Bandits: Non-stationarity and side observations

Constantine Caramanis
constantine@utexas.edu
UT Austin

Joint work with:

Alexia Atsidakou
Soumya Basu
Orestis Papadigenopoulos
Sujay Sanghavi
Sanjay Shakkottai

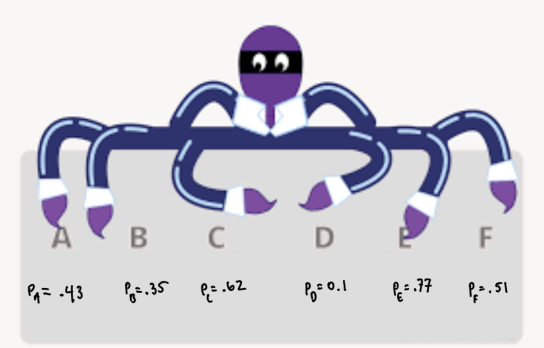
Overview: Multi-Arm Bandits

A Model for Dynamic Decision-Making

Overview: Multi-Arm Bandits

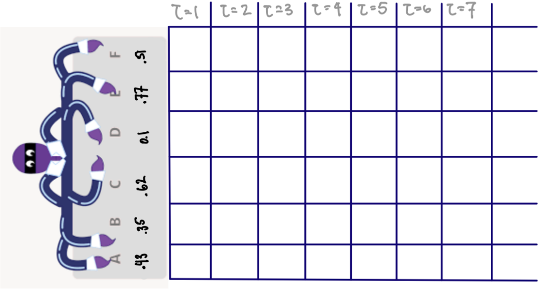
Dynamic Decision Making:

- Actions: $\mathcal{K} = \{1, \ldots, K\}$
- At time t, play action $a_t \in \mathcal{K}$.
- Receive stochastic reward R_t.
- Goal: play to maximize expected reward.



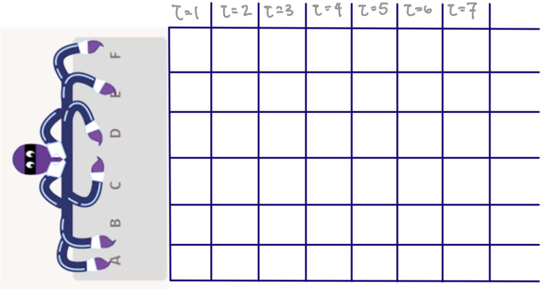
Overview: Multi-Arm Bandits

Which action to play?



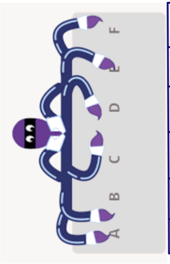
Overview: Multi-Arm Bandits

Which action to play?



Overview: Multi-Arm Bandits

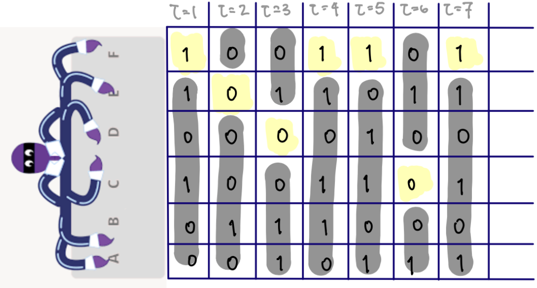
Which action to play?



$\tau=1$	$\tau=2$	$\tau=3$	$\tau=4$	$\tau=5$	$\tau=6$	$\tau=7$	
1	0	0	1	1	0	1	
1	0	1	1	0	1	1	
0	0	0	0	1	0	0	
1	0	0	1	1	0	1	
0	1	1	1	0	0	0	
0	0	1	0	1	1	1	

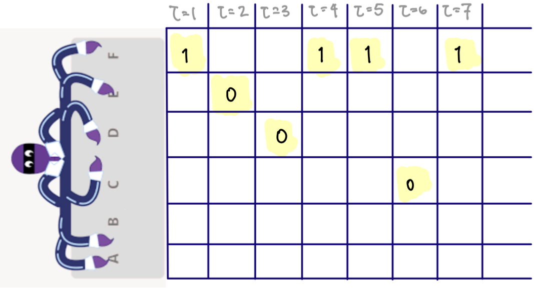
Overview: Multi-Arm Bandits

Which action to play?



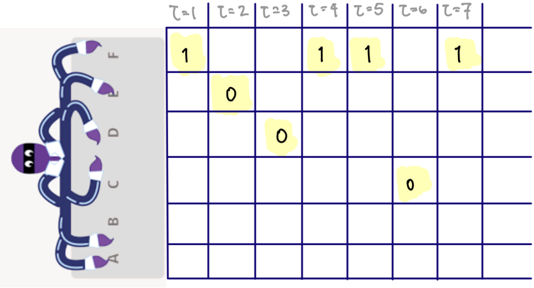
Overview: Multi-Arm Bandits

Which action to play?



Overview: Multi-Arm Bandits

Exploit: Play action F. (Suboptimal)

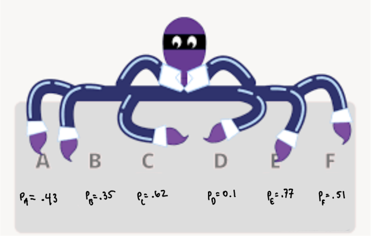


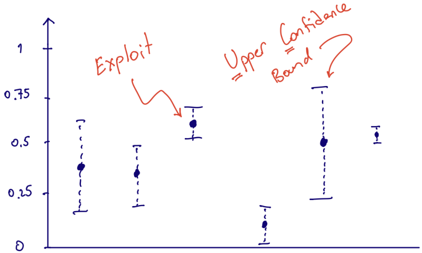
Overview: Multi-Arm Bandits

- Exploit: play action with highest empirical reward (Suboptimal)
- Need to Explore and Exploit
- By now classical problem.
- **Key idea: Optimism: play arm with highest plausible reward.**

Overview: Multi-Arm Bandits

Exploit vs Upper Confidence Bound





Overview: Multi-Arm Bandits

- Exploit: play action with highest empirical reward (Suboptimal)
- Need to Explore and Exploit
- By now classical problem.
- **Key idea: Optimism: play arm with highest plausible reward.**

- (e.g., books by Cesa-Bianchi & Bubeck, Lattimore & Szepasvari)

Overview: Multi-Arm Bandits

- Good news: For classical bandit problems, the UCB algorithm is near-optimal, efficient to implement, and analytically well-understood.
- Bad news: Many interesting practical problems violate classical assumptions. UCB no longer optimal.
- **This talk**: two such problems, and their solutions.

Non-stationarity

Multi-Armed Bandits

Online learning model for studying the tradeoff between exploration *and* exploitation

- Set $\mathcal{K}$ of k *arms* or *actions*
- Each $i \in \mathcal{K}$: unknown *reward distribution* of mean μ_i
- T rounds (can be unknown)
- At each round $t = 1, 2, \ldots, T$, a *player*:
  1. Chooses to play an arm $i \in \mathcal{K}$
  2. Collects the realized reward

Goal: Minimize the *regret*:

$$T \cdot \max_i \mu_i - \mathbb{E}[\text{Player's reward}]$$

Motivation

Example 1

- Suppose we run a boat-to-rent service in a Greek island

Example 1

- Suppose we run a boat-to-rent service in a Greek island
- We own a single boat and each "tour" takes 3 hours

Example 1

- Suppose we run a boat-to-rent service in a Greek island
- We own a single boat and each "tour" takes 3 hours
- At any hour (roughly), a client-type arrives:
  - "Tourist group" offers \$100 (+ tips)
  - "Romantic couple" offers \$50 (+ tips)
  - "Student" offers \$20 (no tips)
  - "'No client" offers \$0

Example 1

- Suppose we run a boat-to-rent service in a Greek island
- We own a single boat and each "tour" takes 3 hours
- At any hour (roughly), a client-type arrives:
    - "Tourist group" offers \$100 (+ tips)
    - "Romantic couple" offers \$50 (+ tips)
    - "Student" offers \$20 (no tips)
    - "'No client" offers \$0
- The arrival probability of each type is (empirically) known

Example 1

- Suppose we run a boat-to-rent service in a Greek island
- We own a single boat and each "tour" takes 3 hours
- At any hour (roughly), a client-type arrives:
  - "Tourist group" offers \$100 (+ tips)
  - "Romantic couple" offers \$50 (+ tips)
  - "Student" offers \$20 (no tips)
  - "'No client" offers \$0
- The arrival probability of each type is (empirically) known
- Suppose the *student* arrives ...

Example 1

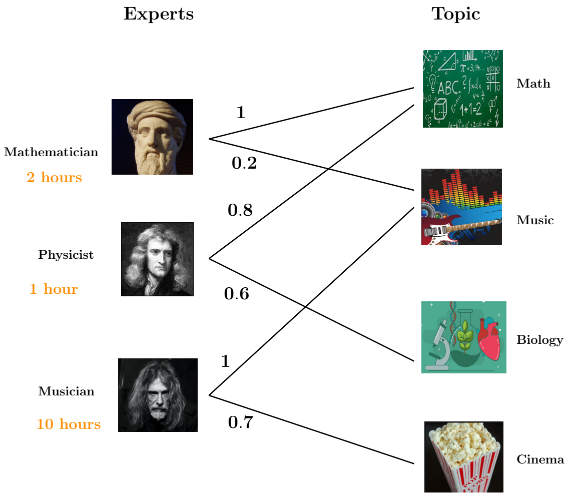
- Suppose we run a boat-to-rent service in a Greek island
- We own a single boat and each “tour” takes 3 hours
- At any hour (roughly), a client-type arrives:
  - “Tourist group” offers \$100 (+ tips)
  - “Romantic couple” offers \$50 (+ tips)
  - “Student” offers \$20 (no tips)
  - “’No client” offers \$0
- The arrival probability of each type is (empirically) known
- Suppose the *student* arrives ...

Should we give her/him the boat?

Example 2

- We run a (monetized) question-answering platform (e.g., JustAnswers, Quora, Chegg)
- k "experts" (mathematician, historian, biologist, polymath)
- m "question-types" (math, philosophy, linguistics)
- Each expert i needs a fixed amount d_i of research hours before answering a question
- Each question-type appears with probability f_j
- For $i \in [k]$ and $j \in [m]$, let μ_{ij} be the probability that expert i gives a satisfactory answer to question-type j
- Questions arrive sequentially (e.g., one at each hour)
- Goal: Assign each question to a non-busy expert to maximize the expected number of satisfactory answers

Example 2


Experts
Topic
Mathematician
2 hours
Physicist
1 hour
Musician
10 hours
1
0.2
0.8
0.6
1
0.7
Math
Music
Biology
Cinema

Problem Definition

Problem Definition

Model:

- Set $\mathcal{K}$ of k arms (or *actions*)
- Set $\mathcal{C}$ of m contexts

Problem Definition

Model:

- Set $\mathcal{K}$ of k arms (or *actions*)
- Set $\mathcal{C}$ of m contexts
- $f_j \in (0, 1)$: frequency of context j
  - known to the player
  - $\sum_{j \in \mathcal{C}} f_j = 1$

Problem Definition

Model:

- Set $\mathcal{K}$ of k arms (or *actions*)
- Set $\mathcal{C}$ of m contexts
- $f_j \in (0, 1)$: frequency of context j
  - known to the player
  - $\sum_{j \in \mathcal{C}} f_j = 1$
- $\mathcal{X}_{ij}$: reward distribution of arm i under context j
  - unknown mean μ_{ij}
  - bounded support in $[0, 1]$

Problem Definition

Model:

- Set $\mathcal{K}$ of k arms (or *actions*)
- Set $\mathcal{C}$ of m contexts
- $f_j \in (0,1)$: frequency of context j
  - known to the player
  - $\sum_{j \in \mathcal{C}} f_j = 1$
- $\mathcal{X}_{ij}$: reward distribution of arm i under context j
  - unknown mean μ_{ij}
  - bounded support in $[0,1]$
- $d_i \in \mathbb{N}$: delay of arm i
  - once played, arm i becomes blocked for the next $d_i - 1$ rounds
  - known and deterministic
  - $d_i = 1$ implies no blocking

Problem Definition

Model:

- Set $\mathcal{K}$ of k arms (or *actions*)
- Set $\mathcal{C}$ of m contexts
- $f_j \in (0, 1)$: frequency of context j
  - known to the player
  - $\sum_{j \in \mathcal{C}} f_j = 1$
- $\mathcal{X}_{ij}$: reward distribution of arm i under context j
  - unknown mean μ_{ij}
  - bounded support in $[0, 1]$
- $d_i \in \mathbb{N}$: delay of arm i
  - once played, arm i becomes blocked for the next $d_i - 1$ rounds
  - known and deterministic
  - $d_i = 1$ implies no blocking
- T: unknown time horizon

Problem Definition

- k arms and m contexts
- f_j: frequency of context j
- d_i: delay of arm i
- μ_{ij} mean reward of arm i under context j
- T: unknown time horizon

At each time $t = 1, 2, \ldots, T$, the player:

1. Observes the realized context of the round $j_t \in \mathcal{C}$
2. Chooses an available action $i \in \mathcal{K}$

Problem Definition

- k arms and m contexts
- f_j: frequency of context j
- d_i: delay of arm i
- μ_{ij} mean reward of arm i under context j
- T: unknown time horizon

At each time $t = 1, 2, \ldots, T$, the player:

1. Observes the realized context of the round $j_t \in \mathcal{C}$
2. Chooses an available action $i \in \mathcal{K}$

Goal: Maximize the expected cumulative reward over T rounds

The Full-Information Problem

The Full-Information Problem

Suppose that the reward distributions of the arms are known to the player a priori (and w.l.o.g. deterministic) . . .

. . . what does a "good" strategy look like?

A (very) simple setting

- Single arm of delay $d \gg 1$.

A (very) simple setting

- Single arm of delay $d \gg 1$.
- Two contexts:
  - Good: reward $\mu \geq 1$ and frequency $\epsilon = 1\%$

A (very) simple setting

- Single arm of delay $d \gg 1$.
- Two contexts:
  - Good: reward $\mu \geq 1$ and frequency $\epsilon = 1\%$
  - Meh: reward 1 and frequency $1 - \epsilon = 99\%$

A (very) simple setting

- Single arm of delay $d \gg 1$.
- Two contexts:
  - Good: reward $\mu \geq 1$ and frequency $\epsilon = 1\%$
  - Meh: reward 1 and frequency $1 - \epsilon = 99\%$

Intuitively, if μ is close to 1, the optimal policy plays the arm whenever it is available (under *both* good and meh) ...

A (very) simple setting

- Single arm of delay $d \gg 1$.
- Two contexts:
  - Good: reward $\mu \geq 1$ and frequency $\epsilon = 1\%$
  - Meh: reward 1 and frequency $1 - \epsilon = 99\%$

Intuitively, if μ is close to 1, the optimal policy plays the arm whenever it is available (under *both* good and meh) ...

... but once $\mu \gg 1$, the optimal policy plays the arm *only* under the good context.

A (very) simple setting

- Single arm of delay $d \gg 1$.
- Two contexts:
  - Good: reward $\mu \geq 1$ and frequency $\epsilon = 1\%$
  - Meh: reward 1 and frequency $1 - \epsilon = 99\%$

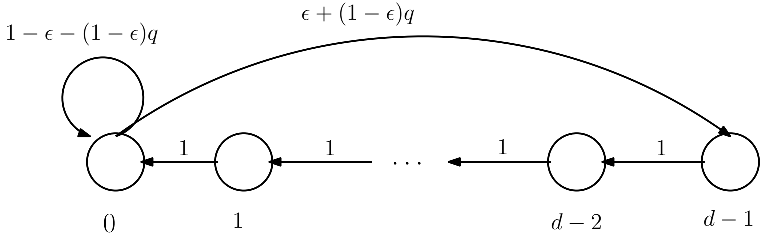
Intuitively, if μ is close to 1, the optimal policy plays the arm whenever it is available (under *both* good and meh) ...

...but once $\mu \gg 1$, the optimal policy plays the arm *only* under the good context.

If the frequency of the good context is ϵ, this phase transition happens *exactly* at $\mu = 1 + \frac{1}{\epsilon(d-1)}$

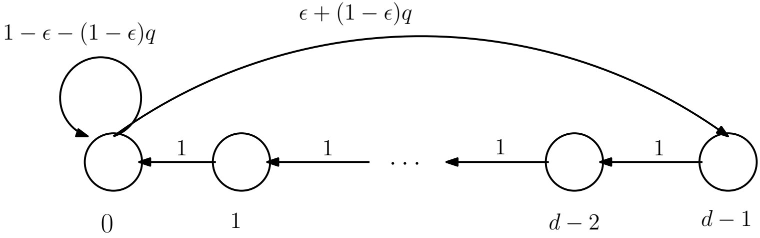
A (very) simple setting

Modeling the optimal policy as a Markov Chain...



A (very) simple setting

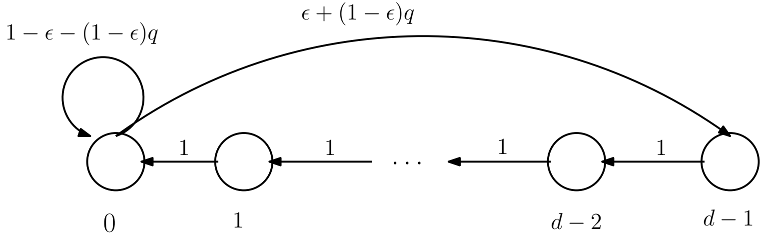
Modeling the optimal policy as a Markov Chain...



- Each state represents the number of rounds until the arm becomes available
- q : probability of playing the arm under context meh

A (very) simple setting

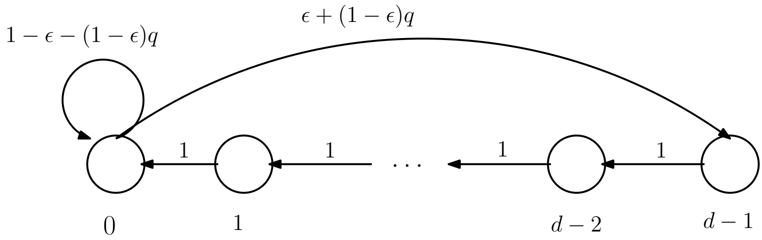
Modeling the optimal policy as a Markov Chain...



- Each state represents the number of rounds until the arm becomes available
- q : probability of playing the arm under context meh
- Expected reward equals $\pi(0)\cdot(\epsilon\mu+(1-\epsilon)q)$

A (very) simple setting

Modeling the optimal policy as a Markov Chain...



- Each state represents the number of rounds until the arm becomes available
- q : probability of playing the arm under context meh
- Expected reward equals $\pi(0) \cdot (\epsilon\mu + (1-\epsilon)q)$
- We can choose q to maximize the total expected reward (asymptotically)

A (very) simple setting

"Greedy" approach: Play the arm whenever it is available

In the above example, for $\mu \gg 1 + \frac{1}{\epsilon(d-1)} \ldots$

- The optimal collects on average $\frac{\epsilon\mu}{1+\epsilon(d-1)}$ (in expectation)
- The greedy collects on average $\frac{\epsilon\mu+1-\epsilon}{d}$
- By setting $\epsilon = \frac{1}{d}$, the competitive ratio scales as $\approx \frac{1}{d}$

A (very) simple setting

"Greedy" approach: Play the arm whenever it is available

In the above example, for $\mu \gg 1 + \frac{1}{\epsilon(d-1)} \ldots$

- The optimal collects on average $\frac{\epsilon\mu}{1+\epsilon(d-1)}$ (in expectation)
- The greedy collects on average $\frac{\epsilon\mu+1-\epsilon}{d}$
- By setting $\epsilon = \frac{1}{d}$, the competitive ratio scales as $\approx \frac{1}{d}$

Takeaways:

- "Greedy" doesn't work
- A "good" policy may *intentionally* skip rounds

An (asymptotic) LP upper bound

Using Linear Programming to upper bound the optimal expected reward.

An (asymptotic) LP upper bound

Using Linear Programming to upper bound the optimal expected reward.

$$\textbf{maximize:} \quad T \cdot \sum_{i \in \mathcal{K}} \sum_{j \in \mathcal{C}} \mu_{i,j} z_{i,j} \qquad \textbf{(LP)}$$

$$\textbf{s.t.:} \quad \sum_{j \in \mathcal{C}} z_{i,j} \leq \frac{1}{d_i} \qquad \forall i \in \mathcal{K} \qquad \textbf{(C1)}$$

$$\sum_{i \in \mathcal{K}} z_{i,j} \leq f_j \qquad \forall j \in \mathcal{C} \qquad \textbf{(C2)}$$

$$z_{i,j} \geq 0 \qquad \forall i \in \mathcal{K}, j \in \mathcal{C}$$

An (asymptotic) LP upper bound

Using Linear Programming to upper bound the optimal expected reward.

$$\textbf{maximize:} \quad T \cdot \sum_{i \in \mathcal{K}} \sum_{j \in \mathcal{C}} \mu_{i,j} z_{i,j} \qquad \textbf{(LP)}$$

$$\textbf{s.t.:} \quad \sum_{j \in \mathcal{C}} z_{i,j} \le \frac{1}{d_i} \qquad \forall i \in \mathcal{K} \qquad \textbf{(C1)}$$

$$\sum_{i \in \mathcal{K}} z_{i,j} \le f_j \qquad \forall j \in \mathcal{C} \qquad \textbf{(C2)}$$

$$z_{i,j} \ge 0 \qquad \forall i \in \mathcal{K}, j \in \mathcal{C}$$

Theorem

(**LP**) *yields a* $\left(1 - \mathcal{O}(\frac{d_{\max}}{T})\right)$*-approximate upper-bound on the optimal (clairvoyant w.r.t. context realizations) expected reward*

Online randomized rounding

Natural approach:

1. Compute a solution z^* to (**LP**)
2. At each round $t = 1, 2, \ldots,$
    1. Observe the context j_t of the round
    2. Sample an arm i with *marginal probability* $z^*_{i,j_t}/f_{j_t}$
    3. If the sampled arm i is available (not blocked), play it.

Online randomized rounding

Natural approach:

1. Compute a solution z^* to (**LP**)
2. At each round $t = 1, 2, \ldots,$
    1. Observe the context j_t of the round
    2. Sample an arm i with *marginal probability* $z^*_{i,j_t}/f_{j_t}$
    3. If the sampled arm i is available (not blocked), play it.

Theorem

The above is a $\frac{d_{\max}}{2d_{\max}-1}$-competitive policy (asymptotically)

Unconditional hardness

Theorem

The player cannot collect (in expectation) more than a $\frac{d_{\max}}{2d_{\max}-1}$*-fraction of the expected reward of an optimal* clairvoyant *policy.*

Unconditional hardness

Theorem

The player cannot collect (in expectation) more than a $\frac{d_{\max}}{2d_{\max}-1}$*-fraction of the expected reward of an optimal* clairvoyant *policy.*

- Based on the simple example of 1 arm and 2 contexts

Unconditional hardness

Theorem

The player cannot collect (in expectation) more than a $\frac{d_{\max}}{2d_{\max}-1}$-fraction of the expected reward of an optimal clairvoyant *policy.*

- Based on the simple example of 1 arm and 2 contexts
- Characterizing the player's best policy is easy

Unconditional hardness

Theorem

The player cannot collect (in expectation) more than a $\frac{d_{\max}}{2d_{\max}-1}$-fraction of the expected reward of an optimal clairvoyant *policy.*

- Based on the simple example of 1 arm and 2 contexts
- Characterizing the player's best policy is easy
- Characterizing the optimal clairvoyant policy is hard

Unconditional hardness

Theorem

The player cannot collect (in expectation) more than a $\frac{d_{\max}}{2d_{\max}-1}$-fraction of the expected reward of an optimal clairvoyant *policy.*

- Based on the simple example of 1 arm and 2 contexts
- Characterizing the player's best policy is easy
- Characterizing the optimal clairvoyant policy is hard
- Key idea: Asymptotically (for $T \to \infty$), it suffices to characterize a *near-optimal* yet simpler clairvoyant policy

Remaining Details

- Improving some asymptotic details.
- Bandit setting: we do not know the means (data for LP) a priori.
- "Contextual Blocking Bandits," Basu, Papadigenopoulos, **C.**, Shakkottai; AISTATS 2021.
  `https://arxiv.org/pdf/2003.03426.pdf`

More General Non-stationarity

Playing a matroid at each round, based on availability (blocking).

- "Combinatorial Blocking Bandits with Stochastic Delays," Atsidakou, Papadigenopoulos, Basu, **C.**, Shakkottai; ICML 2021 `https://arxiv.org/pdf/2105.10625.pdf`
- "Recurrent Submodular Welfare and Matroid Blocking Bandits," Papadigenopoulos, **C.**; NeurIPS 2021 `https://arxiv.org/pdf/2102.00321.pdf`

Recharging bandits

- "Non-Stationary Bandits under Recharging Payoffs: Improved Planning with Sublinear Regret," Papadigenopoulos, **C.**, Shakkottai; NeurIPS 2022. `https://arxiv.org/pdf/2205.14790.pdf`

Recharging Bandits

- Motivation:
  - "*Absence makes the heart grow fonder*"

Recharging Bandits

- Motivation:
  - "*Absence makes the heart grow fonder*"
  - **Examples**:
    - Movie recommendation: cannot watch the same movie every day (even our favorite one)
    - Food: some days need to pass to really enjoy our favorite food again (typically a week)

Recharging Bandits

- Motivation:
  - "*Absence makes the heart grow fonder*"
  - **Examples**:
    - Movie recommendation: cannot watch the same movie every day (even our favorite one)
    - Food: some days need to pass to really enjoy our favorite food again (typically a week)
  - After playing an action
    - (mean) payoff temporarily decreases
    - then (slowly) increases back to a baseline
  - Standard MAB cannot capture this aspect
  - Blocking bandits are a special case.
  - First introduced by Immorlica & Kleinberg 2018.

Recharging Bandits

- Setting:
  - Set $\mathcal{K}$ of n arms
  - Each $i \in \mathcal{K}$ has a (mean) payoff function $p_i(\tau)$
    - τ: # of rounds passed since i was last played (called "delay")
    - $p_i(\tau)$ is monotone non-decreasing in τ
    - Polynomial and known recovery time $\tau_{\max}$ s.t.

$$p_i(\tau) = p_i(\tau_{\max}), \forall \tau > \tau_{\max}$$

Recharging Bandits

- Setting:
  - Set $\mathcal{K}$ of n arms
  - Each $i \in \mathcal{K}$ has a (mean) payoff function $p_i(\tau)$
    - τ: # of rounds passed since i was last played (called "delay")
    - $p_i(\tau)$ is monotone non-decreasing in τ
    - Polynomial and known recovery time $\tau_{\max}$ s.t.

$$p_i(\tau) = p_i(\tau_{\max}), \forall \tau > \tau_{\max}$$

  - Agent plays at most k arms per round (semi-bandit feedback)
    - For each arm played i (under delay τ), collect a payoff with mean $p_i(\tau)$

Recharging Bandits

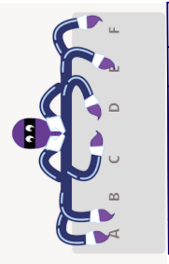
- Setting:
  - Set $\mathcal{K}$ of n arms
  - Each $i \in \mathcal{K}$ has a (mean) payoff function $p_i(\tau)$
    - τ: # of rounds passed since i was last played (called "delay")
    - $p_i(\tau)$ is monotone non-decreasing in τ
    - Polynomial and known recovery time $\tau_{\max}$ s.t.

$$p_i(\tau) = p_i(\tau_{\max}), \forall \tau > \tau_{\max}$$

  - Agent plays at most k arms per round (semi-bandit feedback)
    - For each arm played i (under delay τ), collect a payoff with mean $p_i(\tau)$
  - **Planning**: payoff function known (**NP**-hard in general)
  - **Learning**: payoff function initially unknown
- Goal: Minimize the ρ-regret for T rounds, where ρ is the best-known competitive guarantee for planning

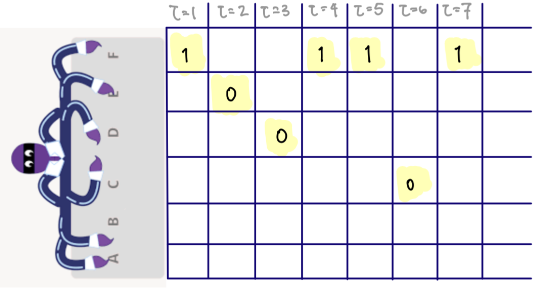
Model: Side observations on a graph

Full vs Bandit Information



$\tau=1$	$\tau=2$	$\tau=3$	$\tau=4$	$\tau=5$	$\tau=6$	$\tau=7$	
1	0	0	1	1	0	1	
1	0	1	1	0	1	1	
0	0	0	0	1	0	0	
1	0	0	1	1	0	1	
0	1	1	1	0	0	0	
0	0	1	0	1	1	1	

Full vs Bandit Information



Side Information

In many applications, information regime is between these two extremes:

- Related products: If you like songs by Tsaous, you may like songs by Tountas.
- Related users: Response by a user on a social network my provide information on connected users.
- etc.

Side observations on a Graph

- $G = (\mathcal{K}, E)$ undirected, unweighted graph
- $\mathcal{K} = \{1, ..., K\}$ actions/arms - nodes

- Playing an action i yields a stochastic reward X_i of initially unknown mean μ_i.

Let $\mathcal{K}_i$: neighbors of node i

- At each time t the player plays A_t and collects (and observes) $X_{A_t}(t)$.

Side observations on a Graph

- $G = (\mathcal{K}, E)$ undirected, unweighted graph
- $\mathcal{K} = \{1, ..., K\}$ actions/arms - nodes

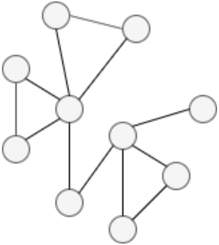
- Playing an action i yields a stochastic reward X_i of initially unknown mean μ_i.

Let $\mathcal{K}_i$: neighbors of node i

- At each time t the player plays A_t and collects (and observes) $X_{A_t}(t)$.
- The player also observes $X_j(t), \ \forall j \in \mathcal{K}_{A_t}$.
- Adversarial setting studied in Mannor & Shamir, 2011.

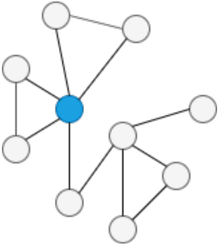
Side observations on a Graph

K actions of unknown mean rewards $\mu_1, ..., \mu_K$



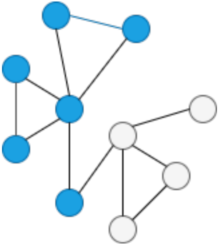
Side observations on a Graph

K actions of unknown mean rewards $\mu_1, ..., \mu_K$



Side observations on a Graph

K actions of unknown mean rewards $\mu_1, \ldots, \mu_K$



Gaussian Bandits with Side Observations

Model (introduced in Wu, György, and Szepesvári 2015):

- K Gaussian arms with (unknown) mean rewards $(\mu_1, \ldots, \mu_K)$
- Known feedback matrix $\Sigma = (\sigma_{i,j})_{i,j \in \mathcal{K}}$
- At each round t, by playing an action $i \in \mathcal{K}$ the player:
  - collects $X_{i,t} \sim \mathcal{N}(\mu_i, \sigma_{i,i}^2)$
  - observes $X_{j,t} \sim \mathcal{N}(\mu_j, \sigma_{i,j}^2)$ for each arm $j \in \mathcal{K}$
  - (rewards are realized independently)

Objective: Maximize the expected cumulative regret

Gaussian Bandits with Side Observations

- Wu, György, and Szepesvári 2015: provide asymptotically optimal regret for the special case where $\sigma_{i,j} \in \{\sigma, \infty\}$ which is equivalent to Graph-structured feedback.

General case:

- Can be modeled as a weighted graph $G = (\mathcal{K}, E, \Sigma)$
- $\mathcal{K} = \{1, \dots, K\}$ actions/nodes
- Edge (i, j) has weight σ_{ij} (can be ∞)

LP-based Lower Bound

Key Idea:

- Need to play each arm enough to distinguish best from second best (etc.)
- In usual setting, we need to play each suboptimal arm: $N_i(t)/\sigma_i^2 \geq 2/\Delta_i^2$ times.
- Now we need to account for different variances:

LP-based Lower Bound

Key Idea:

- Need to play each arm enough to distinguish best from second best (etc.)
- In usual setting, we need to play each suboptimal arm: $N_i(t)/\sigma_i^2 \geq 2/\Delta_i^2$ times.
- Now we need to account for different variances:

$$\sum_{j \in [K]} \frac{N_j(t)}{\sigma_{ji}^2} \geq 2/\Delta_i^2.$$

LP-based Lower Bound

Can use this to build a lower-bounding LP:

- Any algorithm must distinguish strictly suboptimal arms.
- Thus if arm j is played c_j times, for all i we must have:

$$\sum_{j \in [K]} \frac{c_j}{\sigma_{ji}^2} \geq 2/\Delta_i^2.$$

- The optimal algorithm will accomplish this while minimizing suboptimality: $\sum_{i \in \mathcal{K}} c_i \Delta_i(\mu)$

LP-based Lower Bound

Formulation: For any reward vector $\mu \in [0, \infty)^K$, we define:

$$C(\mu) = \left\{ c \in [0, \infty)^K : \begin{array}{l} \sum_{j \in \mathcal{K}} \frac{c_j}{\sigma_{ji}^2} \geq \frac{2}{\Delta_i^2(\mu)}, \forall i \neq i^*(\mu) \\ \sum_{j \in \mathcal{K}} \frac{c_j}{\sigma_{ji}^2} \geq \frac{2}{\Delta_{\min}^2(\mu)}, i = i^*(\mu) \end{array} \right\},$$

where $i^*(\mu) = \text{argmax}_{i \in \mathcal{K}} \mu_i$, $\Delta_i(\mu) = \max_{j \in \mathcal{K}} \mu_j - \mu_i$, and $\Delta_{\min}(\mu) = \min_{i \in \mathcal{K}, \Delta_i(\mu) > 0} \Delta_i(\mu)$.

Let the optimal solution:

$$c^* = \underset{c \in C(\mu)}{\arg\min} \sum_{i \in \mathcal{K}} c_i \Delta_i(\mu).$$

LP-based Lower Bound

Theorem

For environment (μ, Σ), *the regret of any consistent policy satisfies*

$$\liminf_{T \to \infty} \frac{R_T(\mu)}{\log T} \geq \sum_{i \in \mathcal{K}} c_i^* \Delta_i(\mu).$$

LP-based Algorithm

Estimating Empirical Means:

$$\begin{aligned}
\hat{\mu}(t) &= \sum_{\tau=1}^{t-1} \frac{X_\tau}{\sigma_{i_\tau}^2} \Big/ \sum_{\tau=1}^{t-1} \frac{1}{\sigma_{i_\tau}^2} \\
&= \sum_{j\in[K]} \sum_{\tau=1}^{t-1} \frac{X_\tau \mathbb{I}(i_\tau = j)}{\sigma_j^2} \Big/ \zeta(\tau),
\end{aligned}$$

where: $\zeta(\tau) = \sum_{j\in[K]} \frac{N_j(t)}{\sigma_j^2}$.

LP-based Algorithm

Algorithm Idea: Estimate the LP and at the same time implement its solution for exploration.

At each round t, the algorithm performs one of the following:

- Greedy exploitation: Play the arm of best estimated reward
- Uniform exploration: Ensure $C(\widehat{\mu})$ is “close” to $C(\mu)$
- LP-dictated exploration: Follow the actions indicated by (estimated) LP based on $C(\widehat{\mu})$

LP-based Algorithm

At each round t:

Greedy exploitation: If $\left(\frac{N_1(t)}{\log t}, \frac{N_2(t)}{\log t}, \ldots, \frac{N_K(t)}{\log t}\right) \in C(\widehat{\mu})$, then play

$$\boxed{i_t \leftarrow \arg\max_{i \in \mathcal{K}} \widehat{\mu}_i(t)}$$

LP-based Algorithm

At each round t:

n_e: # exploration rounds (initialized at 0)

Uniform exploration: If $\left(\frac{N_1(t)}{\log t}, \frac{N_2(t)}{\log t}, \ldots, \frac{N_K(t)}{\log t}\right) \notin C(\widehat{\mu})$ and

$$\min_{i \in \mathcal{K}} \sum_{\tau=1}^{t-1} \frac{1}{\sigma^2_{i_\tau i}} < o(n_e(t)) \text{ (not uniformly explored)}$$

then play

$$\boxed{i_t \leftarrow \arg\min_{k \in \mathcal{K}} \sigma^2_{ki}, \text{ where } i = \arg\min_{k \in \mathcal{K}} \sum_{\tau=1}^{t-1} \frac{1}{\sigma^2_{i_\tau k}},}$$

and increase n_e by 1

LP-based Algorithm

At each round t:

LP-dictated exploration:

If $\left(\frac{N_1(t)}{\log t}, \frac{N_2(t)}{\log t}, \ldots, \frac{N_K(t)}{\log t}\right) \notin C(\widehat{\mu})$ and arms uniformly explored, then

- Compute $c^*(\widehat{\mu}(t)) \leftarrow \arg\min_{c \in C(\widehat{\mu}(t))} \sum_{i \in \mathcal{K}} c_i \Delta_i(\widehat{\mu}(t))$
- Play arm

$$\boxed{i_t = i \text{ with } N_i(t) < c_i^*(\widehat{\mu}(t)) \log t,}$$

and increase n_e by 1

LP-based Algorithm

Theorem

The regret of the above algorithm satisfies

$$\limsup_{T\to\infty} \frac{R_T(\mu)}{\log T} \leq \sum_{j\in\mathcal{K}} \Delta_j(\mu) c_j^*(\mu) \qquad \textit{(up to constant factors)}$$

Omitted Details

“Asymptotically-Optimal Gaussian Bandits with Side Observations,” Atsidakou, Papadigenopoulos, **C**., Sanghavi, Shakkottai; ICML 2022
https://proceedings.mlr.press/v162/atsidakou22a.html

Parting thoughts

- Bandits are a well-explored framework.
- Classical results critically rely on certain assumptions, such as stationarity.
- Without these, many interesting problems still remain!

`constantine@utexas.edu`
`https://caramanis.github.io/`