Collaborative Filtering

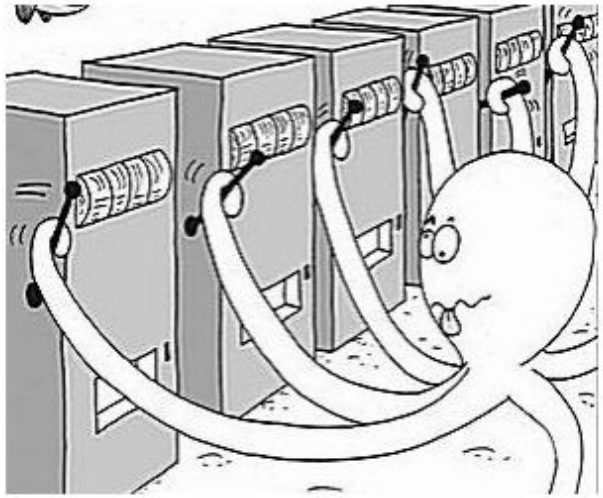
LinUCB for MovieLens dataset

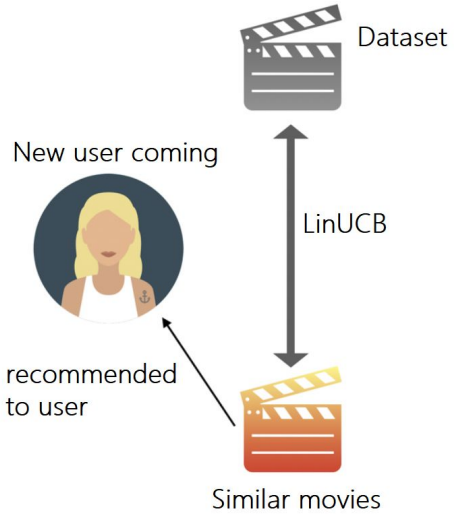
28.10.19: Approach & Preliminary work

Zambra team: Chen Dang, Công Minh Đinh, Oskar Rynkiewicz

Dauphine | PSL
UNIVERSITÉ PARIS

Introduction





Le problème du bandit à k bras représente un agent qui souhaite maximiser sa récompense totale.

Le système de recommandation propose des films aux nouveaux utilisateurs dont les préférences ne sont pas encore connues.

Minimiser l’espérance de regret cumulé

- Le bandit stochastique est défini par des distributions $(P_i)_{i=1}^k$
- Les réalisations de $X_{i,t} \sim P_i$ sont appelés les récompenses $r_{i,t}$
- Stratégie I_t est une distribution conditionnelle pour choisir une bra
- Minimiser l’espérance de regret cumulé

$$R(T) = \mathbb{E}_{I_t}\left[\sum_{t=1}^{T} r_{i^*,t}\right] - \mathbb{E}_{I_t}\left[\sum_{t=1}^{T} r_{i,t}\right]$$

- Les données contextuelles C_t peuvent améliorer la qualité des actions

$$X_t = r(C_t, A_t) + \varepsilon_t$$

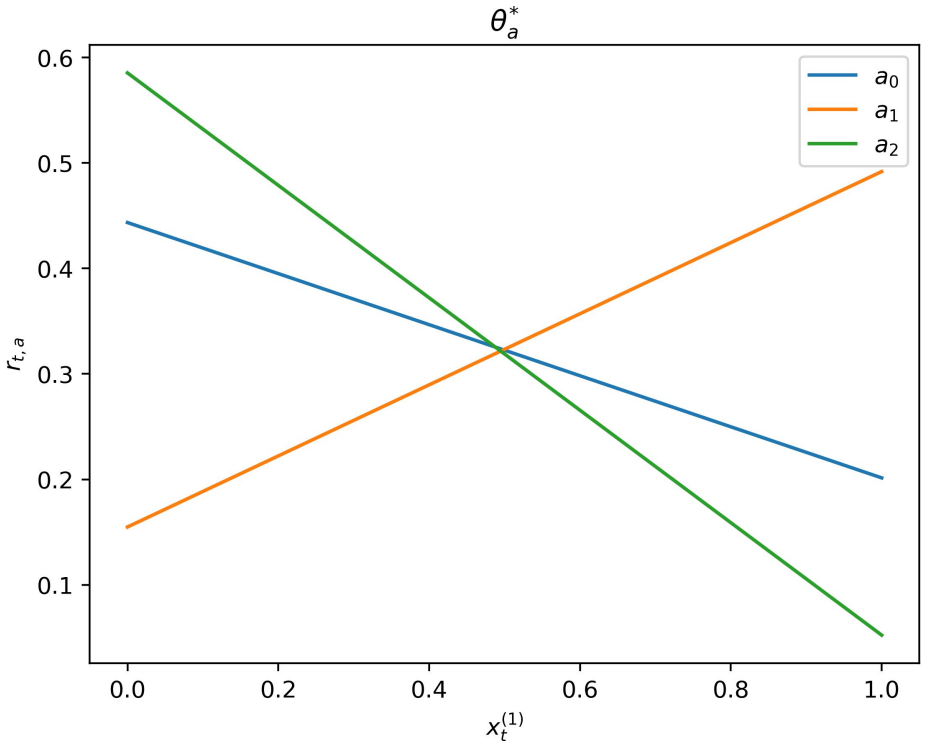
Bandits linéaires avec l’information contextuelle

La récompense de chaque bra est une fonctione linéaire d’un vecteur des attributs:

$$\mathbb{E}\left[\mathbf{r}_{\mathrm{t,a}} \mid \mathbf{x}_{\mathrm{t,a}}\right] = \mathbf{x}_{\mathrm{t,a}}^{\top} \theta_{\mathrm{a}}^{*}$$

Estimer

$$\hat{\theta}_a$$



LinUCB with linear models

0: Inputs: $\alpha \in \mathbb{R}_+$
1: **for** $t = 1, 2, 3, \ldots, T$ **do**
2: Observe features of all arms $a \in \mathcal{A}_t$: $\mathbf{x}_{t,a} \in \mathbb{R}^d$
3: **for all** $a \in \mathcal{A}_t$ **do**
4: **if** a is new **then**
5: $\mathbf{A}_a \leftarrow \mathbf{I}_d$ (d-dimensional identity matrix)
6: $\mathbf{b}_a \leftarrow \mathbf{0}_{d \times 1}$ (d-dimensional zero vector)
7: **end if**
8: $\hat{\boldsymbol{\theta}}_a \leftarrow \mathbf{A}_a^{-1} \mathbf{b}_a$
9: $p_{t,a} \leftarrow \hat{\boldsymbol{\theta}}_a^\top \mathbf{x}_{t,a} + \alpha \sqrt{\mathbf{x}_{t,a}^\top \mathbf{A}_a^{-1} \mathbf{x}_{t,a}}$
10: **end for**
11: Choose arm $a_t = \arg\max_{a \in \mathcal{A}_t} p_{t,a}$ with ties broken arbitrarily, and observe a real-valued payoff r_t
12: $\mathbf{A}_{a_t} \leftarrow \mathbf{A}_{a_t} + \mathbf{x}_{t,a_t} \mathbf{x}_{t,a_t}^\top$
13: $\mathbf{b}_{a_t} \leftarrow \mathbf{b}_{a_t} + r_t \mathbf{x}_{t,a_t}$
14: **end for**

Evaluation methodology

0: Inputs: $T > 0$; policy π; stream of events
1: $h_0 \leftarrow \emptyset$ {An initially empty history}
2: $R_0 \leftarrow 0$ {An initially zero total payoff}
3: **for** $t = 1, 2, 3, \ldots, T$ **do**
4: **repeat**
5: Get next event $(\mathbf{x}_1, ..., \mathbf{x}_K, a, r_a)$
6: **until** $\pi(h_{t-1}, (\mathbf{x}_1, ..., \mathbf{x}_K)) = a$
7: $h_t \leftarrow \text{CONCATENATE}(h_{t-1}, (\mathbf{x}_1, ..., \mathbf{x}_K, a, r_a))$
8: $R_t \leftarrow R_{t-1} + r_a$
9: **end for**
10: Output: R_T / T

Expériences

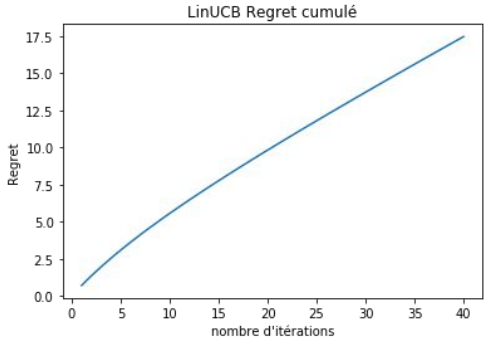
- Les ratings sont redimensionnés entre 0 et 1
- Les films regardés par plus de 1000 utilisateurs ont été sélectionnés (207 films)
- Les utilisateurs ayant regardé plus de 150 films ont été sélectionnés (210 users)
- SVD est appliqué sur la matrice de ratings pour obtenir les features d'utilisateurs et de films, normaliser les features et ajouter un constant 1 comme le biais
- Les 30 features les plus importantes ont été choisies
- Le regret de chaque itération est $regret = \sqrt{\frac{\sum_{(i,u)\in T}(R_{iu} - \hat{R}_{iu})^2}{|T|}}$

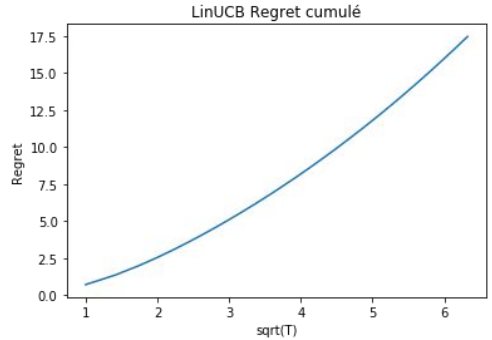
  Dans nos expériences, nous n'avons pas estimé les ratings de tous les films (à cause du temps considérable passé), donc nous calculons que certains films.

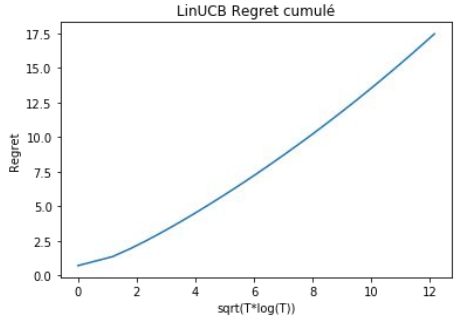
Expériences

$$R(T) = \tilde{O}(\sqrt{KdT})$$

où K: nombre d'arms
d: nombre de features
T: nombre d'itérations





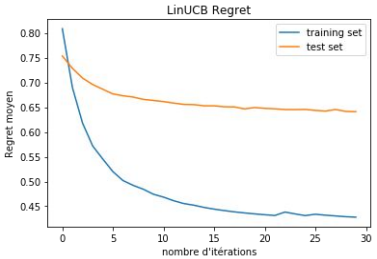


Petit test sur un utilisateur de MovieLens afin de vérifier le bound de regret

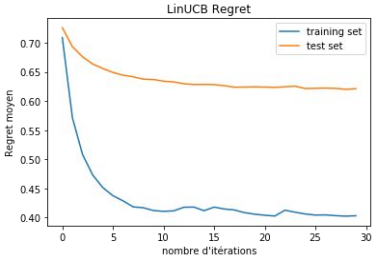
Expériences

Train test sur MovieLens

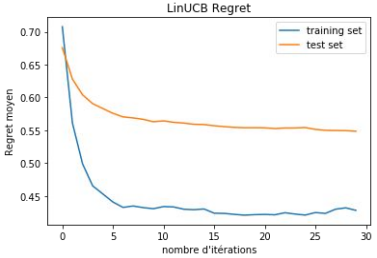
20% utilisateurs en total



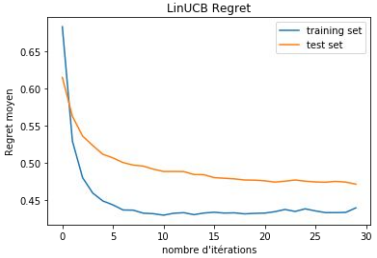
10% users en training set
90% users en test set



20% users en training set
80% users en test set

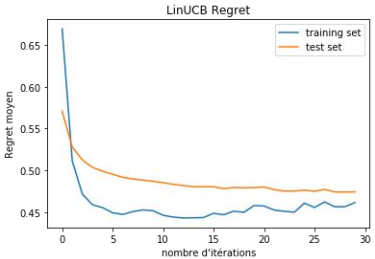


30% users en training set
70% users en test set



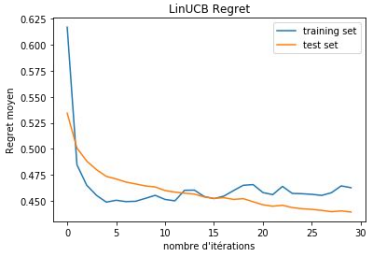
60% users en training set
40% users en test set

50% utilisateurs en total



30% users en training set
70% users en test set

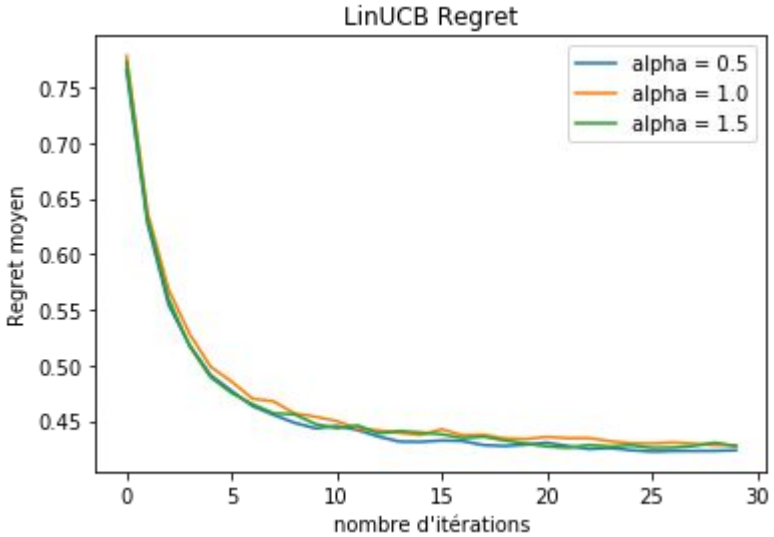
100% utilisateurs en total



30% users en training set
70% users en test set

Expériences

Etude de l'alpha



$$p_{t,a} \leftarrow \hat{\boldsymbol{\theta}}_a^\top \mathbf{x}_{t,a} + \alpha \sqrt{\mathbf{x}_{t,a}^\top \mathbf{A}_a^{-1} \mathbf{x}_{t,a}}$$

Nombre de films : 207
Nombre de users : 10

=> Le changement d'alpha n'a pas un grand impact sur les regrets

À faire

- Augmenter la performance
- Appliquer la méthode LinUCB Hybride
- Tester sur l’ensemble de données MovieLens