Recovering Bandits

Ciara Pike-Burke[1], Steffen Grünewälder[2].

[1] Universitat Pompeu Fabra, [2] Lancaster University.

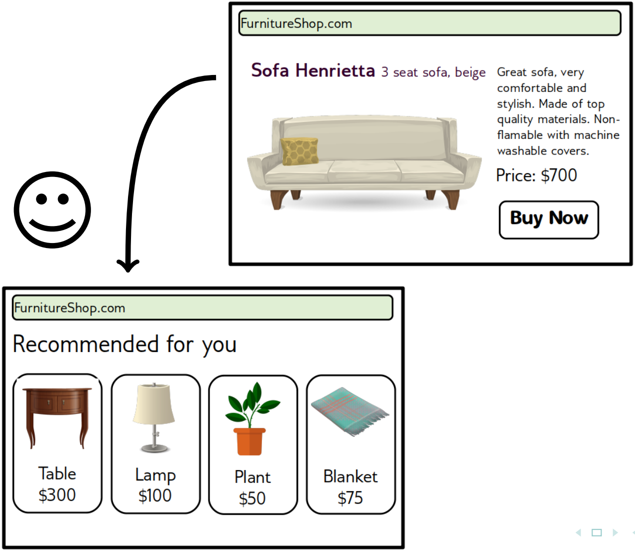
Recovering Bandits

Ciara Pike-Burke[1], Steffen Grünewälder[2].

[1] Universitat Pompeu Fabra, [2] Lancaster University.

Recovering Bandits

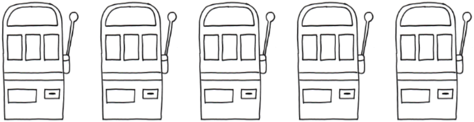
Ciara Pike-Burke[1], Steffen Grünewälder[2].
[1] Universitat Pompeu Fabra, [2] Lancaster University.



Recovering Bandits

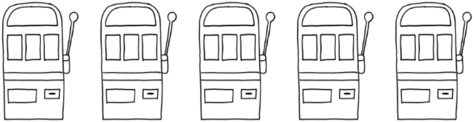
Ciara Pike-Burke[1], Steffen Grünewälder[2].
[1] Universitat Pompeu Fabra, [2] Lancaster University.

When bandit algorithms are used for recommendation, we model each item (or group of items) as an arm.



- In the classical formulation, a key assumption is that the reward of each arm is stationary
- Good bandit algorithms learn to play the best arm constantly.
- However, in many settings, playing a single arm isn't optimal and we want to wait before playing the same arm.

When bandit algorithms are used for recommendation, we model each item (or group of items) as an arm.

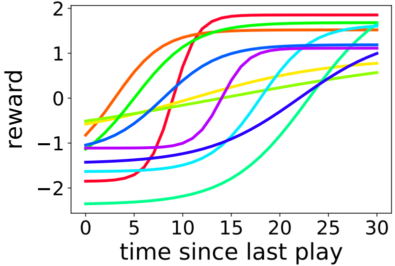


- In the classical formulation, a key assumption is that the reward of each arm is stationary
- Good bandit algorithms learn to play the best arm constantly.
- However, in many settings, playing a single arm isn't optimal and we want to wait before playing the same arm.

We tackle this problem in Recovering Bandits.

Recovering Bandits

We assume that the reward is a function of the time since the arm was last played.



Let $Z_{j,t}$ be the time since arm j was last played. Then, the expected reward is $f_j(Z_{j,t})$

and we observe,

$$Y_{j,t} = f_j(Z_{j,t}) + \epsilon_{j,t}.$$

We assume that these functions can be modeled by a Gaussian Process.

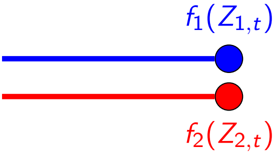
This problem is harder than the stationary bandits problem.

d-step lookahead

In recovering bandits, selecting the arm with highest current reward is not optimal.

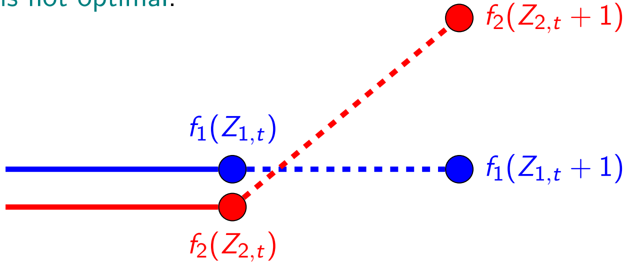
d-step lookahead

In recovering bandits, selecting the arm with highest current reward is not optimal.



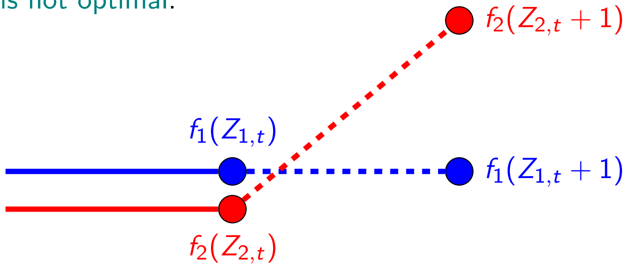
d-step lookahead

In recovering bandits, selecting the arm with highest current reward is not optimal.



d-step lookahead

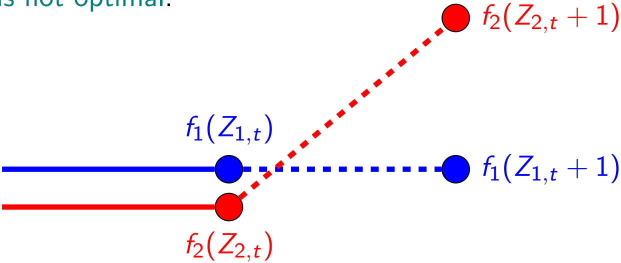
In recovering bandits, selecting the arm with highest current reward is not optimal.



Hence, we look for sequences of arms to maximize the reward over d plays. The corresponding regret is the d-step lookahead regret.

d-step lookahead

In recovering bandits, selecting the arm with highest current reward is not optimal.



Hence, we look for sequences of arms to maximize the reward over d plays. The corresponding regret is the d-step lookahead regret.

The optimal value of d is T. However, this is infeasible.

Proposition: For large d, the best d-step lookahead policy is near optimal.

Algorithms

We present modifications of UCB and Thompson sampling that exploit properties of GPs to select good sequences of d arms.

Theorem: The Bayesian d-step lookahead regret of both algorithms is $\widetilde{O}(\sqrt{dKT})$.

Thus, we only suffer an extra $\sqrt{d}$ regret compared to the easier stationary bandit problem.

Algorithms

We present modifications of UCB and Thompson sampling that exploit properties of GPs to select good sequences of d arms.

Theorem: The Bayesian d-step lookahead regret of both algorithms is $\widetilde{O}(\sqrt{dKT})$.

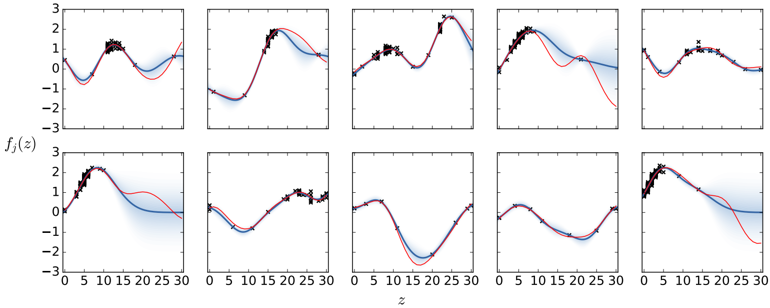
Thus, we only suffer an extra $\sqrt{d}$ regret compared to the easier stationary bandit problem.

Improving computational efficiency

We also provide an adaptation based on optimistic planning that is guaranteed to improve the computational complexity.

Experimental Results

Our algorithms learn to play arms when their rewards are highest.

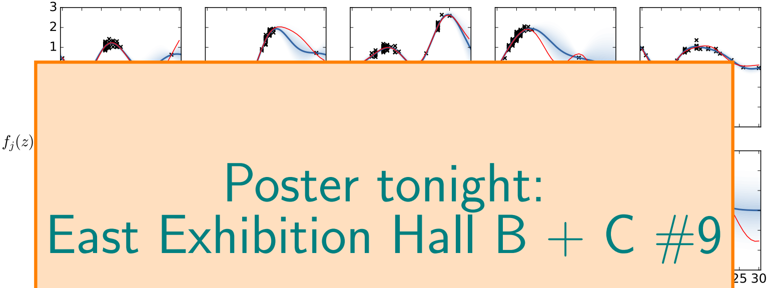


We also show experimentally that:

- our algorithms outperform other methods
- the optimistic planning procedure is more computationally efficient.

Experimental Results

Our algorithms learn to play arms when their rewards are highest.



Poster tonight:
East Exhibition Hall B + C #9

We a

- our algorithms outperform other methods
- the optimistic planning procedure is more computationally efficient.