Introduction to Reinforcement Learning
Part 3: Exploration for decision making, Application to games, optimization, and planning

Rémi Munos

SequeL project: Sequential Learning
http://researchers.lille.inria.fr/∼munos/

INRIA Lille - Nord Europe

Machine Learning Summer School, September 2011, Bordeaux

Outline of Part 3

Exploration for sequential decision making:
Application to games, optimization, and planning

- The stochastic bandit: UCB
- The adversarial bandit: EXP3
- Populations of bandits
  - Computation of equilibrium in games. Application to Poker
  - Hierarchical bandits. MCTS and application to Go.
- Optimism for decision making
  - Lipschitz optimization
  - Lipschitz bandits
  - Optimistic planning in MDPs

The stochastic multi-armed bandit problem

Setting:

- Set of K arms, defined by distributions ν_k (with support in $[0, 1]$), whose law is unknown,
- At each time t, choose an arm k_t and receive reward $x_t \overset{i.i.d.}{\sim} \nu_{k_t}$.
- **Goal**: find an arm selection policy such as to maximize the expected sum of rewards.

Exploration-exploitation tradeoff:

- **Explore**: learn about the environment
- **Exploit**: act optimally according to our current beliefs

The regret

Definitions:

- Let $\mu_k = \mathbb{E}[\nu_k]$ be the expected value of arm k,
- Let $\mu^* = \max_k \mu_k$ the best expected value,
- The cumulative expected **regret**:

$$R_n \stackrel{\text{def}}{=} \sum_{t=1}^{n} \mu^* - \mu_{k_t} = \sum_{k=1}^{K} (\mu^* - \mu_k) \sum_{t=1}^{n} \mathbf{1}\{k_t = k\} = \sum_{k=1}^{K} \Delta_k n_k,$$

where $\Delta_k \stackrel{\text{def}}{=} \mu^* - \mu_k$, and n_k the number of times arm k has been pulled up to time n.

Goal: Find an arm selection policy such as to minimize R_n.

Proposed solutions

This is an old problem! [Robbins, 1952] Maybe surprisingly, not fully solved yet!
Many proposed strategies:

- ϵ-**greedy exploration**: choose apparent best action with proba $1-\epsilon$, or random action with proba ϵ,
- **Bayesian exploration**: assign prior to the arm distributions and select arm according to the posterior distributions (Gittins index, Thompson strategy, ...)
- **Softmax exploration**: choose arm k with proba $\propto \exp(\beta \widehat{X}_k)$ (ex: EXP3 algo)
- **Follow the perturbed leader**: choose best perturbed arm
- **Optimistic exploration**: select arm with highest upper bound

The UCB algorithm

Upper Confidence Bound algorithm [Auer, Cesa-Bianchi, Fischer, 2002]: at each time n, select the arm k with highest $B_{k,n_k,n}$ value:

$$B_{k,n_k,n} \stackrel{\mathrm{def}}{=} \underbrace{\frac{1}{n_k}\sum_{s=1}^{n_k} x_{k,s}}_{\widehat{X}_{k,n_k}} + \underbrace{\sqrt{\frac{3\log(n)}{2n_k}}}_{c_{n_k,n}},$$

with:

- n_k is the number of times arm k has been pulled up to time n
- $x_{k,s}$ is the s-th reward received when pulling arm k.

Note that

- Sum of an *exploitation term* and an *exploration term*.
- $c_{n_k,n}$ is a confidence interval term, so $B_{k,n_k,n}$ is a UCB.

Intuition of the UCB algorithm

Idea:

- **"Optimism in the face of uncertainty"** principle
- Select the arm with highest upper bound (on the true value of the arm, given what has been observed so far).
- The B-values $B_{k,s,t}$ are UCBs on μ_k. Indeed:

$$\mathbb{P}\Big(\widehat{X}_{k,s} - \mu_k \geq \sqrt{\frac{3\log(t)}{2s}}\Big) \leq \frac{1}{t^3},$$

$$\mathbb{P}\Big(\widehat{X}_{k,s} - \mu_k \leq -\sqrt{\frac{3\log(t)}{2s}}\Big) \leq \frac{1}{t^3}$$

Reminder of Chernoff-Hoeffding inequality:

$$\mathbb{P}(\widehat{X}_{k,s} - \mu_k \geq \epsilon) \leq e^{-2s\epsilon^2}$$

$$\mathbb{P}(\widehat{X}_{k,s} - \mu_k \leq -\epsilon) \leq e^{-2s\epsilon^2}$$

Regret bound for UCB

Proposition 1.

Each sub-optimal arm k is visited in average, at most:

$$\mathbb{E}n_k(n) \leq 6\frac{\log n}{\Delta_k^2} + 1 + \frac{\pi^2}{3}$$

times (where $\Delta_k \stackrel{\text{def}}{=} \mu^* - \mu_k > 0$).

Thus the expected regret is bounded by:

$$\mathbb{E}R_n = \sum_k \mathbb{E}[n_k]\Delta_k \leq 6 \sum_{k:\Delta_k>0} \frac{\log n}{\Delta_k} + K(1 + \frac{\pi^2}{3}).$$

Intuition of the proof

Let k be a sub-optimal arm, and k^* be an optimal arm. At time n, if arm k is selected, this means that

$$\begin{aligned}
B_{k,n_k,n} &\geq B_{k^*,n_{k^*},n} \\
\widehat{X}_{k,n_k} + \sqrt{\frac{3\log(n)}{2n_k}} &\geq \widehat{X}_{k^*,n_{k^*}} + \sqrt{\frac{3\log(n)}{2n_{k^*}}} \\
\mu_k + 2\sqrt{\frac{3\log(n)}{2n_k}} &\geq \mu^*, \text{ with high proba} \\
n_k &\leq \frac{6\log(n)}{\Delta_k^2}
\end{aligned}$$

Thus, if $n_k > \frac{6\log(n)}{\Delta_k^2}$, then there is only a small probability that arm k be selected.

Proof of Proposition 1

Write $u = \frac{6\log(n)}{\Delta_k^2} + 1$. We have:

$$\begin{aligned} n_k(n) &\leq u + \sum_{t=u+1}^{n} \mathbf{1}\{k_t = k; n_k(t) > u\} \\ &\leq u + \sum_{t=u+1}^{n} \Big[\sum_{s=u+1}^{t} \mathbf{1}\{\hat{X}_{k,s} - \mu_k \geq c_{t,s}\} + \sum_{s=1}^{t} \mathbf{1}\{\hat{X}_{k^*,s^*} - \mu_k \leq -c_{t,s^*}\} \Big] \end{aligned}$$

Now, taking the expectation of both sides,

$$\begin{aligned} \mathbb{E}[n_k(n)] &\leq u + \sum_{t=u+1}^{n} \Big[\sum_{s=u+1}^{t} \mathbb{P}\big(\hat{X}_{k,s} - \mu_k \geq c_{t,s}\big) + \sum_{s=1}^{t} \mathbb{P}\big(\hat{X}_{k^*,s^*} - \mu_k \leq -c_{t,s^*}\big) \Big] \\ &\leq u + \sum_{t=u+1}^{n} \Big[\sum_{s=u+1}^{t} t^{-3} + \sum_{s=1}^{t} t^{-3} \Big] \leq \frac{6\log(n)}{\Delta_k^2} + 1 + \frac{\pi^2}{3} \end{aligned}$$

Variants of UCB

[Audibert et al., 2008]

- **UCB with variance estimate:** Define the UCB:

$$B_{k,n_k,n} \stackrel{\text{def}}{=} \widehat{X}_{k,t} + \sqrt{2\frac{V_{k,n_k}\log(1.2n)}{n_k}} + \frac{3\log(1.2n)}{n_k}.$$

  Then the expected regret is bounded by:

$$\mathbb{E}R_n \leq 10\Big(\sum_{k:\Delta_k>0} \frac{\sigma_k^2}{\Delta_k} + 2\Big)\log(n).$$

- **PAC-UCB:** Let $\beta > 0$. Define the UCB:

$$B_{k,n_k} \stackrel{\text{def}}{=} \widehat{X}_{k,n_k} + \sqrt{\frac{\log(Kn_k(n_k+1)\beta^{-1})}{n_k}}.$$

  Then w.p. $1-\beta$, the regret is bounded by a constant:

$$R_n \leq 6\log(K\beta^{-1}) \sum_{k:\Delta_k>0} \frac{1}{\Delta_k}.$$

Upper and Lower bounds

UCB:

- Distribution-dependent: $\mathbb{E}R_n = O\Big(\sum_{k:\Delta_k>0} \frac{1}{\Delta_k} \log n \Big)$
- Distribution-independent: $\mathbb{E}R_n = O(\sqrt{Kn \log n})$.

Lower-bounds:

- Distribution-dependent [Lai et Robbins, 1985]:

$$\mathbb{E}R_n = \Omega\Big(\sum_{k:\Delta_k>0} \frac{\Delta_k}{KL(\nu_k || \nu^*)} \log n \Big)$$

- Distribution-independent [Cesa-Bianchi et Lugosi, 2006]:

$$\inf_{\text{Algo}} \sup_{\text{Problem}} R_n = \Omega(\sqrt{nK}).$$

Recent improvements in upper-bounds: optimal bounds!

- MOSS [Audibert & Bubeck, 2009]
- KL-UCB [Garivier & Cappé, 2011], [Maillard et al., 2011]

The adversarial bandit

The rewards are no more i.i.d., but arbitrary!
At time t, simultaneously

- The adversary assigns a reward $x_{k,t} \in [0,1]$ to each arm $k \in \{1, \dots, K\}$
- The player chooses an arm k_t

The player receives the corresponding reward x_{k_t}. His goal is to maximize the sum of rewards.

Can we expect to do almost as good as the best (constant) arm?

Time	1	2	3	4	5	6	7	8	...
Arm pulled	1	2	1	1	2	1	1	1	
Reward arm 1	**1**	0.7	**0.9**	**1**	1	**1**	**0.8**	**1**	
Reward arm 2	0.9	**0**	1	0	**0.4**	0	0.6	0	

Reward obtained: 6.1. Arm 1: 7.4, Arm 2: 2.9.
Regret w.r.t. best constant strategy: $7.4 - 6.1 = 1.3$.

Notion of regret

Define the regret:

$$R_n = \max_{k \in \{1,\ldots,K\}} \sum_{t=1}^{n} x_{k,t} - \sum_{t=1}^{n} x_{k_t}.$$

- Performance assessed in terms of the best constant strategy.
- Can we expect

$$\sup_{rewards} \mathbb{E} R_n / n \to 0?$$

- If the policy of the player is deterministic, there exists a reward sequence such that the performance is arbitrarily poor $\longrightarrow$ Need internal randomization.

EXP3 algorithm

EXP3 algorithm (Explore-Exploit using Exponential weights) [Auer et al, 2002]:

- $\eta > 0$ and $\beta > 0$ are two parameters of the algorithm.
- Initialize $w_1(k) = 1$ for all $k = 1, \dots, K$.
- At each round $t = 1, \dots, n$, player selects arm $k_t \sim p_t(\cdot)$, where

  $$p_t(k) = (1-\beta)\frac{w_t(k)}{\sum_{i=1}^K w_t(i)} + \frac{\beta}{K},$$

  with

  $$w_t(k) = e^{\eta \sum_{s=1}^{t-1} \tilde{x}_s(k)},$$

  where

  $$\tilde{x}_s(k) = \frac{x_s(k)}{p_s(k)} \mathbf{1}\{k_s = k\}.$$

Performance of EXP3

Proposition 2.

Let $\eta \leq 1$ and $\beta = \eta K$. We have $\mathbb{E} R_n \leq \frac{\log K}{\eta} + (e-1)\eta n K$. Thus, by choosing $\eta = \sqrt{\frac{\log K}{(e-1)nK}}$, it comes

$$\sup_{\textit{rewards}} \mathbb{E} R_n \leq 2.63\sqrt{nK \log K}.$$

Properties:

- If all rewards are provided to the learner, with a similar algorithms we have [Lugosi and Cesa-Bianchi, 2006]

$$\sup_{\textit{rewards}} \mathbb{E} R_n = O(\sqrt{n \log K}).$$

Proof of Proposition 2 [part 1]

Write $W_t = \sum_{k=1}^K w_k(t)$. Notice that

$$\mathbb{E}_{k_s \sim p_s}[\tilde{x}_s(k)] = \sum_{i=1}^K p_s(i) \frac{x_s(k)}{p_s(k)} \mathbf{1}\{i = k\} = x_s(k),$$

$$\text{and } \mathbb{E}_{k_s \sim p_s}[\tilde{x}_s(k_s)] = \sum_{i=1}^K p_s(i) \frac{x_s(i)}{p_s(i)} \leq K.$$

We thus have

$$\begin{aligned}
\frac{W_{t+1}}{W_t} &= \sum_{k=1}^K \frac{w_k(t) e^{\eta \tilde{x}_t(k)}}{W_t} = \sum_{k=1}^K \frac{p_k(t) - \beta/K}{1-\beta} e^{\eta \tilde{x}_t(k)} \\
&\leq \sum_{k=1}^K \frac{p_k(t) - \beta/K}{1-\beta} (1 + \eta \tilde{x}_t(k) + (e-2)\eta^2 \tilde{x}_t(k)^2),
\end{aligned}$$

since $\eta \tilde{x}_t(k) \leq \eta K/\beta = 1$, and $e^x \leq 1 + x + (e-2)x^2$ for $x \leq 1$.

Proof of Proposition 2 [part 2]

Thus

$$\frac{W_{t+1}}{W_t} \leq 1 + \frac{1}{1-\beta} \sum_{k=1}^{K} p_k(t)(\eta \tilde{x}_t(k) + (e-2)\eta^2 \tilde{x}_t(k)^2),$$

$$\log \frac{W_{t+1}}{W_t} \leq \frac{1}{1-\beta} \sum_{k=1}^{K} p_k(t)(\eta \tilde{x}_t(k) + (e-2)\eta^2 \tilde{x}_t(k)^2),$$

$$\log \frac{W_{n+1}}{W_1} \leq \frac{1}{1-\beta} \sum_{t=1}^{n} \sum_{k=1}^{K} p_k(t)(\eta \tilde{x}_t(k) + (e-2)\eta^2 \tilde{x}_t(k)^2).$$

But we also have

$$\log \frac{W_{n+1}}{W_1} = \log \sum_{k=1}^{K} e^{\eta \sum_{t=1}^{n} \tilde{x}_t(k)} - \log K \geq \eta \sum_{t=1}^{n} \tilde{x}_t(k) - \log K,$$

for any $k = 1, \ldots, n$.

Proof of Proposition 2 [part 3]

Take expectation w.r.t. internal randomization of the algo, thus for all k,

$$
\begin{aligned}
\mathbb{E}\Big[(1-\beta)\sum_{t=1}^{n}\tilde{x}_t(k) - \sum_{t=1}^{n}\sum_{i=1}^{K} p_i(t)\tilde{x}_t(i)\Big] &\leq (1-\beta)\frac{\log K}{\eta} \\
&+ (e-2)\eta\mathbb{E}\Big[\sum_{t=1}^{n}\sum_{k=1}^{K} p_k(t)\tilde{x}_t(k)^2\Big] \\
\mathbb{E}\Big[\sum_{t=1}^{n} x_t(k) - \sum_{t=1}^{n} x_t(k_t)\Big] &\leq \beta n + \frac{\log K}{\eta} + (e-2)\eta n K \\
\mathbb{E}[R_n(k)] &\leq \frac{\log K}{\eta} + (e-1)\eta n K
\end{aligned}
$$

In summary...

Distribution-dependent bounds:

$$\begin{aligned} \text{UCB:} \quad \mathbb{E}R_n &= O\left(\sum_k \frac{1}{\Delta_k} \log n\right) \\ \text{lower-bound:} \quad \mathbb{E}R_n &= \Omega\left(\log n\right) \end{aligned}$$

Distribution-independent bounds:

$$\begin{aligned} \text{UCB:} \quad \sup_{\textit{distributions}} \mathbb{E}R_n &= O\left(\sqrt{Kn \log n}\right) \\ \text{EXP3:} \quad \sup_{\textit{rewards}} \mathbb{E}R_n &= O\left(\sqrt{Kn \log K}\right) \\ \text{lower-bound:} \quad \sup_{\textit{rewards}} \mathbb{E}R_n &= \Omega\left(\sqrt{Kn}\right) \end{aligned}$$

Remark: The optimal rate $O\left(\sqrt{Kn}\right)$ is achieved by INF [Audibert and Bubeck, 2010]

Population of bandits

- Bandit (or regret minimization) algorithms = tool for rapidly selecting the best action.
- Basic building block for solving more complex problems
- We now consider a population of bandits:

Adversarial bandits

Collaborative bandits

Game between bandits

Consider a 2-players zero-sum repeated game:
A and B play actions: 1 or 2 simultaneously, and receive the reward (for A):

A \ B	1	2
1	2	0
2	-1	1

(A likes consensus, B likes conflicts)

Now, let A and B be bandit algorithms, aiming at minimizing their regret, i.e. for player A:

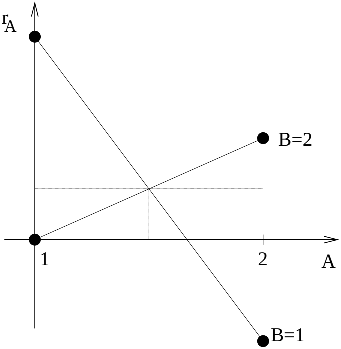
$$R_n(A) \stackrel{\text{def}}{=} \max_{a \in \{1,2\}} \sum_{t=1}^{n} r_A(a, B_t) - \sum_{t=1}^{n} r_A(A_t, B_t).$$

What happens?

Nash equilibrium

Nash equilibrium: (mixed) strategy for both players, such that no player has incentive for changing unilaterally his own strategy.

A \ B	1	2
1	2	0
2	-1	1



Here: A plays 1 with probability $p_A = 1/2$, B plays 1 with probability $p_B = 1/4$.

Regret minimization → Nash equilibrium

Define the regret of A:

$$R_n(A) \stackrel{\text{def}}{=} \max_{a \in \{1,2\}} \sum_{t=1}^{n} r_A(a, B_t) - \sum_{t=1}^{n} r_A(A_t, B_t).$$

and that of B accordingly.

Proposition 3.

If both players perform a (Hannan) consistent regret-minimization strategy (i.e. $R_n(A)/n \to 0$ and $R_n(B)/n \to 0$), then the empirical frequencies of chosen actions of both players converge to a Nash equilibrium.

(Remember that EXP3 is consistent!)

Note that in general, we have convergence towards correlated equilibrium [Foster and Vohra, 1997].

Sketch of proof:

Write $p_A^n \stackrel{\text{def}}{=} \frac{1}{n}\sum_{t=1}^n \mathbf{1}_{A_t=1}$ and $p_B^n \stackrel{\text{def}}{=} \frac{1}{n}\sum_{t=1}^n \mathbf{1}_{B_t=1}$ and $r_A(p,q) \stackrel{\text{def}}{=} \mathbb{E} r_A(A\sim p, B\sim q)$.
Regret-minimization algorithm: $R_n(A)/n \to 0$ means that: $\forall \varepsilon > 0$, for n large enough,

$$\begin{aligned}
\max_{a\in\{1,2\}} \frac{1}{n}\sum_{t=1}^n r_A(a,B_t) - \frac{1}{n}\sum_{t=1}^n r_A(A_t,B_t) &\leq \varepsilon \\
\max_{a\in\{1,2\}} r_A(a,p_B^n) - r_A(p_A^n,p_B^n) &\leq \varepsilon \\
r_A(p,p_B^n) - r_A(p_A^n,p_B^n) &\leq \varepsilon,
\end{aligned}$$

for all $p\in[0,1]$. Now, using $R_n(B)/n \to 0$ we deduce that:

$$r_A(p,p_B^n) - \varepsilon \leq r_A(p_A^n,p_B^n) \leq r_A(p_A^n,q) + \varepsilon, \quad \forall p,q\in[0,1]$$

Thus the empirical frequencies of actions played by both players is arbitrarily close to a Nash strategy.

Texas Hold'em Poker

- In the 2-players Poker game, the Nash equilibrium is interesting (zero-sum game)
- A policy:
  information set (my cards + board + pot) $\rightarrow$ probabilities over decisions (check, raise, fold)
- Space of policies is huge!

Idea: Approximate the Nash equilibrium by using bandit algorithms assigned to each information set.

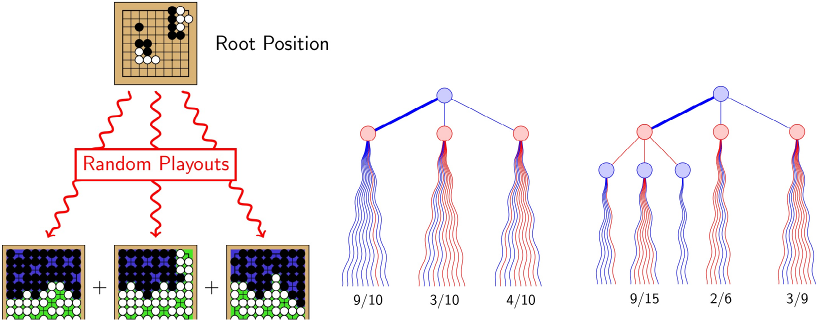
- This provides the world best Texas Hold'em Poker program for 2-player with pot-limit [Zinkevich et al., 2007]

Hierarchy of bandits

We now consider another way of combining bandits:
Hierarchy of bandits: the reward obtained when pulling an arm is itself the return of another bandit in a hierarchy.
Applications to

- tree search,
- optimization,
- planning

Historical motivation for this problem



MCTS in Crazy-Stone (Rémi Coulom, 2005)

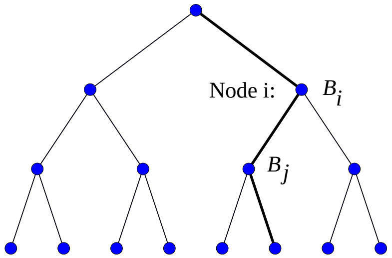
Idea: use bandits at each node.

Hierarchical bandit algorithm

Upper Confidence Bound (UCB) algo at each node

$$B_j \stackrel{\text{def}}{=} X_{j,n_j} + \sqrt{\frac{2\log(n_i)}{n_j}}.$$

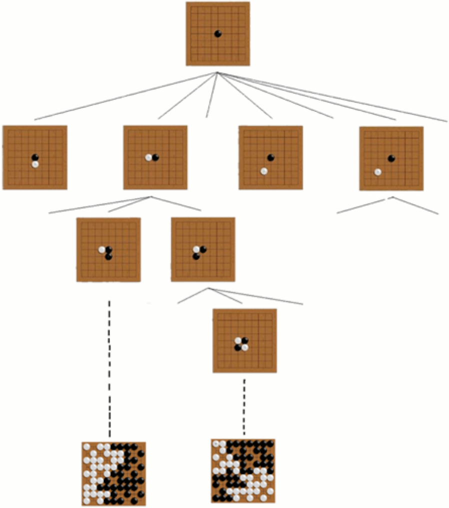
Intuition:
- Explore first the most promising branches
- Average converges to max



- Adaptive Multistage Sampling (AMS) algorithm [Chang, Fu, Hu, Marcus, 2005]
- UCB applied to Trees (UCT) [Kocsis and Szepesvári, 2006]

The MoGo program

[Gelly et al., 2006] + collaborative work with many others.



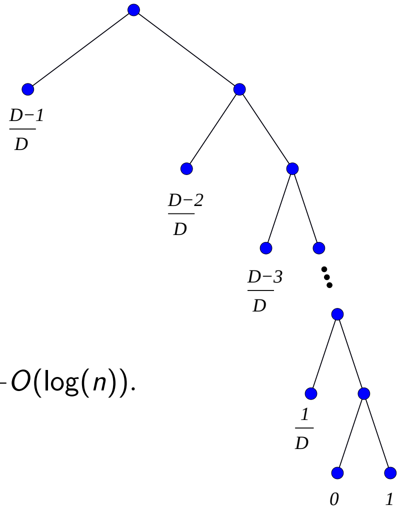
Features:

- Explore-Exploit with UCT
- Monte-Carlo evaluation
- Asymmetric tree expansion
- Anytime algo
- Use of features

Among world best programs!

No finite-time guarantee for UCT

Problem: at each node, the rewards are not i.i.d.
Consider the tree:



The left branches seem better than right branches, thus are explored for a **very** long time before the optimal leaf is eventually reached.
The expected regret is disastrous:

$$\mathbb{E}R_n = \Omega(\underbrace{\exp(\exp(\ldots\exp(}_{D \text{ times}}1)\ldots))) + O(\log(n)).$$

See [Coquelin and Munos, 2007]

Optimism for decision making

Outline:

- Optimization of deterministic Lipschitz functions
- Lipschitz bandits in general spaces: HOO
- Application to planning
  - Deterministic environments
  - Open-loop planning in stochastic environments
  - Closed-loop planning in sparse stochastic environements

Online optimization of a deterministic Lipschitz function

Problem: Find online the maximum of $f : X \to \boldsymbol{R}$, assumed to be Lipschitz: $|f(x) - f(y)| \leq \ell(x, y)$.

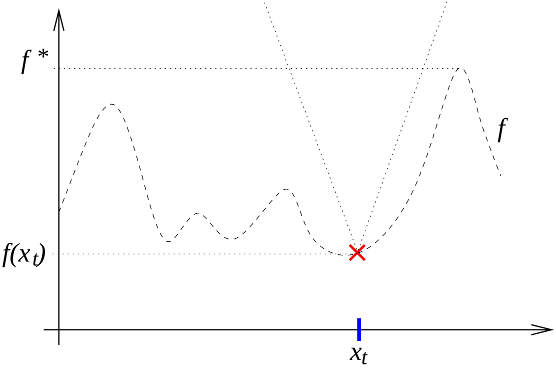
- At each time step t, select $x_t \in X$
- Observe $f(x_t)$
- Goal: find an exploration policy such as to maximize the sum of rewards.

Define the cumulative regret

$$R_n = \sum_{t=1}^{n} f^* - f(x_t),$$

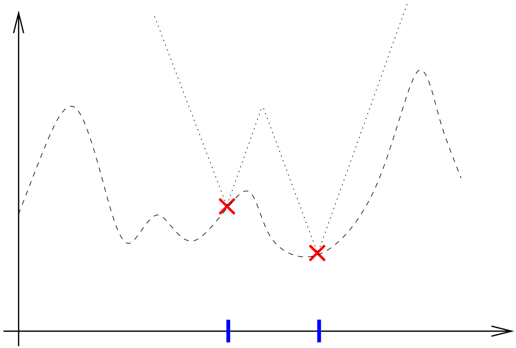
where $f^* = \sup_{x \in X} f(x)$

Example in 1d



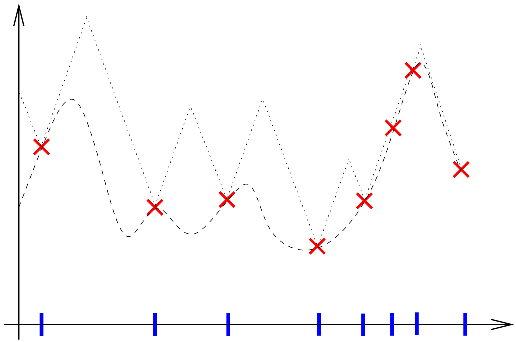
Lipschitz property → the evaluation of f at x_t provides a first upper-bound on f.

Example in 1d (continued)



New point $\rightarrow$ refined upper-bound on f.

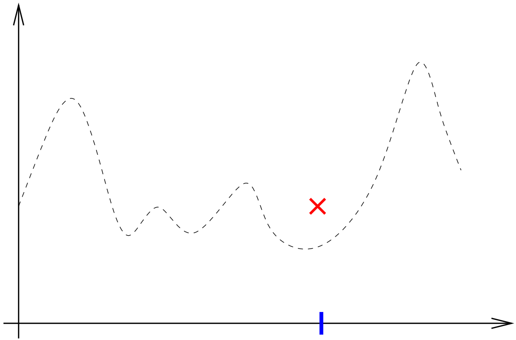
Example in 1d (continued)



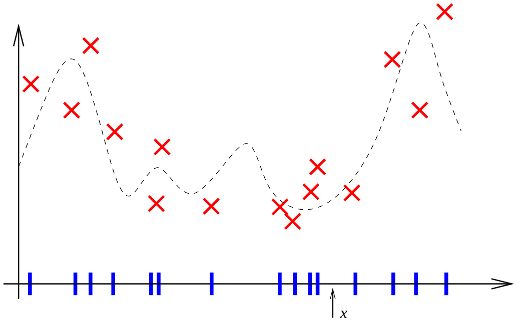
Question: where should one sample the next point?
Answer: select the point with highest upper bound!
"Optimism in the face of (partial observation) uncertainty"

Lipschitz optimization with noisy evaluations

f is still Lipschitz, but now, the evaluation of f at x_t returns a noisy evaluation r_t of $f(x_t)$, i.e. such that $\mathbb{E}[r_t|x_t] = f(x_t)$.

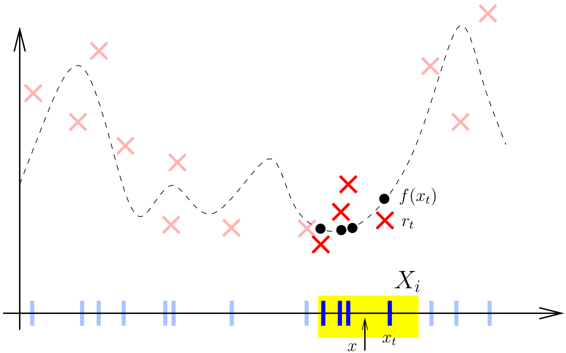


Where should one sample next?



How to define a high probability upper bound at any state x?

UCB in a given domain

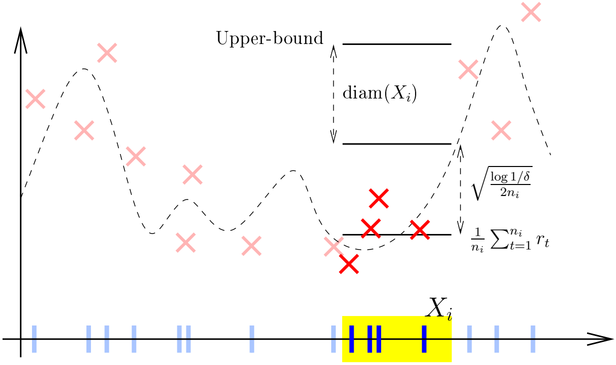


For a fixed domain $X_i \ni x$ containing n_i points $\{x_t\} \in X_i$, we have that $\sum_{t=1}^{n_i} r_t - f(x_t)$ is a Martingale. Thus by Azuma's inequality,

$$\frac{1}{n_i}\sum_{t=1}^{n_i} r_t + \sqrt{\frac{\log 1/\delta}{2n_i}} \geq \frac{1}{n_i}\sum_{t=1}^{n_i} f(x_t) \geq f(x) - diam(X_i),$$

since f is Lipschitz (where $diam(X_i) = \sup_{x,y \in X_i} \ell(x,y)$).

High probability upper bound



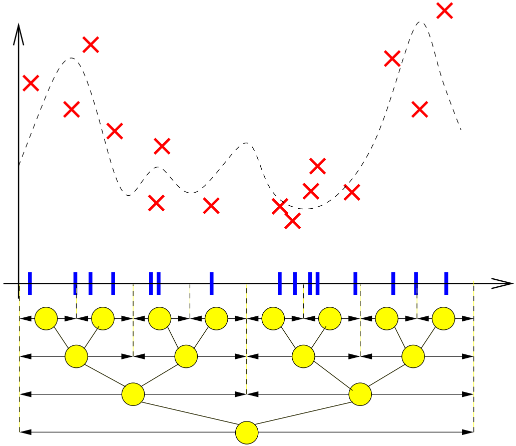
$$\text{w.p. } 1-\delta, \qquad \frac{1}{n_i}\sum_{t=1}^{n_i} r_t + \sqrt{\frac{\log 1/\delta}{2n_i}} + diam(X_i) \geq \sup_{x\in X_i} f(x).$$

Tradeoff between size of the confidence interval and diameter.

By considering several domains we can derive a tighter upper bound.

A hierarchical decomposition

Use a tree of partitions at all scales:



$$B_i(t) \stackrel{\text{def}}{=} \min\left\{\hat{\mu}_i(t) + \sqrt{\frac{2\log(t)}{t_i}} + \text{diam}(i), \max_{j\in\mathcal{C}(i)} B_j(t)\right\}$$

Multi-armed bandits in a semi-metric space

More generally:
Let X be space equipped with a semi-metric $\ell(x, y)$. Let $f(x)$ be a function such that:

$$f(x^*) - f(x) \leq \ell(x, x^*),$$

where $f(x^*) = \sup_{x \in X} f(x)$.

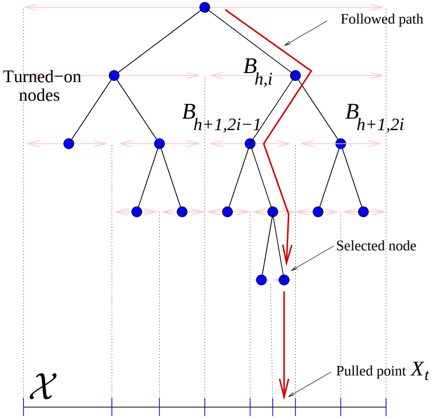
X**-armed bandit problem**: At each round t, choose a point (arm) $x_t \in X$, receive reward r_t independent sample drawn from a distribution $\nu(x_t)$ with mean $f(x_t)$.
Goal: minimize regret:

$$R_n \stackrel{\text{def}}{=} \sum_{t=1}^{n} f(x^*) - r_t$$

Hierarchical Optimistic Optimization (HOO)

[Bubeck et al., 2011]: Consider a tree of partitions of X, where each node i corresponds to a subdomain X_i.

HOO Algorithm:
Let $\mathcal{T}_t$ denote the set of expanded nodes at round t.
- $\mathcal{T}_1 = \{\text{root}\}$ (space X)
- At t, select a leaf i_t of $\mathcal{T}_t$ by maximizing the B-values,
- $\mathcal{T}_{t+1} = \mathcal{T}_t \cup \{i_t\}$
- Select $x_t \in X_{i_t}$ (arbitrarily)
- Observe reward $r_t \sim \nu(x_t)$ and update the B-values:



$$B_i \stackrel{\text{def}}{=} \min\left[\widehat{X}_{i,n_i} + \sqrt{\frac{2\log(n)}{n_i}} + diam(i), \max_{j\in\mathcal{C}(i)} B_j\right]$$

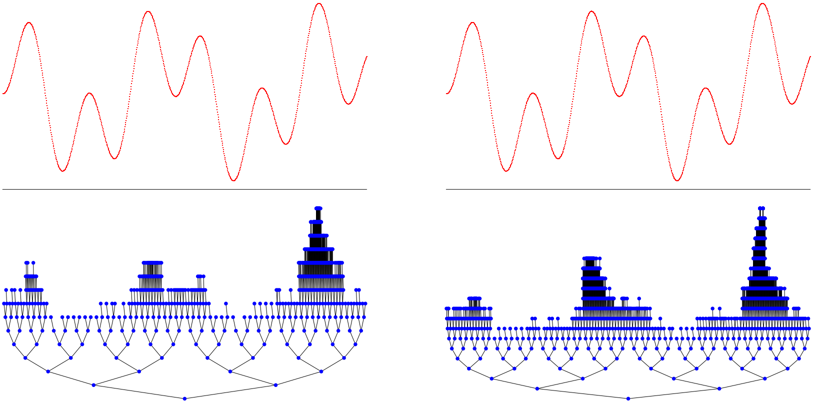
Properties of HOO

Properties:

- For any domain $X_i \ni x^*$, the corresponding B_i values is a (high probability) upper bound on $f(x^*)$.
- We don't really care if for sub-optimal domains X_i, the B_i values is an upper bound on $\sup_{x \in X_i} f(x)$ or not.
- The tree grows in an asymmetric way, leaving mainly unexplored the sub-optimal branches,
- Only the optimal branch is essentially explored.

Example in 1d

$r_t \sim \mathcal{B}(f(x_t))$ a Bernoulli distribution with parameter $f(x_t)$



Resulting tree at time $n = 1000$ and at $n = 10000$.

Analysis of HOO

Let d be the **near-optimality dimension** of f in X: i.e. such that the set of ε-optimal states

$$X_\varepsilon \stackrel{\text{def}}{=} \{x \in X, f(x) \geq f^* - \varepsilon\}$$

can be covered by $O(\varepsilon^{-d})$ balls of radius ε.
Then

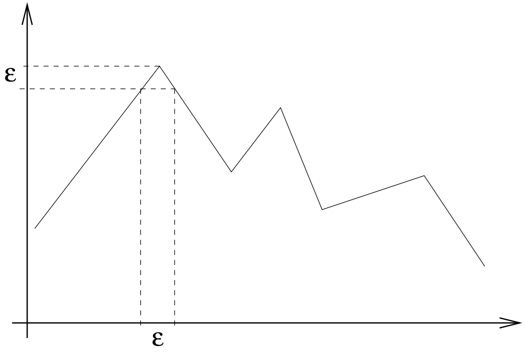
$$\mathbb{E}R_n = \widetilde{O}(n^{\frac{d+1}{d+2}}).$$

(Similar to Zooming algorithm of [Kleinberg, Slivkins, Upfall, 2008], but weaker assumption about f and ℓ, and does not require a sampling oracle)

Example 1:

Assume the function is locally peaky around its maximum:

$$f(x^*) - f(x) = \Theta(||x^* - x||).$$

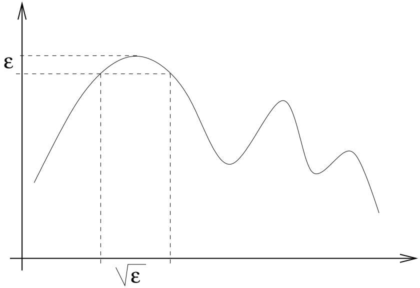


It takes $O(\epsilon^0)$ balls of radius ϵ to cover X_ε. Thus $d = 0$ and the regret is $\sqrt{n}$.

Example 2:

Assume the function is locally quadratic around its maximum:

$$f(x^*) - f(x) = \Theta(||x^* - x||^\alpha), \text{ with } \alpha = 2.$$



- For $\ell(x,y) = ||x-y||$, it takes $O(\epsilon^{-D/2})$ balls of radius ϵ to cover X_ε (of size $O(\epsilon^{D/2})$). Thus $d = D/2$ and the regret is $n^{\frac{D+2}{D+4}}$.
- For $\ell(x,y) = ||x-y||^2$, it takes $O(\epsilon^0)$ ℓ-balls of radius ϵ to cover X_ε. Thus $d = 0$ and the regret is $\sqrt{n}$.

Known smoothness around the maximum

Consider $X = [0,1]^d$. Assume that f has a finite number of global maxima and is locally α-smooth around each maximum x^*, i.e.

$$f(x^*) - f(x) = \Theta(||x^* - x||^\alpha).$$

Then, by choosing $\ell(x,y) = ||x-y||^\alpha$, X_ε is covered by $O(1)$ balls of "radius" ε. Thus the near-optimality dimension $d = 0$, and the regret of HOO is:

$$\mathbb{E}R_n = \widetilde{O}(\sqrt{n}),$$

i.e. the rate of growth is **independent of the ambient dimension**.

Conclusions on bandits in general spaces

The near-optimality dimension may be seen as an excess order of smoothness of f (around its maxima) compared to what is known:

- **If the smoothness order of the function is known** then the regret of HOO algorithm is $\widetilde{O}(\sqrt{n})$
- **If the smoothness is underestimated**, for example f is α-smooth but we only use $\ell(x, y) = ||x - y||^{\beta}$, with $\beta < \alpha$, then the near-optimality dimension is $d = D(1/\beta - 1/\alpha)$ and the regret is $\widetilde{O}(n^{(d+1)/(d+2)})$
- **If the smoothness is overestimated**, the weak-Lipschitz assumption is violated, thus there is no guarantee (e.g., UCT)

Applications

- **Online supervized learning:** At time t, HOO selects $h_t \in H$. The environment chooses $(x_t, y_t) \sim P$. The resulting loss $\ell(h_t(x_t), y_t)$ is a noisy evaluation of $\mathbb{E}_{(x,y)\sim P}[\ell(h(x), y)]$. HOO generates sequences of hypotheses (h_t) whose cumulated performances are close to that of the best hypothesis $h^* \in H$.
- **Policy optimization for MDPs or POMDPs:** Consider a class of parameterized policies π_α. At time t, HOO algo selects α_t and a trajectory is generated using π_{α_t}.
  The sum of rewards obtained is a noisy evaluation of the value function $V^{\pi_{\alpha_t}}$.
  Thus HOO performs almost as well as if using the best parameter α^*.

Application to planning in MDPs

Setting:

- Assume we have a generative model of an MDP.
- The state space is large: no way to represent the value function
- Search for the best policy, given a computational budget (e.g., number of calls to the model).
- Ex: from current state s_t, search for the best possible immediate action a_t, play this action, observe next state s_{t+1}, and repeat

Works:

- Optimistic planning in deterministic systems
- Open-Loop optimistic planning

Planning in deterministic systems

Controlled *deterministic* system with discounted rewards:

$$s_{t+1} = f(s_t, a_t), \text{ where } a_t \in A.$$

Goal is to maximize $\sum_{t \geq 0} \gamma^t r(s_t, a_t)$.
Online planning:

- From the current state s_t, return the best possible immediate action a_t, computed by using a given computational budget (eg, CPU time, number of calls to the model).
- Play a_t in the real world, and repeat from next state s_{t+1}.

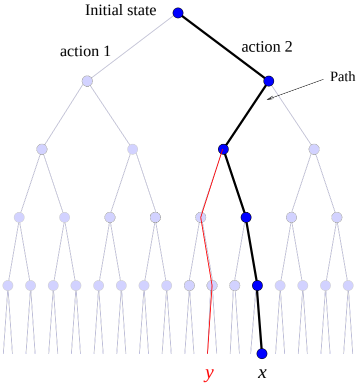
Given n calls to a generative model, return actions $a_t(n)$.

$$\text{Simple regret:} \quad r_n \stackrel{\text{def}}{=} \max_{a \in A} Q^*(s_t, a) - Q^*(s_t, a_t(n)).$$

Look-ahead tree for planning in deterministic systems

From the current state, build the look-ahead tree:

- Root of the tree = current state s_t
- Search space X = set of paths (infinite sequence of actions)
- Value of any path x: $f(x) = \sum_{t \geq 0} \gamma^t r_t$
- Metric: $\ell(x, y) = \frac{\gamma^{h(x,y)}}{1-\gamma}$
- Prop: f is Lipschitz w.r.t. ℓ
- Use optimistic search to explore the tree with budget n resources



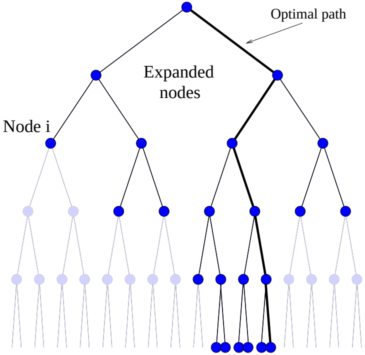
Optimistic exploration

(HOO algo in deterministic setting)

- For any node i of depth d, define the B-values:

$$B_i \stackrel{\text{def}}{=} \sum_{t=0}^{d-1} \gamma^t r_t + \frac{\gamma^d}{1-\gamma} \geq v_i$$

- At each round n, expand the node with highest B-value
- Observe reward, update B-values,
- Repeat until no more available resources
- Return maximizing action



Analysis of the regret

[Hren and Munos, 2008] Define β such that the proportion of ϵ-optimal paths is $O(\epsilon^\beta)$ (this is related to the near-optimal dimension). Let

$$\kappa \stackrel{\text{def}}{=} K\gamma^\beta \in [1, K].$$

- If $\kappa > 1$, then
$$r_n = O\left(n^{-\frac{\log 1/\gamma}{\log \kappa}}\right).$$
(whereas for uniform planning $R_n = O\big(n^{-\frac{\log 1/\gamma}{\log K}}\big)$.)
- If $\kappa = 1$, then we obtain the exponential rate $r_n = O\big(\gamma^{\frac{(1-\gamma)^\beta}{c} n}\big)$, where c is such that the proportion of ϵ-path is bounded by $c\epsilon^\beta$.

Open Loop Optimistic Planning

Setting:

- **Rewards are stochastic** but depend on sequence of actions (and not resulting states)
- Goal : find the sequence of actions that maximizes the expected discounted sum of rewards
- Search space: open-loop policies (sequences of actions)

[Bubeck et Munos, 2010] OLOP algorithm has expected regret

$$\mathbb{E}r_n = \begin{cases} \tilde{O}\left(n^{-\frac{\log 1/\gamma}{\log \kappa}}\right) & \text{if } \gamma\sqrt{\kappa} > 1, \\ \tilde{O}\left(n^{-\frac{1}{2}}\right) & \text{if } \gamma\sqrt{\kappa} \leq 1. \end{cases}$$

Remarks:

- For $\gamma\sqrt{\kappa} > 1$, this is the same rate as for deterministic systems!
- This is not a consequence of HOO

Possible extensions

Applications of hierarchical bandits:

- Planning in MDPs when the number of next states is finite [Bușoniu et al., 2011]
- Planning in POMDPs when the number of observations is finite
- Combine planning with function approximation: local ADP methods.
- Many applications in MCTS (Monte-Carlo Tree Search): See Teytaud, Chaslot, Bouzy, Cazenave, and many others.

Related references

- J.Y. Audibert, R. Munos, and C. Szepesvari, *Tuning bandit algorithms in stochastic environments*, ALT, 2007.
- P. Auer, N. Cesa-Bianchi, and P. Fischer, *Finite time analysis of the multiarmed bandit problem*, Machine Learning, 2002.
- S. Bubeck and R. Munos, *Open Loop Optimistic Planning*, COLT 2010.
- S. Bubeck, R. Munos, G. Stoltz, Cs. Szepesvari, *Online Optimization in X-armed Bandits*, NIPS 2008. Long version *X-armed Bandits* JMLR 2011.
- L. Bușoniu, R. Munos, B. De Schutter, R. Babuška *Optimistic Planning for Sparsely Stochastic Systems*, ADPRL 2011.
- P.-A. Coquelin and R. Munos, *Bandit Algorithm for Tree Search*, UAI 2007.
- S. Gelly, Y. Wang, R. Munos, and O. Teytaud, *Modification of UCT with Patterns in Monte-Carlo Go*, RR INRIA, 2006.

Related references (cont'ed)

- J.-F. Hren and R. Munos, *Optimistic planning in deterministic systems.* EWRL 2008.
- M. Kearns, Y. Mansour, A. Ng, *A Sparse Sampling Algorithm for Near-Optimal Planning in Large Markov Decision Processes*, Machine Learning, 2002.
- R. Kleinberg, A. Slivkins, and E. Upfal, *Multi-Armed Bandits in Metric Spaces*, ACM Symposium on Theory of Computing, 2008.
- L. Kocsis and Cs. Szepesvri, *Bandit based Monte-Carlo Planning*, ECML 2006.
- T. L. Lai and H. Robbins, *Asymptotically Efficient Adaptive Allocation Rules*, Advances in Applied Mathematics, 1985.