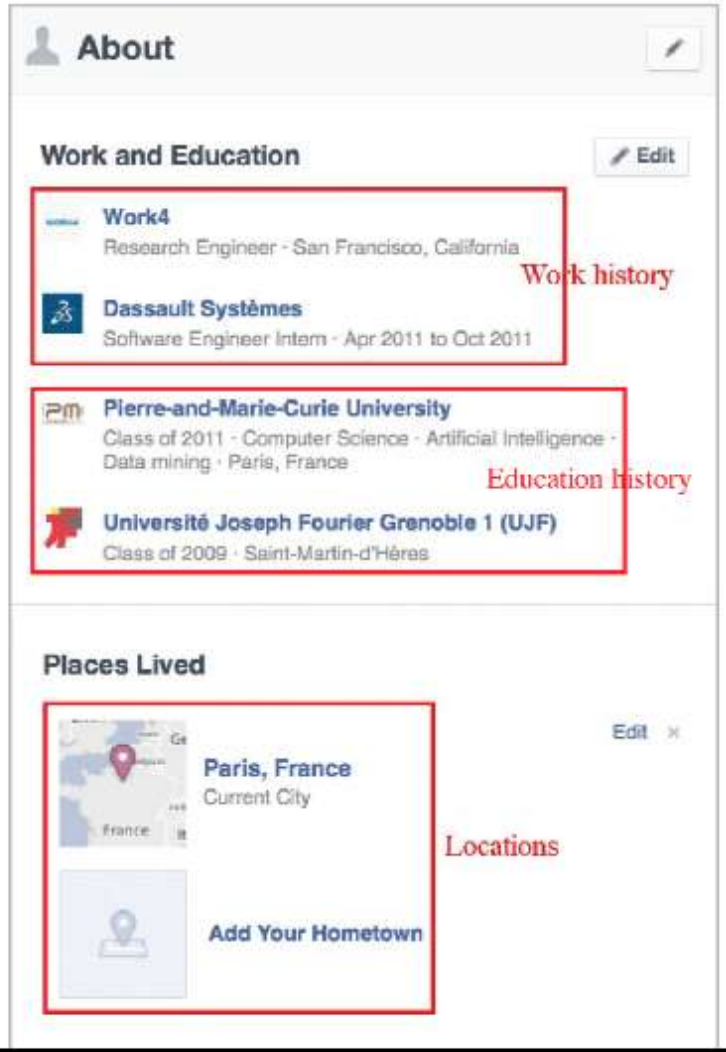


Figure : Modeling of the process of recommending jobs to social network users.

Examples of profiles of users and jobs



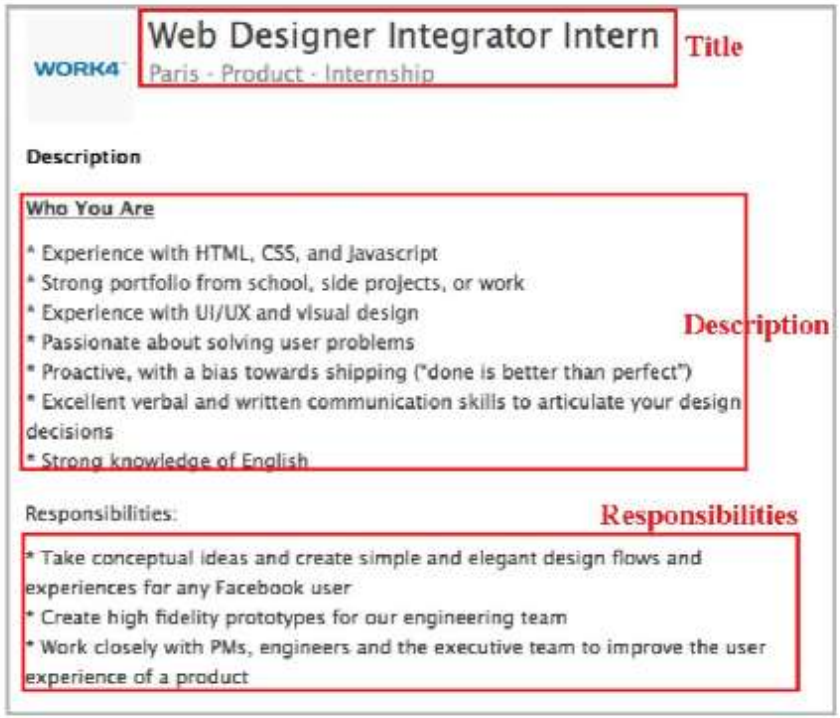


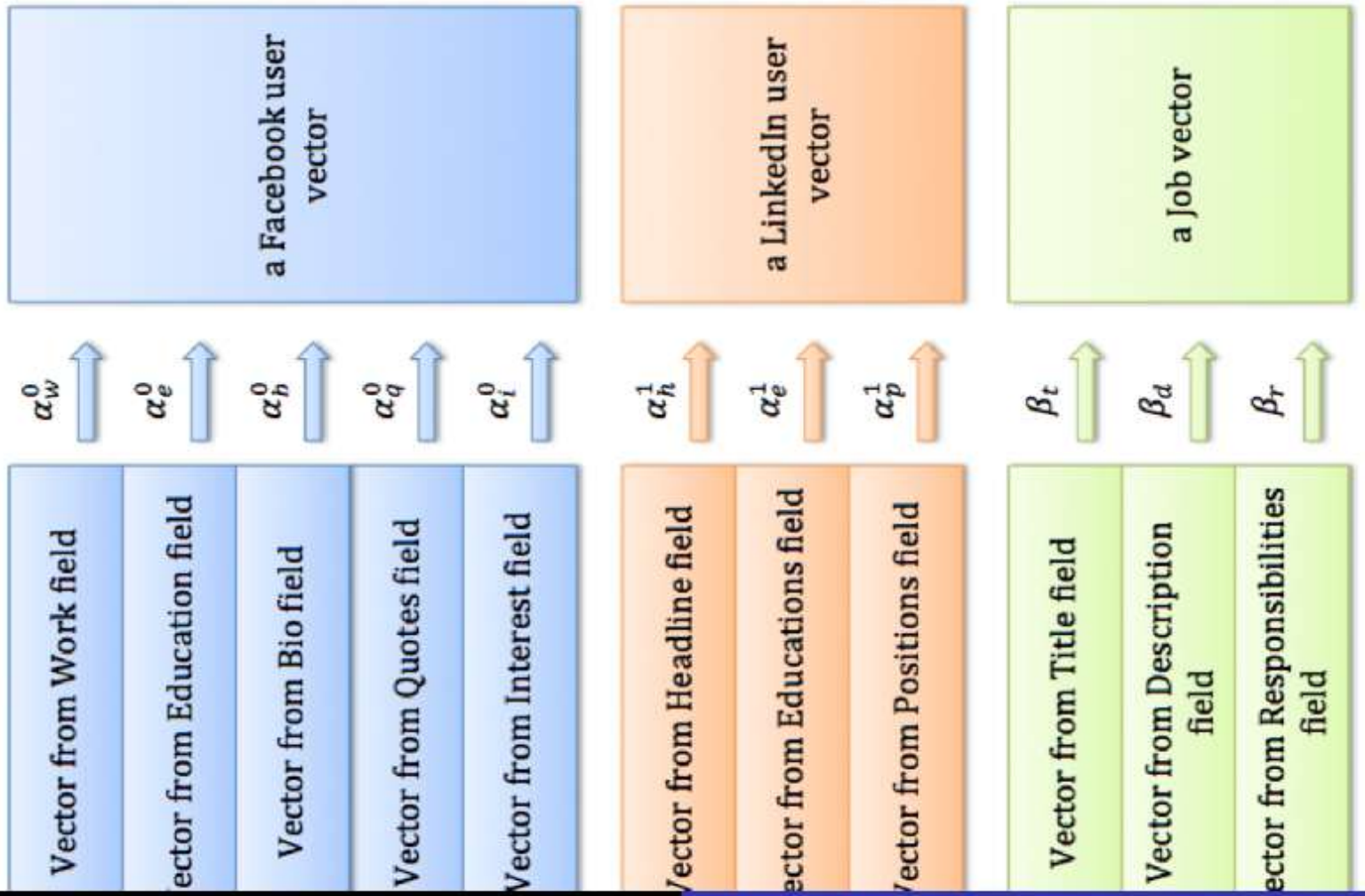
Figure : An example of a job

Summary statistics

		Datasets			
		ALL	Validation	Review	Candidate
	Total number of instances	86,524	54,247	14,414	17,863
	Distinct users	41,303	27,408	7,572	9,232
	Distinct jobs	10,527	1,326	2,171	7,699
Proportion	label 0	0.70	0.91	0.75	0.00
	label 1	0.30	0.09	0.25	1.00
	Facebook users	0.44	0.27	0.30	0.97
	LinkedIn users	0.56	0.73	0.70	0.03

Table : Summary statistics of our datasets; we assumed that users only applied to jobs that match their profiles in Candidate dataset.

Job recommender based on a Bag-of-words model



Job recommender based on a taxonomy-based vector model

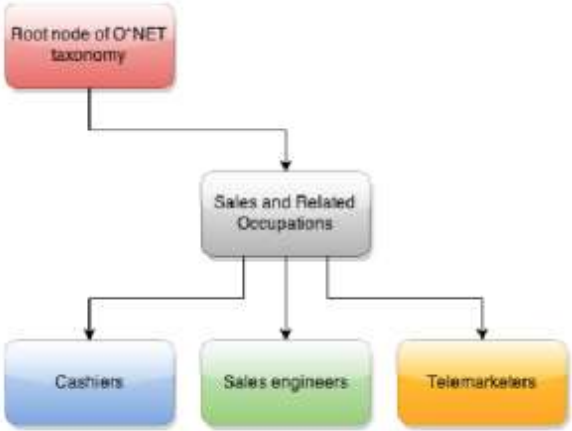


Figure : An O*NET subtree.

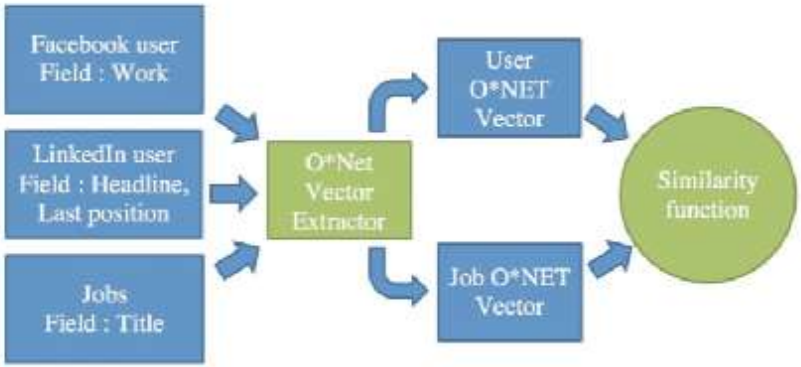


Figure : Scheme of our taxonomy-based job recommender systems.

Engines using O*NET vectors

- Engine-6: O*NET vector
  - Similarity function: Cosine similarity

En guise de conclusion

Quelques problèmes difficiles:

- évaluation
- prise en compte du contexte (eg trajectoires, contraintes applicatives)
- prise en compte du temps
- recommandation structurées
- préférences explicites de l'utilisateurs
- explication à l'utilisateur

En guise de conclusion

- La recommandation pose de nombreux problèmes théoriques et pratiques (intégration, évaluation, passage à l'échelle);
- les règles "métier" sont plus importantes que le choix de l'algorithme;
- peu de travaux académiques s'intéressent à la recommandation, la plupart se penchant sur l'approximation de notes;
- le formalisme présenté unifie de nombreuses approches présentées de façon séparée dans la littérature, et facilite la conception de nouveaux modèles et leur combinaison efficace.