



# FILTRAGE COLLABORATIF – PROBLÈME DES BANDITS

Emma AMBLARD

Eva BOUBA

Charlotte DURAND



# Le problème des bandits

- K machines à sous donnant une récompense suivant une loi de probabilités inconnue d'espérance  $\mu_i$ , avec  $0 \leq i \leq K$ .
- Objectif : Maximiser le gain obtenu , ce qui est équivalent à minimiser le regret :

$$r_n = n\mu^* - \mathbb{E} \left[ \sum_{t=1}^n X_t \right]$$

- Choix à faire entre exploration et exploitation des observations faites lors de précédents tours

# Algorithmes de résolution

- $\varepsilon$ -greedy :
  - exploration avec probabilité  $\varepsilon$ , exploitation avec probabilité  $1 - \varepsilon$
- UCB (Upper Confidence Bound) :
  - À chaque instant  $t$ , choix du bras ayant la meilleure borne de confiance supérieure

$$UCB_i(t-1, \delta) = \begin{cases} \infty & \text{si } T_i(t-1) = 0 \\ \hat{\mu}_i(t-1) + \sqrt{\frac{2 \log(\frac{1}{\delta})}{T_i(t-1)}} & \text{sinon} \end{cases}$$

---

**Algorithm 1** UCB

---

**for**  $t \in 1, \dots, n$  **do**

    Choisir l'action  $A_t = \operatorname{argmax}_i UCB_i(t-1, \delta)$

    Observer la récompense  $X_t$  et mettre à jour l'intervalle de confiance

**end for**

---

# Algorithmes de résolution

## ■ Linear UCB :

- *On suppose que la récompense  $X_t$  est linéaire par rapport à un vecteur de contexte représentant le bras sélectionné :*

$$X_t = \langle \theta_*, A_t \rangle + \eta_t$$

---

**Algorithm 2** LinUCB

---

$V_0 \leftarrow \lambda I_d$

$\hat{\theta}_0 \leftarrow 0_{\mathbb{R}^d}$

**for**  $t \in 1, \dots, n$  **do**

Observer le contexte  $A_t$  pour chaque bras  $a \in \mathcal{A}_t$

**for all**  $a \in \mathcal{A}_t$  **do**

$UCB_{t,a} \leftarrow \langle \hat{\theta}_{t-1}^T, a \rangle + \sqrt{\beta_t} \sqrt{a^T V_{t-1}^{-1} a}$

**end for**

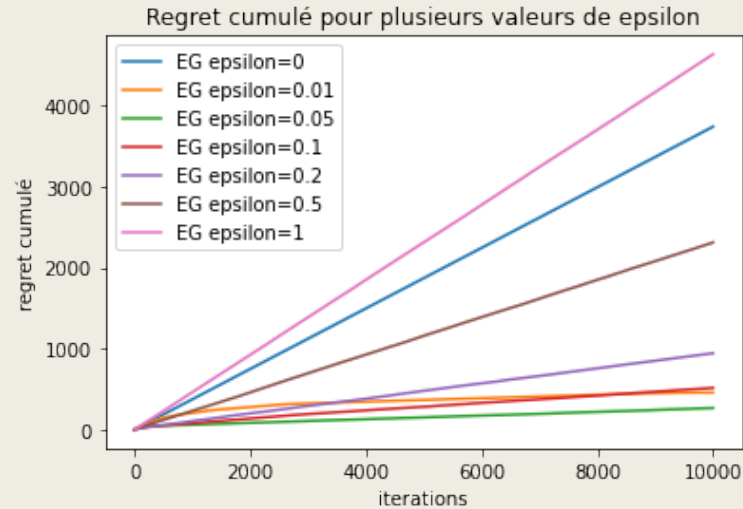
Sélectionner le bras  $A_t = \operatorname{argmax}_{a \in \mathcal{A}_t} UCB_{t,a}$  en choisissant aléatoirement en cas d'égalité, et observer une récompense  $X_t$

$V_t \leftarrow V_{t-1} + A_t A_t^T$

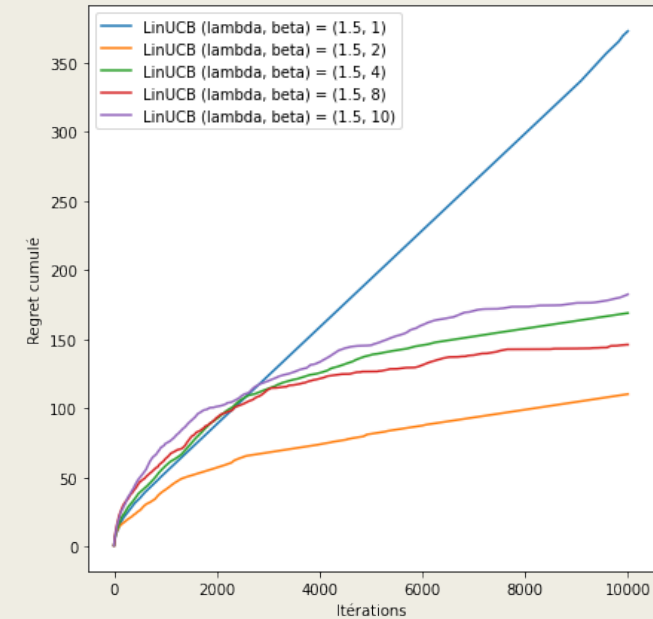
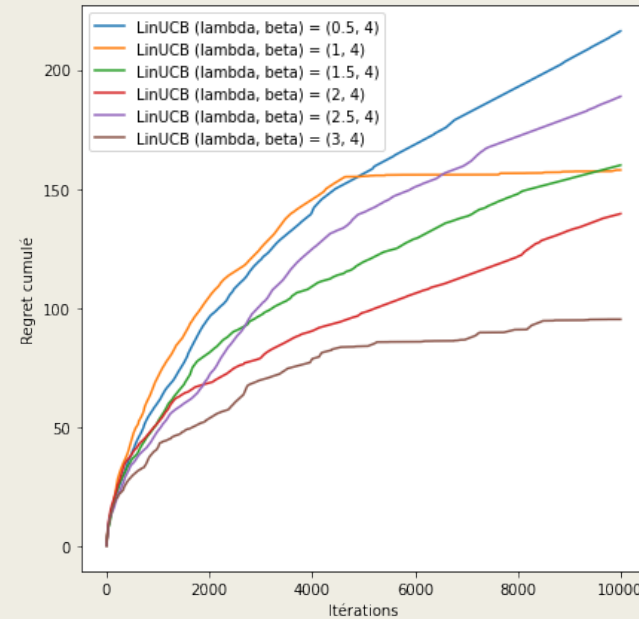
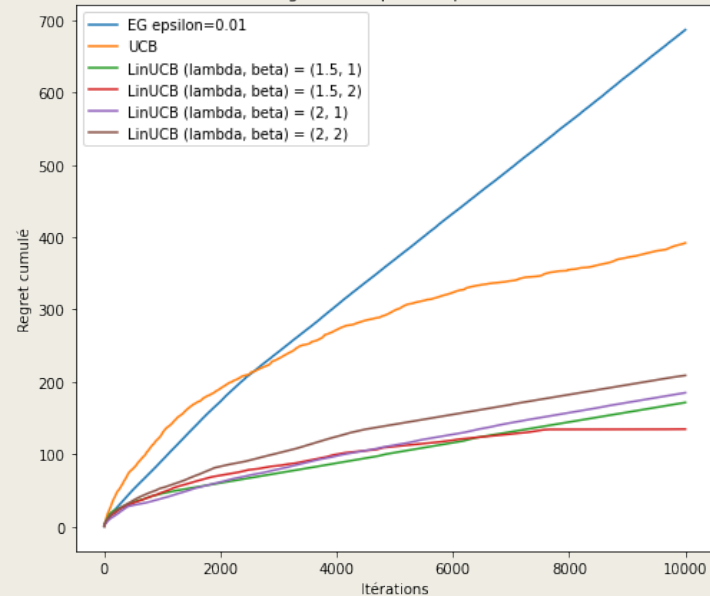
$\hat{\theta}_t \leftarrow V_t^{-1} \sum_{s=1}^t A_s X_s$

**end for**

# Test des algorithmes sur des données aléatoires



Regret cumulé en fonction des différents algorithmes pour un problème à 10 bras suivant des lois gaussiennes



- $\epsilon$ -greedy n'est pas une très bonne stratégie
- UCB est moins bon que LinUCB
- LinUCB fonctionne bien mais il faut optimiser ses paramètres

# Application de LinUCB au dataset MovieLens

1 bras = 1 film

But : maximiser la somme des notes que les utilisateurs mettent aux films recommandés

Pb : on ne peut recommander plusieurs fois le même film à un utilisateur : on a besoin d'autres informations, comme les notes déjà collectées sur les films

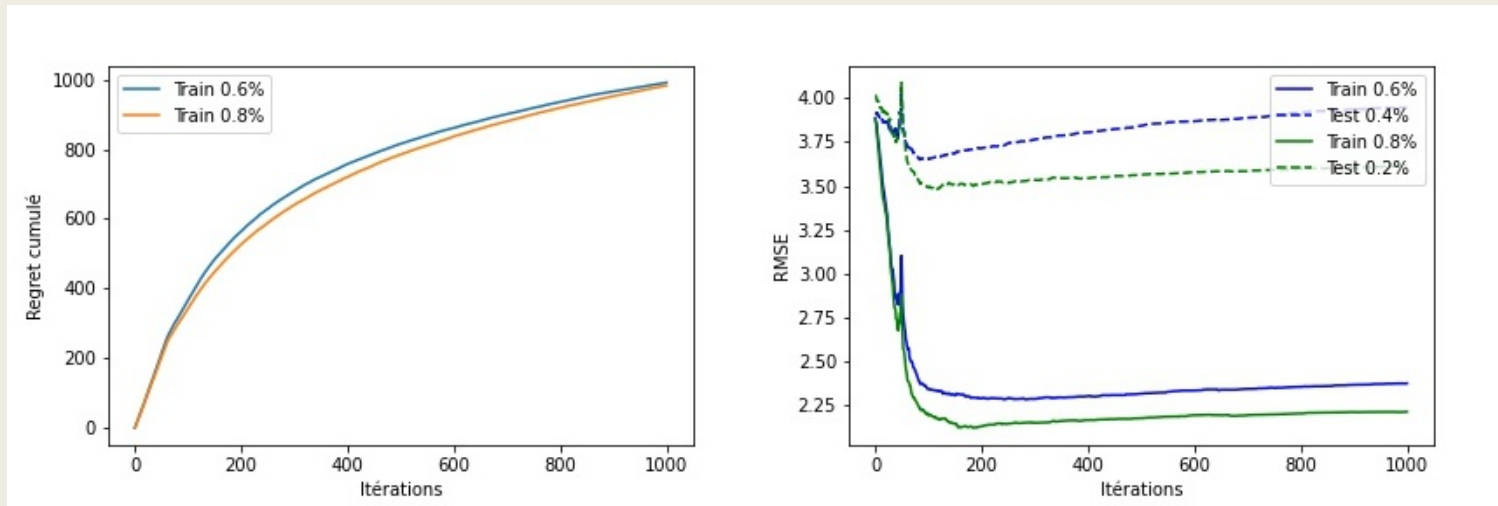
## Utilisation de l'algorithme LinUCB

- 1- Factorisation de la matrice de ratings (dimension  $d$ ) : on utilise la matrice  $V$  qui correspond à la matrice des films
- 2- Sélection du contexte : 80% de la matrice  $R$
- 3- Séparation du reste en training set et testing set et application de LinUCB

On mesure alors la RMSE sur le testing set

# Résultats

Lin UCB sur MovieLens Dataset 1M avec  $\beta=100$



Lin UCB sur MovieLens Dataset 1M avec  $\beta = 250$

