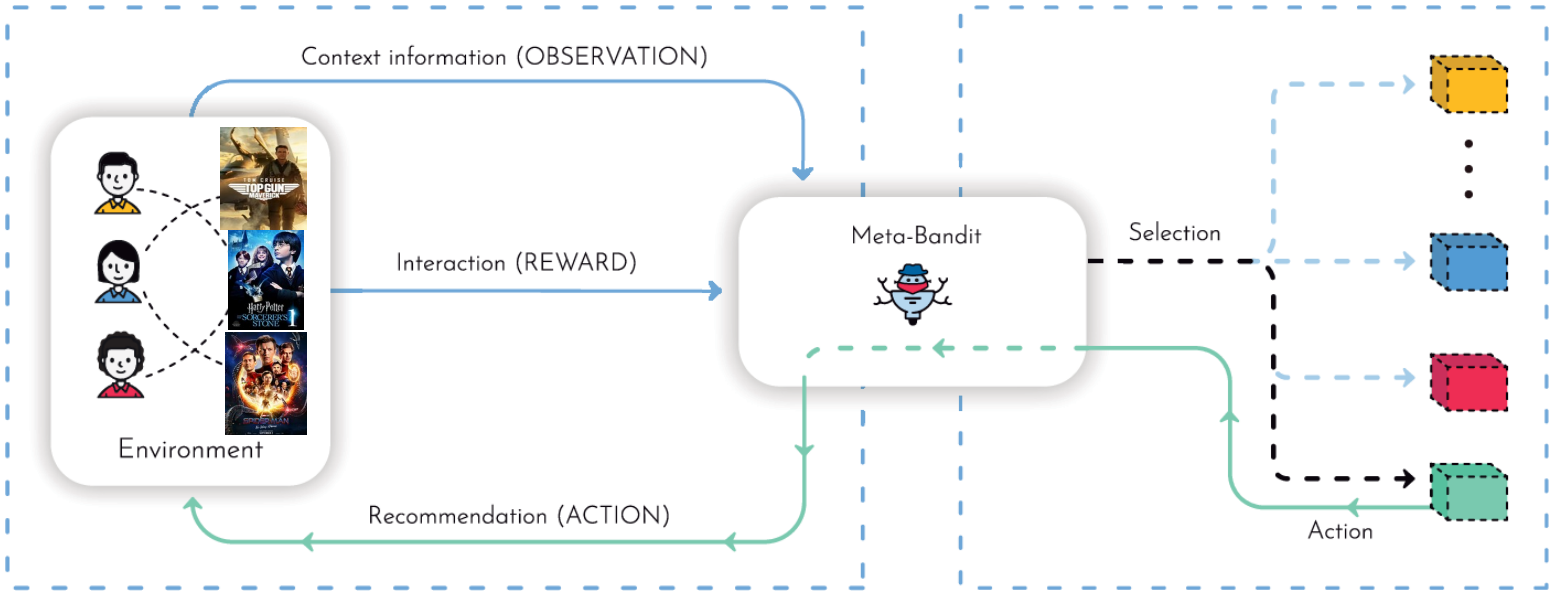
Data Science Project : Systems Recommendation

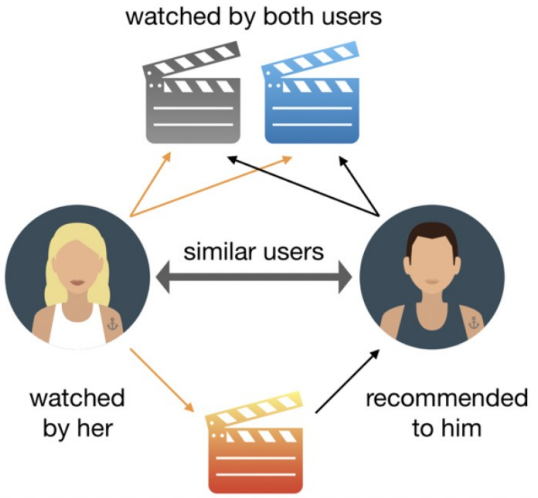
3rd approch : Linear Bandit (LB)



Adrien Golebiewski – Linda Gutsche – Sihan Xie

Dauphine | PSL
UNIVERSITÉ PARIS

Context definition and statement of the problem



Collaborative based recommendation system

Unable to deal with

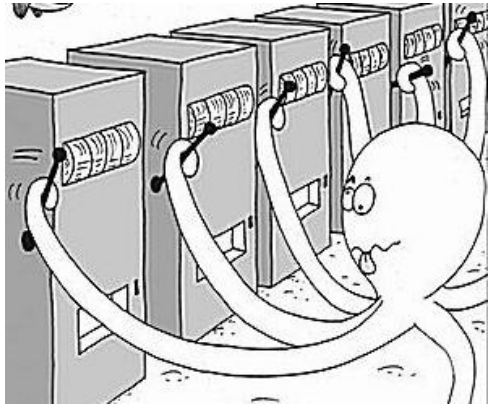
New User problem – Cold Start

Solution to tackle cold start ?

Dilemna - Make your choice :

Known movie (Exploitation) **OR** new movie (Exploration) ?

Multi-armed bandit framework



Our goals:

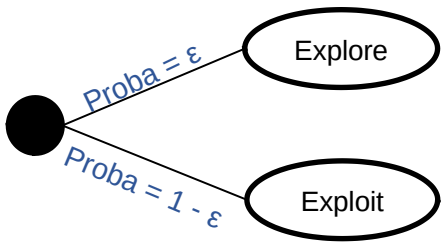
- **Tackle the cold-start** user problem by recommending a movie that will appeal to the user in question, rather than the best movie.
- Applying the M.A.B algorithms on the real Dataset MovieLens in order to make recommendations to **one user**.

Dauphine | PSL UNIVERSITÉ PARIS

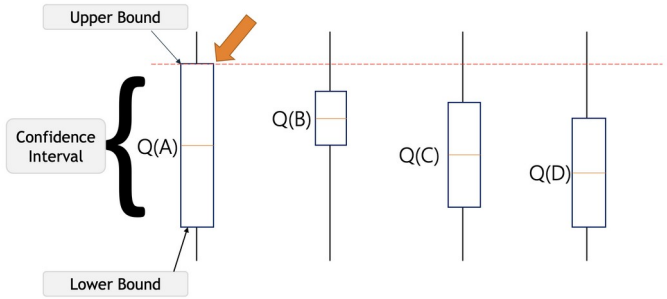
Our approach: test and compare different algorithms

Context-free Bandit Model

ε-greedy



UCB 1



Select the movie with the largest **Upper Confidence Bound**

Contextual Bandit Model

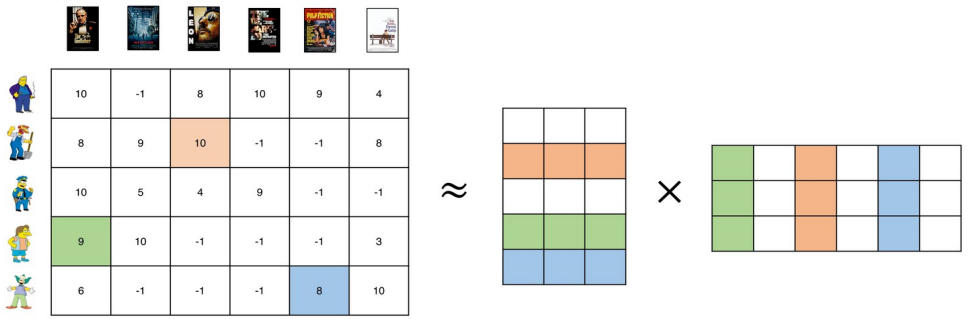
LinUCB

From A Contextual-Bandit Approach to Personalized News Article Recommendation, Li et al., 2010

For each trial t=1,2,…, T

1. A **context** is built for each available movie based on the current user() and each movie()
2. Select based on observed payoffs in previous trials and receive payoff ($\mathbf{E}[r_{t,a}|\mathbf{x}_{t,a}] = \mathbf{x}_{t,a}^{\top}\boldsymbol{\theta}_a^*$.
3. Improve arm-selection strategy with the new observation

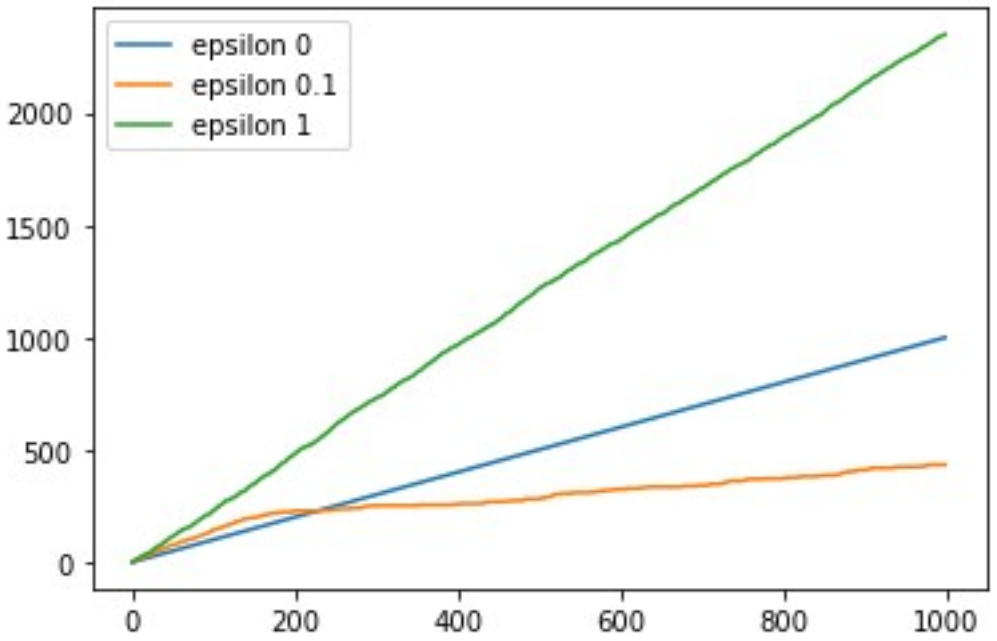
Learning transfer from one context to another



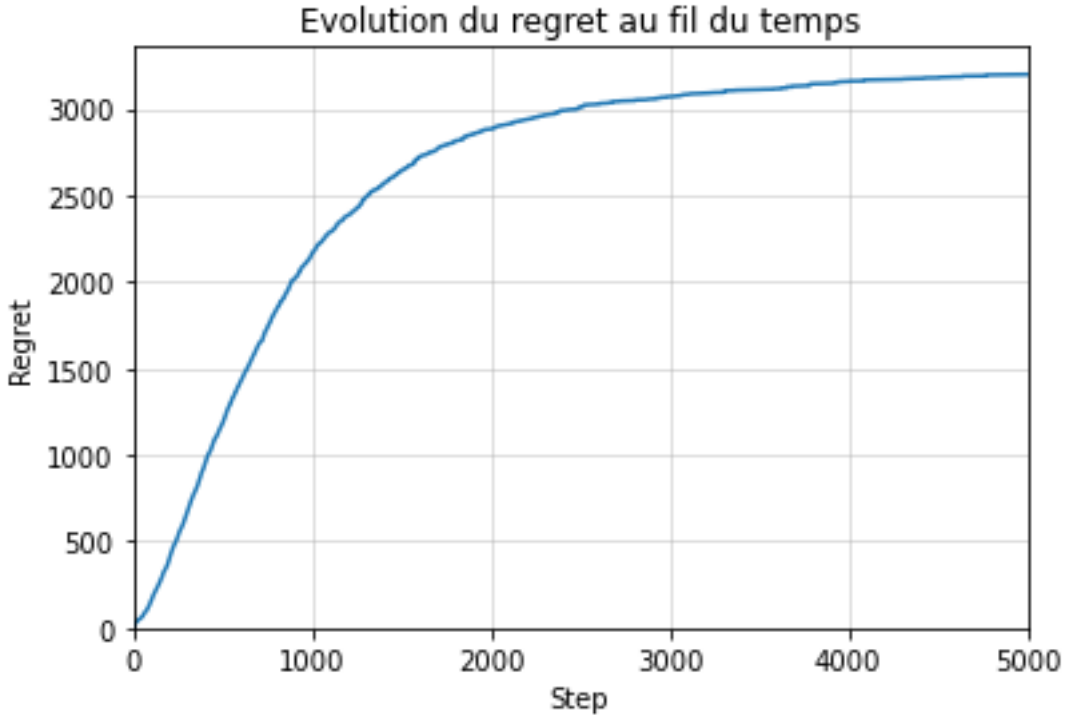
User-Movie Interaction Matrix | User Matrix | Movie Matrix

Dauphine | PSL UNIVERSITÉ PARIS

Some intermediary results in graphs



ε-greedy



linUCB

Evaluation metrics:

- **Cumulative regret** (equivalent to payoff)
- Evolution of **per-step regret**
- **Time** of execution
- **Distribution** of the recommendations

To do

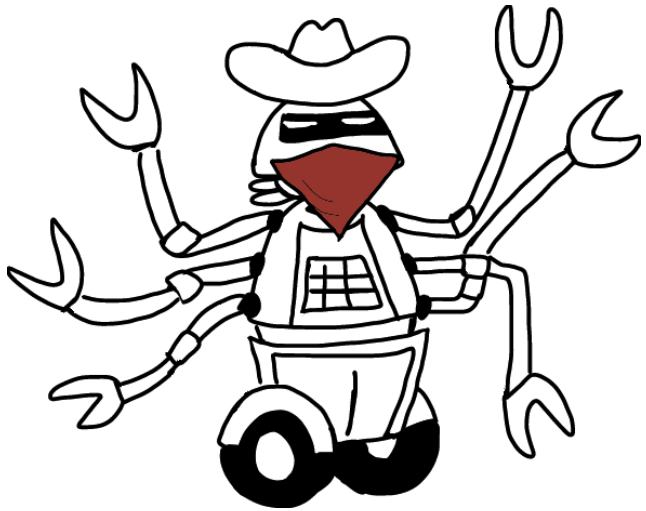
Current limits :

- **considered movies**
(limit on number of reviews)

- **chosen user**
(limit on number of reviews)

- **films we can recommend**
(we need know the payoff)

- recommendation for **only one** user



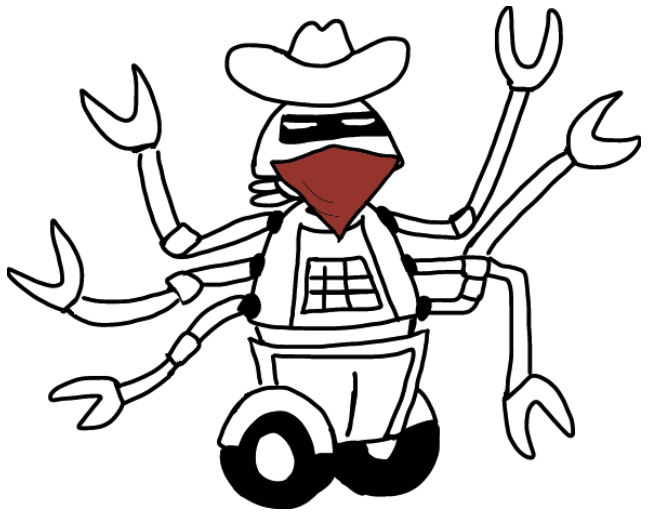
To do

Current limits :

- **considered movies** (limit on number of reviews)
- **chosen user** (limit on number of reviews)
- **films we can recommend** (we need know the payoff)
- recommendation for **only one** user

Experiments :

- Adjust the **parameters** that appear in the algorithms
- Test what happens when we **forbid recommending twice** the same film



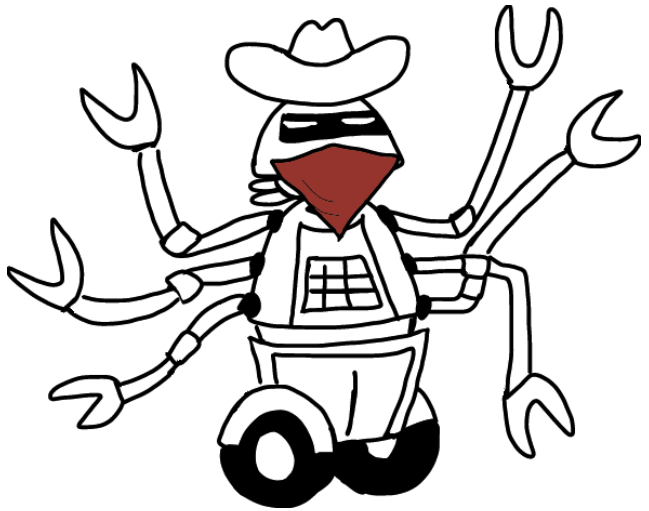
To do

Current limits :

- **considered movies** (limit on number of reviews)
- **chosen user** (limit on number of reviews)
- **films we can recommend** (we need know the payoff)
- recommendation for **only one** user

Experiments :

- Adjust the **parameters** that appear in the algorithms
- Test what happens when we **forbid recommending twice** the same film



Other algorithms :

- Such as hybrid LinUCB, Bayes, EXP3, NeuralUCB, etc.