

5. Apprentissage pour le filtrage collaboratif

Il semble que le nombre de choix qui nous sont ouverts augmente constamment. Films, livres, recettes, nouvelles du monde, autant d'ensembles dans lesquels nous devons opérer une sélection sans avoir la possibilité de considérer toutes les informations nécessaires. Comment alors choisir ? Sur une base individuelle, seule l'expérience accumulée sur la valeur de choix passés pourrait nous aider à guider nos choix futurs. Un processus d'amélioration au mieux linéaire, en supposant que les critères et ensemble de choix restent constants. Heureusement, nous ne sommes pas seuls face aux mêmes choix. Ainsi, si quelqu'un a des goûts similaires aux nôtres et qu'il a aimé tel film récent, les chances que nous aimions également ce film semblent plus grandes qu'en l'absence de cette information. Il est donc possible de tirer profit des informations disponibles sur les choix des autres agents pour induire des préférences sur nos propres choix. Avec maintenant la disponibilité d'Internet et de grandes bases de données sur les préférences des utilisateurs, il devient envisageable d'étendre à grande échelle, la notion de bouche à oreille. La formalisation et l'exploitation de cette intuition sont l'objet des travaux en filtrage collaboratif.

5.1 Les approches

Comme dans l'induction supervisée, on retrouve les approches utilisant un espace d'hypothèses et les approches opérant directement dans l'espace des utilisateurs ou des items.

Le but du filtrage collaboratif est de suggérer de nouveaux items ou de prédire l'utilité d'items inconnus pour un utilisateur donné, en se fondant sur les évaluations déjà exprimées (éventuellement implicitement) par cet utilisateur à propos d'autres items. On supposera ainsi qu'il existe une liste de m utilisateurs $\mathcal{U} = \{u_1, u_2, \dots, u_m\}$ et une liste de n items $\mathcal{I} = \{i_1, i_2, \dots, i_n\}$. Chaque utilisateur u_l a exprimé une évaluation sur un sous-ensemble \mathcal{I}_{u_l} d'items. On notera X_l^j l'évaluation de l'item i_j par l'utilisateur u_l , et $Pred_{l,j}$ la prédiction d'utilité pour l'utilisateur u_l de l'item i_j .

5.1.1 Les approches par modèle de l'utilisateur

L'approche classique pour les systèmes de recommandation consiste à construire des modèles des utilisateurs en fonction d'informations les concernant, comme leur intervalle de revenus, leur niveau culturel, leur âge, leurs habitudes culturelles, etc., et à utiliser ces modèles pour prédire la réponse d'un utilisateur ainsi caractérisé face à un objet ou item (livre, restaurant, ...).

Une technique courante est ainsi de construire un modèle régressif linéaire afin de prédire l'appréciation d'un utilisateur u_l pour un item i_j , par exemple :

$$pred(u_l, i_j) = \alpha_1(\text{âge}(u_l)) + \alpha_2(\text{revenu}(u_l)) + \alpha_3(\text{intervalle_prix}(i_j)) + \dots$$

Il est également possible d'apprendre la fonction $pred$ en utilisant des SVM (voir chapitre 14) ou des méthodes de classification bayésienne.

Cependant cette approche comporte plusieurs difficultés. Outre le problème du choix du modèle (par exemple comment prendre en compte le paramètre date dans l'année dans un modèle linéaire ?), et celui du passage à l'échelle lorsque le nombre d'utilisateurs et d'items est énorme, il peut être très **difficile d'incorporer dans le modèle des variables cruciales pour l'appréciation**. Par exemple, il se peut qu'une recette de cuisine soit préférée à une autre, décrivant pourtant la confection du même plat, parce qu'elle est plus lisible, plus facile à comprendre. Mais comment un système va-t-il pouvoir mesurer cette facilité ? Faudra-t-il prévoir de décrire tous ces facteurs critiques souvent implicites : lisibilité, atmosphère, effet de mode, ... Cela devient

vite inenvisageable. C'est pourquoi on va plutôt utiliser des *approches avec variables latentes*, non explicitées, mais permettant d'organiser les utilisateurs ou les items.

5.1.2 Les approches par similarité et plus proches voisins

On peut distinguer les *approches centrées utilisateurs* et les *approches centrées items*.

Dans l'**approche centrée utilisateurs**, un utilisateur est caractérisé par les appréciations qu'il a déjà émises à propos de certains items. Par exemple, la note qu'il a attribuée aux restaurants dans lesquels il a déjà dîné. Lorsque cet utilisateur va chercher l'appréciation probable qu'il porterait sur un nouveau restaurant, le système de recommandation va identifier les autres utilisateurs dont le profil est le plus proche de l'utilisateur en fonction des notes exprimées, et va ensuite à partir de ces « plus proches voisins », utiliser leurs notes (s'il y en a) sur le restaurant visé pour calculer une recommandation.

La correspondance entre les utilisateurs et les items se représente généralement sous la forme d'une table.

	Item 1	Item 2	Item 3	Item 4	...
Jean	-	2	7	8	...
Marie	4	1	-	7	...
Christian	3	8	-	4	...

On suppose que les appréciations vont de 0 à 10 (*très mauvais* à *excellent*) et que le tiret « - » représente l'absence d'évaluation. En effet, la plupart des utilisateurs ne vont noter qu'un très petit nombre d'items (par exemple Amazon vend des millions de livres, un lecteur même assidu ne peut en noter au plus que quelques milliers). Dans de nombreux cas, la matrice sera donc creuse, voire très creuse.

Supposons que nous voulions maintenant utiliser cette matrice pour calculer une similarité entre utilisateurs puis pour induire $Pred(u_l, i_j)$ pour un utilisateur u_l et un item i_j donnés. Il faut alors étudier les questions suivantes :

- Quelle mesure de similarité utiliser ?
- Combien de « voisins » prendre pour référence ?
- Quelle méthode de combinaison d'avis employer pour inférer une nouvelle évaluation ?

Mesure de similarité entre utilisateurs

Chaque utilisateur peut être considéré comme un vecteur incomplet dont nous ne connaissons que quelques composantes. Il est cependant possible de calculer une similarité entre de tels vecteurs en se restreignant aux seules composantes qu'ils ont en commun.

Si l'on suppose que les notes attribuées par les utilisateurs U_i et U_j aux items sont des variables aléatoires X_i et X_j qui suivent une distribution conjointe inconnue, il est possible de définir le coefficient de corrélation entre X_i et X_j par la *formule de Bravais-Pearson* :

$$\rho = \frac{Cov(X_i, X_j)}{\sqrt{Var(X_i)Var(X_j)}}$$

avec $Cov(X_i, X_j) = \mathbb{E}[(X_i - \mathbb{E}(X_i))(X_j - \mathbb{E}(X_j))]$.

Cette corrélation prend ses valeurs dans $[-1, +1]$, une valeur positive indiquant que les variables varient dans le même sens, tandis qu'une valeur négative signifie que les individus qui ont des

scores élevés pour la première variable auront tendance à avoir des scores faibles pour la deuxième et inversement.

En disposant d'un échantillon de taille n , $(X_i^1, X_j^1), (X_i^2, X_j^2), \dots, (X_i^n, X_j^n)$ tiré d'une distribution conjointe, la quantité :

$$r = \frac{\sum_k (X_i^k - \bar{X}_i)(X_j^k - \bar{X}_j)}{\sqrt{\sum_k (X_i^k - \bar{X}_i)^2} \sqrt{\sum_k (X_j^k - \bar{X}_j)^2}}$$

est une estimation de ρ .

— EXEMPLE **Calcul de la similarité** —

Selon cette mesure, la similarité entre *Jean* et *Marie* est :

$$\begin{aligned} r(\text{Jean}, \text{Marie}) &= \frac{(2-5)(1-4) + (8-5)(7-4)}{\sqrt{(2-5)^2 + (8-5)^2} \sqrt{(1-4)^2 + (7-4)^2}} \\ &= \frac{(-3)(-3) + (3)(3)}{\sqrt{9+9} \sqrt{9+9}} = \frac{18}{18} = 1 \end{aligned}$$

avec $X_{\text{Jean}} = (2+8)/2 = 5$ et $X_{\text{Marie}} = (1+7)/2 = 4$ quand on prend les articles notés en commun.

Entre *Jean* et *Christian*, elle est : $r(\text{Jean}, \text{Christian}) = \frac{-12}{\sqrt{18} \cdot \sqrt{8}} = -12/(3 \cdot 2 \cdot 2) = -1$

Et entre *Marie* et *Christian* : $r(\text{Marie}, \text{Christian}) = \frac{-12}{\sqrt{18} \cdot \sqrt{14}} = -4/(\sqrt{2} \cdot \sqrt{14}) \approx -0.756$

Il apparaît ainsi que *Jean* et *Marie* sont positivement corrélés, tandis que *Jean* et *Christian*, de même que *Marie* et *Christian*, sont des paires corrélées négativement. Il faut noter que s'il peut sembler étrange que *Jean* et *Christian* soient parfaitement négativement corrélés, il faut prendre en compte leur moyenne sur les composantes 2 et 4 : à savoir 5 pour *Jean* et 6 pour *Christian*. Par rapport à ces moyennes, les notes de *Jean* pour les *items* 2 et 4 sont -3 et +3, tandis que pour *Christian* elles sont 2 et -2. Il y a bien tendance exactement inverse.

De fait, il a été empiriquement observé que la corrélation de Pearson n'est pas la mesure de corrélation la mieux adaptée pour le filtrage collaboratif, un **meilleur choix étant de prendre la corrélation de Pearson à la puissance 2.5** [BHK98].

Calcul d'une recommandation

La plupart des algorithmes de recommandations utilisent une combinaison des évaluations d'utilisateurs *proches* au sens de la mesure de similarité employée. Par exemple, l'approche la plus simple consiste à retenir les k plus proches voisins au sens de cette similarité r et de calculer une moyenne pondérée de leurs évaluations pour fournir une prédiction d'utilité.

$$\text{Pred}(u_l, i_j) = \bar{X}_l + \frac{\sum_{v=1}^k r(u_l, u_v) (X_v^j - \bar{X}_v)}{\sum_{v=1}^k |r(u_l, u_v)|} \quad (20.11)$$

où \bar{X}_l (resp. \bar{X}_v) est la moyenne des notes attribuées par l'utilisateur u_l (resp. u_v) à tous les items.

 EXEMPLE Calcul d'une recommandation

Supposons que nous voulions prédire la note que donnerait *Jean* à l'*item 1* et que nous prenions *Marie* et *Christian* comme voisins.

$$Pred(\text{Jean}, \text{item 1}) = \frac{17}{3} + \frac{(1 \cdot (4 - 4)) + (-1 \cdot (3 - 5))}{1 + 1} = 5.67 + 1 \approx 6.67$$

Intuitivement, puisque *Christian* n'a pas aimé l'*item 1*, et qu'il est négativement corrélé avec *Jean*, alors cela devrait amener à penser que *Jean* va plutôt aimer *item 1*, d'où une valeur accordée à *item 1* supérieure à la moyenne de *Jean*. L'évaluation de *Marie*, quant à elle, ne modifie rien car elle évalue l'*item 1* à sa moyenne : 4.

L'**approche centrée items** est duale de l'approche centrée utilisateurs. Elle consiste en effet à utiliser une *mesure de similarité entre items* pour déterminer les k items les plus similaires à l'item i_j pour lequel on cherche à calculer $Pred(u_l, i_j)$. On utilise alors une formule de pondération qui, dans le cas le plus simple, est :

$$Pred(u_l, i_j) = \bar{X}^i + \frac{\sum_{v=1}^k r(i_j, i_v) (X_l^v - \bar{X}^v)}{\sum_{v=1}^k |r(i_j, i_v)|} \quad (20.12)$$

 EXEMPLE

Supposons que nous voulions prédire la note que donnerait *Jean* à l'*item 1* et que nous prenions *Item 2* et *Item 4* comme voisins (*Item 3* ne peut pas être voisin de *Item 1* car il ne partage aucune composante utilisateur).

En utilisant la mesure de corrélation de Pearson, on trouve que $r(\text{Item 1}, \text{Item 2}) = -1$ et $r(\text{Item 1}, \text{Item 4}) = 1$. On a alors :

$$Pred(\text{Jean}, \text{Item 1}) = \frac{4 + 3}{2} + \frac{(-1 \cdot (2 - 11/3)) + (1 \cdot (8 - 19/3))}{1 + 1} \approx 3.5 + 1.67 = 5.17$$

On observe que, en utilisant ces formules de similarité et de combinaison, le résultat n'est pas le même que dans l'approche centrée utilisateur.

5.2 Les difficultés

Les difficultés liées au filtrage collaboratif proviennent essentiellement :

- de la *dimension de la matrice* utilisateurs-items qui peut éventuellement être énorme (typiquement de l'ordre de 10^4 lignes \times 10^5 colonnes) ;
- du *caractère creux de cette matrice* (typiquement avec moins de 1 % d'entrées) ;
- du choix à effectuer de la *mesure de similarité*, en particulier si les données comportent des relations.

Par ailleurs, il n'est pas facile de *mesurer la performance* d'un système de filtrage-collaboratif, ce qui est nécessaire pour identifier la technique la mieux adaptée à une tâche particulière.

5.2.1 Traitement des matrices

Afin de traiter de très grandes matrices, dans lesquelles la liste des items est relativement peu variable comparée à la liste des utilisateurs, plusieurs techniques sont employées.

L'une d'elles consiste à précalculer les voisins dans l'espace des items (peu variables) afin de limiter les calculs en ligne de $pred(u_l, i_j)$. Une autre consiste à effectuer une classification non

supervisée préalable des items (ou des utilisateurs) afin de limiter effectivement la dimensionalité du problème. Cette technique entraîne cependant souvent une dégradation des performances, mais elle n'a sans doute pas dit son dernier mot. Finalement, des techniques de réduction de dimensionalité classiques sont également utilisées en prétraitement (voir chapitres 3 et 18).

5.2.2 Mesures de similarité

Le choix de la mesure de similarité est naturellement crucial dans l'approche par voisinage. Elle doit d'abord s'appuyer sur une sélection avisée des descripteurs utilisés. Généralement, cet ensemble de descripteurs découle directement de la base de données disponible (par exemple, pour une base de films : *auteur, titre, acteurs, producteur* ...). Il se peut cependant que l'ensemble d'attributs pertinents ne soit pas facile à déterminer. Par exemple, dans le cas d'items décrits par des textes (articles, recettes ...), les mots ne doivent pas prendre tous le même poids. Il est alors fréquent de les pondérer grâce au score tf-idf (*Term Frequency* \times *Inverse Document Frequency*) .

- Étant donnée la fréquence f_{ij} du terme t_i dans le document d_j , $TF_{ij} = \frac{f_{ij}}{\max_k f_{kj}}$.
- De même, étant donné n_i le nombre de documents mentionnant le terme i et N le nombre total de documents, $IDF_i = \log \frac{N}{n_i}$.
- Le score tf-idf pour le terme t_i et le document d_j est $w_{ij} = TF_{ij} \times IDF_i$.
- Le profil d'un document (item) est alors caractérisé par l'ensemble des termes de score tf-idf les plus élevés avec leur score.

Les mesures de similarité les plus populaires [HKR02] sont le coefficient de corrélation de Pearson exposé plus haut (ou sa version simplifiée cosinus) et le **coefficient de corrélation de rang de Spearman**. Celui-ci compare deux variables X_i et X_j après transformation de leurs réalisations $(X_i^1, X_i^2, \dots, X_i^n)$ et $(X_j^1, X_j^2, \dots, X_j^n)$ en liste de rangs $(rg_i^1, rg_i^2, \dots, rg_i^n)$ et $(rg_j^1, rg_j^2, \dots, rg_j^n)$ dans lesquelles rg_l^m est le rang de l'élément X_l^m dans la liste X_l ordonnée selon une certaine relation d'ordre (par exemple l'ordre de préférence de films).

Le coefficient de corrélation est basé sur la différence des rangs obtenus par les réalisations sur les deux variables selon la formule :

$$r_s = 1 - \frac{6 \sum_{m=1}^n D^2}{n(n^2 - 1)} \quad (20.13)$$

où D représente la différence de rang sur les deux variables pour une observation donnée.

— EXEMPLE **Coefficient de corrélation de rang de Spearman** —

$$\begin{aligned} r_s((1.8, 3.4, 2.5, 4.1), (6.0, 1.2, 2.2, 3.7)) &= r_s((1, 3, 2, 4), (4, 1, 2, 3)) \\ &= 1 - \frac{6(3^2 + 2^2 + 0 + 1^2)}{4(4^2 - 1)} = -0.4 \end{aligned}$$

Les deux variables sont donc négativement corrélées.

L'avantage du coefficient de Spearman est qu'il est non paramétrique, c'est-à-dire qu'il ne fait aucune présupposition sur la distribution de fréquence des variables. Il ne suppose pas non plus, à l'inverse du coefficient de Pearson, que la relation entre les variables est linéaire.

Il est possible que les données disponibles impliquent l'existence de relations au-delà du graphe bipartites des utilisateurs et des items. Ainsi, dans la figure 20.7, on suppose que les items, outre qu'ils sont liés avec les utilisateurs, appartiennent à des catégories (C_1 ou C_2).

Dans ce cas, des mesures de similarité plus sophistiquées permettent de tenir compte de la structure de graphe entre les éléments du domaine (voir section 4 du chapitre 18).

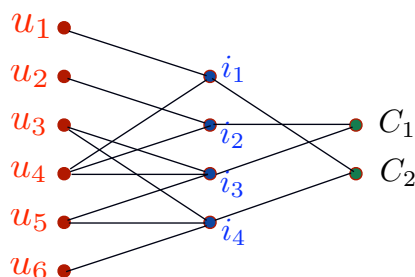


FIG. 20.7: Graphe montrant les relations entre utilisateurs, items et catégories dans une hypothétique base de données. On voudrait pouvoir exprimer le fait que l'évaluation d'un utilisateur à propos d'un item devrait propager une information sur son évaluation potentielle d'autres items de la même catégorie. Pour cela il faut une mesure de similarité prenant en compte la structure de graphe.

5.2.3 Évaluation des algorithmes de filtrage collaboratif

L'évaluation des algorithmes de prédiction en filtrage collaboratif procède généralement par validation croisée. On découpe les données disponibles en sous-ensembles et on utilise tous ces sous-ensembles sauf un pour l'apprentissage, et le dernier pour la mesure de performance, et on répète ce procédé plusieurs fois en changeant le sous-ensemble utilisé pour le test (voir chapitre 3).

En pratique, la démarche peut être la suivante :

1. On choisit, aléatoirement, un certain nombre d'utilisateurs, par exemple, la moitié de ceux-ci. Ces utilisateurs sont employés par l'algorithme d'apprentissage. Les autres utilisateurs constituent un ensemble test.
2. Les utilisateurs de l'ensemble test sont alors considérés un à un. Pour chaque utilisateur u , on ne fournit à l'algorithme qu'une partie des évaluations, par exemple, toutes les évaluations sauf une. On note l'utilisateur ainsi amputé de son évaluation sur l'item i par u'_i .
3. On mesure alors l'erreur commise par l'algorithme lorsqu'il tente de prédire la note accordée par l'utilisateur à l'item i : $pred(u', i)$, par rapport à la vraie évaluation x_u^i : $|pred(u', i) - x_u^i|$.
4. En calculant la moyenne sur tous les utilisateurs de l'ensemble test et tous les articles qu'ils ont notés, on obtient la mesure *All-But 1 Mean Average Error* (MAE).

La mesure MAE est la plus employée, mais [HKTR04] propose une revue des méthodes d'évaluation existantes.

Il est à noter qu'il n'est pas facile de comparer plusieurs algorithmes de recommandation entre eux, en particulier lorsque l'évaluation a lieu en ligne, c'est-à-dire alors même que le système est employé par les utilisateurs. Il existe en effet des boucles de rétroaction entre les recommandations produites par le système et les préférences exprimées par les utilisateurs. Par exemple, un item populaire sera souvent recommandé, donc souvent évalué par les utilisateurs, et son poids dans les mesures de similarité et le calcul des recommandations ultérieures sera de ce fait augmenté, ce qui biaisera le système.

Finalement, la précision des prédictions n'est pas le seul critère important d'évaluation d'un système. D'abord, les évaluations des utilisateurs varient généralement avec le temps. Une grande précision de prédiction est donc illusoire. Ensuite, le temps de calcul est souvent un facteur déterminant. Les utilisateurs ne sont pas prêts à attendre beaucoup plus qu'une seconde (trois

secondes semble le maximum bien toléré) les recommandations. Par ailleurs, ce qui compte le plus n'est pas la précision des évaluations prédites, mais le classement et le fait qu'il n'y ait pas de faux positifs dans les items recommandés. Or, finalement, cette contrainte entre en conflit avec une autre attente des utilisateurs. En effet, il ne suffit pas de recommander aux utilisateurs ce qu'ils attendent. Supposons que l'utilisateur soit un admirateur de Jacques Brel ; si le système lui recommande 10 disques de Jacques Brel, il aura l'impression que le système ne lui apporte rien. Il faut donc que le système soit capable de suggérer des « nouveautés ». Dans le cadre musical, des systèmes tels que *Music Access* ou *Music Browser*, dus à François Pachet, représentent des tentatives très intéressantes de prise en compte de la similarité entre morceaux musicaux et des goûts exprimés de l'utilisateur pour soit organiser une liste de morceaux musicaux, soit suggérer l'écoute de morceaux inconnus.

5.2.4 Limitations générales

D'autres difficultés sont à considérer :

- Le filtrage collaboratif court le risque de la sur-adaptation, ne recommandant jamais d'items qui ne figurent pas dans le profil initial de l'utilisateur. Si l'on ne peut reprocher au système de ne pas deviner les centres d'intérêt potentiellement multiples de l'utilisateur, il est cependant intéressant de chercher à élargir les recommandations offertes.
- Une difficulté liée à la précédente est celle de la construction d'un profil pour les nouveaux utilisateurs. Comment intégrer un nouvel utilisateur rapidement sans lui demander de renseigner une fiche qui peut vite sembler trop longue à remplir ?

Une approche qui peut répondre en partie à ces problèmes est d'utiliser des évaluations implicites des utilisateurs. Ainsi, au lieu de demander explicitement à un utilisateur de noter des items, il devient possible avec les technologies actuelles d'évaluer ces notes en mesurant des indices, tels que les achats effectués sur un site, le temps passé sur des pages du site, etc.

Notes historiques supplémentaires

La notion fondamentale d'**apprentissage actif** a une longue histoire en apprentissage artificiel. À notre connaissance, mais il conviendrait de mener une recherche plus approfondie, les premiers à parler d'apprentissage actif sont Simon et Lea [SL74] et Winston [Win75]. Simon et Lea argumentent ainsi que l'apprentissage artificiel est différent d'autres tâches de résolution de problème dans la mesure où l'apprentissage implique à la fois une recherche dans l'espace des hypothèses et dans l'espace des exemples. De ce fait, l'exploration de l'espace des hypothèses peut avoir un impact sur la recherche dans l'espace des exemples. Winston, quant à lui, détermine que les meilleurs exemples d'apprentissage sont les nuances critiques (*near-misses*) dans le cadre de son système ARCH basé sur une représentation des exemples et des hypothèses par réseau sémantique. Plus tard, des résultats théoriques ont montré qu'une réduction substantielle du nombre d'exemples d'apprentissage requis pouvait être obtenue par un échantillonnage bien informé [Ang88, Val84]. Le terme d'apprentissage actif a été utilisé plus récemment par Cohn et ses collègues [CAL94] pour décrire les protocoles dans lesquels c'est l'apprenant lui-même qui sélectionne les exemples potentiellement les plus informatifs.

Le terme **collaborative filtering** fut proposé par David Golberg et ses collaborateurs chez Xerox en 1992 [GNOT92].

Deux ans plus tard, en 1994, Paul Resnick du MIT (*Massachusetts Institute of Technology*)

et ses collaborateurs de l'Université du Minnesota proposèrent l'architecture GroupLens⁶ pour recommander des articles dans les newsgroup.

La librairie Amazon a popularisé le filtrage collaboratif avec « sa fonction : les utilisateurs qui ont aimé ce livre ont aussi aimé tel autre livre ». L'ingénieur responsable de ce projet, Greg Linden, a d'ailleurs un blog très intéressant (en anglais).

En 1998, Brin et Page publièrent leur algorithme PageRank et lancèrent Google. La même année, chez Microsoft, John S. Breese et ses collaborateurs publièrent un article charnière, *Empirical Analysis of Predictive Algorithms for Collaborative Filtering* [BHK98] dans lequel figure une comparaison détaillée des divers algorithmes de filtrage collaboratif.

Avant 2001, les algorithmes de filtrage collaboratif étaient soit basés sur les réseaux bayésiens, les réseaux de neurones, etc., soit sur une approche utilisateur-utilisateur. En 2001, Amazon innovait avec la publication d'un brevet introduisant le filtrage collaboratif basé sur l'article ; la même année, le groupe GroupLens publiait aussi, indépendamment, le même type d'algorithme [SKKR01].

En 2006, la compagnie Netflix a annoncé qu'elle accorderait un prix d'un million de dollars à celui qui améliorerait de 10 % leur outil de recommandation. La compagnie rend ainsi disponible un ensemble de données qui permettent de tester des systèmes. La compétition prendra fin en 2011.

Résumé

- Un *apprenant actif* a l'initiative du choix des exemples d'apprentissage. De nombreuses approches heuristiques ont été développées. Même si l'avantage en termes de nombre d'exemples requis pour apprendre n'est pas garanti, il s'observe cependant généralement. Les études théoriques doivent élargir le cadre i.i.d. classique pour en rendre compte. C'est encore un champ de recherche ouvert et... actif.
- L'*apprentissage en ligne* concerne l'apprentissage à partir de flux de données, éventuellement issus de distributions non stationnaires. Là aussi, le cadre classique i.i.d. doit être dépassé. Il y a là l'occasion d'un renouvellement important du paradigme de l'apprentissage artificiel qui s'ajoute à l'enjeu de pouvoir traiter de nombreux nouveaux domaines d'application.
- Dans le cadre de l'apprentissage à partir de *flux de données*, l'un des problèmes les plus importants est de calculer des résumés des données passées qui s'adaptent aux variations du processus génératif. Il y a là de très intéressantes questions de calcul et d'informatique qui ont produit de nombreux algorithmes très astucieux.
- L'apprentissage supervisé peut viser à calculer des *étiquettes complexes*, telles que des arbres ou des séquences. Dans ce cadre, il faut en particulier savoir mesurer des distances entre structures et rendre compte des dépendances à l'intérieur de l'espace de sortie \mathcal{Y} et avec l'espace des entrées \mathcal{X} . Les méthodes à noyaux sont actuellement les plus étudiées pour cette tâche.

⁶ voir <http://www.inf.ed.ac.uk/teaching/courses/tts/papers/resnick.pdf>

- L'apprentissage pour le filtrage collaboratif introduit un nouveau type d'apprentissage. Il requiert le calcul de similarité entre items et entre utilisateurs à partir d'une matrice énorme et très creuse. L'estimation des performances est un problème en soi, demandant une métrique spécifique, rendant compte du caractère en ligne de la tâche et évitant les boucles de rétro-action trompeuses. Il s'agit d'un champ de recherche promis à un bel avenir.