

# Multi-Armed Bandit

Giacomo Boracchi, Francesco Trovò

May 11th, 2022

Politecnico di Milano, DEIB

[francesco1.trovo@polimi.it](mailto:francesco1.trovo@polimi.it)

# Lecture Overview

*Multi-Armed Bandit*

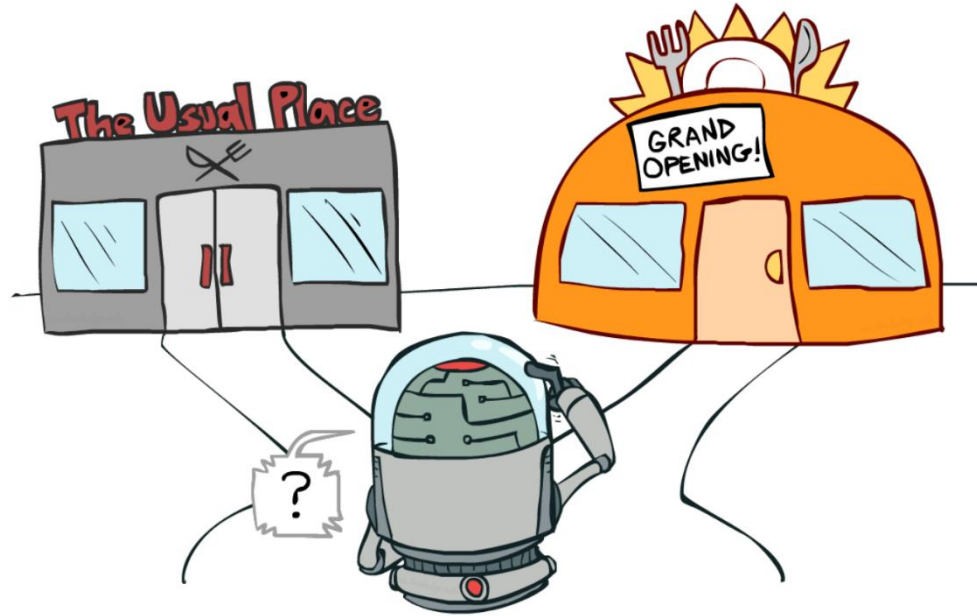
*Adversarial*

*Stochastic*

*Beyond MAB*

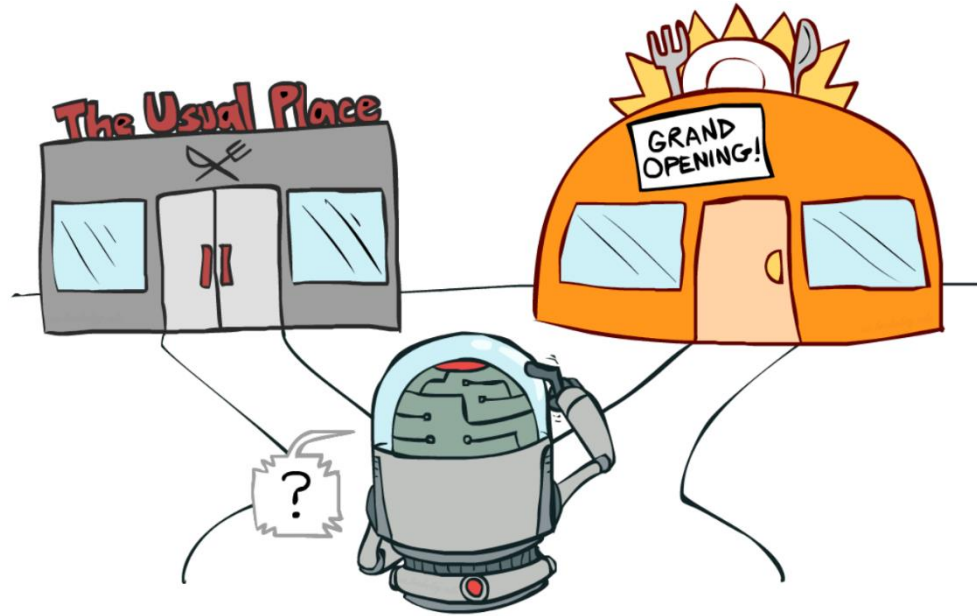
# Motivating Example

# Restaurant Selection Problem



- Going to the usual place you know what you might order and that it is a good choice for you
  - Going to a new place might provide you with your next favorite place in town
- I cannot judge a place from a single try, I need to collect some evidence

# Standard Solution: A/B Testing



Divide my dinners equally (even days, go to the favorite place, odd days, go to the new one)

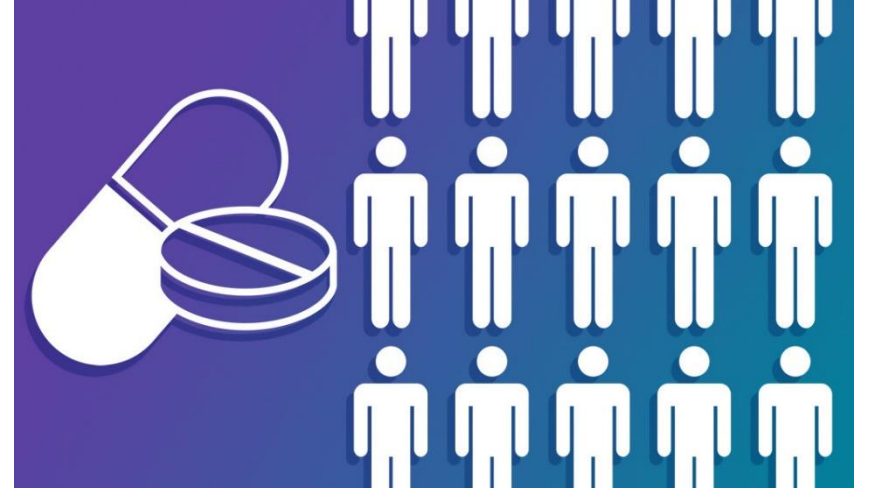
After a predetermined period, always select the place you like the most

You have a small probability that the restaurant you selected in the end is the worst one and you will go there forever...

# Clinical Trials

In the current situation we have:

- Two different groups (drug and placebo)
- Both have the same number of patients
- According to previous statistics, the desired power and confidence we design a study selecting the proper number of patients



Problems:

1. If the drug is effective, half of the patients are getting the placebo
2. If the drug is harmful, half of the patients are getting it

Goal: try to minimize the number of patients getting the wrong treatment

# Questions

Can we use traditional ML methods to solve these problem?

NO: we do not have a dataset at the beginning

Can we use expert learning techniques to select the restaurant?

NO: we do not have full feedback

Is the environment adversarial? Are we competing against an opponent?

NOT REALLY: in some situations, we are fighting ignorance (lack of information)

Is the online approach proper for this setting?

YES: we want to improve our decision policy as soon as a new data arrives

# Multi Armed Bandit

Model and Regret



# Different Online Learning Problems

- Expert learning

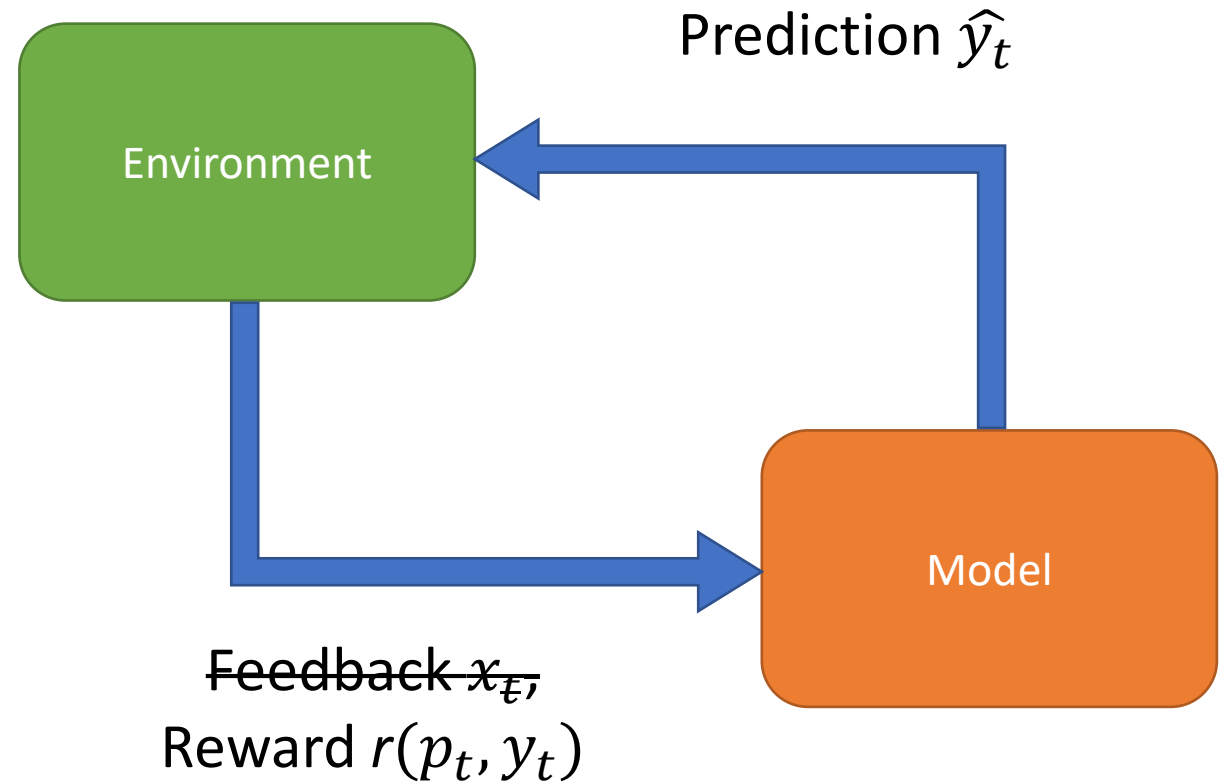
loss for all the possible choices

$$x_t = \{l(p, y_t), \forall p \in D\}$$

- **Multi-Armed Bandit**

No feedback  $x_t = ()$

- Partial Monitoring
- ...



In the MAB setting, we also need to take care of information gathering!

# A New Issue

If we want to have information about an option (an expert, a choice), we need to try it

The problem is intrinsically more difficult than the full feedback one

## Definition: Regret

Given an algorithm  $A$ , selecting a prediction  $\hat{y}_t$  at round  $t$ , and a clairvoyant algorithm  $A^*$ , selecting a prediction  $y_t^*$  at round  $t$ , the Regret of  $A$  over a time horizon of  $T$  rounds is:

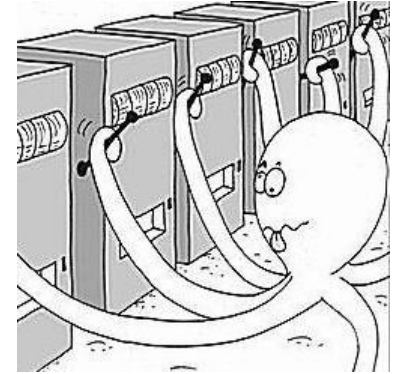
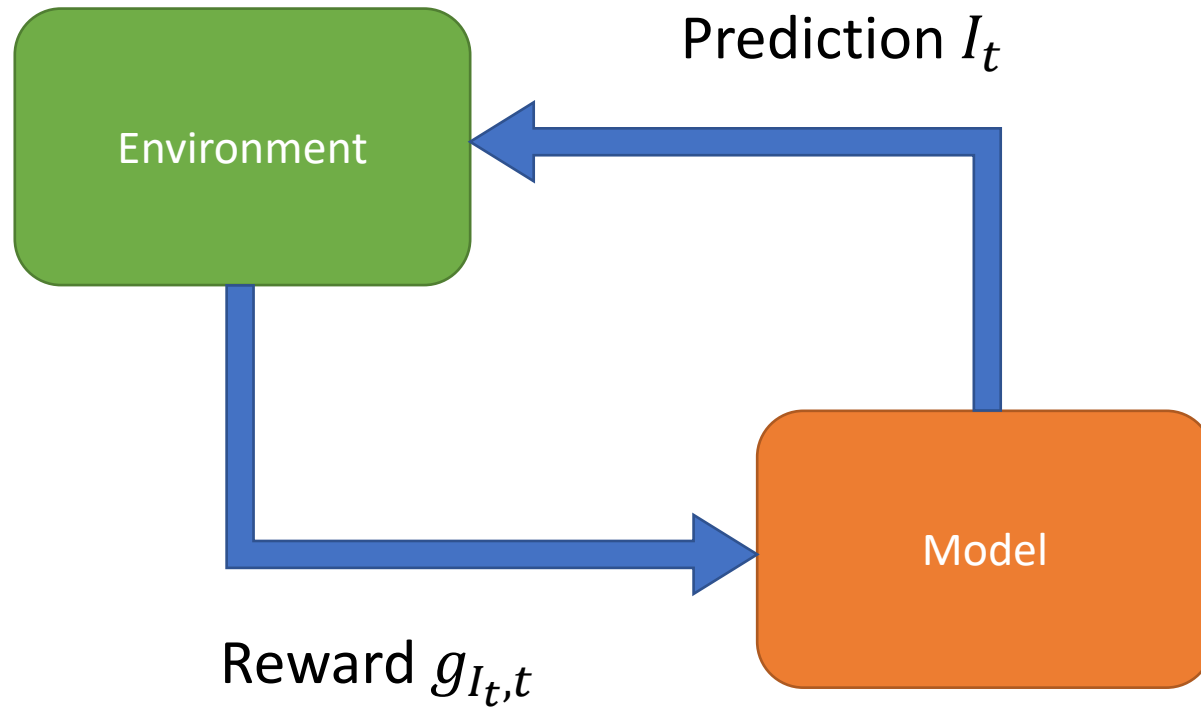
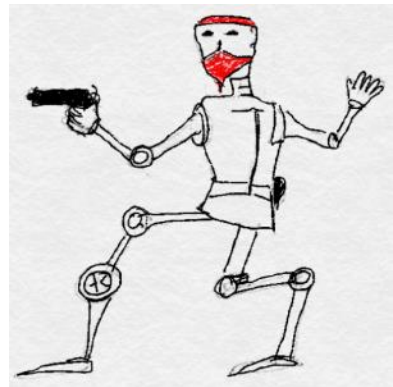
$$R_T(A) = \sum_{t=1}^T [r(y_t^*, y_t) - r(\hat{y}_t, y_t)]$$

To keep the regret limited we need to balance between:

**exploration and exploitation**

# Multi Armed Bandit Framework

We assume to have a finite decision set: the prediction is an index



# Multi-armed Bandit

Adversarial Setting

# MAB Setting

The learner chooses among a finite set of  $K$  options, called **arms**

The environment chooses a reward  $g_{i,t} \forall t \in \{1, \dots, n\}, \forall i \in \{1, \dots, K\}$

At each round  $t \in \{1, \dots, n\}$

the learner chooses  $I_t \in \{1, \dots, K\}$

the learner gets a reward  $g_{I_t,t} \in \{0, 1\}$  (for simplicity)

the learner updates the model according to the reward

# Adversarial Setting

The rewards are chosen by an opponent who

- Knows the algorithm used by the learner
- Chooses all the rewards in advance

In this scenario, the regret we use is against the best constant choice among the ones we have

## Definition: Adversarial Regret

Given an algorithm  $A$ , selecting an arm  $I_t$  at round  $t$  the Regret of  $A$  over a time horizon of  $n$  rounds is:

$$R_n(A) = \min_{i \in \{1, \dots, N\}} \sum_{t=1}^n E[g_{i,t} - g_{I_t,t}]$$

where the expectation is taken w.r.t. the stochasticity of the algorithm  $A$

# Link to the Expert setting

- We have a discrete prediction space with a finite number of options
- We have a generic loss function

The idea is to use a modified version of the REWA forecaster updating only the observed loss (or gain)

## Exp3 Algorithm

initialize the arms weights  $p_{i,1} = \frac{1}{K}$  and cumulated reward  $G_{i,1} = 0$

at each round  $t$

play an arm  $I_t$  according to  $p_{1,t}, \dots, p_{K,t}$

update  $G_{I_t,t} \leftarrow G_{I_t,t-1} + \frac{g_{I_t,t}}{p_{I_t,t}}$

update the probability distribution for each arm  $p_{i,t+1} \leftarrow \frac{\exp(\eta_t G_{i,t})}{\sum_{j=1}^K \exp(\eta_t G_{j,t})}$

# Analysis of the Exp3 Algorithm

## Theorem

The Exp3 algorithm applied to an adversarial MAB problem with  $K$  arms and

parameter  $\eta_t = \sqrt{\frac{2 \log K}{nK}}$  suffers a regret of:

$$R_n \leq \sqrt{2nK \log K}$$

If the Exp3 algorithm is run with parameter  $\eta_t = \sqrt{\frac{\log N}{tK}}$ , it suffers a regret of:

$$R_n \leq 2\sqrt{nK \log K}$$

Note the dependence on the number of arms w.r.t. the expert case (number of experts)

Both results holds in expectation w.r.t. the stochasticity of the algorithm

This is usually called **pseudo-regret**



# Lower Bound on the Regret

## Theorem

Let  $\sup$  be the supremum over all distribution of rewards such that, for  $i \in \{1, \dots, K\}$  the rewards  $Y_{i,t} \in \{0, 1\}$  are i.i.d., and let  $\inf$  be the infimum over all the forecasters, we have

$$\inf \sup \left( \max_{i \in \{1, \dots, K\}} E \sum_{t=1}^n Y_{i,t} - E \left[ \sum_{t=1}^n Y_{I_t,t} \right] \right) \geq \frac{1}{20} \sqrt{nK}$$

where the expectation is w.r.t. the random generation of the reward and of the forecaster

If we compare it with the previous results, we notice that the orders are optimal except for logarithmic factors and constants

# A Stronger Result

We would like to have results in high probability

This cannot be guaranteed since some arms might be selected with an arbitrarily small probability indefinitely: we need to randomize more than ever...

Exp3.P Algorithm

initialize the arms weights  $p_{i,1} = \frac{1}{K}$  and cumulated reward  $G_{i,1} = 0$

at each round  $t$

play an arm  $I_t$  according to  $p_{1,t}, \dots, p_{K,t}$

update  $G_{I_t,t} \leftarrow G_{I_t,t-1} + \frac{g_{I_t,t} + \beta}{p_{I_t,t}}$  and  $G_{i,t} \leftarrow G_{i,t-1} + \frac{\beta}{p_{i,t}}$  for  $i \neq I_t$

update the probability distribution for each arm

$$p_{i,t+1} \leftarrow (1 - \gamma) \frac{\exp(\eta_t G_{i,t})}{\sum_{j=1}^K \exp(\eta_t G_{j,t})} + \frac{\gamma}{K}$$

# Regret of the Exp3.P Algorithm

The results provided by the previous bound are unimprovable, except for logarithmic factors

## Theorem

The Exp3.P algorithm applied to an adversarial MAB problem with  $K$  arms and

parameters  $\eta = \sqrt{\frac{\log(K \delta^{-1})}{nK}}$ ,  $\gamma = 0.95 \sqrt{\frac{\log K}{nK}}$  and  $\beta = \sqrt{\frac{K \log K}{n}}$  suffers a regret of:

$$R_n \leq 5.15 \sqrt{nK \log(K \delta^{-1})}$$

with probability at least  $1 - \delta$

# Multi-armed Bandit

Stochastic Setting

# A new Perspective

The environment is stochastic and stationary, but the learner receives a partial feedback on its decision

Example: a multiple-class classification problem in which we receive only the information on the correctness of the prediction



Prediction  $I_t = \text{cat}$   
Reward  $g_{I_t,t} = 1$  (YES)



Prediction  $I_t = \text{cat}$   
Reward  $g_{I_t,t} = 0$  (NO)

We might have a set of samples from which to learn (**Learning From Bandit Feedback** problem), but in most the cases we do not have prior information on the problem

# Examples of Stochastic MAB Problems

## **Clinical Trial**

- Exploration: Try new treatments
- Exploitation: Choose the treatment that provides the best results

## **Slot machine** (a.k.a. one-armed bandit) selection

- Exploration: Try all the available slot machines
- Exploitation: Pull the one which provided you the highest payoff so far

## **Game Playing**

- Exploration: Play an unexpected move
- Exploitation: Play the move you think is the best

## **Advertisement**

- Exploration: Present a new ad to an Internet user
- Exploitation: Display the most profitable ad you displayed so far to the user

# Stochastic MAB Setting

At each round  $t \in \{1, \dots, n\}$

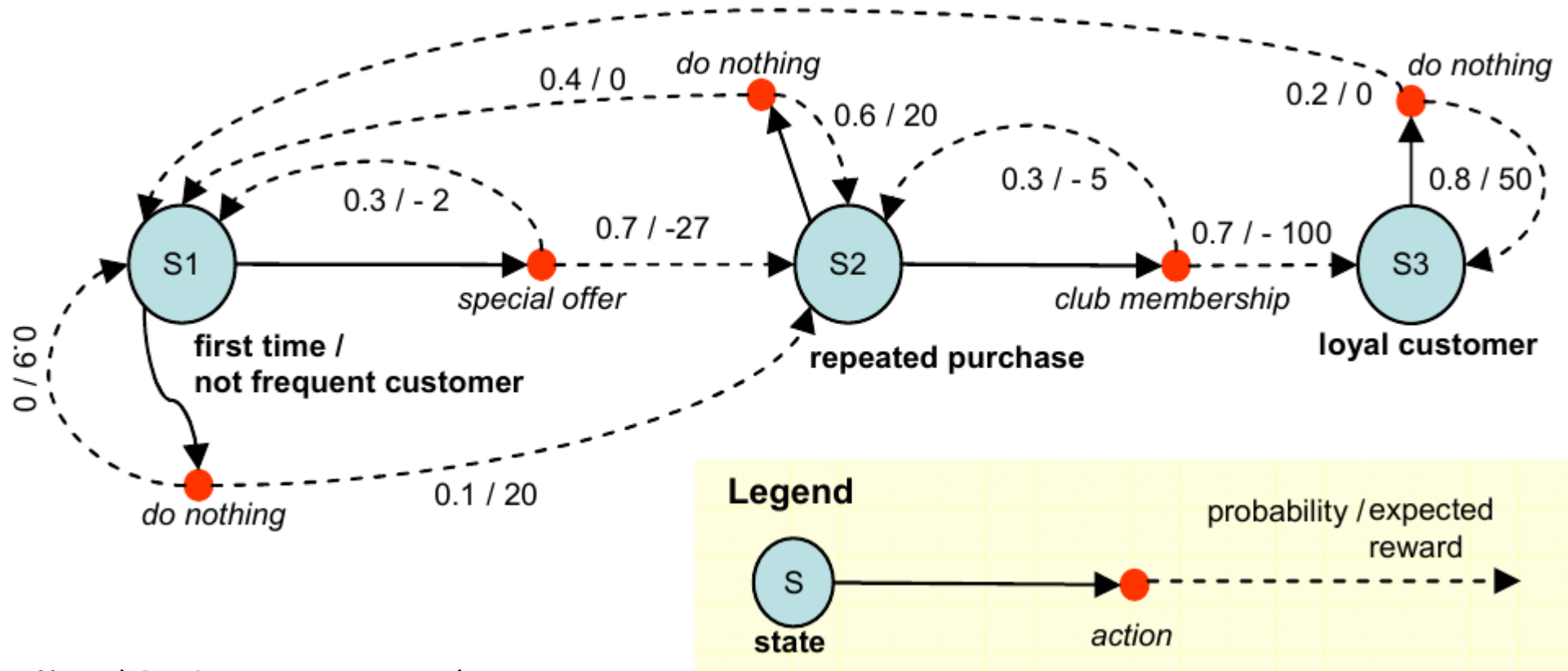
the learner chooses  $I_t \in \{1, \dots, K\}$

**the learner gets a reward  $g_{I_t,t} \sim \mathbf{D}_{I_t}$**  (distribution over support  $\Omega$ )

the learner updates the model according to the reward

The problem seems to be less complex than before: the rewards are drawn from fixed distributions

# Markov Decision Problems



Formally:  $\langle S, A, P, R, \gamma, \mu_0 \rangle$

Powerful tools to model phenomena involving temporal dependence

**Markov assumption:** the behaviour depends only on the current state



# MDP to MAB Mapping

We can see the Multi-Armed Bandit setting as a specific case of an MDP

$$\langle S, A, P, R, \gamma, \mu_0 \rangle$$

$S$  is a set of states (single state)

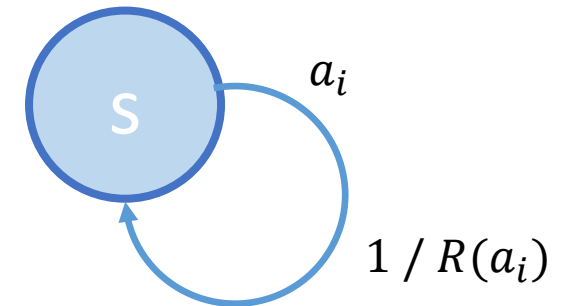
$A$  is a set of actions (arms)  $\{1, \dots, K\}$

$P$  is a state transition probability matrix

$R$  is a reward function  $R(s, a_i) = R(a_i)$

$\gamma$  is a discount factor (for finite time horizon  $\gamma = 1$ )

$\mu_0$  is a set of initial state probabilities



Any strategy for solving sample-based MDPs (RL) is also a strategy to solve MAB

# Stochastic Regret

Define:

- Optimal arm

$$i^* = \arg \max_i E[g_{i,1}]$$

- Expected reward

$$\mu_i = E[g_{i,1}]$$

- Optimal expected reward

$$\mu^* = \mu_{i^*}$$

## Definition: Stochastic Regret

Given an algorithm  $A$ , selecting an arm  $I_t$  at round  $t$  the Regret of  $A$  over a time horizon of  $T$  rounds is:

$$R_n(A) = \sum_{t=1}^n [\mu^* - E(\mu_{I_t})]$$

where the expectation is w.r.t. the forecaster stochasticity

# Rewriting the Regret

Define:

- $\Delta_i = \mu^* - \mu_i$  the expected difference between the reward of the optimal arm and the  $i$ -th arm
- $N_{i,t}$  the number of times we selected the arm  $i$  after  $t$  rounds

$$R_n(A) = \sum_{t=1}^n [\mu^* - E(\mu_{I_t})] = \sum_{t=1}^n E(\mu^* - \mu_{I_t}) = \sum_{i=1}^K \Delta_i E(N_{i,n})$$

We would like to limit the number of times we select a suboptimal arm (common strategy used in the upper bound proofs)

This also implies that, if we want to have a sublinear regret, we should play the suboptimal arms a sublinear number of times ( $E[N_{i,n}] = O(n^\alpha)$  with  $\alpha < 1$ )

# Lower Bound

## Theorem: Stochastic MAB Lower Bound

Let  $A$  be a strategy s.t.  $E[N_{i,n}] = O(n^\alpha)$  for any set of Bernoulli distributions, any arm s.t.  $\Delta_i > 0$ , and  $\alpha > 0$ . Then for any set of Bernoulli distributions  $D_i$  the following holds:

$$\liminf_{n \rightarrow +\infty} \frac{R_n(A)}{\log n} \geq \sum_{i, \Delta_i > 0} \frac{\Delta_i}{KL(\mu_i, \mu^*)}$$

Where  $KL(\cdot, \cdot)$  denotes the Kullback-Leibler divergence between two Bernoulli distributions

- $KL(p, q) = p \log \frac{p}{q} + (1 - p) \log \frac{1-p}{1-q}$
- Can we design an algorithm for which  $\alpha = 0$ ?

NO: we would have a small probability to fail and an infinite time to regret it

# Frequentist vs. Bayesian Solution Approach

Two different formulation for the problem

- Frequentist formulation

$\mu_1, \dots, \mu_K$  are unknown parameters we want to somehow estimate

a policy selects at each time step an arm based on the observation history

- Bayesian formulation

$\mu_1, \dots, \mu_K$  are random variables with prior distributions  $\pi_1, \dots, \pi_K$

a policy selects at each time step an arm based on the observation history

and the provided priors (so based on the posteriors)

# Greedy Algorithm

Select the next arm to play according to the empirical expected value of the reward of the arms, formally:

$$I_t = \arg \max_{i \in \{1, \dots, K\}} \frac{\sum_{h=1}^t g_{i,h} 1\{I_h = i\}}{\sum_{h=1}^t 1\{I_h = i\}}$$

Theorem: Greedy algorithm Regret

The Greedy algorithm suffers  $O(n)$  regret

Proof: by counterexample if the optimal arm, at the first pull provides a null reward while at least one other arm provides a non-zero reward

Problem: we are not considering **the uncertainty** over the arm reward estimates

# Optimism in the Face of Uncertainty

Small  $N_{i,t}$  corresponds to a large uncertainty on the reward estimates

Large  $N_{i,t}$  corresponds to a small uncertainty on the reward estimates

We want to use the concentration inequalities like the following one

## Theorem: Mean of Gaussian distributions

Let  $X_1, \dots, X_t$  be i.i.d. Gaussian random variables with mean  $E[X_i] = \mu$  and variance  $\sigma^2$ . Let  $\bar{X} = \sum_i \frac{X_i}{t}$  be the sample mean. Then for each  $\epsilon > 0$ :

$$P\left(\frac{|\bar{X} - \mu|}{\frac{\sigma}{\sqrt{t}}} \geq z_{1-\frac{\alpha}{2}}\right) \leq \alpha$$

# UCB1: Genesis

## Theorem: Hoeffding Bound

Let  $X_1, \dots, X_t$  be i.i.d. random variables with support in  $[0,1]$  and mean  $E[X_i] = \mu$  and let  $\bar{X} = \sum_i \frac{X_i}{t}$  be the sample mean. Then for each  $\epsilon > 0$ :

$$P(\mu > \bar{X} + \epsilon) \leq \exp(-2t \epsilon^2)$$

We would like to include the information about the uncertainty using the confidence bound

Instead of selecting the arm with the largest sample mean  $\bar{g}_{i,t}$  we use an optimistic estimate of the real mean  $\mu_i$ :

$$u_{i,t} = \bar{g}_{i,t} + b_{i,t}$$



# Computing the Upper Bounds

Let us choose the probability  $p$  that an arm is exceeding the bound at time  $t$  in the case we pulled it for  $N_{i,t}$  times:

$$P(\mu_i > \bar{g}_{i,t} + b_{i,t}) \leq \exp(-2N_{i,t} b_{i,t}^2) = p(t)$$

The bound becomes

$$b_{i,t} = \sqrt{\frac{-\log p(t)}{2N_{i,t}}}$$

Overall, we would like that the algorithm makes a limited number (constant) of mistakes for the  $i$ -th arm for each:

- Round  $t \in \{1, \dots, n\}$
- Possible number of pulls  $N_{i,t} \in \{1, \dots, t\}$

# Choice of the Probability

Therefore, we would like to have:

$$\sum_{t=1}^n \sum_{N_{i,t}}^t p(t) = \text{const}$$

Choosing  $p(t) = \frac{1}{t^4}$  the summation leads to  $\frac{\pi^2}{6}$  and the bound becomes:

$$b_{i,t} = \sqrt{\frac{2 \log t}{N_{i,t}}}$$

This is an exploration term that:

- Decreases as we pull an arm
- Increases if the arm is not pulled

# UCB1

at each round  $t$

play the arm  $I_t$  having the largest  $u_{i,t} = \bar{g}_{i,t} + \sqrt{\frac{2 \log t}{N_{i,t}}}$

get reward  $g_{I_t,t}$

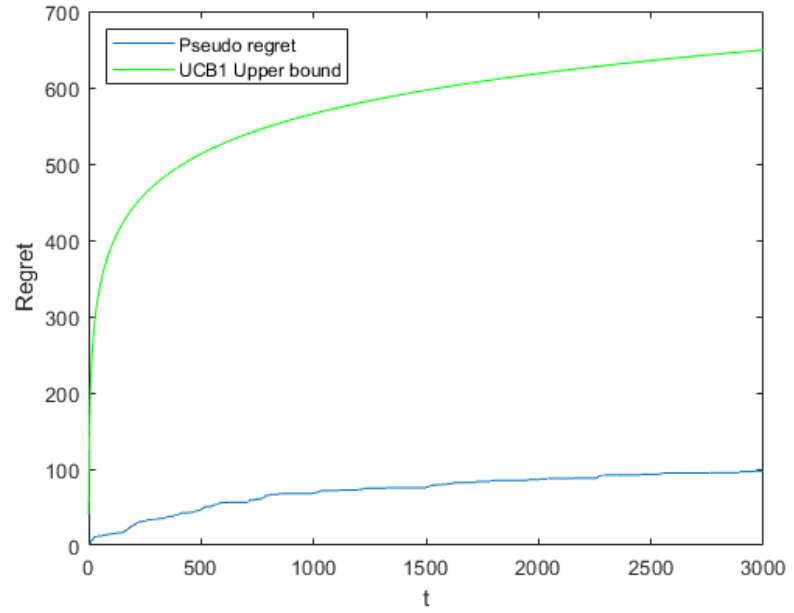
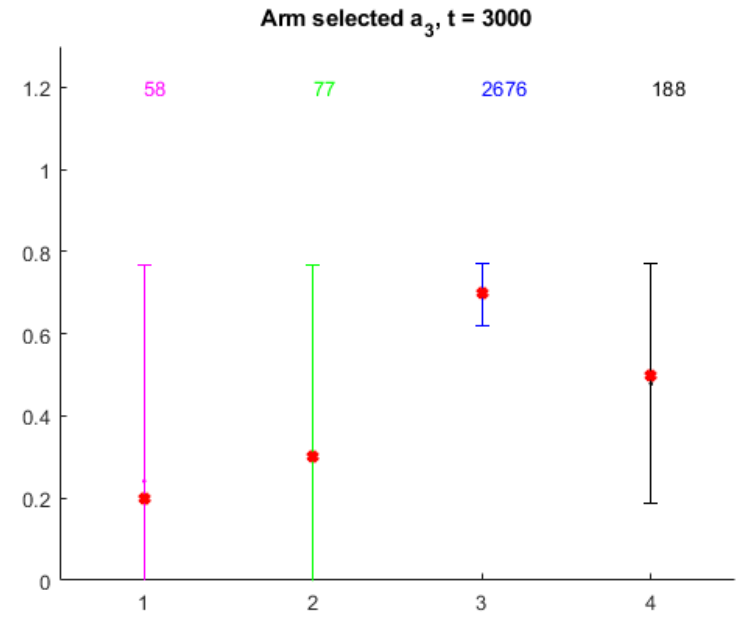
update the sample mean  $\bar{g}_{I_t,t}$  and the bounds for all the arms

## Theorem

The UCB1 algorithm applied to a stochastic MAB problem with  $K$  arms suffers a regret of:

$$R_n \leq 8 \log n \sum_{i, \Delta_i > 0} \frac{1}{\Delta_i} + \left(1 + \frac{\pi^2}{3}\right) \sum_{i, \Delta_i > 0} \Delta_i$$

# Execution Example



# Comment on the UCB1 Bound

- The bound is distribution dependent (computed using  $\Delta_i$ ) (the equivalent bound without the dependence on the distribution is  $O(\sqrt{n})$ )
- The difficulty of the problem seems to be intrinsically dependent on the  $\Delta_i$ , since the more they are small, the larger the bound is
- Up to a constant, the bound is matching the lower bound, so UCB1 is an optimal algorithm

Can we do better?

Yes: on specific MAB we might improve over the lower bound

Yes: on specific runs we might even get null regret

# Other UCB-base Algorithms

- UCB2: instead of changing the arm at each round, we use increasingly large epochs, during which the arm chosen is still the same
- UCBV: using the Bernstein's bound we include the information about the variance of the rewards
- KL-UCB: use explicitly the Kullback-Leibler divergence to compute the upper confidence bound (requires the solution of an optimization problem for each arm for each round)
- Bayes-UCB: use Beta distributions to determine the bounds
- ...

# Bayesian Approach

Different approach:

- Each expected reward  $\mu_i$  is associated with a distribution
- We update the distribution as soon as we have a new reward from the arm

Requires the update of a Bayesian prior, using the likelihood of the reward, at each round, which in principle is a computationally complex operation

We rely on conjugate prior/posterior:

- Gaussian / Gaussian (known variance)
- Gamma / Gaussian (known mean)
- **Beta / Bernoulli**

# Beta Distribution

$$Beta(\alpha, \beta)$$

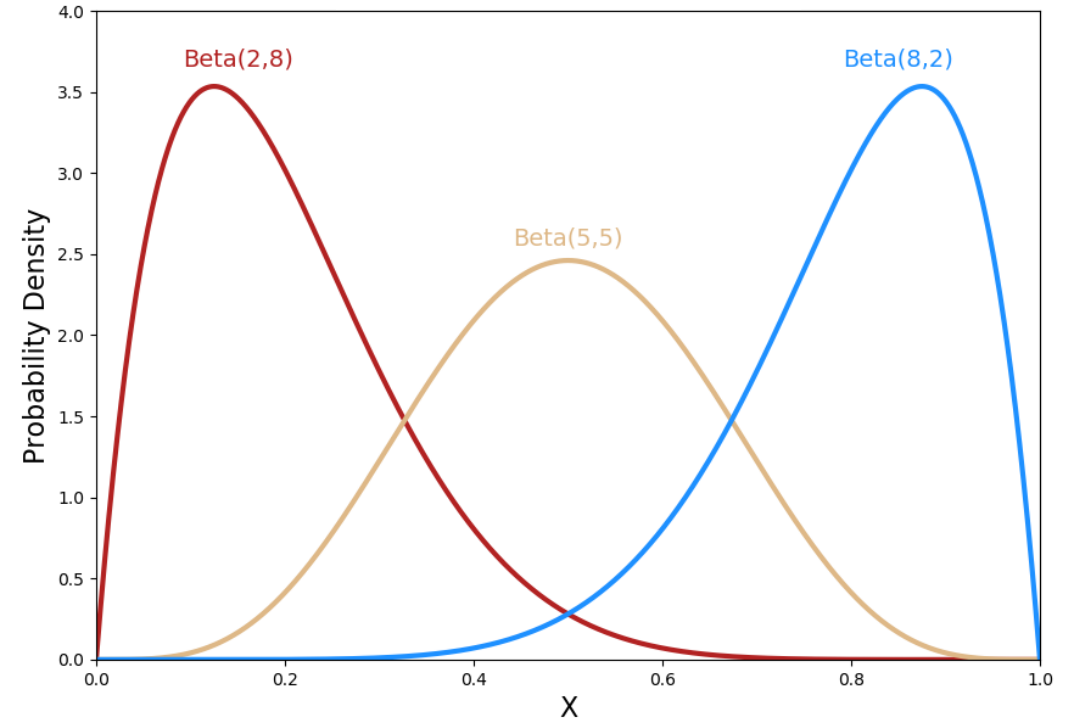
Domain:  $\Omega = [0,1]$

$$\text{Pdf: } p(x, \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)}$$

Expected value:  $\frac{\alpha}{\alpha+\beta}$

$$\text{Variance: } \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

Quantiles: computed numerically





# Thompson Sampling

set a prior  $\pi_{i,1} = \text{Beta}(1,1)$  for each arm  $i$

at each round  $t$

select a sample  $\theta_i$  from each distribution  $\pi_{i,t}$

play the arm  $I_t$  having the largest  $\theta_i$

get reward  $g_{I_t,t}$

update the parameters of the prior corresponding to the arm  $I_t$ :

- increasing  $\alpha$  by one if we observed a success
- increasing  $\beta$  by one if we observed a failure

# Regret for the TS algorithm

## Theorem

The TS algorithm applied to a stochastic MAB problem with  $K$  arms suffers a regret of:

$$R_n \leq (1 + \epsilon) \sum_{i, \Delta_i > 0} \frac{\Delta_i (\log n + \log \log n)}{KL(\mu_i, \mu^*)} + C(\epsilon, \mu_1, \dots, \mu_K)$$

for each  $\epsilon > 0$

It is asymptotically optimal, even if we look at the constants

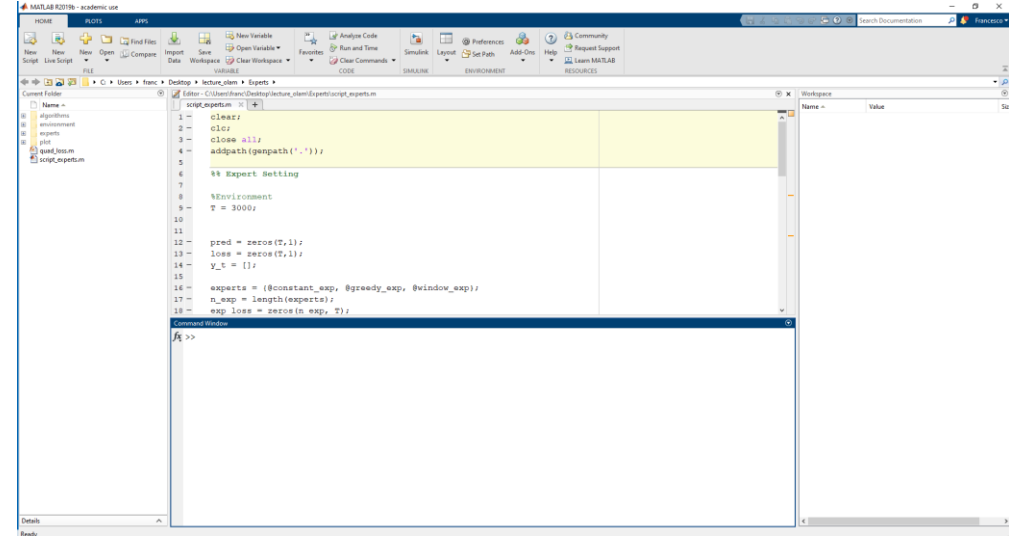
The algorithm was designed by Thompson in 1933

The proof for this theorem have been provided independently by Kaufmann and Agrawal in 2012

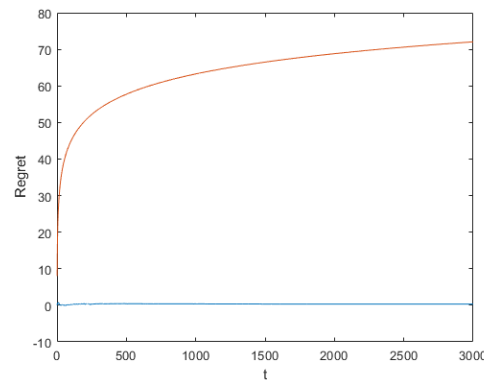
# Matlab Exercise

Given a MAB environment:

- Implement the UCB1 algorithm
- Implement the TS algorithm



Draw the regret for both algorithms and compare it with their upper bounds



# Multi-armed Bandit

Beyond Stochastic Settings

# MAB Algorithm Applicability

The problem solved with the previous algorithm is an extreme simplification of the real settings

- Contextual MAB
- Correlated rewards
- Large/Infinite Number of Arms
- Non-stationarity of the environment
- Truthful Bandit
- ...

# Contextual Bandits

We are provided with side information for each round

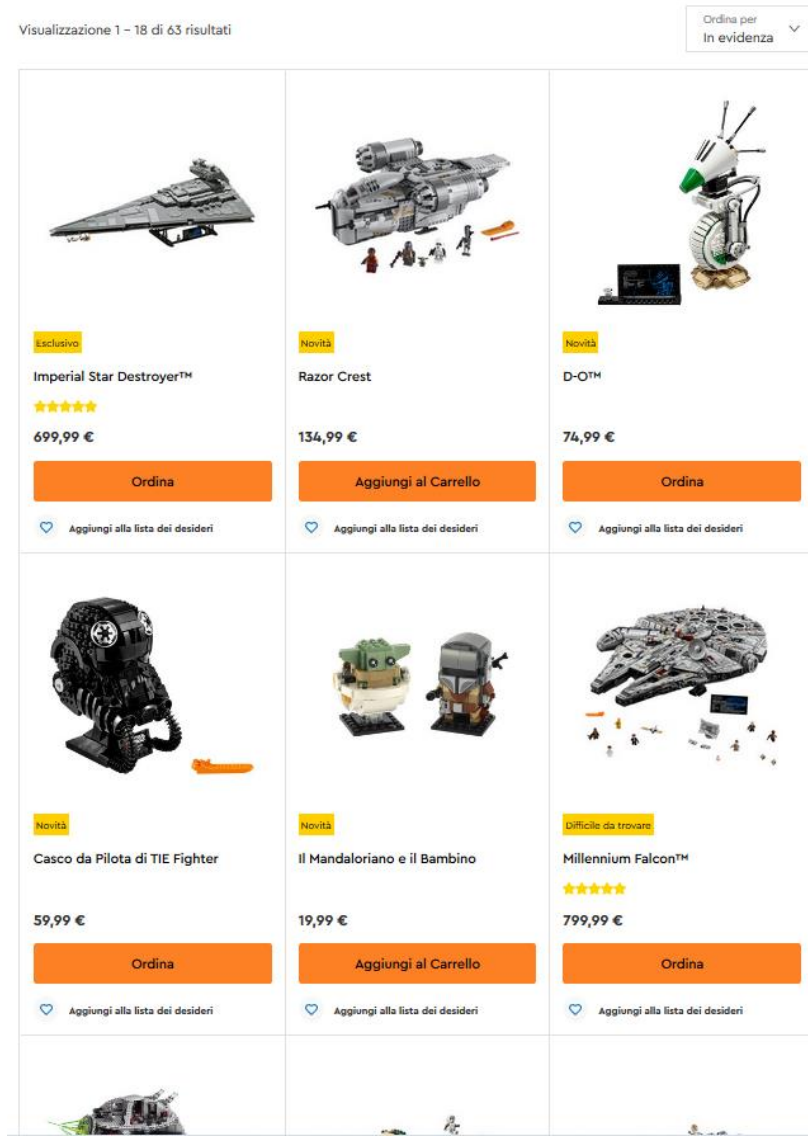
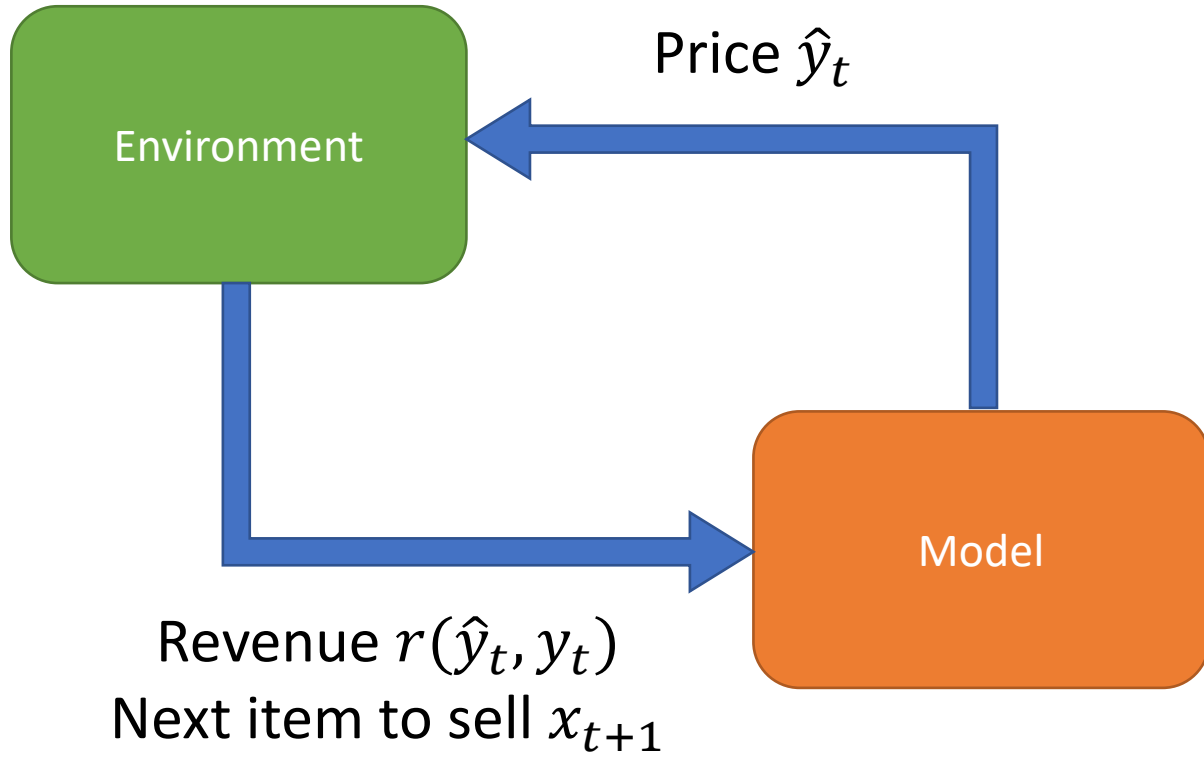
Example: personalized news article recommendation in which the task is to select, from a pool of candidates, a news article to display whenever a new user visits a website

- Arms: different news
- Context: information about the user
- Reward: click on the news

We require to have some structure on the reward function, otherwise we are solving a MAB for each context

# Pricing as an Online Learning Problem

$$r(\hat{y}_t, y_t) = 1 - l(\hat{y}_t, y_t)$$



# Linear Bandit Problem

In this setting for each round  $t$

- play an arm from  $X_t \in \mathbb{R}^d$
- get reward  $Y = \langle X_t, \theta \rangle + \eta_t$  where  $\eta_t$  is a random noise

The arms rewards are correlated by the unknown parameter  $\theta$

Idea:

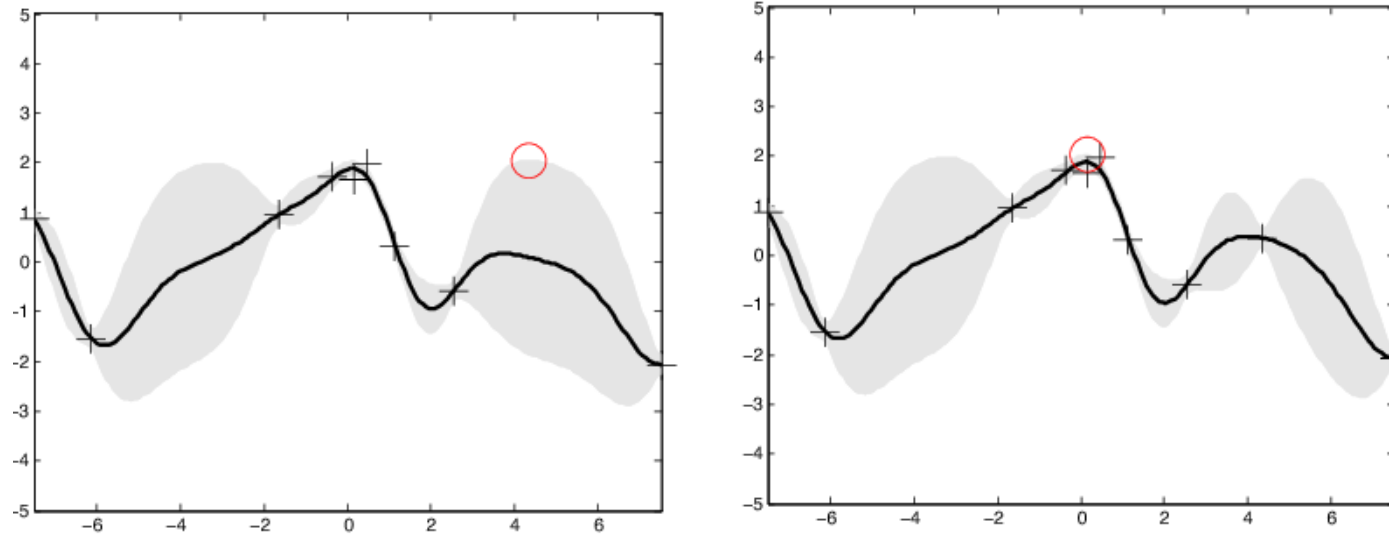
- Compute the best possible estimates of the parameter  $\theta$  using the LS method
- Add a bound holding for all the  $d$  dimensions at the same time



# Correlated Arms

What if the correlation is not linear and has a more complex structure?

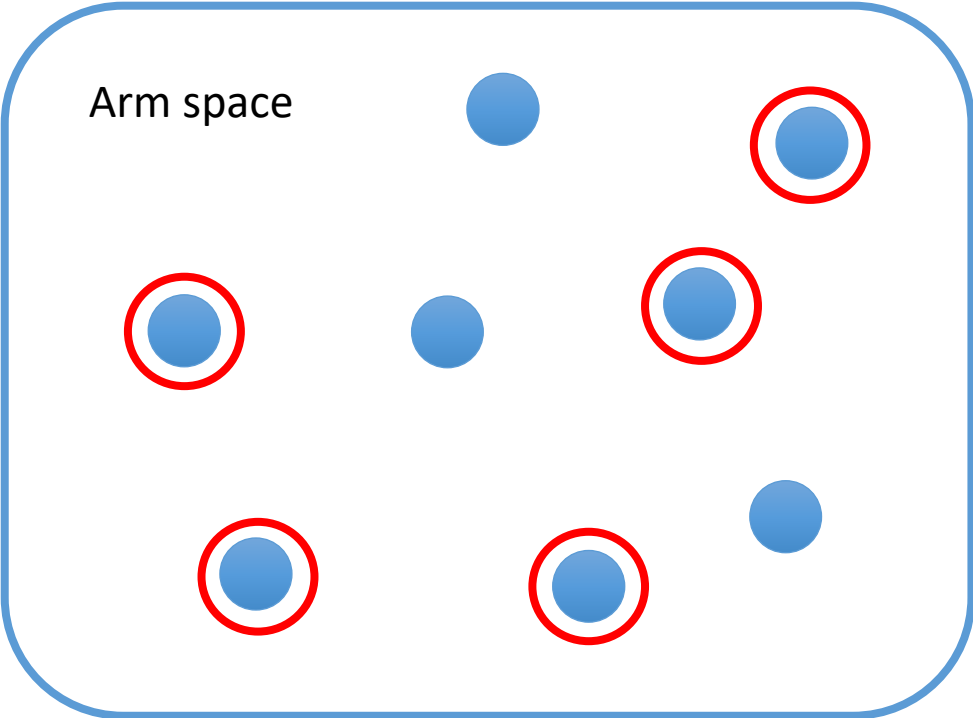
GP-MAB: the reward structure is determined by a Gaussian Process Kernel




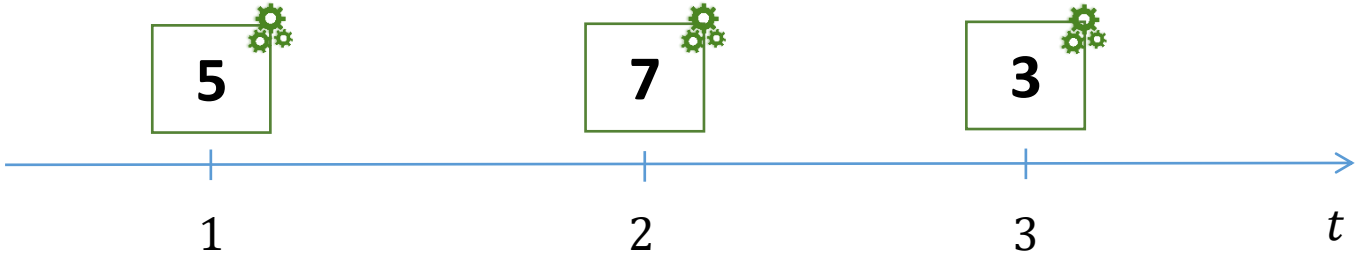
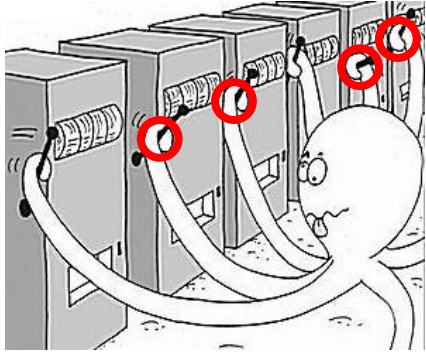
GP-UCB: selects the arm with the largest value using an upper bound computed using the GP (using the estimated mean and variance)

GP-TS: generates samples from the GP and selects the arm providing the largest one

# Combinatorial Bandit Problem



Optimization problem 



# Continuous Bandit Problem

The arm space is a compact in  $\mathbb{R}^d$  and the expected reward is a function  $\mu: \mathbb{R}^d \rightarrow \mathbb{R}$

Similar problem to non-convex optimization, with a different goal

- Optimization: get to a solution as close as possible to the maximum in the smallest **number of rounds**
- Bandit: minimize the times we are selecting suboptimal arm to maximize the cumulative reward over rounds

If the function does not have any regularity the task is unfeasible

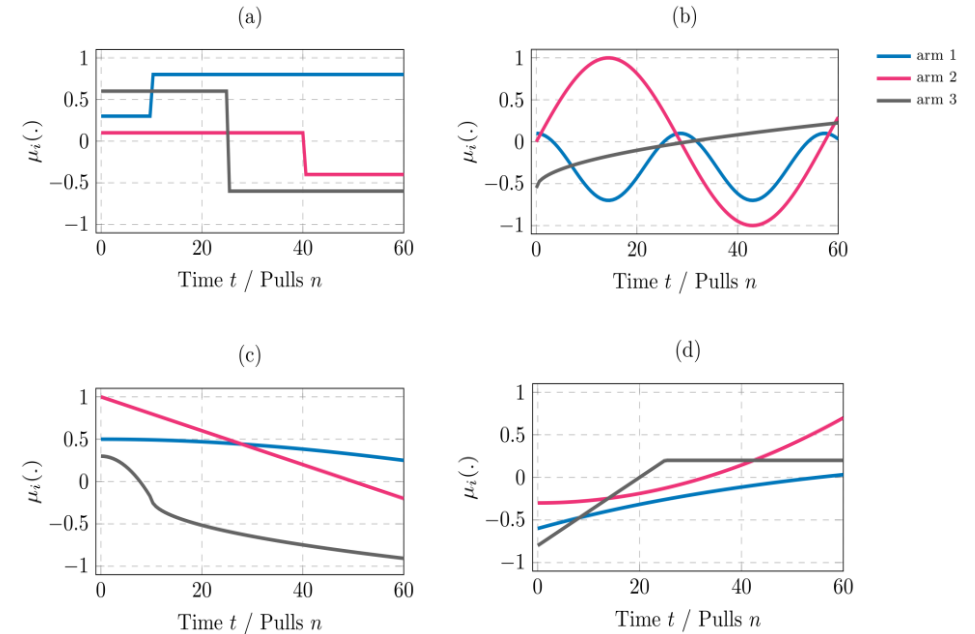
Regularity assumptions:

- Lipschitz bandit:  $|\mu(\mathbf{x}) - \mu(\mathbf{y})| \leq L \|\mathbf{x} - \mathbf{y}\|$
- GP bandit

# Non-Stationary Bandits

The arms reward might change:

- Abruptly (single arm)
- Abruptly (all the arms at a time)
- Smoothly



Passive approaches for abrupt changes:

- SW-UCB: selects only the sample in a prespecified window (with  $O(\sqrt{n})$  rounds) and select according to them
- $\delta$ -UCB: discount the reward we got in the past by a factor of  $\delta$

Passive approaches for abrupt and smooth changes:

- SW-TS: selecting only the samples in a prespecified window

# Truthful Bandit



Problem of ad placement:

we would like to allocate some ads, but the allocation depends on:

- Unknown parameters we would like to estimate
- The report of a bid value from the advertiser

This learning process is dependent on the choice of (possibly adversarial) advertisers:  
combination of Online Learning and Mechanism Design

Solution: divide the exploration and exploitation phases

- Exploration: round robin
- Exploitation: use the best option we found so far

# Beyond Multi-Armed Bandit

# Partial Monitoring

The feedback is not the reward, but a quantity correlated to the reward and a cost associated to the action

Example: when we are selling a product:

- we have only an information if the item has been purchased
- no information about the buyer threshold (largest price still feasible to sell the product)
- we do not know the loss we incurred

# Best Arm Identification

The target is to identify the arm providing the largest expected reward among the available ones:

- Selection criterion
- Stopping criterion
- Final guess

Solutions:

- Select according to UCB-like algorithm, up to the round in which we have at least a **fixed confidence** it is the optimal one
- Use a round robin selection criterion, dividing into epochs the **fixed time horizon**, and eliminate one arm at the end of each epoch



# Online Security Games

Physical Security problem



A cop (defender) should select the targets with having the most chance to be attacked by a burglar (attacker)

We need to determine the strategy profile of the attacker (e.g., rational, partial rationality, stochastic)

In the morning we have a feedback about each target

# Cops and Robbers

Special feedback: the learner observes the losses associated with all actions except the played action

Except for constant factors, this problem is no harder than the full information setting where all losses are observed

The minimax regret of cops and robbers satisfies:

$$R_n \leq \sqrt{2n \log(K)}$$

# Bibliography

B. Sébastien, N. Cesa-Bianchi. “Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems.” *Machine Learning* 5.1 (2012): 1-122.

T. Lattimore, C. Szepesvári. “Bandit algorithms.” Cambridge University Press, 2020

J.Y. Audibert, S. Bubeck, R. Munos., “Best arm identification in multi-armed bandits.” In Proceedings of the Conference on Learning Theory (COLT), 2010

A. Nuara, F. Trovò, N. Gatti, M. Restelli. “A Combinatorial-Bandit Algorithm for the Online Joint Bid / Budget Optimization of Pay-per-Click Advertising Campaigns.” In the Proceedings of the Conference on Artificial Intelligence (AAAI), 2018

L. Bisi, G. De Nittis, F. Trovò, M. Restelli, N. Gatti. “Regret Minimization Algorithms for the Follower’s Behaviour Identification in Leadership Games.” In the Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI), 2017