LOG6308 — Systèmes de recommandation

Systèmes de recommandations, Approches filtres collaboratifs

Michel C. Desmarais

Génie informatique et génie logiciel
École Polytechnique de Montréal

Automne, 2017
(version 29 août 2017)

Systèmes de recommandations, Approches filtres collaboratifs

Le problème

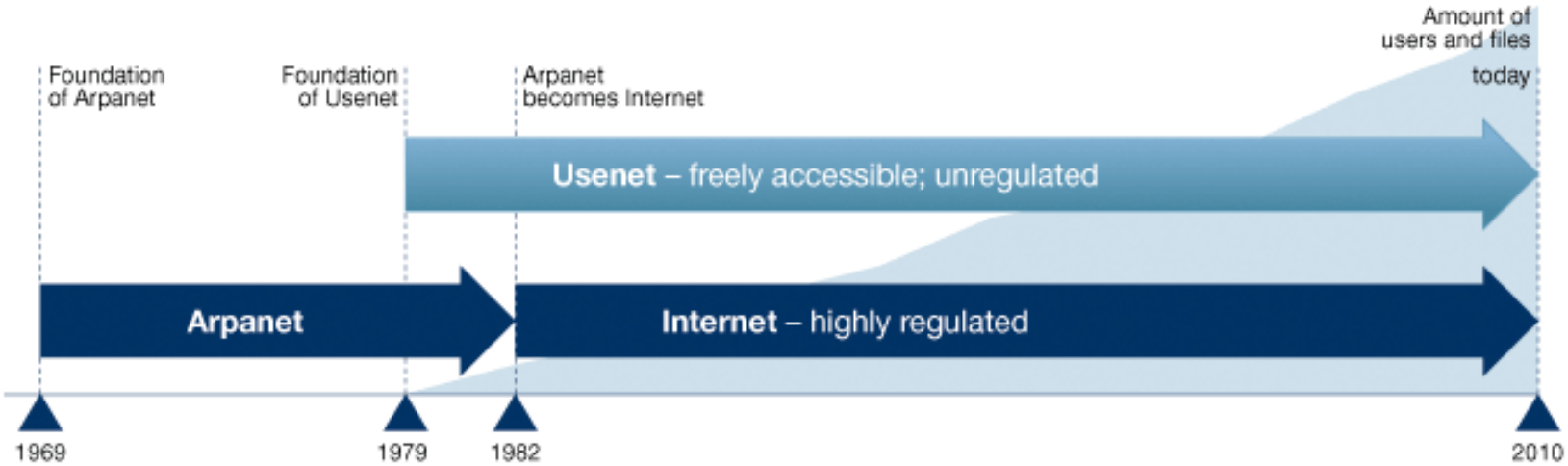
- Recommander ou filtrer l'information pertinente.
- Par exemple :

  Filtres : éliminer ou classifier les courriels indésirés ; filtrer les nouvelles selon nos intérêts.

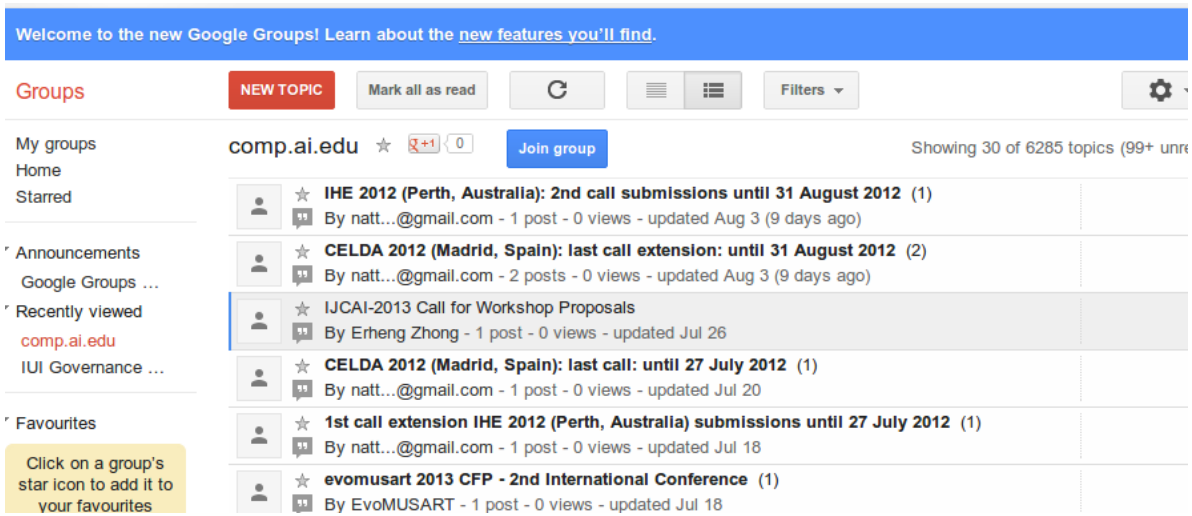
  Recommandations : suggestions d'achats croisés ; publicité personnalisée ;
- Avec l'émergence du web social, les algorithmes de systèmes de recommandation sont de plus répandus

Historique I

Filtrer les messages pertinents sur **Usenet** (GroupLens Project CSCW 1994)



Historique II



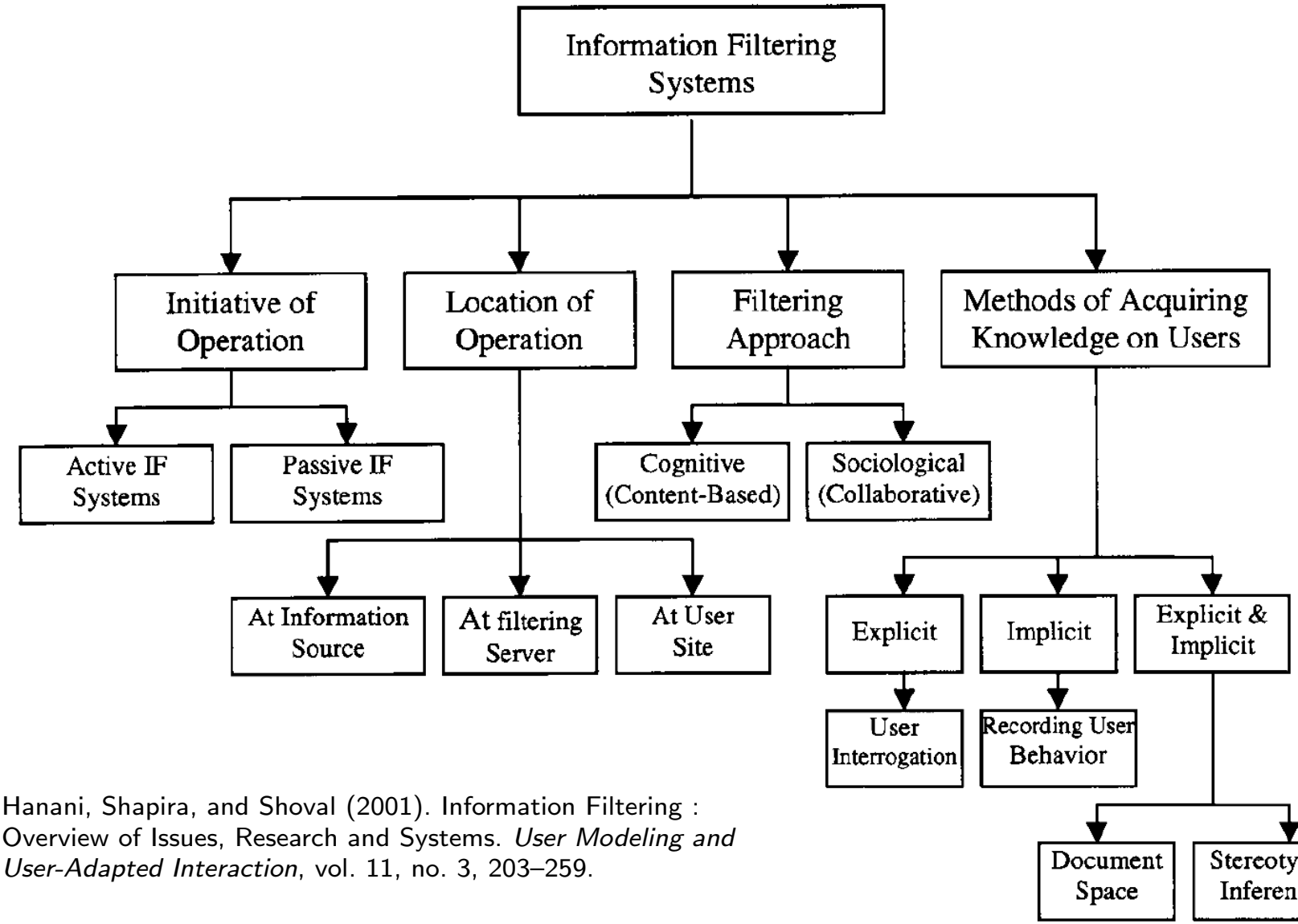
ÉCOLE POLYTECHNIQUE MONTRÉAL

Historique III

Quelques grands moments des systèmes de recommandations :

- **Amazon.com**
  - Le plus grand détaillant en ligne
  - Fait un usage intensif de différents types de recommandations
- **Prix Netflix** :
  - **1M$**
  - améliorer les recommandations de film de 10% faites par Cinematch (de RMSE 0.96 à 0.86)
  - lancé en octobre 2006 et gagné en septembre 2009
  - Deux montréalais dans l'équipe gagnante (c'est la fille d'un d'eux à Poly qui me l'a appris !)

Taxonomie de Hanani et coll. (2001)



Hanani, Shapira, and Shoval (2001). Information Filtering : Overview of Issues, Research and Systems. *User Modeling and User-Adapted Interaction*, vol. 11, no. 3, 203–259.

Taxonomie de Hanani et coll. (2001)

- *Initiative*
  Le filtre d'information peut être *passif* ou *actif*. Un système passif agit pour réduire un flot d'information, comme c'est le cas pour un filtre de polluriel. À l'inverse, un système actif créé un flot d'information en extrayant de l'information de différentes sources pour la livrer à l'utilisateur.
- *Localisation*
  Le filtre peut se trouver à trois endroits différents :
  (1) au poste de travail (client)
  (2) à la source d'information
  (3) sur un serveur de filtre

  Un filtre passif est généralement situé sur le poste client (1), tandis qu'un filtre actif sera plutôt situé à la source (2) ou sur un serveur de filtre (3)

Taxonomie de Hanani et coll. (2001)

- *Méthodes d'acquisition de l'information sur l'utilisateur*

Explicite : l'utilisateur spécifie ses préférences.
Implicite : les préférences sont déduites du comportement.
Combinaison : l'utilisateur peut spécifier des préférences mais un traitement est réalisé pour établir des informations implicites (ex. modèle d'espace de documents, regroupements en stéréotypes, etc.)

Les approches de filtrage I

L'approche collaborative :

- On se base sur des données explicites (ex. des votes) ou implicites (ex. des achats, des consultations de pages) pour et l'on cherche des similarités entre les utilisateurs ou les items.

Approche utilisateur-utilisateur :

- Classifier les gens selon leurs intérêts, leurs votes, leur comportements, ou toute autre dimension pertinente.
- Effectuer des recommandations basées sur le groupe le plus près, le plus représentatif d'un individu. On recommandera ainsi un élément très caractéristique du groupe qui n'est pas dans le profil de l'individu.

Approche item-item :

- Recherche des items qui ont des profils d'intérêt similaires.
- Recommandation en fonction de l'item affiché ou d'un historique d'items.

Les approches de filtrage II

L'approche "contenu" :

- Analyse du contenu ou des propriétés des items.
- Toujours le principe de recherche de similarité entre des items ou des utilisateurs.

Approches *memory-based* vs. *model-based*

Outre la taxonomie de Hanani et coll. présentée, Breese et coll. (1998) font aussi la distinction entre les types d'algorithmes :

- *Memory-based* ou basé mémoire qui effectue une recherche BD
  On utilise la base de données complète d'utilisateurs (ou d'items) pour effectuer les recommandations
- *Model-based* ou basé sur l'apprentissage
  On utilise la BD pour entraîner un modèle qui ensuite permet la prédiction de l'item pertinent

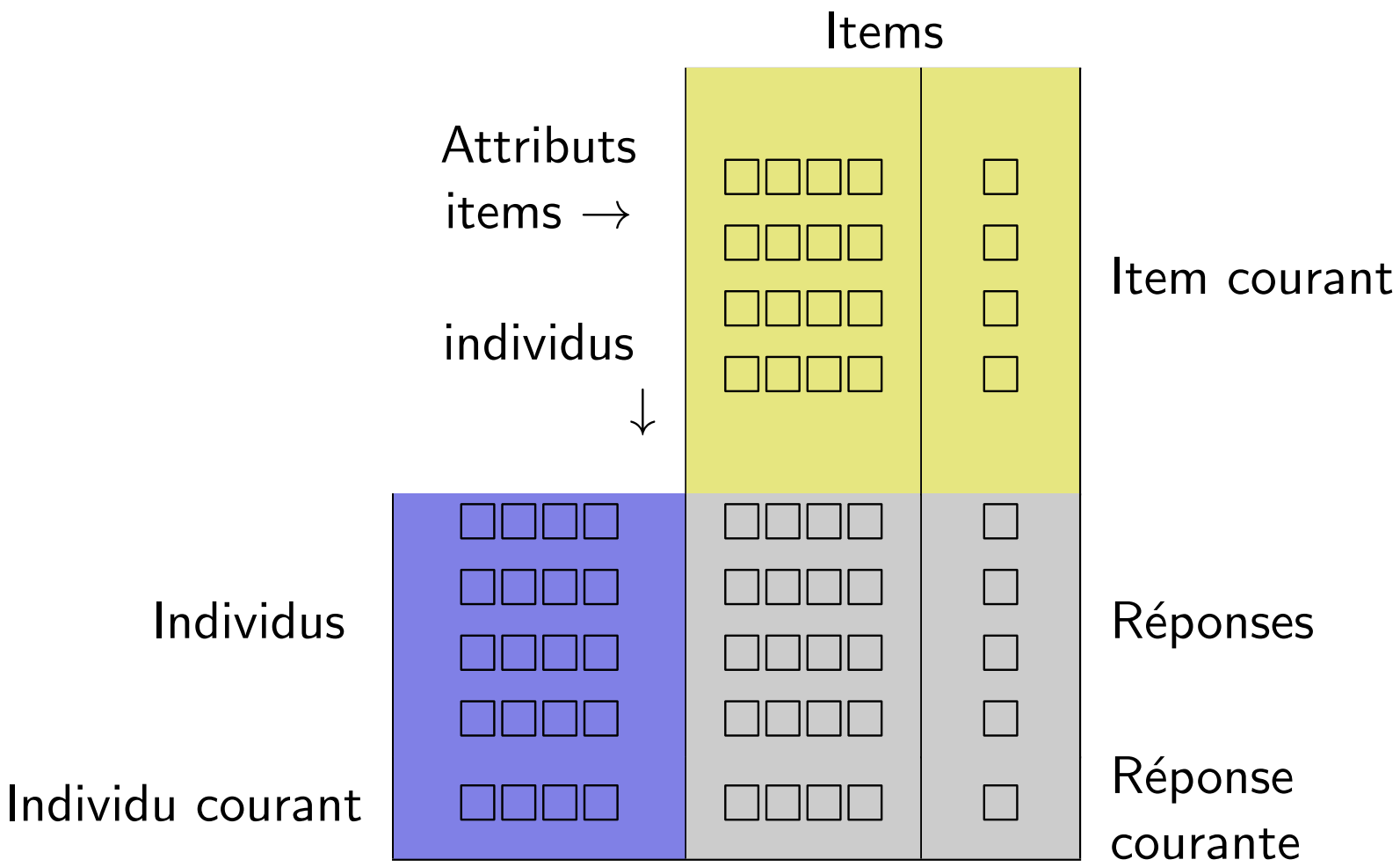
Quelques technologies classiques ou en opération

- MovieLens
- CD now (maintenant Amazon)
- Amazon
- last.fm
- youtube
- Mate1.com
- e-180.com

Algorithmes et composants

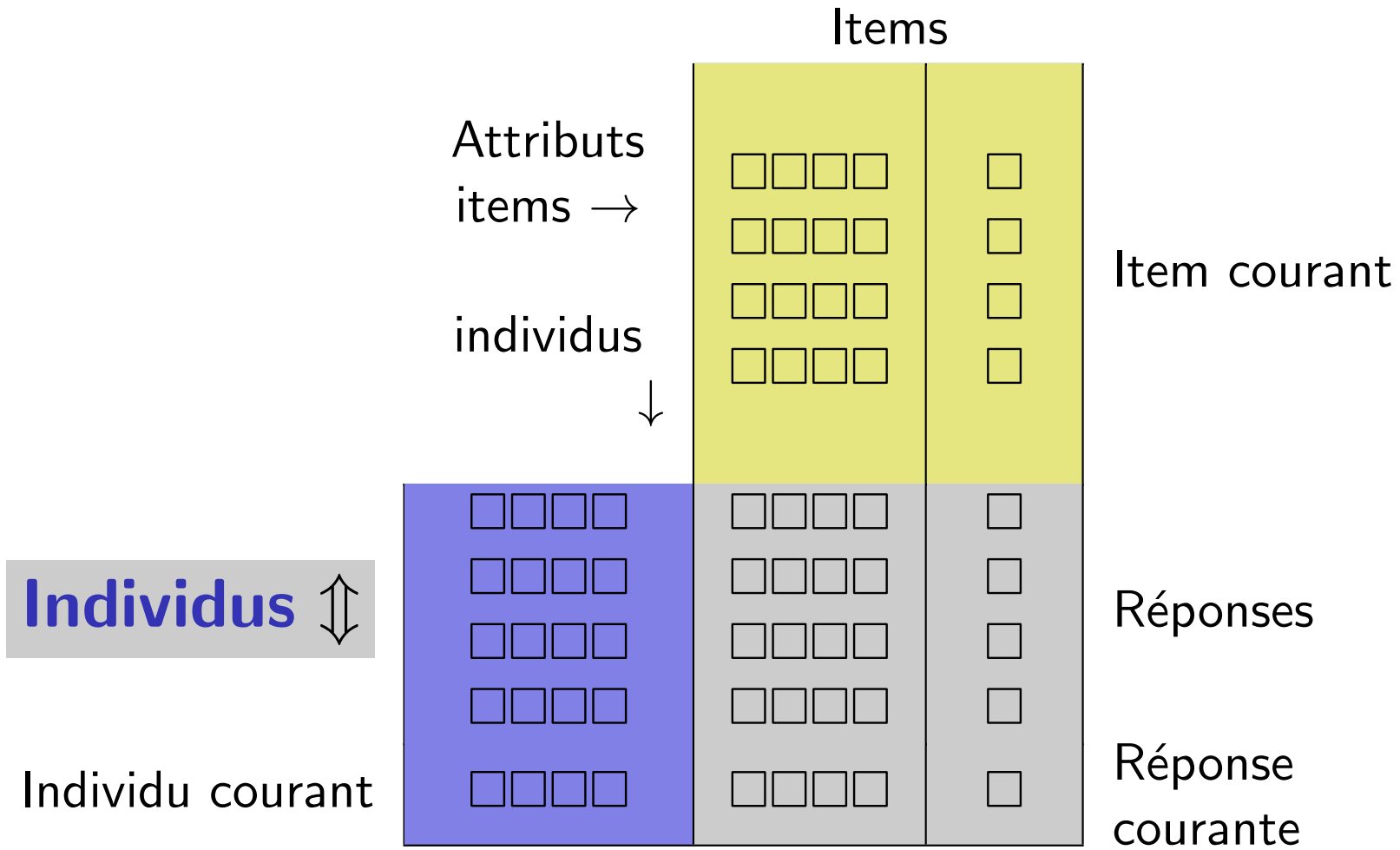
- *Espaces vectoriels* : on crée une matrice *utilisateurs-items* similaire à une matrice terme-document. De cette matrice, on calcule des distances (similarités) entre des utilisateurs (lignes—utilisateur-utilisateur) ou des items (colonnes—item-item). On trouve les voisins dans cet espace et on applique différents algorithmes pour prédire l'intérêt d'items. Les algorithmes qui reposent sur les espaces vectoriels sont nombreux et les plus répandus.
- *Méthodes bayésiennes* : approche de probabilité conditionnelle.
- *Systèmes à base de règles* : approche basée sur un travail de modélisation du domaine et le développement de règles.

Types d'information



Systèmes de recommandations, Approches filtres collaboratifs

Types d'information



Un exemple

Prenons les votes de 4 utilisateurs pour 4 items. On cherche à estimer le vote de l'utilisateur actif 1, U_1, à l'item 3, I_3.

Votes pour 4 utilisateurs

	Item			
U	**1**	**2**	**3**	**4**
U_1	5	1	?	2
U_2	4	1	1	3
U_3	4	2	1	2
U_4	1	4	3	2

Algorithmes *utilisateur-utilisateur*

Objectif : prédire les votes d'un utilisateur spécifique à partir d'une BD de votes d'autres utilisateurs.

La valeur estimée de l'utilisateur a pour un item j, $E(v_{a,j})$, est la somme pondérée des votes des autres utilisateurs, v_i, qui ont des votes communs :

$$E(v_{a,j}) = \overline{v}_a + \kappa \sum_{i}^{n} w_{a,i}(v_{i,j} - \overline{v}_i) \tag{1}$$

où n est le nombre usagers ayant des votes communs et $\overline{v}_i$ représente le vote moyen d'un utilisateur i et $\overline{v}_a$ le vote moyen de l'utilisateur a.

Le poids $w_{i,a}$ peut représenter une distance, une corrélation ou un coefficient de similarité quelconque entre un utilisateur i et l'utilisateur actif a. La constante κ normalise la somme des poids à 1 (donc, $\kappa = \frac{1}{\sum_i |w_{a,i}|}$).

La corrélation pour $w_{a,i}$

Une estimation du poids $w_{a,i}$ est celle de la corrélation de Pearson.

La corrélation de Pearson est une mesure statistique très commune qui a été originalement utilisée pour les filtres collaboratifs par le projet GroupLens (Resnick et al., 1994). La corrélation entre les utilisateurs a et i est :

$$w_{cor(a,i)} = \frac{\mathrm{cov}(a,i)}{\sigma_a \sigma_i} = \frac{\sum_j (v_{a,j} - \overline{v}_a)(v_{i,j} - \overline{v}_i)}{\sqrt{\sum_j (v_{a,j} - \overline{v}_a)^2 \sum_j (v_{i,j} - \overline{v}_i)^2}} \qquad (2)$$

On calcule la corrélation pour tous les utilisateurs qui ont un vote commun avec l'utilisateur actif a.

Un exemple

Prenons les votes de 4 utilisateurs pour 4 items. On cherche à estimer le vote de l'utilisateur actif 1, U_1, à l'item 3, I_3.

Votes pour 4 utilisateurs

	Item			
U	**1**	**2**	**3**	**4**
U_1	5	1	$?$	2
U_2	4	1	1	3
U_3	4	2	1	2
U_4	1	4	3	2

Un exemple, estimation avec la corrélation U_1 et U_i

Votes pour 4 utilisateurs

U	Item 1	2	3	4	$\overline{v}_i$	$w_{cor(1,i)}$
U_1	5	1	?	2	2,67	
U_2	4	1	1	3	2,67	0,89
U_3	4	2	1	2	2,67	0,97
U_4	1	4	3	2	2,33	-0,89
κ						1/2,75

On calcule le vote moyen : $\overline{v}_i = \frac{1}{|I_i|} \sum_{j \in I_i} v_{i,j}$ et la corrélation entre U_1 et U_i avec l'équation (2) (excluant l'item 3). La constante κ est l'inverse de la somme des valeurs absolues des poids, $\frac{1}{\sum_{i=2} |w_{cor(1,i)}|}$.

Puis, en appliquant la formule d'estimation du vote, équation (1), on obtient : $E(v_{1,3}) = 1,32$.

Détails des calculs

Votes pour 4 utilisateurs

U	Item 1	2	3	4	$\overline{v}_i$	$w_{cor(1,i)}$
U_1	5	1	?	2	2,67	
U_2	4	1	1	3	2,67	0,89
U_3	4	2	1	2	2,67	0,97
U_4	1	4	3	2	2,33	-0,89
κ						1/2,75

$$
\begin{aligned}
E(v_{1,3}) &= \overline{v}_1 + \kappa \sum_{i=2}^{n} w_{cor(1,i)}(v_{i,j} - \overline{v}_i) \\
&= 2,67 + \frac{0,89(4-2,67) + 0,97(1-2,67) + -0,89(3-2,33)}{|0,89| + |0,97| + |-0,89|} \\
&= \mathbf{1,32}
\end{aligned}
$$

ÉCOLE POLYTECHNIQUE MONTRÉAL

Le cosinus pour $w(1, i)$

Nous pouvons aussi utiliser le cosinus comme mesure du poids $w_{1,i}$:

$$
\begin{aligned}
w_{cos(1,i)} &= \frac{\sum_j v_{1,j} v_{i,j}}{\sqrt{\sum_{k \in I_1} v_{1,k}^2} \sqrt{\sum_{k \in I_i} v_{i,k}^2}} \\
&= \frac{\mathbf{v}_1 \mathbf{v}_i}{||\mathbf{v}_1|| \; ||\mathbf{v}_i||}
\end{aligned}
$$

Chaque utilisateur représente ainsi un vecteur dans un espace de vote et on cherche celui qui a le vecteur le plus près (parallèle).

Exemple, estimation avec le cosinus U_1 et U_i

Votes pour 4 utilisateurs

U	Item 1	2	3	4	$\overline{v}_i$	$w_{cos(1,i)}$
U_1	5	1	?	2	2,67	
U_2	4	1	1	3	2,00	0,97
U_3	4	2	1	2	2,25	0,97
U_4	1	4	3	2	2,50	0,52
κ						1/2,45

En prenant $w_{cos(1,i)} = \cos(1, i)$ on obtient le tableau ci-dessus.
En appliquant la formule d'estimation du vote, équation (1), on obtient : $E(v)_{13} = 1,49$.

ÉCOLE POLYTECHNIQUE MONTRÉAL

Estimation du vote par la méthode des voisins rapprochés (*k-nearest neighbour*)

Principe : on utilise le vote moyen, ou pondéré, des n utilisateurs les plus rapprochés dans l'espace vectoriel des votes. La distance euclédienne est généralement utilisée pour cette fin :

$$d(a,i) = \sqrt{\sum_j (v_{a,j} - v_{i,j})^2} \tag{3}$$

Pour chaque item j, on calcule la racine carrée la somme des carrés des différences entre les votes des utilisateurs a et i.

Les utilisateurs les plus proches voisins sont ceux ayant les premières n valeurs. On détermine alors la valeur du vote de l'utilisateur a par la moyenne, potentiellement pondérée, de ces utilisateurs.

Exemple, estimation avec le cosinus et voisins=2

Votes pour 4 utilisateurs

U	Item 1	2	3	4	$\overline{v}_i$	d(1,i)	$w_{cos(1,i)}$
U_1	5	1	**?**	2	2,67	0,00	
U_2	*4*	*1*	*1*	*3*	*2,00*	*1,41*	*0,97*
U_3	*4*	*2*	*1*	*2*	*2,25*	*1,41*	*0,97*
U_4	1	4	3	2	2,50	5,00	0,52
κ							*1/1,94*

Les deux utilisateurs les plus proches de U_1 sont U_2 et U_3. Leur distance avec U_1 est $\sqrt{2}$ dans les deux cas.
En ne conservant que ces deux plus proches voisins, la réponse est : $E(v)_{1,3} = 1,0$.

ÉCOLE POLYTECHNIQUE MONTRÉAL

Correction pour le nombre de votes communs

La valeur d'une corrélation ou d'un cosinus basée sur un plus grand nombre de votes commun devrait avoir un poids plus important qu'une valeur basée sur un plus petit nombre. Il est donc fréquent de faire la correction suivante au poids $w_{u,v}$:

$$w'_{u,v} = \frac{max(v_{u,v}, \gamma)}{\gamma} \cdot w_{u,v}$$

où $v_{u,v}$ est le nombre de votes communs entre les utilisateurs u et v, et où γ est une constante représentant le nombre minimum de votes pour effectuer cette correction, par exemple, Herloker et coll. (1999) utilisent $\gamma = 5$.

Systèmes de recommandations, Approches filtres collaboratifs

Principe général de l'approche item-item

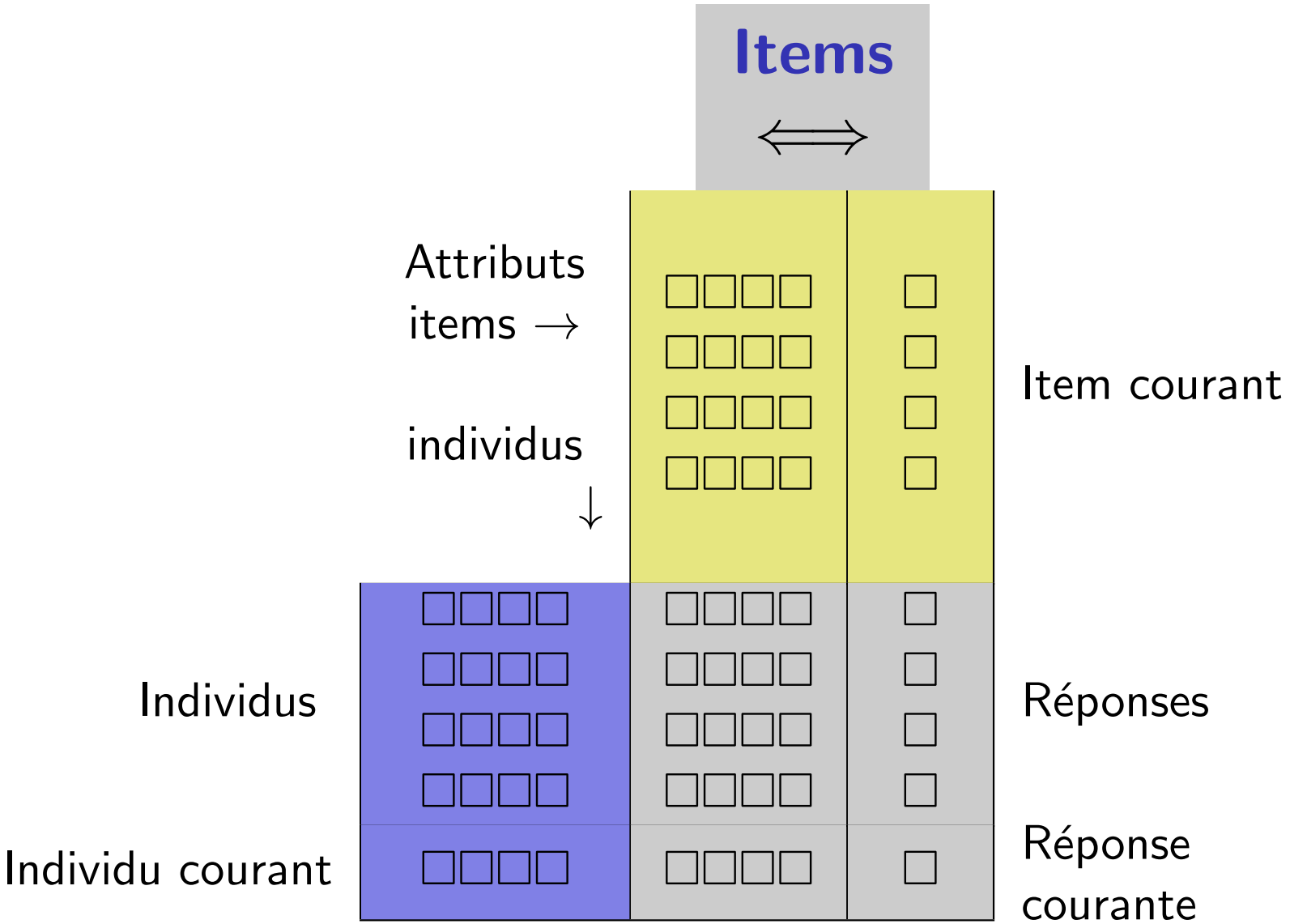
Les algorithmes précédents cherchent des similarités entre utilisateurs pour ensuite proposer les items les plus populaires des utilisateurs voisins, notamment par une somme pondérée similarité-utilisateur par item pour suggérer les items (équation (1)).

L'approche item-item cherche plutôt des similarités entre les items. Dès qu'un utilisateur s'intéresse à un item, on lui suggère des items similaires.

Ici encore, les similarités peuvent être estimés par le cosinus ou la corrélation entre des items, sauf qu'on transpose la matrice pour faire le calcul de la similarité item-item plutôt qu'utilisateur-utilisateur.

Finalement, on procède de façon analogue à l'approche utilisateur-utilisateur en utilisant une somme pondérée.

Types d'information



Transposition de la matrice

Votes pour 4 utilisateurs

	Item			
U_i	I_1	I_2	I_3	I_4
U_1	5	1	?	2
U_2	4	1	1	3
U_3	4	2	1	2
U_4	1	4	3	2

transposition

$\Longrightarrow$

Matrice inversée des votes pour 4 utilisateurs

	Utilisateur			
I_i	U_1	U_2	U_3	U_4
I_1	5	4	4	1
I_2	1	1	2	4
I_3	?	1	1	3
I_4	2	3	2	2

Algorithme item-item, exemple pour l'item 3

Matrice transposée des votes pour 4 utilisateurs

I_i	Utilisateur **1**	**2**	**3**	**4**	$\overline{v}_i$	$d(I_3, I_i)$	$w_{cor(I_3,I_i)}$
I_1	5	4	4	1	3,00	4,69	-1,00
I_2	*1*	*1*	*2*	*4*	*2,33*	*1,41*	*0,94*
I_3	*?*	1	1	3	1,67	0,00	1,00
I_4	*2*	*3*	*2*	*2*	*2,33*	*2,45*	*-0,50*
κ							*1/1,44*

En ne gardant que $I2$ et $I4$ comme voisins rapprochés, on prédirait la valeur de $I_{3,1}$ basée sur le principe de l'équation (1) comme suit :

$$I_{3,1} = 1,67 + \frac{0,94(1-2,33) + -0,50(2-2,33)}{(|0,94| + |-0,50|)} = 0,92$$

ÉCOLE POLYTECHNIQUE MONTRÉAL

Algorithme item-item, exemple pour l'item 3

Matrice transposée des votes pour 4 utilisateurs

I_i	Utilisateur **1**	**2**	**3**	**4**	$\overline{v}_i$	$d(I_3, I_i)$	$w_{cor(I_3,I_i)}$
I_1	5	4	4	1	3,00	4,69	-1,00
I_2	*1*	*1*	*2*	*4*	*2,33*	*1,41*	*0,94*
I_3	*?*	1	1	3	1,67	0,00	1,00
I_4	*2*	*3*	*2*	*2*	*2,33*	*2,45*	*-0,50*
κ							*1/1,44*

Mais devrait-on vraiment prendre la corrétation de -0,5 ?

Exemple avec le cosinus

L'exemple précédent peut aussi être calculé en prenant le cosinus comme mesure de similarité. La colonne $w_{cor(I_3,I_i)}$ du tableau précédent est alors remplacée par $w_{cos(I_3,I_i)}$:

$$w_{cos(I_3,I_i)} = (0,58\ 0,99\ 1,00\ 0,80)^T$$

et le résultat donne : $I_{3,1} = 0,78$

Cependant, Sarwar et coll. (2001, p. 288) cautionnent que la mesure du cosinus ne tient pas compte des différences individiduelles entre les utilisateurs lorsqu'ils indiquent leurs préférences. Certains utilisateurs ont tendance à être très critiques, d'autres très généreux dans leurs votes. Ils suggèrent donc d'utiliser une formule modifiée du cosinus qui normalise pour ce facteur :

$$w_{ncos(i,j)} = \frac{\sum_{u \in U}(R_{u,i} - \overline{R}_u)(R_{u,j} - \overline{R}_u)}{\sqrt{\sum_{u \in U}(R_{u,i} - \overline{R}_u)^2}\sqrt{\sum_{u \in U}(R_{u,j} - \overline{R}_u)^2}}$$

Exemple avec le cosinus normalisé

Matrice transposée des votes et *normalisée par utilisateur* pour obtenir $w_{ncos(1,i)}$

	Utilisateur				
I_i	**1**	**2**	**3**	**4**	$w_{ncos(I_3,I_i)}$
I_1	2,33	1,75	1,75	-1,5	-0.96
I_2	*-1,67*	*-1,25*	*-0,25*	*1,5*	*0,73*
I_3	*?*	-1,25	-1,25	0,50	1,00
I_4	*-0,67*	*0,75*	*-0,25*	*-0,5*	*-0,51*
$\overline{R}_u$	*2,67*	*2,25*	*2,25*	*2.50*	
κ					*1/1,24*

En remplaçant $w_{cos(I_3,I_i)}$ par $w_{ncos(I_3,I_i)}$ dans la matrice non normalisée, la valeur de $I_{3,1}$ devient alors : $I_{3,1} = 1,62$.

ÉCOLE POLYTECHNIQUE MONTRÉAL

Au-delà du calcul de la valeur du vote

En supposant un utilisateur qui s'intéresse à un item particulier, comment déterminer d'autres items qui pourraient l'intéresser ?

Supposons un item d'intérêt i pour l'utilisateur u, un approche possible serait :

- Trouver les N items les plus similaires à i
- De ces N items, choisir les R items dont on calcule le vote le plus élevé pour u

Fréquence inverse utilisateur

Extensions aux modèles *basé mémoire*

- À l'instar de la transformation TF-IDF pour la recherche d'information, on peut conclure que la similarité de votes pour un item comportant un grand nombre de votes n'a pas le même poids qu'un vote similaire pour un item comportant très peu de votes.
- La transformation TF-IDF dans le contexte des filtres collaboratifs transforme le poids original pour l'item j, w_j, en nouveau poids pondéré, w'_j :

$$w'_j = w_j \ \log(\frac{n}{n_j})$$

où n est le nombre total d'individus et n_j est le nombre d'individus qui ont exprimé un vote pour l'item j.

Systèmes de recommandations, Approches filtres collaboratifs

Mesures de performance I

- **Prédiction de la valeur individuelle des votes**
  - Mesure de l'écart entre la valeur prédite et la valeur réelle.
  - Peut être la moyenne de la valeur absolue de l'écart, la racine carrée de la somme des différences au carré, ou autre mesure de distance.
  - La **moyenne des erreurs au carré / erreur quadratique moyenne** (*RMSE*) :

$$\sqrt{\frac{\sum_i^N (\hat{x}_i - x_i)^2}{N}}$$

où $\hat{x}_i$ est la valeur prédite et x_i est la valeur réelle.

  - Mesure alternative **Erreur absolue moyenne** :

$$\frac{\sum_i^N |(\hat{x}_i - x_i)|}{N}$$

Mesures de performance II

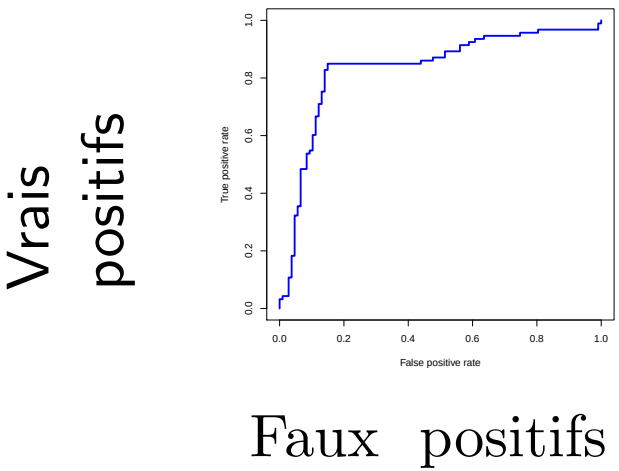
- **Prédiction d'une liste de recommandations**
  Dans le cas où la prédiction est une liste de recommandation, l'approche de la recherche d'information peut être utilisée avec les mesures correspondantes :
  - **Rappel**
    Proportion des documents pertinents à recommander qui sont correctement identifiés.
  - **Précision**
    Proportion de documents identifiés qui sont pertinents à recommander.
- Breese et coll. utilisent une notion *d'utilité* qui correspond à l'écart positif entre le rang d'une recommandation neutre et la recommandation prédite et rapporte le ratio en pourcentage entre l'utilité prédite et l'utilité maximale.

Le ROC et AUC

Mesures très utiles car ne nécessite pas de fixer de seuils pour la classification des résultats.

- **ROC, Receiver Operator Characteristic**

  Évolution du taux de vrais positifs contre celui des faux positifs



- **AUC, Area Under the Curve**

  Surface sous la courbe ROC.

Simulations

- Validation croisée :
  - séparation de données réelles en deux blocs :
    1. entrainement
    2. validation
  - prédiction du comportement réel avec les données d'entraînement
  - répétition en échantillonnant aléatoirement les blocs (ex. 10 répétitions)
- Validation individuelle (*leave one out*)
  - On élimine du bloc de validation et d'entrainement les données de l'individu *cible*
  - On rapporte le score moyen.
- Aux fins de comparaison, on utilise les mêmes blocs d'entrainement et de validation pour chaque algorithme afin de réduire les écarts dus à l'échantillonnage.