Aggregation of MAB Learning Algorithms for OSA

Lilian Besson

Advised by Christophe Moy Émilie Kaufmann

PhD Student
Team SCEE, IETR, CentraleSupélec, Rennes
& Team SequeL, CRIStAL, Inria, Lille

IEEE WCNC - 16th April 2018

Introduction

- Cognitive Radio (CR) is known for being one of the possible solution to tackle the spectrum scarcity issue
- Opportunistic Spectrum Access (OSA) is a good model for CR problems in **licensed bands**
- Online learning strategies, mainly using multi-armed bandits (MAB) algorithms, were recently proved to be efficient `[Jouini 2010]`
- But there is many different MAB algorithms... which one should you choose in practice?

$\Longrightarrow$ we propose to use an online learning algorithm to also decide which algorithm to use, to be more robust and adaptive to unknown environments.

Outline

1. Opportunistic Spectrum Access
2. Multi-Armed Bandits
3. MAB algorithms
4. Aggregation of MAB algorithms
5. Illustration

Please

Ask questions *at the end* if you want!

1. Opportunistic Spectrum Access

- Spectrum scarcity is a well-known problem
- Different range of solutions...
- Cognitive Radio is one of them
- Opportunistic Spectrum Access is a kind of cognitive radio

Communication & interaction model

- Primary users are occupying K radio channels
- Secondary users can sense and exploit free channels: want to **explore** the channels, and learn to **exploit** the best one
- Discrete time for everything $t \geq 1, t \in \mathbb{N}$

2. Multi-Armed Bandits

Model

- Again $K \geq 2$ resources (*e.g.*, channels), called **arms**
- Each time slot $t = 1, \ldots, T$, you must choose one arm, denoted $A(t) \in \{1, \ldots, K\}$
- You receive some reward $r(t) \sim \nu_k$ when playing $k = A(t)$
- **Goal:** maximize your sum reward $\sum_{t=1}^{T} r(t)$
- Hypothesis: rewards are stochastic, of mean μ_k. *E.g.*, Bernoulli

Why is it famous?

Simple but good model for **exploration/exploitation** dilemma.

3. MAB algorithms

- Main idea: index $I_k(t)$ to approximate the quality of each arm k
- First example: *UCB algorithm*
- Second example: *Thompson Sampling*

3.1 Multi-Armed Bandit algorithms

Often *index* based

- Keep *index* $I_k(t) \in \mathbb{R}$ for each arm $k = 1, \dots, K$
- Always play $A(t) = \arg\max I_k(t)$
- $I_k(t)$ should represent our belief of the *quality* of arm k at time t

Example: "Follow the Leader"

- $X_k(t) := \sum_{s<t} r(s)\mathbf{1}(A(s) = k)$ sum reward from arm k
- $N_k(t) := \sum_{s<t} \mathbf{1}(A(s) = k)$ number of samples of arm k
- And use $I_k(t) = \hat{\mu}_k(t) := \frac{X_k(t)}{N_k(t)}$.

3.2 First example of algorithm *Upper Confidence Bounds* algorithm (UCB)

- Instead of using $I_k(t) = \frac{X_k(t)}{N_k(t)}$, add an exploration term

$$I_k(t) = \frac{X_k(t)}{N_k(t)} + \sqrt{\frac{\alpha \log(t)}{2N_k(t)}}$$

Parameter α: tradeoff exploration *vs* exploitation

- Small α: focus more on **exploitation**
- Large α: focus more on **exploration**

Problem: how to choose "the good α" for a certain problem?

3.3 Second example of algorithm *Thompson sampling* (TS)

- Choose an initial belief on μ_k (uniform) and a prior p^t (*e.g.*, a Beta prior on $[0, 1]$)
- At each time, update the prior p^{t+1} from p^t using Bayes theorem
- And use $I_k(t) \sim p^t$ as *random* index

Example with Beta prior, for binary rewards

- $p^t = \mathrm{Beta}(1 + \text{nb successes}, 1 + \text{nb failures})$.
- Mean of $p^t = \frac{1+X_k(t)}{2+N_k(t)} \simeq \hat{\mu}_k(t)$.

How to choose "the good prior" for a certain problem?

4. Aggregation of MAB algorithms

Problem

- How to choose which algorithm to use?
- But also... Why commit to one only algorithm?

Solutions

- Offline benchmarks?
- Or online selections from a pool of algorithms?

$\hookrightarrow$ Aggregation?

Not a new idea, studied from the 90s in the ML community.

- Also use online learning to *select the best algorithm*!

5. Some illustrations

- Artificial simulations of stochastic bandit problems
- Bernoulli bandits but not only
- Pool of different algorithms (UCB, Thompson Sampling etc)
- Compared with other state-of-the-art algorithms for *expert aggregation* (Exp4, CORRAL, LearnExp)
- What is plotted it the *regret* for problem of means $\mu_1, \ldots, \mu_K$:

$$R_T^{\mu}(\mathcal{A}) = \max_k (T\mu_k) - \sum_{t=1}^{T} \mathbb{E}[r(t)]$$

- Regret is known to be lower-bounded by $C(\mu)\log(T)$
- and upper-bounded by $C'(\mu)\log(T)$ for efficient algorithms

On a simple Bernoulli problem

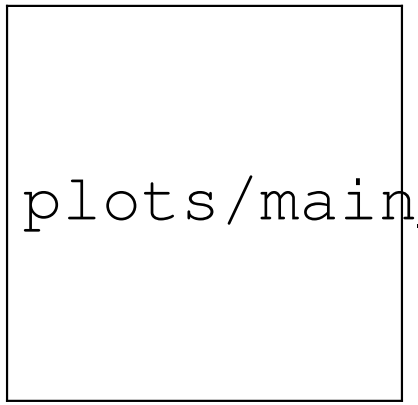

Figure 1: bg original 105%

On a “hard” Bernoulli problem

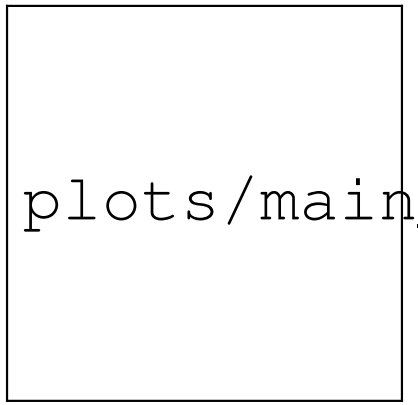

Figure 2: bg original 105%

On a mixed problem

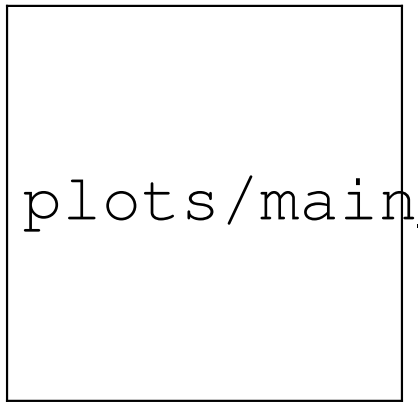

Figure 3: bg original 105%

Conclusion (1/2)

- Online learning can be a powerful tool for Cognitive Radio, and many other real-world applications
- Many formulation exist, a simple one is the Multi-Armed Bandit
- Many algorithms exist, to tackle different situations
- It's hard to know before hand which algorithm is efficient for a certain problem...
- Online learning can also be used to select *on the run* which algorithm to prefer, for a specific situation!

Conclusion (2/2)

- Our algorithm **Aggregator** is efficient and easy to implement
- For N algorithms $\mathcal{A}_1, \ldots, \mathcal{A}_N$, it costs $\mathcal{O}(N)$ memory, and $\mathcal{O}(N)$ extra computation time at each time step
- For stochastic bandit problem, it outperforms empirically the other state-of-the-arts (Exp4, CORRAL, LearnExp).

See our paper

`HAL.Inria.fr/hal-01705292`

See our code for experimenting with bandit algorithms

Python library, open source at `SMPyBandits.GitHub.io`

Thanks for listening!