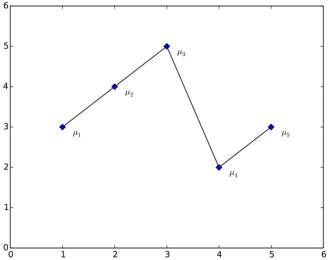
Discrete PAC Optimization: Application to MCTS

Aurélien Garivier, Emilie Kaufmann and Wouter Koolen

SPADRO Meeting
May 17th, 2016

Generic PAC optimization

- K Bernoulli distributions that can be sampled from
- a question $\mathcal{Q}$ about their means $\mu_1, \ldots, \mu_K$ (answer A^*)



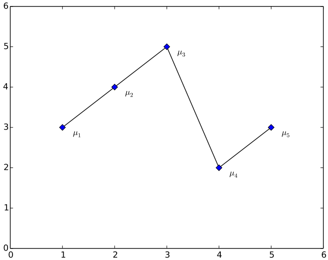
Goal: design a sequential decision strategy

sampling rule (A_t) / stopping rule τ / answering rule $\hat{A}$

such that $\mathbb{P}\left(\hat{A} = A^*\right) \geq 1 - \delta$, $\quad$ (δ − PAC algorithm) and
$\mathbb{E}[\tau]$ as small as possible.

Example: Best Arm Identification in bandit models

$\mathcal{Q}$: Which arm has highest mean? i.e. find $a^* = \text{argmax}_a \ \mu_a$



The sequential decision strategy:

- sampling rule: arm A_t chosen at time t ($\Rightarrow X_t \sim \mathcal{B}(\mu_{A_t})$)
- stopping rule τ
- recommendation rule $\hat{a}$ (based on τ samples)

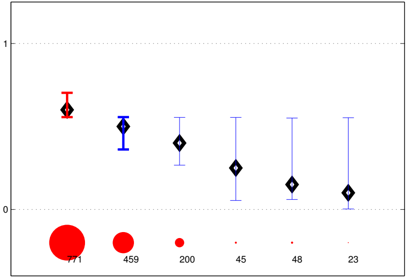
such that

$$\mathbb{P}(\hat{a} = a^*) \geq 1 - \delta, \quad \text{and} \quad \mathbb{E}[\tau] \ \text{ as small as possible}$$

LUCB: an algorithm for Best Arm Identification

An algorithm based on confidence intervals

$$\mathcal{I}_a(t) = [\mathrm{LCB}_a(t), \mathrm{UCB}_a(t)].$$



- At round t, draw

$$L_t = \underset{a}{\arg\max}\ \hat{\mu}_a(t)$$
$$C_t = \underset{a \neq L_t}{\arg\max}\ \mathrm{UCB}_a(t)$$

- Stop at round t if

$$\mathrm{LCB}_{L_t}(t) > \mathrm{UCB}_{C_t}(t)$$

Theorem [Kalyanakrishan et al.]

For well chosen confidence intervals, LUCB is δ-PAC and

$$\mathbb{E}[\tau] = O\left(\left[\frac{1}{(\mu_1 - \mu_2)^2} + \sum_{a=2}^{K} \frac{1}{(\mu_1 - \mu_a)^2}\right] \log(1/\delta)\right)$$

Optimal best arm identification

Let $d(x, y) = \mathrm{KL}(\mathcal{B}(x), \mathcal{B}(y))$.

Theorem

For any δ-PAC algorithm,

$$\mathbb{E}_{\boldsymbol{\mu}}[\tau] \geq T^*(\boldsymbol{\mu}) \log\left(\frac{1}{2.4\delta}\right),$$

where

$$T^*(\boldsymbol{\mu})^{-1} = \sup_{w \in \Sigma_K} \inf_{\boldsymbol{\lambda} \in \mathrm{Alt}(\boldsymbol{\mu})} \left(\sum_{a=1}^{K} w_a d(\mu_a, \lambda_a)\right).$$

Moreover, we propose a δ-PAC algorithm such that

$$\limsup_{\delta \to 0} \frac{\mathbb{E}_{\boldsymbol{\mu}}[\tau_\delta]}{\log(1/\delta)} = T^*(\boldsymbol{\mu})$$

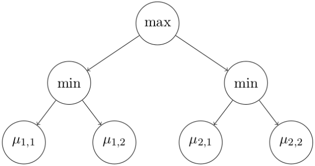
A. Garivier, E. Kaufmann, Optimal Best Arm Identification with Fixed Confidence, COLT 2016

Towards another discrete PAC optimization problem

Imagine a two-player game in which

- when A chooses action $i \in \{1, \dots, K\}$
- and then player B choose action $j \in \{1, \dots, K_i\}$,

the probability that A wins is $\mu_{i,j}$.



Best action for A given that B is strategic:

$$i^* \in \underset{i \in \{1,\dots,K\}}{\text{argmax}} \ \min_{j \in \{1,\dots,K_i\}} \ \mu_{i,j}.$$

(*maximin action*)

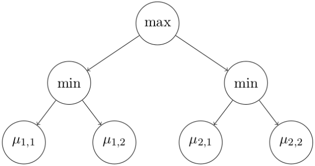
Goal: Learn i^* by sequentially choosing pairs of actions (i, j) and observing samples from $\mathcal{B}(\mu_{i,j})$ ("rollouts")

Towards another discrete PAC optimization problem

Imagine a two-player game in which

- when A chooses action $i \in \{1, \dots, K\}$
- and then player B choose action $j \in \{1, \dots, K_i\}$,

the probability that A wins is $\mu_{i,j}$.



Best action for A given that B is strategic:

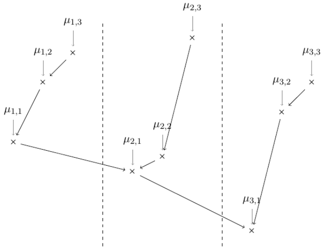
$$i^* \in \underset{i \in \{1,\dots,K\}}{\text{argmax}} \ \underset{j \in \{1,\dots,K_i\}}{\min} \ \mu_{i,j}.$$

(*maximin action*)

Goal: Learn i^* by sequentially choosing pairs of actions (i,j) and observing samples from $\mathcal{B}(\mu_{i,j})$ ("rollouts") $\Rightarrow$ Depth 2 MCTS

Maximin action identification

$\mathcal{Q}$: What is the maximin action? i.e. find $i^* = \arg\max_i \min_j \ \mu_{i,j}$



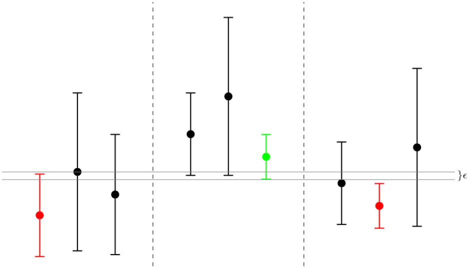
Build a strategy $(P_t, \tau, \hat{\imath})$ such that

$$\forall \boldsymbol{\mu}, \ \mathbb{P}_{\boldsymbol{\mu}} \left(\min_{j \in \{1 \dots K_{i^*}\}} \mu_{i^*,j} - \min_{j \in \{1 \dots K_{\hat{\imath}}\}} \mu_{\hat{\imath},j} \leq \epsilon \right) \geq 1 - \delta,$$

and $\mathbb{E}_{\boldsymbol{\mu}}[\tau]$ is as small as possible.

A. Garivier, E. Kaufmann, W. Koolen, *Maximin Action Identification: a new bandit framework for games*, COLT 2016

The Maximin-LUCB algorithm



- Pick one representative per action $P_i = (i, j_i)$,

$$j_i = \underset{j}{\arg\max}\, \mathrm{LCB}_{(i,j)}(t)$$

- Letting $\hat{i}(t) = \arg\max_i \min_j \hat{\mu}_{(i,j)}(t)$, draw

$$L_t = (\hat{i}(t), j_{\hat{i}(t)}) \quad \text{and} \quad C_t = \underset{P \in \{(i,j_i)\}_{i \neq \hat{i}(t)}}{\arg\max} \mathrm{UCB}_P(t)$$

- Stop if $\mathrm{LCB}_{L_t}(t) > \mathrm{UCB}_{C_t}(t) - \epsilon$

Sample complexity analysis

$$\mathrm{LCB}_P(t) = \hat{\mu}_P(t) - \sqrt{\frac{\beta(t,\delta)}{2N_P(t)}}, \quad \mathrm{UCB}_P(t) = \hat{\mu}_P(t) + \sqrt{\frac{\beta(t,\delta)}{2N_P(t)}}$$

Theorem (two actions per player)

Let $\alpha > 0$. There exists $C > 0$ such that for the choice

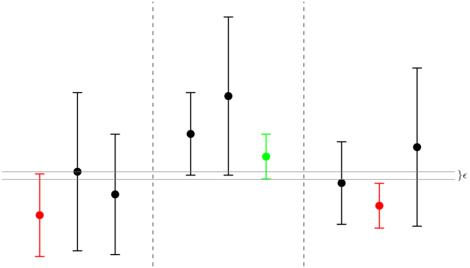
$$\beta(t,\delta) = \log(Ct^{1+\alpha}/\delta),$$

M-LUCB is δ-PAC and

$$\limsup_{\delta \to 0} \frac{\mathbb{E}_{\boldsymbol{\mu}}[\tau_\delta]}{\log(1/\delta)} \le 8(1+\alpha)H^*(\boldsymbol{\mu})$$

$$H^*(\boldsymbol{\mu}) = \frac{2}{(\mu_{1,1}-\mu_{2,1})^2} + \frac{1}{(\mu_{1,2}-\mu_{2,1})^2} + \frac{1}{\max\left[(\mu_{1,1}-\mu_{2,1})^2, (\mu_{2,2}-\mu_{2,1})^2\right]}$$

Perspective on M-LUCB



- Pick one representative per action $P_i = (i, j_i)$,

$$j_i = \underset{j}{\arg\max}\, \mathrm{LCB}_{(i,j)}(t)$$

- Perform a LUCB step on $(P_1, \dots, P_K)$

$\Rightarrow$ Use a better BAI algorithm ?

$\Rightarrow$ Can we keep this "representative" idea beyond depth 2?