

# Sequential Learning

## Lecture 7 : Structured bandits

Rémy Degenne  
(remy.degenne@inria.fr)



Université  
de Lille



Centrale Lille, 2025/2026

# Recap from last class

Several important ideas to tackle the **exploration/exploitation challenge** in a simple multi-armed bandit model with independent arms :

- ▶ Explore then Commit
- ▶  $\varepsilon$ -greedy
- ▶ Optimistic algorithms : Upper Confidence Bounds strategies
- ▶ Bayesian algorithms : Thompson Sampling

Some of these can be extended to more realistic **structured** models that are suited for different applications.

# More about UCB : worst case bound

**Context :**  $\sigma^2$  sub-Gaussian rewards

$$\text{UCB}_a(t) = \hat{\mu}_a(t) + \sqrt{\frac{8\sigma^2 \log(t)}{N_a(t)}}$$

## Theorem

The UCB algorithm associated to the above index satisfies

$$\mathbb{E}[N_a(T)] \leq \frac{32\sigma^2}{(\mu_\star - \mu_a)^2} \log(T) + 2.$$

if the rewards distributions are  $\sigma^2$  sub-Gaussian.

## Regret bounds :

- ▶ Distribution-dependent :  $\mathbb{E}[R_T] \leq \sum_{a: \Delta_a > 0} \frac{32\sigma^2}{\Delta_a} \log(T) + 2\Delta_a.$
- ▶ Worst-case :  $\mathbb{E}[R_T] \leq C\sqrt{KT \log T}.$

# Outline

## 1 Contextual Bandits

## 2 Solving Linear Bandits

- Lin-UCB
- Linear Thompson Sampling

## 3 Other variants of the classical MAB

# Contextual Bandits

## Example : movie recommendation



What movie should Netflix recommend to a particular user, given the ratings provided by previous users ?

- to make good recommendation, we should **take into account the characteristics of the movies / users**

**Contextual bandit problem** : at time  $t$

- ▶ a context  $c_t$  is observed
- ▶ an arm  $A_t$  is chosen
- ▶ a reward  $R_t$  that depends on  $c_t, A_t$  is received.

# RL as a contextual bandit

## RL can be cast as a contextual bandit

- ▶ a context  $c_t$  is observed  $\rightarrow c_t = s_t$ , the state
- ▶ an arm  $A_t$  is chosen
- ▶ a reward  $R_t$  that depends on  $c_t, A_t$  is received. reward depends on state and action

# RL as a contextual bandit

## RL can be cast as a contextual bandit

- ▶ a context  $c_t$  is observed  $\rightarrow c_t = s_t$ , the state
- ▶ an arm  $A_t$  is chosen
- ▶ a reward  $R_t$  that depends on  $c_t, A_t$  is received. reward depends on state and action

## But you should not do that

- ▶ All information about transitions is lost !
- ▶ No link between successive contexts

# Independent bandits

**Contextual bandit problem** : at time  $t$

- ▶ a context  $c_t$  is observed
- ▶ an arm  $A_t$  is chosen
- ▶ a reward  $R_t$  that depends on  $c_t, A_t$  is received.

**Idea** : 1 context = 1 bandit

- ▶ Run many bandit algorithms, one per context
- ▶ Regret depends on the number of context  $\rightarrow$  not practical

We have to **link different context together**, to avoid the independent bandits situation.



# Mixing bandits and regression models

A **contextual bandit model** incorporates two components :

- ▶ a sequential interaction protocol :  
pick an arm, receive a reward
- ▶ a **regression model** for the dependency between context and reward

# Mixing bandits and regression models

A (stochastic) **contextual bandit model** incorporates two components :

- ▶ a sequential interaction protocol :  
pick an arm, receive a (random) reward
- ▶ a **regression model** for the dependency between context and reward

# Mixing bandits and regression models

A **(stochastic) contextual bandit model** incorporates two components :

- ▶ a sequential interaction protocol :  
pick an arm, receive a **(random)** reward
- ▶ a **regression model** for the dependency between context and reward

## General stochastic contextual bandit model

In each round  $t$ , the agent

- ▶ observes a context  $c_t \in \mathcal{C}$  *(user characteristics)*
- ▶ selects an arm  $A_t \in \mathcal{A}_t$  *(an item out of a possibly changing pool)*
- ▶ the agent receives a reward

$$r_t = f_{A_t}(c_t) + \varepsilon_t$$

where  $\varepsilon_t$  is an independent noise :  $\mathbb{E}[\varepsilon_t] = 0$ .

$f_a : \mathcal{C} \rightarrow \mathbb{R}$  maps a context  $c$  to the average reward of arm  $a$ ,  $f_a(c)$

# Examples

## Example 1

- user  $t$  : descriptor  $c_t \in \mathbb{R}^p$
- item  $a$  : descriptor  $\theta_a \in \mathbb{R}^p$

$$r_t = \theta_{A_t}^\top c_t + \varepsilon_t$$

Linear function  $f_a(c) = \theta_a^\top c$

**Observation** : if  $\mathcal{A}_t = \{1, \dots, K\}$  is a fixed set of items

- ▶ the model is parameterized by  $\theta_1, \theta_2, \dots, \theta_K \in (\mathbb{R}^p)^K$
- ▶ it can also be rewritten  $r_t = \theta_\star^\top (x_{t,A_t}) + \varepsilon_t$  with

$$\theta_\star = \begin{pmatrix} \theta_1 \\ \dots \\ \theta_a \\ \dots \\ \theta_K \end{pmatrix} \in \mathbb{R}^{p \times K}, \quad x_{t,a} = \begin{pmatrix} 0 \\ \dots \\ c_t \\ \dots \\ 0 \end{pmatrix} \in \mathbb{R}^{p \times K}$$

$x_{t,a}$  : feature vector for the user-item pair  $(t, a)$

# Examples

## Example 2

- user  $t$  : descriptor  $c_t \in \mathbb{R}^p$
- item  $a$  : descriptor  $x_a \in \mathbb{R}^{p'}$
- build a user-item feature vector for  $(t, a) : x_{t,a} \in \mathbb{R}^d$   
(feature engineering)

$$r_t = \theta_\star^\top x_{t,A_t} + \varepsilon_t$$

### Observation :

- ▶ the model is parameterized by  $\theta_\star \in \mathbb{R}^d$
- ▶ in each round  $t$ , the user-item feature vectors belong to the set

$$\mathcal{X}_t = \{x_{t,a}, a \in \mathcal{A}_t\} \subseteq \mathbb{R}^d$$

- ▶ picking an arm  $a \leftrightarrow$  picking a feature vector  $x_t \in \mathcal{X}_t$

$$r_t = \theta_\star^\top x_t + \varepsilon_t$$

# Examples

## Example 2

- user  $t$  : descriptor  $c_t \in \mathbb{R}^p$
- item  $a$  : descriptor  $x_a \in \mathbb{R}^{p'}$
- build a user-item feature vector for  $(t, a) : x_{t,a} \in \mathbb{R}^d$   
(feature engineering)

$$r_t = \theta_\star^\top x_{t,A_t} + \varepsilon_t$$

### Observation :

- ▶ the model is parameterized by  $\theta_\star \in \mathbb{R}^d$
- ▶ in each round  $t$ , the user-item feature vectors belong to the set

$$\mathcal{X}_t = \{x_{t,a}, a \in \mathcal{A}_t\} \subseteq \mathbb{R}^d$$

- ▶ picking an arm  $a \leftrightarrow$  picking a feature vector  $x_t \in \mathcal{X}_t$

$$r_t = f_\star(x_t) + \varepsilon_t$$

# Two formulations

## Contextual MAB, version 1

In each round  $t$ , the agent

- ▶ observes a context  $c_t \in \mathcal{C}$
- ▶ selects an arm  $A_t \in \mathcal{A}_t$  *(set of arm can vary in each round)*
- ▶ the agent receives a reward  $r_t = f_{A_t}(c_t) + \varepsilon_t$

Unknown : regression functions  $(f_a)$  for all possible arm  $a$

## Contextual MAB (more general)

In each round  $t$ , the agent

- ▶ is given a set of arms  $\mathcal{X}_t$  *(can be different in each round)*
- ▶ selects an arm  $x_t \in \mathcal{X}_t$
- ▶ the agent receives a reward  $r_t = f_*(x_t) + \varepsilon_t$

Unknown : regression function  $f_*$

# Two formulations

## Contextual MAB, version 1

In each round  $t$ , the agent

- ▶ observes a context  $c_t \in \mathcal{C}$
- ▶ selects an arm  $A_t \in \mathcal{A}_t$  *(set of arm can vary in each round)*
- ▶ the agent receives a reward  $r_t = f_{A_t}(c_t) + \varepsilon_t$

Unknown : regression functions ( $f_a$ ) for all possible arm  $a$

## Contextual MAB (more general)

In each round  $t$ , the agent

- ▶ is given a set of arms  $\mathcal{X}_t$  *(can be different in each round)*
- ▶ selects an arm  $x_t \in \mathcal{X}_t$
- ▶ the agent receives a reward  $r_t = f_*(x_t) + \varepsilon_t$

Unknown : regression function  $f_*$

→ **Goal** : learn the unknown function  $f_*$ ... while maximizing rewards !



# Outline

1 Contextual Bandits

2 Solving Linear Bandits

- Lin-UCB

- Linear Thompson Sampling

3 Other variants of the classical MAB

# Contextual linear bandits

In each round  $t$ , the agent

- ▶ receives a (finite) set of arms  $\mathcal{X}_t \subseteq \mathbb{R}^d$
- ▶ chooses an arm  $x_t \in \mathcal{X}_t$
- ▶ gets a reward  $r_t = \theta_\star^\top x_t + \varepsilon_t$

where

- $\theta_\star \in \mathbb{R}^d$  is an unknown regression vector
- $\varepsilon_t$  is a centered noise, independent from past data

**Assumption** :  $\sigma^2$ - sub-Gaussian noise

$$\forall \lambda \in \mathbb{R}, \mathbb{E} [e^{\lambda X}] \leq e^{\frac{\lambda^2 \sigma^2}{2}}$$

e.g., Gaussian noise, bounded noise.

# Contextual linear bandits

In each round  $t$ , the agent

- ▶ receives a (finite) set of arms  $\mathcal{X}_t \subseteq \mathbb{R}^d$
- ▶ chooses an arm  $x_t \in \mathcal{X}_t$
- ▶ gets a reward  $r_t = \theta_\star^\top x_t + \varepsilon_t$

where

- $\theta_\star \in \mathbb{R}^d$  is an unknown regression vector
- $\varepsilon_t$  is a centered noise, independent from past data

## (Pseudo)-regret for contextual bandit

maximizing expected total reward  $\leftrightarrow$  minimizing the expectation of

$$R_T(\mathcal{A}) = \sum_{t=1}^T \left( \max_{x \in \mathcal{X}_t} \theta_\star^\top x - \theta_\star^\top x_t \right)$$

→ in each round, comparison to a possibly different optimal action !

# Linear Regression

**Goal** : find estimate  $\hat{\theta}_t$  of  $\theta_*$ , given observations  $(x_1, y_1), \dots, (x_t, y_t)$ , where  $y_s = \theta_*^\top x_s + \varepsilon_s$ .

**Loss function** : squared loss + regularization

**Goal** : find  $\operatorname{argmin}_{\theta} \sum_{s=1}^t (y_s - \theta^\top x_s)^2 + \lambda \|\theta\|^2$ .



# Tools

Algorithms will rely on estimates / confidence regions / posterior distributions for  $\theta_\star \in \mathbb{R}^d$ .

- ▶ design matrix (with regularization parameter  $\lambda > 0$ )

$$B_t^\lambda = \lambda I_d + \sum_{s=1}^t x_s x_s^\top$$

- ▶ regularized least-square estimate

$$\hat{\theta}_t^\lambda = (B_t^\lambda)^{-1} \left( \sum_{s=1}^t r_s x_s \right)$$

**Remark** : easy online update ! use the Sherman-Morrison formula.

- ▶ estimate of the expected reward of an arm  $x \in \mathbb{R}^d$  :  $x^\top \hat{\theta}_t^\lambda$
- sufficient for Follow the Leader, but not for smarter algorithms !

# Outline

1 Contextual Bandits

2 Solving Linear Bandits

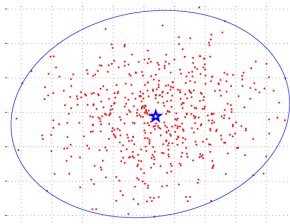
- Lin-UCB

- Linear Thompson Sampling

3 Other variants of the classical MAB

# How to build (tight) confidence interval on the mean rewards ?

**Idea** : rely on a **confidence ellipsoid** around  $\hat{\theta}_t^\lambda$



$$\theta_\star \in \left\{ \theta \in \mathbb{R}^d : \|\theta - \hat{\theta}_t^\lambda\|_A \leq \beta_t \right\}$$

**Why ?** For all invertible matrix positive semi-definite matrix  $A$ ,

$$\forall x \in \mathbb{R}^d, \quad \left| x^\top \theta_\star - x^\top \hat{\theta}_t^\lambda \right| \leq \|x\|_{A^{-1}} \left\| \theta_\star - \hat{\theta}_t^\lambda \right\|_A$$

$$\|x\|_A = \sqrt{x^\top A x}$$

# How to build (tight) confidence interval on the mean rewards ?

**Wanted :**  $\theta_\star \in \left\{ \theta \in \mathbb{R}^d : \|\theta - \hat{\theta}_t^\lambda\|_A \leq \beta_t \right\}$

Example of threshold [Abbasi-Yadkori et al., 2011]

Assuming that the noise  $\varepsilon_t$  is  $\sigma^2$ -sub-Gaussian, and that for all  $t$  and  $x \in \mathcal{X}_t$ ,  $\|x\| \leq L$ , we have

$$\mathbb{P} \left( \exists t \in \mathbb{N}^* : \|\theta_\star - \hat{\theta}_t^\lambda\|_{B_t^\lambda} > \beta(t, \delta) \right) \leq \delta$$

with  $\beta(t, \delta) = \sigma \sqrt{2 \log(1/\delta) + d \log(1 + t \frac{L}{d\lambda})} + \sqrt{\lambda} \|\theta_\star\|$ .

→ Letting

$$C_t(\delta) = \left\{ \theta \in \mathbb{R}^d : \|\theta - \hat{\theta}_t^\lambda\|_{B_t^\lambda} \leq \beta(t, \delta) \right\},$$

one has  $\mathbb{P}(\forall t \in \mathbb{N}, \theta_\star \in C_t(\delta)) \geq 1 - \delta$ .



# A Lin-UCB algorithm

**Consequence :**

$$\mathbb{P}\left(\forall t \in \mathbb{N}^*, \forall x \in \mathcal{X}_{t+1}, \underbrace{x^\top \theta_\star}_{\substack{\text{unknown mean} \\ \text{of arm } x}} \leq \underbrace{x^\top \hat{\theta}_t^\lambda + \|x\|_{(B_t^\lambda)^{-1}} \beta(t, \delta)}_{\text{Upper Confidence Bound}}\right) \geq 1 - \delta.$$

One can assign to each arm  $x \in \mathcal{X}_{t+1}$

$$\text{UCB}_x(t) = \underbrace{x^\top \hat{\theta}_t^\lambda}_{\substack{\text{empirical mean} \\ \text{(exploitation term)}}} + \underbrace{\|x\|_{(B_t^\lambda)^{-1}} \beta(t, \delta)}_{\text{exploration bonus}}$$

## Lin-UCB

In each round  $t + 1$ , the algorithm selects

$$x_{t+1} = \underset{x \in \mathcal{X}_{t+1}}{\operatorname{argmax}} \left[ x^\top \hat{\theta}_t^\lambda + \|x\|_{(B_t^\lambda)^{-1}} \beta(t, \delta) \right]$$

(many algorithms of this style, with different choices of  $\beta(t, \delta)$ )

# Theoretical guarantees

We want to bound the **pseudo-regret**

$$R_T(\text{Lin-UCB}) = \sum_{t=1}^T \left( \max_{x \in \mathcal{X}_t} \theta_\star^\top x - \theta_\star^\top x_t \right)$$

or its expectation, the **regret**  $\mathcal{R}_T(\text{Lin-UCB}) = \mathbb{E}[R_T(\text{Lin-UCB})]$ .

## Lemma

One can prove that, with probability larger than  $1 - \delta$ ,

$$\forall T \in \mathbb{N}^*, R_T(\text{Lin-UCB}) \leq C\beta(T, \delta) \sqrt{dT \log(T)}$$

- ▶ with the choice of  $\beta(t, \delta)$  presented before, with high probability

$$R_T(\text{Lin-UCB}) = \mathcal{O}(d\sqrt{T} \log(T) + \sqrt{dT \log(T) \log(1/\delta)})$$

- ▶ choosing  $\delta = 1/T$ ,  $\mathcal{R}_T(\text{Lin-UCB}) = \mathcal{O}(d\sqrt{T} \log(T))$

# Outline

1 Contextual Bandits

2 Solving Linear Bandits

- Lin-UCB

- Linear Thompson Sampling

3 Other variants of the classical MAB

# A Bayesian view on Linear Regression

## Bayesian model :

- ▶ likelihood :  $r_t = \theta_\star^\top x_t + \varepsilon_t$
- ▶ prior :  $\theta_\star \sim \mathcal{N}(0, \kappa^2 I_d)$

Assuming further that the noise is Gaussian :  $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$ , the **posterior distribution** of  $\theta_\star$  has a closed form :

$$\theta_\star | x_1, r_1, \dots, x_t, r_t \sim \mathcal{N}(\hat{\theta}_t^\lambda, \sigma^2 (B_t^\lambda)^{-1})$$

with

- $B_t^\lambda = \lambda I_d + \sum_{s=1}^t x_s x_s^\top$
- $\hat{\theta}_t^\lambda = (B_t^\lambda)^{-1} (\sum_{s=1}^t r_s x_s)$  is the regularized least square estimate

with a regularization parameter  $\lambda = \frac{\sigma^2}{\kappa^2}$ .

# Thompson Sampling for Linear Bandits

Recall the Thompson Sampling principle :

“draw a possible model from the posterior distribution and act optimally in this sampled model”

## Thompson Sampling in linear bandits

In each round  $t + 1$ ,

$$\begin{aligned}\tilde{\theta}_t &\sim \mathcal{N}\left(\hat{\theta}_t^\lambda, \sigma^2 (B_t^\lambda)^{-1}\right) \\ x_{t+1} &= \operatorname{argmax}_{x \in \mathcal{X}_{t+1}} x^\top \tilde{\theta}_t\end{aligned}$$

**Numerical complexity** : one need to draw a sample from a multivariate Gaussian distribution, e.g.

$$\tilde{\theta}_t = \hat{\theta}_t^\lambda + \sigma (B_t^\lambda)^{-1/2} X$$

where  $X$  is a vector with  $d$  independent  $\mathcal{N}(0, 1)$  entries.

# Theoretical guarantees

[Agrawal and Goyal, 2013] analyze a *variant* of Thompson Sampling using some “posterior inflation” :

$$\begin{aligned}\tilde{\theta}_t &\sim \mathcal{N}\left(\hat{\theta}_t^1, \nu^2 (B_t^1)^{-1}\right) \\ x_{t+1} &= \operatorname{argmax}_{x \in \mathcal{X}_{t+1}} x^\top \tilde{\theta}_t\end{aligned}$$

where  $\nu = \sigma \sqrt{9d \ln(T/\delta)}$ .

## Theorem

If the noise is  $\sigma^2$ -sub-Gaussian, the above algorithm satisfies

$$\mathbb{P}\left(R_T(\text{TS}) = \mathcal{O}\left(d^{3/2} \sqrt{T} \left[\ln(T) + \sqrt{\ln(T) \ln(1/\delta)}\right]\right)\right) \geq 1 - \delta.$$

- ▶ slightly worse than Lin-UCB... how about in practice ?
- ▶ do we need the posterior inflation ?

## Beyond linear bandits

Depending on the application, other parameteric models may be better suited than the simple linear model, for example the **logistic model**.

$$\begin{aligned}\mathbb{P}(r_t = 1|x_t) &= \frac{1}{1 + e^{-\theta_{\star}^{\top} x_t}} \\ \mathbb{P}(r_t = 0|x_t) &= \frac{e^{-\theta_{\star}^{\top} x_t}}{1 + e^{-\theta_{\star}^{\top} x_t}}\end{aligned}$$

e.g., clic / no-clic on an add depending on a user/add feature  $x_t \in \mathbb{R}^d$

- ▶ [Filippi et al., 2010] : first UCB style algorithm for Generalized Linear Bandit models
- ▶ Thompson Sampling for logistic bandits [Dumitrescu et al., 2018]
- ▶ going further : UCB/TS for neural bandits !

# Outline

## 1 Contextual Bandits

## 2 Solving Linear Bandits

- Lin-UCB
- Linear Thompson Sampling

## 3 Other variants of the classical MAB



# Many possible structures

$\mathcal{X}$ -armed bandits :  $\mathcal{X}_t = \mathcal{X}$  arbitrary metric space

$$r_t = f_\star(x_t) + \varepsilon_t$$

with non-parametric assumption on  $f_\star$ .

## Examples :

- ▶  $f_\star$  is a **Lipschitz** function :

$$|f_\star(x) - f_\star(y)| \leq Ld(x, y)$$

where  $d$  is a metric on  $\mathcal{X}$ .

[Bubeck et al., 2008]

- ▶  $f_\star$  is a **unimodal** function
- ▶  $f_\star$  is drawn from a **Gaussian process** prior

[Srinivas et al., 2009]

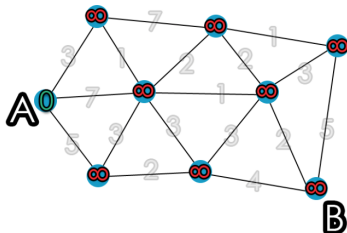
- ▶ ...

# Beyond one arm : Combinatorial bandits

classical bandit : **one arm** is selected in each round

combinatorial bandit : possibility to select a **group of arms** (action)

e.g., [Chen et al., 2013]



**Example :**

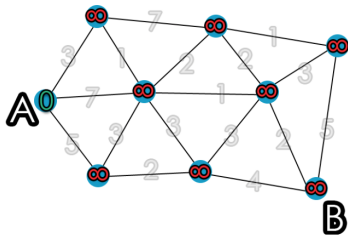
- ▶ arms : edges in a graph
- ▶ actions : paths from A to B
- ▶ reward : some function of the edges's rewards in the chosen path  
(e.g. - (total travelling distance))

# Beyond one arm : Combinatorial bandits

classical bandit : **one arm** is selected in each round

combinatorial bandit : possibility to select a **group of arms** (action)

e.g., [Chen et al., 2013]



**Combinatorial bandit** :  $\text{Actions} \subseteq \mathcal{P}(\{1, \dots, K\})$ .

In round  $t$ , the agent

- ▶ selects an action  $\text{Act}_t \in \text{Actions}$
- ▶ a reward  $r_{a,t}$  is generated for every arm  $a \in \text{Act}_t$
- ▶ the agent receives as a reward  $\sum_{a \in \text{Act}_t} r_{a,t}$  (or some other function)

# Beyond one agent : Multi-Player bandits

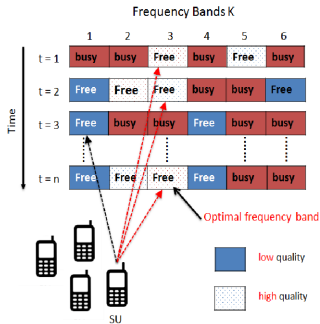
classical bandit : **one agent** select and arm in each round

multi-player bandit : **several agents** play on the same bandit

e.g., [Besson and Kaufmann, 2018]

**Example :** (cognitive radio system)

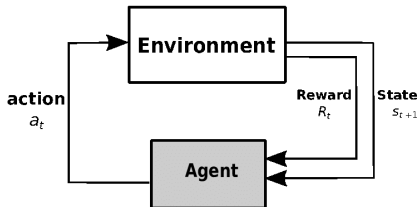
- ▶ arm : availability of a radio channel
- ▶ agent : a radio device, picking a channel in each round
- ▶ reward : quality of the communication
- if two agents select the same arm, the reward is reduced...



# Beyond one state : Reinforcement Learning

In most bandit models, the agent repeatedly faces the **same set of actions** (or at least the set of available actions in round does not depend on the past decisions).

- no longer true in **reinforcement learning**, in which an action also triggers a transition to a new **state**



# Bandits without rewards ?



$\mathcal{B}(\mu_1)$



$\mathcal{B}(\mu_2)$



$\mathcal{B}(\mu_3)$



$\mathcal{B}(\mu_4)$



$\mathcal{B}(\mu_5)$

For the  $t$ -th patient in a clinical study,

- ▶ chooses a treatment  $A_t$
- ▶ observes a response  $X_t \in \{0, 1\} : \mathbb{P}(X_t = 1) = \mu_{A_t}$

**Maximize rewards**  $\leftrightarrow$  cure as many patients as possible

**Alternative goal :** identify as quickly as possible the best treatment  
(without trying to cure patients during the study)

# Bandits without rewards ?



$\mathcal{B}(\mu_1)$

$\mathcal{B}(\mu_2)$

$\mathcal{B}(\mu_3)$

$\mathcal{B}(\mu_4)$

$\mathcal{B}(\mu_5)$

For the  $t$ -th patient in a clinical study,

- ▶ chooses a treatment  $A_t$
- ▶ observes a response  $X_t \in \{0, 1\} : \mathbb{P}(X_t = 1) = \mu_{A_t}$

**Maximize rewards**  $\leftrightarrow$  cure as many patients as possible

**Alternative goal** : identify as quickly as possible the best treatment  
(without trying to cure patients during the study)

→ Pure exploration, Best arm identification [Bubeck et al., 2011]



Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011).

Improved algorithms for linear stochastic bandits.

*Advances in neural information processing systems*, 24.



Agrawal, S. and Goyal, N. (2013).

Thompson sampling for contextual bandits with linear payoffs.

In *International conference on machine learning*, pages 127–135. PMLR.



Besson, L. and Kaufmann, E. (2018).

Multi-player bandits revisited.

In *Algorithmic Learning Theory*, pages 56–92. PMLR.



Bubeck, S., Munos, R., and Stoltz, G. (2011).

Pure exploration in finitely-armed and continuous-armed bandits.

*Theoretical Computer Science*, 412(19) :1832–1852.



Bubeck, S., Stoltz, G., Szepesvári, C., and Munos, R. (2008).

Online optimization in x-armed bandits.

*Advances in Neural Information Processing Systems*, 21.



Chen, W., Wang, Y., and Yuan, Y. (2013).



Combinatorial multi-armed bandit : General framework and applications.  
In *International conference on machine learning*, pages 151–159. PMLR.



Dumitrescu, B., Feng, K., and Engelhardt, B. (2018).

Pg-ts : Improved thompson sampling for logistic contextual bandits.  
*Advances in neural information processing systems*, 31.



Filippi, S., Cappe, O., Garivier, A., and Szepesvári, C. (2010).

Parametric bandits : The generalized linear case.  
*Advances in Neural Information Processing Systems*, 23.



Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. (2009).

Gaussian process optimization in the bandit setting : No regret and experimental design.  
*arXiv preprint arXiv :0912.3995*.