Algorithmique des réseaux sociaux **2013/2014**

Cours 7 — 2-9 Décembre

Enseignant: Marc Lelarge *Scribe: Simon Forest, Robin Champenois*

Pour information

– Page web du cours
`http://www.di.ens.fr/~lelarge/soc.html`

7.1 Introduction : Histoire de Netflix

Netflix est une entreprise de location de DVD qui a commencé ses activités en 1997. Au début, les DVD loués étaient envoyés par courrier aux clients. Ceux-ci pouvaient garder leur DVD pendant une durée indeterminée. Une fois le film regardé, les clients renvoyaient le DVD par courrier, et indiquaient dans l'enveloppe le prochain film souhaité.

En 2008, Netflix est passé au streaming. En 2013, Netflix a 23 millions de clients.

Progressivement, Netflix s'est rendu compte que recommander des films aux utilisateurs augmentait les ventes. Netflix a donc eu l'idée d'intégrer un système de recommandation automatique, sans expert : on recommande un film à un utilisateur suivant les films qu'il a aimés et les films qu'il n'a pas aimés. Netflix conçoit un premier algorithme de recommandation appelé Cinematch.

Afin de mesurer les performances d'un tel algorithme, on utilise la mesure d'erreur *RMSE* : Root mean square error. L'erreur e définie par cette mesure est

$$e = \sqrt{\sum_{r_{u,i} connu} \frac{(r_{u,i} - \hat{r}_{u,i})^2}{C}}$$

où C est le nombre de votes $r_{u,i}$ connus dans les données de test.

Dans le but d'améliorer les performances de son système de recommandation, Netflix organise un concours de programmation. À la clé, une récompense de 1000000 $. Les règles du jeu sont les suivantes : les participants se voient confier un ensemble de votes d'utilisateurs de Netflix sur des films (100 millions de votes). Les participants doivent alors soumettre leur propre algorithme de recommandation, et améliorer l'algorithme Cinematch. Si une amélioration d'au moins 10% de l'erreur est constatée, le prix est remporté. Les utilisateurs ont accès à un ensemble de données de test (1,4 millions de votes) sur lequel ils peuvent constater les performances de leur algorithme. Ils peuvent ensuite soumettre leur algorithme à Netflix,

qui leur renvoie leurs résultats sur l'ensemble Quiz, privé, gardé par Netflix (1,4 millions de votes). L'évaluation finale se fait sur un dernier ensemble de test, gardé secret (1,4 millions de votes).

Le problème est donc le suivant : *À partir d'un ensemble de notes données par des utilisateurs sur des films, prédirepour chaque couple (utilisateur,film) qui n'est pas connu la note que va donner cet utilisateur au film.* Pour gagner, il faut minimiser l'erreur RMSE.

L'algorithme naïf qui consiste à renvoyer la moyenne des votes de tous les utilisateurs sur un film i comme valeur de prédiction $\hat{r}_{u,i}$ obtient une erreur $e = 1.0540$. L'algorithme Cinematch obtient un score $e = 0.9514$.

En Septembre 2009, Netflix récompense l'équipe «BellKor's pragmatic chaos», qui a réalisé un score de 0.8553, soit une amélioration de 10.09 % sur l'algorithme de Netflix.

Dans la suite, nous allons développer les solutions à ce genre de problèmes de collaborative filtering.

Modèle de voisinage : on recommande un utilisateur suivant les utilisateurs qui ont un profile voisin du sien

Modèle des facteurs latents : on cherche une structure cachée de petite dimension qui modèlise bien les votes que l'on obtient

7.2 Prédicateur basique

On note $\bar{r}$ la moyenne de toutes les notes dont on dispose, N le nombre d'utilisateurs, M le nombre de films, V_u le nombre de votes de l'utilisateur u que l'on a dans notre base de données, V_i le nombre de votes pour le film i que l'on a dans notre base de données.

Dans le modèle basique, on suppose que chaque utilisateur u note tous les films qu'il a regardé avec un écart à la moyenne b_u qui lui est propre. Aussi, chaque film a un écart à la moyenne qui lui est propre b_i. On obtient alors un prédicateur $\hat{r}$ des notes attribuées par chaque utilisateur à chaque film tel que

$$\hat{r}_{u,i} = \bar{r} + b_u + b_i$$

Remarque : On pourrait prendre :

$$b_u = \frac{\sum_i r_{u,i}}{V_u} - \bar{r}$$

$$b_i = \frac{\sum_u ru, i}{V_i} - \bar{r}$$

mais cela n'est pas optimal.

Pour trouver les valeurs $(b_i)_i$ et $(b_u)_u$ idoines, on va formuler le problème sous la forme d'un problème d'optimisation.

On veut minimiser l'erreur RMSE suivant $(b_i)_i$ et $(b_u)_u$:

$$\min_{b_u,b_i} \sum_{u,i} (r_{u,i} - \hat{r}_{u,i})^2$$

Ceci est un problème standard de minimisation des moindres carrés : On cherche à minimiser $\|Ab-c\|_2^2$ suivant b, connaissant la matrice A et le vecteur c avec $b \in \mathbb{R}^{N+M}$, $A \in 0,1^{C\times(N+M)}$, et $c \in \mathbb{R}^C$ où C est le nombre de votes dont on dispose.

On a alors

$$\|Ab - c\| = b^t A^t Ab - 2b^t A^t c + c^t c$$

que l'on dérive suivant b. On cherche ensuite un point d'annulation de la dérivée (pour trouver le minimum), et on trouve :

$$A^t Ab = A^t c$$

Si les colonnes de A sont indépendantes, alors $A^t A$ est définie positive et on peut retrouver $b = (A^t A)^{-1} A^t c$.

Cependant, avec cette méthode, on peut avoir un problème de surapprentissage : les valeurs (b_u) et (b_i) donnent de bonnes réponses pour les données d'entraînement mais se généralisent assez mal sur les données de test. On peut parfois supprimer le surapprentissage en ajoutant une contrainte d'optimisation supplémentaire. On résout donc cette fois le problème d'optimisation suivant :

$$\min_{b_u,b_i} = \sum_{u,i} (r_{u,i} - \hat{r}_{u,i})^2 + \lambda(\sum_u b_u^2 + \sum_i b_i^2)$$

en ajustant le paramètre λ de façon à avoir des valeurs (b_u) et (b_i) efficaces pour ce que l'on cherche à faire, tout en évitant le problème de surapprentissage.

Une fois les valeurs (b_u) et (b_i) déterminées, on note $\tilde{r}_{u,i} = r_{u,i} - (\bar{r} + b_u + b_i)$ et $\tilde{R} = (\tilde{r}_{u,i})_{u,i})$

7.3 Méthode du voisinage

Avec cette méthode, on cherche à établir une notion de similarité entre utilisateurs pour pouvoir prédire la note que va mettre l'utilisateur u au film i suivant les notes qu'ont mises les utilisateurs similaires à u.

Définition 7.3.1 (Score de similarité) *Soient $\tilde{r}_i$ et $\tilde{r}_j$ deux colonnes de $\tilde{R}$ associées aux films i et j. On définit le score de similarité $s_{i,j}$ entre les deux films i et j comme étant la valeur :*

$$s_{i,j} = \frac{\tilde{r}_i^t \tilde{r}_j}{\|\tilde{r}_i\|_2 \|\tilde{r}_j\|_2} = \frac{\sum_u \tilde{r}_{u,i}\tilde{r}_{u,j}}{\sqrt{(\sum_u r_{u,i}^2)(\sum_u r_{u,j}^2)}}$$

en ne sommant bien sûr la valeur $r_{u,i}$ que si on la connaît.

On note $S = (s_{i,j})_{i,j}$. Pour un film i donné, on classe les autres films i par ordre décroissant suivant $|s_{i,j}|$. On prend ensuite les L premiers voisins suivant ce classement comme voisins du film i, dont l'ensemble est noté $\mathcal{L}_i$.

Ainsi, la note prédictive de l'utilisateur u sur le film i sera la note du prédicateur basique plus une somme pondérée par $w_{i,j}$ des films voisins.

Un choix naturel est $w_{i,j} = s_{i,j}$. Le nouveau prédicateur s'écrit donc

$$\tilde{r}_{u,i} = (\bar{r} + b_u + b_i) + \frac{\sum_{j\in\mathcal{L}_i} s_{i,j}\tilde{r}_{u,j}}{\sum_{j\in\mathcal{L}_i} |s_{i,j}|}$$

7.4 Méthode des facteurs latents

Dans cette partie, on développe une autre idée basée sur une structure cachée que peuvent avoir les données : un utilisateur aurait un ensemble de préférences pour des types de films, et un film serait une combinaison de types de films.

On modélise cela par un vecteur p_u de dimension K pour chaque utilisateur modélisant ses goûts, ainsi que pour chaque film un vecteur q_i de dimension K modélisant ses attraits. Le produit scalaire $p_u^t q_i$ est alors une prédiction $\hat{r}_{u,i}$.

On cherche alors à résoudre le problème d'optimisation suivant :

$$\min_{P,Q} \sum_{u,i} (r_{u,i} - p_u^t q_i)^2$$

où P est une matrice $N \times K$ contenant les p_u et où Q est une matrice $K \times M$ contenant les q_i.

On peut aussi rajouter une pénalisation $\lambda(\sum_u \|p_u\|_2^2 + \sum_i \|q_i\|_2^2)$ pour ajuster l'équilibre entre performances et surapprentissage.

7.5 Décomposition en valeurs singulières

Ici, on cherche à trouver une décomposition de la forme $A = UDV^T$ où les colonnes de U et V sont orthonormales et D est diagonale à entrées positives.

Cette décomposition peut se voir d'un point de vue géométrique : A matrice $n \times d$ vue comme n points dans $\mathbf{R}^d$ La question posée est alors : quel est le meilleur sous-espace de dim k approximant ces points ? C'est-à-dire quel est le sous-espace qui minimise la somme des carrés des distances de ces points au sous-espace ?

Pour $k = 1$: Quelle est la meilleure ligne passant par l'origine ? Si on choisit une droite $\mathcal{D}$ portée par un vecteur $\mathbf{v}$, et si on note π la projection sur la droite, on a : $|\mathbf{a}_i|^2 = |\pi(\mathbf{a}_i)|^2 + d(\mathbf{a}_i, \mathcal{D})^2$

donc $\sum_{i=1}^n |\mathbf{a}_i|^2 = \sum_{i=1}^n |\pi(\mathbf{a}_i)|^2 + \sum_{i=1}^n d(\mathbf{a}_i, \mathcal{D})^2$

Soit $\mathbf{v}$ tel que $|\mathbf{v}| = 1$. La longueur de la projection de la i$^{\text{e}}$ ligne $\mathbf{a}_i$ de la matrice A est $|\mathbf{a}_i\mathbf{v}|^2$. Donc la somme des carrés est $|A\mathbf{v}|^2$.

On définit le premier vecteur singulier $\mathbf{v}_1 = \text{argmax}_{|\mathbf{v}|=1} |A\mathbf{v}|$. La valeur $\sigma_1(A) = |A\mathbf{v}_1| \geq 0$ est *la première valeur singulière* de A.

Pour trouver la deuxième valeur singulière, on cherche $\mathbf{v}_2 \perp \mathbf{v}_1, |\mathbf{v}_2| = 1$ qui maximise $|A\mathbf{v}|$, donc le second vecteur singulier est $\mathbf{v}_2 = \text{argmax}_{|\mathbf{v}|=1, \mathbf{v}\perp\mathbf{v}_1} |A\mathbf{v}|$

$\sigma_2(A) = |A\mathbf{v}_2|$ est alors *la seconde valeur singulière.*

Etant donnés $\mathbf{v}_1, ..., \mathbf{v}_{k-1}$, on définit $\mathbf{v}_k = \text{argmax}_{\substack{|\mathbf{v}|=1 \\ \mathbf{v}\perp\mathbf{v}_1...\mathbf{v}_{k-1}}} |A\mathbf{v}|$, $\sigma_k(\mathbf{v}) = |A\mathbf{v}_k|$

On définit r comme le plus petit indice k tel que $\sigma_{k+1}(A) = 0$

Théorème 7.5.1 *Soit A matrice $n \times d$ et V_k le sous-espace engendré par $\mathbf{v}_1, ..., \mathbf{v}_k$. Alors pour tout $k \leq r$, V_k est un meilleur sous-espace de dimension k approximant A.*

Démonstration. Pour $k = 1$: Rien à montrer.

Pour $k = 2$: Soit W de dim 2 un meilleur sous-esp pour A. $(\mathbf{w}_1, \mathbf{w}_2)$ BON de W telle que $\mathbf{w}_2 \perp \mathbf{v}_1$.

$|A\mathbf{w}_1|^2 + |A\mathbf{w}_2|^2 =$ somme des carrés des projections des lignes de A sur W.

Par définition de $\mathbf{v}_1$, $|A\mathbf{w}_1|^2 \leq |A\mathbf{v}_1|^2$, et par définition de $\mathbf{v}_2$, $|A\mathbf{w}_2|^2 \leq |A\mathbf{v}_2|^2$

Les vecteurs $(\mathbf{v}_1, ..., \mathbf{v}_r)$ génèrent l'expace des lignes de A car $\mathbf{a}_i \cdot \mathbf{v} = 0$ pour $\mathbf{v} \perp \mathbf{v}_1, ..., \mathbf{v}_r$.

$$\begin{array}{rrcl} & \sum_{i=1}^{n}(\mathbf{a}_j \cdot \mathbf{v}_i)^2 & = & \sum_{i=1}^{r}\sum_{j=1}^{n}(\mathbf{a}_j \cdot \mathbf{v}_i)^2 \\ \text{Donc} & & = & \sum_{i=1}^{r}|A\mathbf{v}_i|^2 \\ & & = & \sum_{i=1}^{r}\sigma_i^2(A) \end{array}$$

□

Définition 7.5.1 *On définit la norme de Frobenius d'une matrice A par* $\|A\|_F = \sqrt{\sum \sigma_i(A)^2}$

Lemme 7.5.1 $\|A\|_F^2 = \sum_{i=1}^{r}\sigma_i^2(A)$

On définit les vecteurs singuliers à gauche de A par $\mathbf{u}_i = \frac{1}{\sigma(i)A\mathbf{v}(i)}$ où les $\mathbf{v}_i$ sont les vecteurs singulières à droite.

Théorème 7.5.2 *Soit A de rang r. Les vecteurs singuliers à gauche $\mathbf{u}_1, ..., \mathbf{u}_n$ sont orthogonaux.*

Démonstration. Par induction sur r. Pour $r = 1$, rien à montrer.

Pour $r \geq 2$: $B = A - \sigma_1\mathbf{u}_2\mathbf{v}_1^T$. On a $B\mathbf{v}_1 = A\mathbf{v}_1 - \sigma_1 u_i = 0$.

Donc $\mathbf{z}$, le premier vecteur singulier de B, est orthogonal à $\mathbf{v}_1$ car si il a une composante selon $\mathbf{v}_1$, notée $\mathbf{v}_1 \neq 0$, $\left|B\frac{\mathbf{z}-\mathbf{z}_i}{|\mathbf{z}-\mathbf{z}_i|}\right| = \frac{|B\mathbf{z}|}{|\mathbf{z}-\mathbf{z}_i|} > |B\mathbf{z}|$.

Pour tout $\mathbf{v}\perp\mathbf{v}_1$, on a $B\mathbf{v} = A\mathbf{v}$, donc le premier vecteur singulier de B est le second du A, etc.

Donc $\mathbf{v}_2, ..., \mathbf{v}_r$ et $\mathbf{u}_2, ..., \mathbf{u}_r$ sont les vecteurs singuliers à droite et gauche de B(exo !).

Il faut montrer que $\mathbf{u}_1\perp\mathbf{u}_i$ pour $i \geq 2$.

Si, au contraire, $\exists i$ tel que $\mathbf{u}_1^T\mathbf{u}_i > 0$, $\forall\,\varepsilon > 0$,

$$A\left(\frac{\mathbf{v}_1 + \varepsilon\,\mathbf{v}_i}{|\mathbf{v}_1 + \varepsilon\,\mathbf{v}_i|}\right) = \frac{\sigma_1\mathbf{u}_1 + \varepsilon\,\sigma_i\mathbf{u}_i}{\sqrt{1+\varepsilon^2}}$$

Composante selon $\mathbf{u}_1$:

$$\begin{array}{rcl} \mathbf{u}_1^T\left(\frac{\sigma_1\mathbf{u}_1 + \varepsilon\,\sigma_i\mathbf{u}_i}{\sqrt{1+\varepsilon^2}}\right) & = & (\sigma_1 + \varepsilon\,\sigma_i\mathbf{u}_1^T\mathbf{u}_i)(1 - \frac{\varepsilon^2}{2} + O(\varepsilon^4)) \\ & = & \sigma_1 + \varepsilon\,\sigma_i\mathbf{u}_1^T\mathbf{u}_i + O(\varepsilon^2) > \sigma_1 \end{array}$$

Contradiction, donc $\mathbf{u}_1\perp\mathbf{u}_i$ pour $i \geq 2$ □

Théorème 7.5.3 *A se décompose suivant ses valeurs singulières :*

$$A = \sum_{i=1}^{r}\sigma_i\mathbf{u}_i\mathbf{v}_i^T = UDV^T$$

Démonstration. $\mathbf{v}_j$ vecteur singulier

$A\mathbf{v}_j = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \mathbf{v}_j = \sigma_j \mathbf{u}_j$

□

7.6 Meilleure approximation de rang k

Définition 7.6.1 *On définit la norme d'opérateur d'une matrice A par $\|A\|_2 = \max_{|v|=1} |Av| = \sigma_1(A)$*

Soit A matrice $n \times d$, $A = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$

Pour $k \in \{1...r\}$, soit $A_k = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$ de rang k.

Théorème 7.6.1 *Pour toute matrice B de rang au plus k, $\|A - A_k\|_F \leq \|A - B\|_F$.*

Lemme 7.6.1 *Les lignes de A_k sont les projections des lignes de A sur V_k.*

Démonstration. Projection de $\mathbf{a}_j = \sum_{i=1}^{k} (\mathbf{a}_j \cdot \mathbf{v}_i) \mathbf{v}_i^T$

Donc $\sum_{i=1}^{k} A\mathbf{v}_i \mathbf{v}_i^T = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T = A_k$

□

Démonstration. B de rang $\leq k$ minimisant $\|A - B\|_F$. Soit V l'espace engendré par les lignes de B : $\dim V \leq k$

Comme B minimise $\|A - B\|_F^2$, chaque ligne de B est la projection de la ligne de A sur V.

Donc $\|A - B\|_F^2$ =somme des carrés des distances des lignes de A à V.

Comme A_k minimise cette distance pour tout sous-espace de dimension k, on a bien la proposition

□

Théorème 7.6.2 *Pour toute matrice B de rang $\leq k$, $\|A - A_k\|_2 \leq \|A - B\|_2$*

Lemme 7.6.2 $\|A - A_k\|_2^2 = \sigma_{k+1}^2$

Démonstration. (Exo) □

Démonstration. Si A est de rang $\leq k$, c'est bon.

Sinon, on suppose A de rand $\geq k + 1$, $\|A - A_k\|_2^2 = \sigma_{k+1}^2$

Pour tout B de rang $\leq k$, $\ker B = \{\mathbf{v}, B\mathbf{v} = 0\}$: $\dim \ker B \geq d - k$.

Soit $\mathbf{v}_1 ... \mathbf{v}_{k+1}$ les $k+1$ premiers vecteurs singuliers de A. Il existe $|\mathbf{z}| = 1$ tel que $z \in \ker B \cap Vect(v_1, \cdots, v_{k+1})$.

On a donc $\|A - B\|_2^2 \geq |(A - B)\mathbf{z}|^2 = |A\mathbf{z}|^2$.

$$\begin{array}{rcl} |Az|^2 & = & |\sum_i \sigma_i \mathbf{u}_i \mathbf{v}_i^t \mathbf{z}|^2 \\ & = & \sum_{i=1}^{k+1} \sigma_i^2 (\mathbf{v}_i^t \mathbf{z})^2 \\ & \geq & \sigma_{k+1}^2 \sum_{i=1}^{k+1} (\mathbf{v}_i^t \mathbf{z})^2 \\ & = & \sigma_{k+1}^2 \end{array}$$

□