Practical Algorithms for Multiplayer Bandits

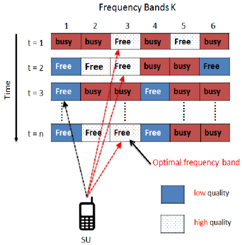
Emilie Kaufmann (ULille, CRIStAL)

Allerton Conference September 25th, 2019

Motivation : Cognitive Radio

Goal : allow radio devices to smartly select communication channels in frequency bandits already used by other devices

- licensed bands : Opportunistic Spectrum Access [Jouini et al. 09]
  arm ↔ availability of a chanel from primary users

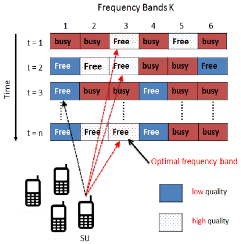


- un-licensed bands : IoT communications
  arm ↔ background traffic

Motivation : Cognitive Radio

Goal : allow radio devices to smartly select communication channels in frequency bandits already used by other devices

- licensed bands : Opportunistic Spectrum Access [Jouini et al. 09]
  arm ↔ availability of a chanel from primary users



- un-licensed bands : IoT communications
  arm ↔ background traffic

➜ what if **multiple device** want to communicate at the same time ?

Outline

1. The multi-player bandit model
2. Homogeneous case : the Rand-Top-M algorithm
3. Heterogeneous case : M-ETC-Elim

Outline

The multi-player multi-armed bandit model

At round $t = 1, \dots, T$, each agent $m = 1, \dots, M$:

- selects arm $A^m(t)$ [based on **his past observation**],
- possibly experiment a collision

$$C^m(t) := \{\exists m' \neq m : A^{m'}(t) = A^m(t)\}$$

- and receives the reward

$$R^m(t) = \underbrace{X^m_{A^m(t),t}}_{\text{rewards of the chosen arm}} \times \underbrace{\left(1 - \mathbb{1}\left(C^m(t)\right)\right)}_{\text{...received if no collision occurs}} .$$

Channel qualities for agent m :

Channel 1	$X^m_{1,1}$	$X^m_{1,2}$	$\dots$	$X^m_{1,t}$	$\dots$	$X^m_{1,T}$	$\sim \mathcal{B}\left(\mu^m_1\right)$
Channel 2	$X^m_{2,1}$	$X^m_{2,2}$	$\dots$	$X^m_{2,t}$	$\dots$	$X^m_{2,T}$	$\sim \mathcal{B}\left(\mu^m_2\right)$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	
Channel K	$X^m_{K,1}$	$X^m_{K,2}$	$\dots$	$X^m_{K,t}$	$\dots$	$X^m_{K,T}$	$\sim \mathcal{B}\left(\mu^m_K\right)$

The multi-player multi-armed bandit model

At round $t = 1, \dots, T$, each agent $m = 1, \dots, M$:

- selects arm $A^m(t)$ [based on **his past observation**],
- possibly experiment a collision

$$C^m(t) := \{\exists m' \neq m : A^{m'}(t) = A^m(t)\}$$

- and receives the reward

$$R^m(t) = \underbrace{X^m_{A^m(t),t}}_{\text{rewards of the chosen arm}} \times \underbrace{(1 - \mathbb{1}(C^m(t)))}_{\text{...received if no collision occurs}} .$$

Goal : design an arm selection strategy for each agent maximizing the global reward of the system

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{m=1}^{M} R^m(t)\right]$$

Assumption : $X^m_{k,t} \sim$ Bernoulli distribution with mean μ^m_k

Two different setting

- **Homogeneous setting :** $\forall m \neq m', \mu_k^m = \mu_k^{m'} = \mu_k$

➜ optimal bandit algorithm + orthogonalization mechanism

Lilian Besson & E.K. *Multi-player bandit revisited*, ALT 2018

- **Heterogeneous setting :** agents may have different utilities

➜ jointly identify a near-optimal matching from agents to arms

Etienne Boursier, E.K., Abbas Mehrabian & Vianney Perchet
A Practical Algorithm for Multiplayer Bandits when Arm Means Vary Among Players., arXiv :1902.01239




Outline

Regret in the homogeneous case

Arms sorted by decreasing utility : $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_K$

$$R_{\boldsymbol{\mu}}(\mathcal{A}, T) := \underbrace{\left(\sum_{k=1}^{M} \mu_k\right) T}_{\text{oracle total reward}} - \mathbb{E}_{\boldsymbol{\mu}}^{\mathcal{A}}\left[\sum_{t=1}^{T}\sum_{m=1}^{M} R^m(t)\right]$$

Regret decomposition

$$R_{\boldsymbol{\mu}}(\mathcal{A}, T) = \sum_{k=M+1}^{K} (\mu_M - \mu_k)\mathbb{E}[N_k(T)] + \sum_{k=1}^{M}(\mu_k - \mu_M)\left(T - \mathbb{E}[N_k(T)]\right) + \sum_{k=1}^{K} \mu_k \mathbb{E}[\mathcal{C}_k(T)].$$

- $N_k(T)$ total number of selections of arm k
- $\mathcal{C}_k(T)$ total number of collisions experienced on arm k

Regret in the homogeneous case

Arms sorted by decreasing utility : $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_K$

$$R_{\boldsymbol{\mu}}(\mathcal{A}, T) := \underbrace{\left(\sum_{k=1}^{M} \mu_k\right) T}_{\text{oracle total reward}} - \mathbb{E}_{\boldsymbol{\mu}}^{\mathcal{A}}\left[\sum_{t=1}^{T}\sum_{m=1}^{M} R^m(t)\right]$$

Regret decomposition

$$R_{\boldsymbol{\mu}}(\mathcal{A}, T) \leq C \sum_{k=M+1}^{K} \mathbb{E}\left[N_k(T)\right] + D \sum_{k=1}^{M} \mathbb{E}\left[\mathcal{C}_k(T)\right].$$

We need to control :

- the number of selections of sub-optimal arms
- the number of collisions on optimal arms

The MC-Top-M algorithm

Feedback model

Agent m observes :

- the sensing information of the chosen arm, $X_{A^m(t),t}$
- his reward $R^m(t)$

The MC-Top-M algorithm

Feedback model

Agent m observes :

- the sensing information of the chosen arm, $X_{A^m(t),t}$
- his reward $R^m(t)$

At round t, player m uses his past sensing information to :

- compute an Upper Confidence Bound for each mean μ_k, $\mathrm{UCB}_k^m(t)$
- use the UCBs to estimate the M best arms

$$\hat{M}^m(t) := \{\text{arms with } M \text{ largest } \mathrm{UCB}_k^m(t)\}$$

Other UCB-based algorithms :

TDFS [Lui and Zhao 2010], Rho-Rand [Anandkumar et al. 2011]

The MC-Top-M algorithm

Two simple ideas :

- ➜ always pick $A^m(t) \in \hat{M}^m(t-1)$
- ➜ try not to switch arm too often

$$s^m(t) = \{\text{player } m \text{ is ``fixed'' at the end of round } t\}$$

$\rightarrow$ inspired by Musical Chair [Rosenski et al. 2016]

The MC-Top-M algorithm

Two simple ideas :

- always pick $A^m(t) \in \hat{M}^m(t-1)$
- try not to switch arm too often

$$s^m(t) = \{\text{player } m \text{ is "fixed" at the end of round } t\}$$

$\rightarrow$ inspired by Musical Chair [Rosenski et al. 2016]

MC-Top-M

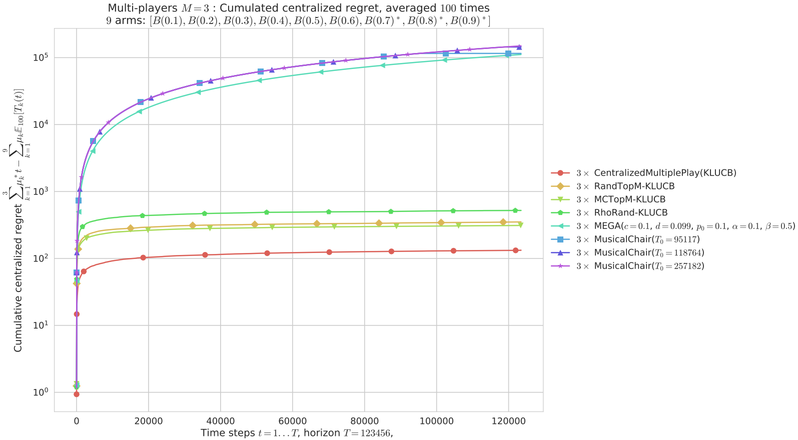
- if $A^m(t-1) \notin \hat{M}^m(t-1)$,
  set $s^m(t) = \text{False}$ and carefully select a new arm in $\hat{M}^m(t-1)$.
- else if $\overline{s^m(t-1)} \cap C^m(t-1)$, pick a new arm at random

$$A^m(t) \sim \mathcal{U}(\hat{M}^m(t-1)) \text{ and } s^m(t) = \text{False}$$

- else, draw the previous arm, and fix on it

$$A^m(t) = A^m(t-1) \text{ and } s^m(t) = \text{True}$$

Practical results



(log scale on the y axis)

Theoretical results

MC-Top-M with kl-based confidence intervals [Cappé et al. 13]

$$\mathrm{UCB}_k^m(t) = \max\left\{q : N_k^m(t)\mathrm{kl}\left(\hat{\mu}_k^m(t), q\right) \leq \ln(t)\right\},$$

where $\mathrm{kl}(x,y) = \mathrm{KL}\left(\mathcal{B}(x), \mathcal{B}(y)\right) = x \ln \frac{x}{y} + (1-x)\ln\frac{1-x}{1-y}$.

Control of the sub-optimal selections

For all $k \in \{M+1, \dots, K\}$,

$$\mathbb{E}[N_k^m(T)] \leq \frac{\ln(T)}{\mathrm{kl}(\mu_k, \mu_M)} + C_{\boldsymbol{\mu}}\sqrt{\ln(T)}.$$

Control of the collisions

$$\mathbb{E}\left[\sum_{k=1}^{K} \mathcal{C}_k(T)\right] \leq \left(\sum_{a,b:\mu_a<\mu_b} \frac{M^2\,(2M+1)}{\mathrm{kl}(\mu_a,\mu_b)}\right)\ln(T) + O(\ln T).$$

logarithmic regret !

Theoretical results

MC-Top-M with kl-based confidence intervals [Cappé et al. 13]

$$\mathrm{UCB}_k^m(t) = \max\left\{q : N_k^m(t)\mathrm{kl}\left(\hat{\mu}_k^m(t), q\right) \leq \ln(t)\right\},$$

where $\mathrm{kl}(x, y) = \mathrm{KL}\left(\mathcal{B}(x), \mathcal{B}(y)\right) = x \ln \frac{x}{y} + (1-x) \ln \frac{1-x}{1-y}$.

Control of the sub-optimal selections

For all $k \in \{M+1, \dots, K\}$,

$$\mathbb{E}[N_k^m(T)] \leq \frac{\ln(T)}{\mathrm{kl}(\mu_k, \mu_M)} + C_{\boldsymbol{\mu}}\sqrt{\ln(T)}.$$

Control of the collisions

$$\mathbb{E}\left[\sum_{k=1}^{K} \mathcal{C}_k(T)\right] \leq \left(\sum_{a,b:\mu_a<\mu_b} \frac{M^2(2M+1)}{\mathrm{kl}(\mu_a, \mu_b)}\right) \ln(T) + O(\ln T).$$

logarithmic regret !

Optimality ?

Control of the sub-optimal selections

For all $k \in \{M+1, \ldots, K\}$,

$$\mathbb{E}[N_k^m(T)] \leq \frac{\ln(T)}{\mathrm{kl}(\mu_k, \mu_M)} + C_{\boldsymbol{\mu}}\sqrt{\ln(T)}.$$

- is this the best we can do ?

Optimality ?

Control of the sub-optimal selections

For all $k \in \{M+1, \dots, K\}$,

$$\mathbb{E}[N_k^m(T)] \leq \frac{\ln(T)}{\mathrm{kl}(\mu_k, \mu_M)} + C_{\boldsymbol{\mu}} \sqrt{\ln(T)}.$$

- is this the best we can do ? **NO !**

➜ best achievable sub-optimal selections for an algorithm "not exploiting too much the collision information"

- one can propose an algorithm such that

$$\sum_{m=1}^{M} \mathbb{E}[N_k^m(T)] \simeq \frac{\ln(T)}{\mathrm{kl}(\mu_k, \mu_M)}$$

by exploiting forced collisions to perform implicit communications

Boursier and Perchet, SIC-MMAB, NeurIPS 2019

Outline

Regret in the heterogeneous case

Utility matrix :
$$\boldsymbol{\mu} = (\mu_k^m)_{\substack{1 \leq k \leq K \\ 1 \leq m \leq M}}$$

Given $\pi : [M] \to [K]$ a matching from agent to arms,

$$U(\pi) := \sum_{m=1}^{K} \mu_{\pi(m)}^m \quad \text{and} \quad U_\star = \max_\pi U(\pi).$$

Regret

$$R_{\boldsymbol{\mu}}(\mathcal{A}, T) = TU_\star - \mathbb{E}_{\boldsymbol{\mu}}^{\mathcal{A}}\left[\sum_{t=1}^{T}\sum_{m=1}^{M} R^m(t)\right]$$

M-ETC-Elim

Feedback model

Agent m observes :

- the collision indicator $\mathbb{1}(C^m(t))$
- his reward $R^m(t)$

M-ETC-Elim

Feedback model

Agent m observes :

- the collision indicator $\mathbb{1}(C^m(t))$
- his reward $R^m(t)$

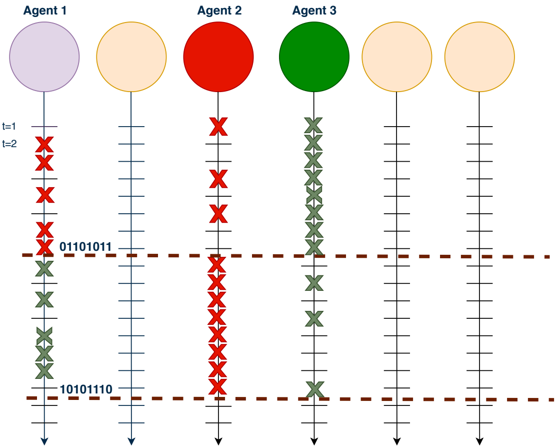
Main ingredients :

- Initialization phase : assigns M different arms (and M ranks) to the M agents. Agent 1 is the **Leader**, other are **Followers**
- Exploration phases : each agent is assigned a list of arms to explore (= sample a certain number of times)
- Communication phases :
  - **Leader** $\longrightarrow$ **Follower** : send the list of arm to explore
  - **Follower** $\longrightarrow$ **Leader** : report the empirical mean of the explored arms

Communications ?

Idea : leverage forced collisions to perform communications

[Boursier et al. 19, Nayyar et al. 18, Tibrewal et al. 19]



Agent can send **sequence of bits** to each other.

- transmit 0 : select his own communicating arm
- transmit 1 : select the other's communicating arm

Ranks → **order** of communications

The algorithm

- Initialization : **Leader** and **Followers** are designated, players all have communicating arms and ranks. Leader initializes candidate edges

$$\mathcal{E} = \{(m, k), m \in \{1, \ldots, M\}, k \in \{1, \ldots, K\}\}$$

- For $p = 1, 2, \ldots$
  - ➜ **Leader performs computations** based on estimates $(\tilde{\mu}_k^m)_{(m,k)\in\mathcal{E}}$
    $\tilde{\pi}_\star = \mathrm{argmax}_\pi \tilde{U}(\pi)$ and $\tilde{\pi}_{(m,k)} = \mathrm{argmax}_{\{\pi:\pi(m)=k\}} \tilde{U}(\pi)$
    - if $\tilde{U}(\tilde{\pi}_\star) - \tilde{U}(\tilde{\pi}_{(m,k)}) > 4M\sqrt{\ln(2M^2KT^2)/2^{1+p^c}}$, remove (m, k) from $\mathcal{E}$
    - else, add $\tilde{\pi}_{(m,k)}$ to $\mathcal{C}$
  - ➜ **Leader communicate to eack Follower** the list of arms to explore

$$\mathcal{L}_m = \{\pi(m) \text{ for } \pi \in \mathcal{C}\}$$

  - ➜ **All agents explore** each arm in their list 2^{p^c} times
  - ➜ **Follower communicate to the Leader**, for their explored arms,

$$\tilde{\mu}_k^m : (p^c + 1)/2 \text{ most significant bits of } \hat{\mu}_k^m$$

- in case $|\mathcal{C}| = 1$, the agents enter an **exploitation phase**

Theoretical results

Theorem

(a) M-ETC-Elim with parameter $c \in \{1, 2, \ldots\}$ satisfies

$$R_{\boldsymbol{\mu}}(T) = O\left(MK\left(\frac{M^2 \ln(KT)}{\Delta}\right)^{1+1/c}\right).$$

(b) If the maximum matching is unique, for M-ETC-Elim with $c = 1$

$$R_{\boldsymbol{\mu}}(T) = O\left(\frac{M^3 K \ln(KT)}{\Delta}\right).$$

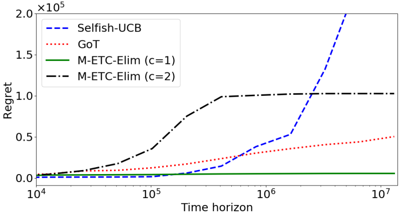
where $\Delta = \min_{\pi : U(\pi) < U_\star} (U_\star - U(\pi)) > 0$.

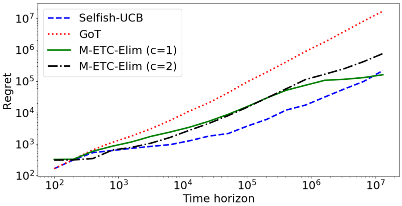
- ➜ an algorithm achieving $O(\ln^{1+\kappa}(T))$ regret for every $\kappa > 0$
- ➜ logarithmic regret in the presence of a unique maximum matching !

improves over [Bistritz and Leshem, NeurIPS 18]

Practical results

$$U_1 = \begin{pmatrix} 0.1 & 0.05 & 0.9 \\ 0.1 & 0.25 & 0.3 \\ 0.4 & 0.2 & 0.8 \end{pmatrix} \qquad U_2 = \begin{pmatrix} 0.5 & 0.49 & 0.39 & 0.29 & 0.5 \\ 0.5 & 0.49 & 0.39 & 0.29 & 0.19 \\ 0.29 & 0.19 & 0.5 & 0.499 & 0.39 \\ 0.29 & 0.49 & 0.5 & 0.5 & 0.39 \\ 0.49 & 0.49 & 0.49 & 0.49 & 0.5 \end{pmatrix}$$





Conclusion

We proposed :

- efficient algorithms with (quasi) logarithmic regret for the homogeneous and heterogeneous setting...
- ... under different feedback model

Future work :

- efficient algorithm with provable regret guarantees when each player only observes the reward $R^m(t)$