Sequential Decision

Session 4: Recap and bandit challenge

Igor Colin

October 2026

ESIEE Paris

Session 3

Quiz

1. With a uniform prior, an arm has been pulled 8 times, with 6 successes. Give its posterior and the posterior mean.

2. Thompson sampling has two arms: arm 1 with posterior mean 0.7, arm 2 with posterior mean 0.5. Can it play arm 2?

3. Write the K-armed bandit as a linear bandit: give d, the feature vectors and $\theta^\star$.

Quiz

1. With a uniform prior, an arm has been pulled 8 times, with 6 successes. Give its posterior and the posterior mean.
   $\mathrm{Beta}(7, 3)$, with mean $7/10 = 0.7$.

2. Thompson sampling has two arms: arm 1 with posterior mean 0.7, arm 2 with posterior mean 0.5. Can it play arm 2?

3. Write the K-armed bandit as a linear bandit: give d, the feature vectors and $\theta^\star$.

Quiz

1. With a uniform prior, an arm has been pulled 8 times, with 6 successes. Give its posterior and the posterior mean.
   $\mathrm{Beta}(7, 3)$, with mean $7/10 = 0.7$.

2. Thompson sampling has two arms: arm 1 with posterior mean 0.7, arm 2 with posterior mean 0.5. Can it play arm 2?
   Yes, with probability $\mathbb{P}(\tilde{\theta}_2 > \tilde{\theta}_1) > 0$, which is larger when the posterior of arm 2 is wide.

3. Write the K-armed bandit as a linear bandit: give d, the feature vectors and $\theta^\star$.

Quiz

1. With a uniform prior, an arm has been pulled 8 times, with 6 successes. Give its posterior and the posterior mean.
   $\mathrm{Beta}(7, 3)$, with mean $7/10 = 0.7$.

2. Thompson sampling has two arms: arm 1 with posterior mean 0.7, arm 2 with posterior mean 0.5. Can it play arm 2?
   Yes, with probability $\mathbb{P}(\tilde{\theta}_2 > \tilde{\theta}_1) > 0$, which is larger when the posterior of arm 2 is wide.

3. Write the K-armed bandit as a linear bandit: give d, the feature vectors and $\theta^\star$.
   $d = K$, $x_a = e_a$ and $\theta^\star = (\mu_1, \dots, \mu_K)$.

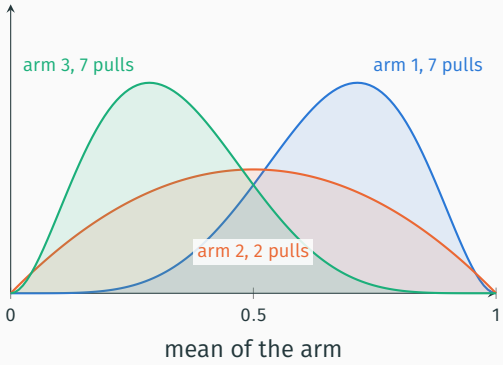
Today

Part	Content	Time
Recap	Thompson sampling, linear bandits, LinUCB	20 min
Tutorial	Tutorial 3, exercise 3: LinUCB on an example	30 min
Lab 3	Thompson sampling, ridge, LinUCB, linear Thompson sampling	1 h 30
Lab 4	Bandit challenge	40 min

- Tutorial 3, exercise 3: formula (4) and question 4 of exercise 2 are admitted
- Course page: download the notebooks of Labs 3 and 4, and the setup archive again for the new `seqlearn`; add `ipywidgets` to your environment (instructions on the page)

Thompson sampling

Thompson sampling

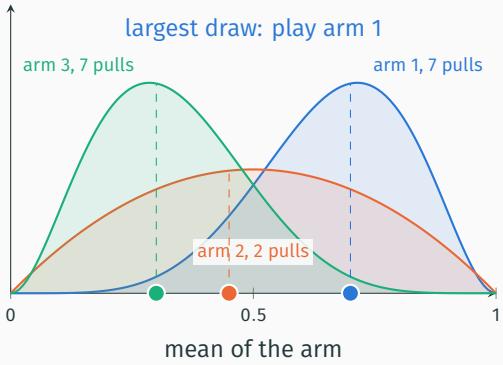


Strategy (Thompson, 1933)

At each round:

1. draw $\tilde{\theta}_a$ from the posterior of each arm a;
2. play $A_t \in \arg\max_a \tilde{\theta}_a$;
3. update the posterior of A_t.

Thompson sampling



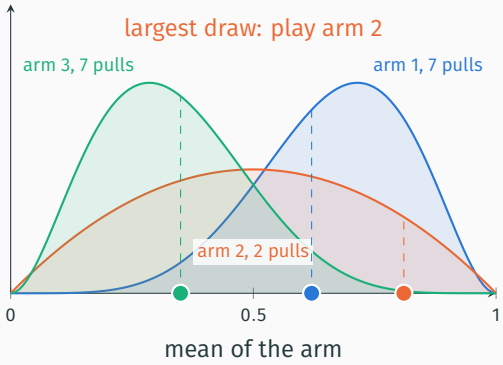
Strategy (Thompson, 1933)

At each round:

1. draw $\tilde{\theta}_a$ from the posterior of each arm a;
2. play $A_t \in \arg\max_a \tilde{\theta}_a$;
3. update the posterior of A_t.

- Most draws favor the arm with the best posterior

Thompson sampling



Strategy (Thompson, 1933)

At each round:

1. draw $\tilde{\theta}_a$ from the posterior of each arm a;
2. play $A_t \in \arg\max_a \tilde{\theta}_a$;
3. update the posterior of A_t.

- Most draws favor the arm with the best posterior
- A wide posterior sometimes gives the largest draw: this is the exploration

UCB and Thompson sampling

	UCB	Thompson sampling
Uncertainty	confidence interval	posterior distribution
Choice	argmax of the upper bound	argmax of a random draw
Decision given the history	deterministic	randomized
Tuning	exploration parameter α	prior
Delayed or batched feedback	repeats the same arm	spreads its choices

Sampling and argmax

Taking the argmax of an estimate (greedy) locks; adding a bonus (UCB) or drawing from a distribution (Thompson sampling) explores. The same choice comes back in language models: greedy decoding takes the most likely token, sampling draws from the distribution.

Linear bandits

The linear model

Linear bandit

K arms with known feature vectors $x_1, \dots, x_K \in \mathbb{R}^d$ and an unknown parameter $\theta^\star \in \mathbb{R}^d$. At each round t, the agent chooses A_t and observes X_t with

$$\mathbb{E}[X_t \mid \text{history}] = \langle \theta^\star, x_{A_t} \rangle \ .$$

K-armed bandit

- $d = K$, $x_a = e_a$
- $\theta^\star = (\mu_1, \dots, \mu_K)$

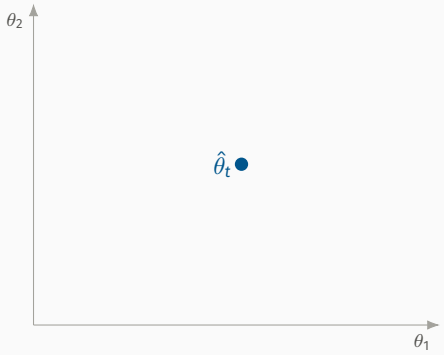
Catalog

- x_a: features of item a, with $d \ll K$
- $\theta^\star$: tastes of the audience

Regret: as before, with $\mu_a = \langle \theta^\star, x_a \rangle$,

$$R_T = T \max_a \langle \theta^\star, x_a \rangle - \mathbb{E}\Big[\sum_{t=1}^{T} \langle \theta^\star, x_{A_t} \rangle \Big] \ .$$

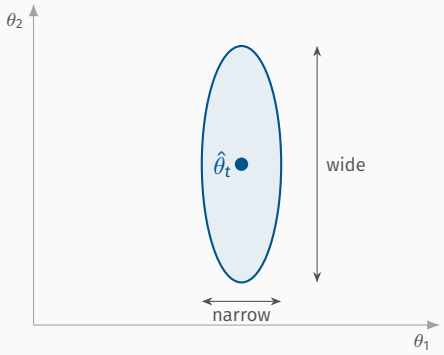
The confidence ellipsoid



With high probability, $\theta^\star$ belongs to

$$\mathcal{C}_t = \left\{ \theta : (\theta - \hat{\theta}_t)^\top \mathbf{V}_t (\theta - \hat{\theta}_t) \leq \beta^2 \right\} \ .$$

The confidence ellipsoid

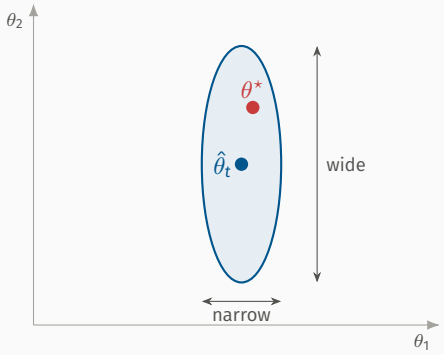


With high probability, $\theta^\star$ belongs to

$$\mathcal{C}_t = \left\{ \theta : (\theta - \hat{\theta}_t)^\top \mathbf{V}_t (\theta - \hat{\theta}_t) \leq \beta^2 \right\} .$$

- An ellipsoid centered at $\hat{\theta}_t$: the confidence interval of dimension d
- Here the pulls were mostly along the first coordinate: θ_1 is well known, θ_2 is not

The confidence ellipsoid

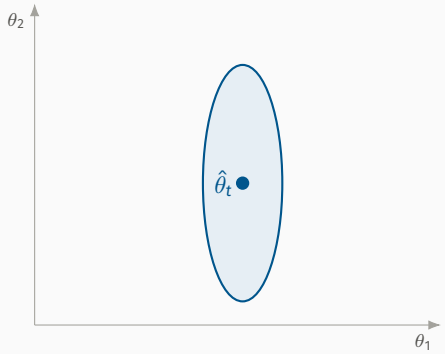


With high probability, $\theta^\star$ belongs to

$$\mathcal{C}_t = \left\{ \theta : (\theta - \hat{\theta}_t)^\top \mathbf{V}_t (\theta - \hat{\theta}_t) \leq \beta^2 \right\} .$$

- An ellipsoid centered at $\hat{\theta}_t$: the confidence interval of dimension d
- Here the pulls were mostly along the first coordinate: θ_1 is well known, θ_2 is not
- Each pull shrinks the ellipsoid in the direction of the feature vector that was played

LinUCB

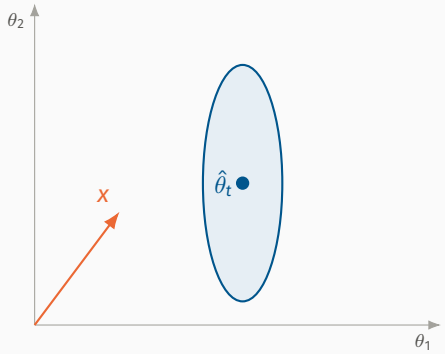


Optimism: evaluate each arm with the most favorable parameter of the ellipsoid.

Index of the arm with features x

$$U_t(x) = \max_{\theta \in \mathcal{C}_t} \langle \theta, x \rangle = \langle \hat{\theta}_t, x \rangle + \beta \sqrt{x^\top \mathbf{V}_t^{-1} x}$$

LinUCB



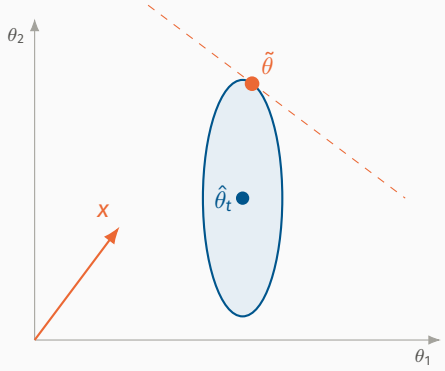
Optimism: evaluate each arm with the most favorable parameter of the ellipsoid.

Index of the arm with features x

$$U_t(x) = \max_{\theta \in \mathcal{C}_t} \langle \theta, x \rangle = \langle \hat{\theta}_t, x \rangle + \beta \sqrt{x^\top \mathbf{V}_t^{-1} x}$$

- Play $A_t \in \arg\max_a U_t(x_a)$

LinUCB



Optimism: evaluate each arm with the most favorable parameter of the ellipsoid.

Index of the arm with features x

$$U_t(x) = \max_{\theta \in \mathcal{C}_t} \langle \theta, x \rangle = \langle \hat{\theta}_t, x \rangle + \beta \sqrt{x^\top \mathbf{V}_t^{-1} x}$$

- Play $A_t \in \arg\max_a U_t(x_a)$
- The bonus is the half-width of the ellipsoid in the direction x: large for directions that were rarely played
- For $x_a = e_a$: bonus $\beta / \sqrt{\lambda + N_a}$, as in UCB

Limits of Thompson sampling

For K Bernoulli arms, Thompson sampling is as cheap and simple as UCB. Beyond:

Limits of Thompson sampling

For K Bernoulli arms, Thompson sampling is as cheap and simple as UCB. Beyond:

- **Theory.** The guarantees of linear Thompson sampling require inflating the posterior variance by a factor of order $\sqrt{d}$, hence $d^{3/2}\sqrt{T}$ instead of $d\sqrt{T}$; without inflation, the regret can be linear on some instances (Hamidi and Bayati, 2020)

Limits of Thompson sampling

For K Bernoulli arms, Thompson sampling is as cheap and simple as UCB. Beyond:

- **Theory.** The guarantees of linear Thompson sampling require inflating the posterior variance by a factor of order $\sqrt{d}$, hence $d^{3/2}\sqrt{T}$ instead of $d\sqrt{T}$; without inflation, the regret can be linear on some instances (Hamidi and Bayati, 2020)
- **Computation.** A draw from $\mathcal{N}(\hat{\theta}_t, v^2 \mathbf{V}_t^{-1})$ costs a Cholesky factorization, $O(d^3)$. For a Gaussian process (session 4), one draw is a whole function: a joint Gaussian over the n candidates, $O(n^3)$, where GP-UCB only needs the variance at each point

Limits of Thompson sampling

For K Bernoulli arms, Thompson sampling is as cheap and simple as UCB. Beyond:

- **Theory.** The guarantees of linear Thompson sampling require inflating the posterior variance by a factor of order $\sqrt{d}$, hence $d^{3/2}\sqrt{T}$ instead of $d\sqrt{T}$; without inflation, the regret can be linear on some instances (Hamidi and Bayati, 2020)
- **Computation.** A draw from $\mathcal{N}(\hat{\theta}_t, v^2\mathbf{V}_t^{-1})$ costs a Cholesky factorization, $O(d^3)$. For a Gaussian process (session 4), one draw is a whole function: a joint Gaussian over the n candidates, $O(n^3)$, where GP-UCB only needs the variance at each point
- **Models.** Without conjugacy (logistic model, neural network) the posterior has no closed form: Laplace approximation, MCMC, bootstrap or ensembles

Limits of Thompson sampling

For K Bernoulli arms, Thompson sampling is as cheap and simple as UCB. Beyond:

- **Theory.** The guarantees of linear Thompson sampling require inflating the posterior variance by a factor of order $\sqrt{d}$, hence $d^{3/2}\sqrt{T}$ instead of $d\sqrt{T}$; without inflation, the regret can be linear on some instances (Hamidi and Bayati, 2020)
- **Computation.** A draw from $\mathcal{N}(\hat{\theta}_t, v^2\mathbf{V}_t^{-1})$ costs a Cholesky factorization, $O(d^3)$. For a Gaussian process (session 4), one draw is a whole function: a joint Gaussian over the n candidates, $O(n^3)$, where GP-UCB only needs the variance at each point
- **Models.** Without conjugacy (logistic model, neural network) the posterior has no closed form: Laplace approximation, MCMC, bootstrap or ensembles
- **Other objectives.** Identifying the best arm (session 4): Thompson sampling focuses on the best arm and samples the second one too little. Off-policy evaluation (session 5): the probability of each action has no closed form

Bandit challenge

Bandit challenge

One agent, three tracks, horizon $T = 5000$. The agent does not know which track it plays.

Track	K	d	Structure
`slots`	20	20	unrelated arms with small means, $x_a = e_a$
`catalog`	300	8	fixed features, as in Lab 3
`users`	6	24	features depend on the user of the round

Score of a track: mean final regret of the agent divided by the one of UCB1 on the same instances, in percent. UCB1 scores 100; the lower, the better. The score of the challenge is the mean of the three.

The track `users`

6 items, 4 moods: $\theta^\star$ holds the click rate of each item for each mood (24 unknown numbers). Example with 2 items and 2 moods, so $d = 4$:

	sport	news
football	0.8	0.1
politics	0.1	0.6

$$\theta^\star = (0.8, 0.1, 0.1, 0.6)$$

The track users

6 items, 4 moods: $\theta^\star$ holds the click rate of each item for each mood (24 unknown numbers). Example with 2 items and 2 moods, so $d = 4$:

	sport	news
football	0.8	0.1
politics	0.1	0.6

$\theta^\star = (0.8, 0.1, 0.1, 0.6)$

User 1  0.9 / 0.1

$x_{\text{foot}} = (0.9, 0.1, 0, 0)$, $x_{\text{pol}} = (0, 0, 0.9, 0.1)$

means 0.73 and 0.15: **football**

The track `users`

6 items, 4 moods: $\theta^\star$ holds the click rate of each item for each mood (24 unknown numbers). Example with 2 items and 2 moods, so $d = 4$:

	sport	news
football	0.8	0.1
politics	0.1	0.6

$\theta^\star = (0.8, 0.1, 0.1, 0.6)$

User 1  0.9 / 0.1

$x_{\text{foot}} = (0.9, 0.1, 0, 0)$, $x_{\text{pol}} = (0, 0, 0.9, 0.1)$
means 0.73 and 0.15: **football**

User 2  0.2 / 0.8

$x_{\text{foot}} = (0.2, 0.8, 0, 0)$, $x_{\text{pol}} = (0, 0, 0.2, 0.8)$
means 0.24 and 0.50: **politics**

The track `users`

6 items, 4 moods: $\theta^\star$ holds the click rate of each item for each mood (24 unknown numbers). Example with 2 items and 2 moods, so $d = 4$:

	sport	news
football	0.8	0.1
politics	0.1	0.6

$\theta^\star = (0.8, 0.1, 0.1, 0.6)$

User 1  0.9 / 0.1

$x_{\text{foot}} = (0.9, 0.1, 0, 0)$, $x_{\text{pol}} = (0, 0, 0.9, 0.1)$
means 0.73 and 0.15: **football**

User 2  0.2 / 0.8

$x_{\text{foot}} = (0.2, 0.8, 0, 0)$, $x_{\text{pol}} = (0, 0, 0.2, 0.8)$
means 0.24 and 0.50: **politics**

Same linear model, the feature vectors change at each round

$\mathbb{E}[X_t \mid \text{history}] = \langle \theta^\star, x_{t,A_t} \rangle$, where $x_{t,a}$ is the profile of the user of round t in the block of item a. The regret compares with the best item of each round. The agents of Lab 3 play this track unchanged.

Rules

- Write your agent in one file `agent_yournames.py`, from the notebook of Lab 4
- It defines `make_agent(n_arms, dim, horizon, rng)`, imports only `numpy` and `seqlearn`, and takes all its randomness from `rng`
- The agent learns only from the features and the rewards, and `evaluate` runs in less than 3 minutes on a laptop

Starting point

The agents of Lab 3 already play the three tracks. Their parameters were tuned on another problem.