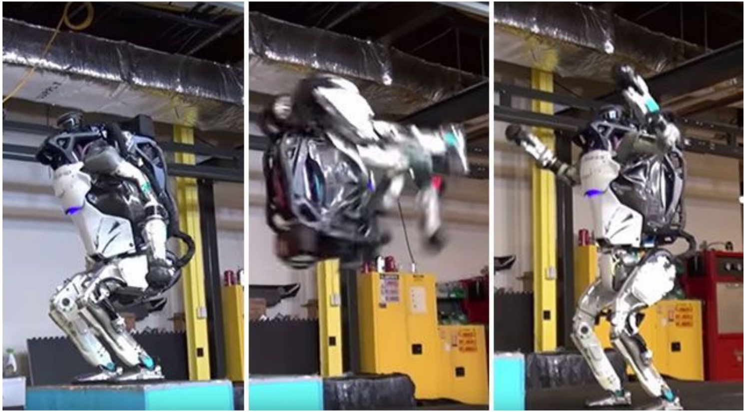
POLITECNICO
MILANO 1863

Exploration in Policy Search by Multiple Importance Sampling

Lorenzo Lupo
lorenzo.lupo@mail.polimi.it

April 16th, 2019

How can robots learn to backflip?



Algorithmic Trading

Self-driving Cars

Plan

1. Basics of Reinforcement Learning
2. Exploration in Policy Search
3. Problem Formalization
4. OPTIMIST

The Reinforcement Learning Framework

Environment

$\mathcal{P}, \mathcal{R}$

The Reinforcement Learning Framework

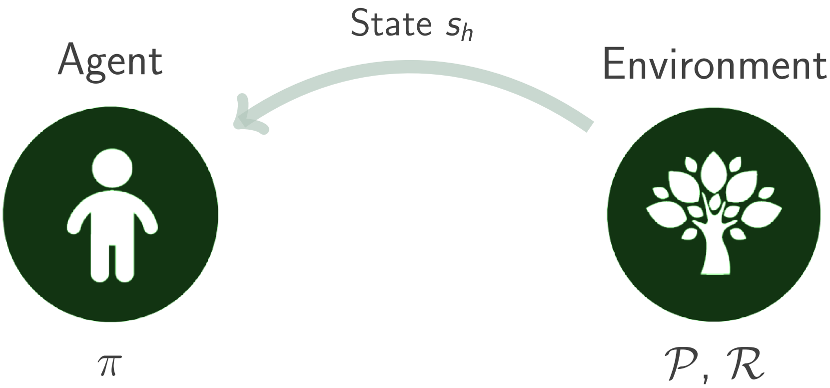
Agent

π

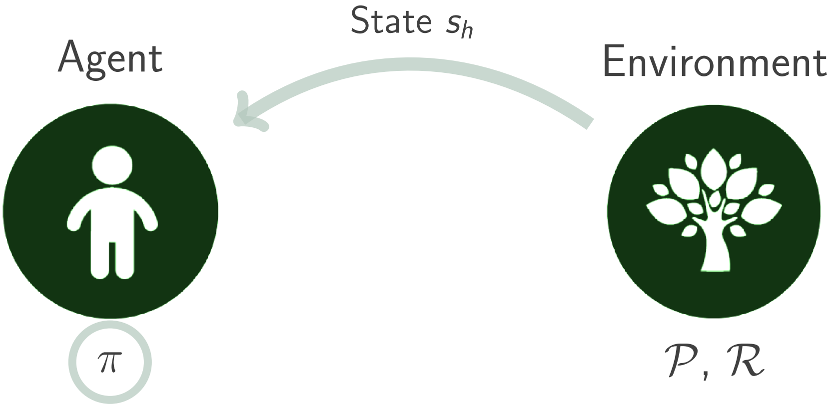
Environment

$\mathcal{P}, \mathcal{R}$

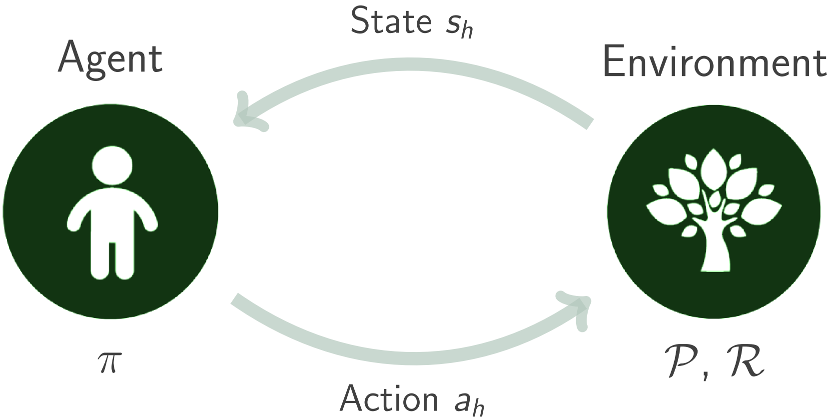
The Reinforcement Learning Framework



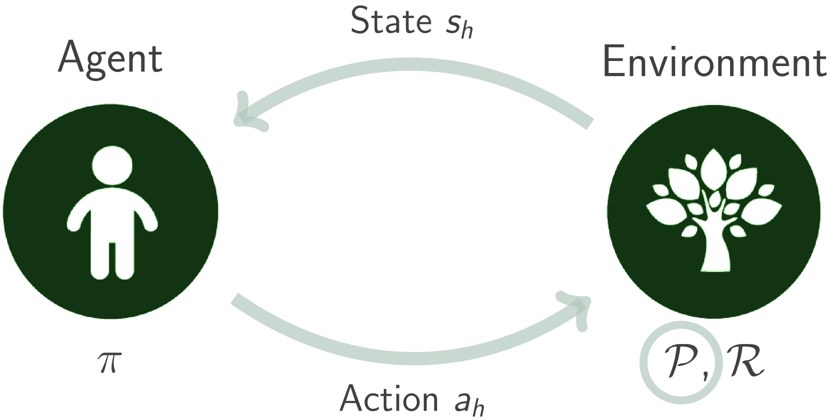
The Reinforcement Learning Framework



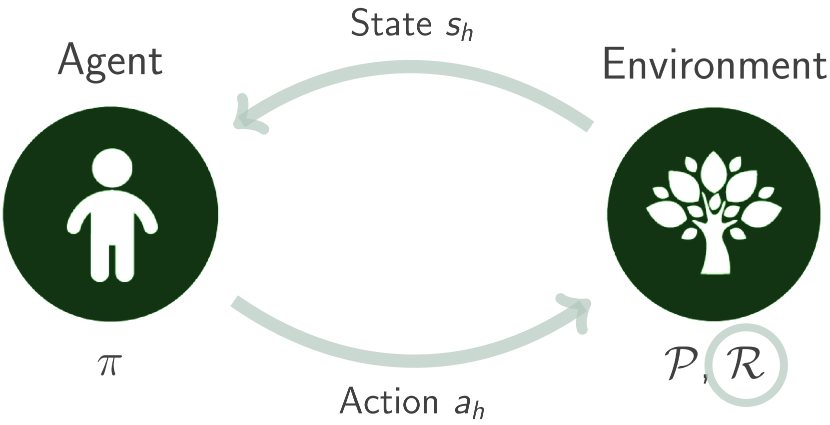
The Reinforcement Learning Framework



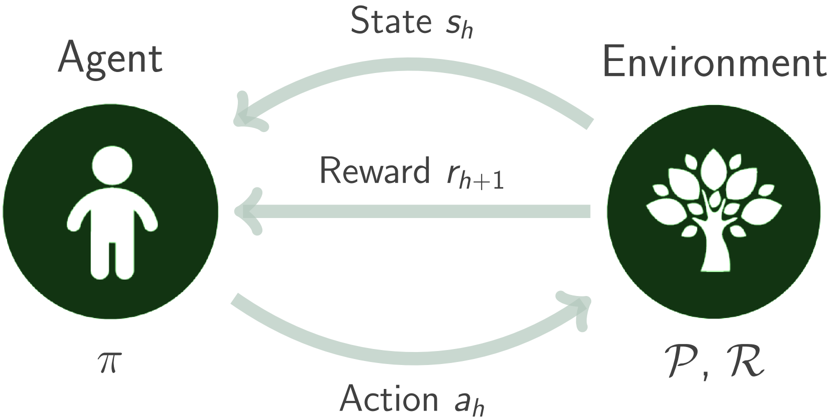
The Reinforcement Learning Framework



The Reinforcement Learning Framework



The Reinforcement Learning Framework



Policy Search Formulation

Cumulative return of a trajectory τ:

$$\mathcal{R}(\tau) = \sum_{h=0}^{H-1} \gamma^h r_{h+1}, \text{ with } \tau = [s_0, a_0, s_1, a_1, \ldots, s_{H-1}, a_{H-1}]$$

Policy Search Formulation

Cumulative return of a trajectory τ:

$$\mathcal{R}(\tau) = \sum_{h=0}^{H-1} \gamma^h r_{h+1}, \text{ with } \tau = \left[s_0, a_0, s_1, a_1, \ldots, s_{H-1}, a_{H-1}\right]$$

Parametric policy:

$$\pi_{\boldsymbol{\theta}} : \mathcal{S} \to \Delta(\mathcal{A}), \text{ i.e., } \pi_{\boldsymbol{\theta}}(a|s) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\left(\frac{a - \boldsymbol{\theta}^T \phi(s)}{\sigma}\right)^2\right)$$

Policy Search Formulation

Cumulative return of a trajectory τ:

$$\mathcal{R}(\tau) = \sum_{h=0}^{H-1} \gamma^h r_{h+1}, \text{ with } \tau = [s_0, a_0, s_1, a_1, \ldots, s_{H-1}, a_{H-1}]$$

Parametric policy:

$$\pi_{\boldsymbol{\theta}} : \mathcal{S} \to \Delta(\mathcal{A}), \text{ i.e., } \pi_{\boldsymbol{\theta}}(a|s) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{1}{2}\left(\frac{a - \boldsymbol{\theta}^T \phi(s)}{\sigma}\right)^2\right)$$

Performance:

$\mu(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\tau \sim p_{\boldsymbol{\theta}}} [\mathcal{R}(\tau)]$, where $p_{\boldsymbol{\theta}}$ is the **distribution over trajectories** $\tau \in \mathcal{T}$ induced by $\pi_{\boldsymbol{\theta}}$

Policy Search Formulation

Cumulative return of a trajectory τ:

$$\mathcal{R}(\tau) = \sum_{h=0}^{H-1} \gamma^h r_{h+1}, \text{ with } \tau = [s_0, a_0, s_1, a_1, \ldots, s_{H-1}, a_{H-1}]$$

Parametric policy:

$$\pi_{\boldsymbol{\theta}} : \mathcal{S} \to \Delta(\mathcal{A}), \text{ i.e., } \pi_{\boldsymbol{\theta}}(a|s) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\left(\frac{a - \boldsymbol{\theta}^T \phi(s)}{\sigma}\right)^2\right)$$

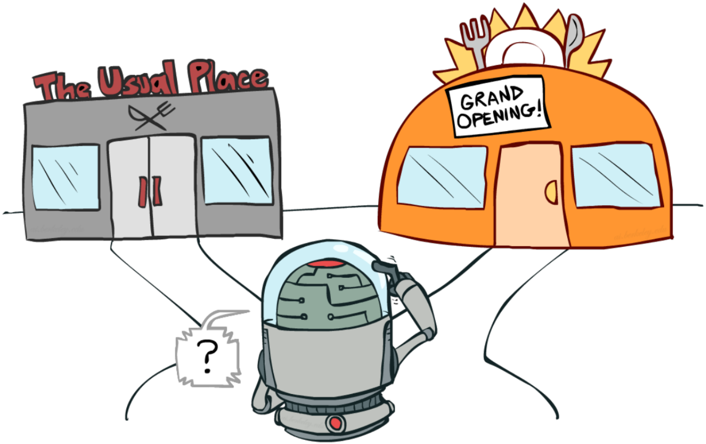
Performance:

$\mu(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\tau \sim p_{\boldsymbol{\theta}}} [\mathcal{R}(\tau)]$, where $p_{\boldsymbol{\theta}}$ is the **distribution over trajectories** $\tau \in \mathcal{T}$ induced by $\pi_{\boldsymbol{\theta}}$

Objective:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta} \in \Theta} \mu(\boldsymbol{\theta}).$$

Exploration VS Exploitation



Plan

1. Basics of Reinforcement Learning
2. Exploration in Policy Search
3. Problem Formalization
4. OPTIMIST

Exploration in Policy Search

Undirected exploration

Generate actions based on randomness, without any knowledge of the learning process.

Exploration in Policy Search

Undirected exploration

Generate actions based on randomness, without any knowledge of the learning process.

- Ex1: by adopting stochastic policies.

Exploration in Policy Search

Undirected exploration

Generate actions based on randomness, without any knowledge of the learning process.

- Ex1: by adopting stochastic policies.
- Ex2: by augmenting rewards with the entropy of the policy:

$$\mathcal{R}(\tau) = \sum_{h=0}^{H-1} \gamma^h r_{h+1} + \mathcal{H}(\pi_{\boldsymbol{\theta}}(\cdot|s_h)).$$

Exploration in Policy Search

Undirected exploration

Generate actions based on randomness, without any knowledge of the learning process.

- Ex1: by adopting stochastic policies.
- Ex2: by augmenting rewards with the entropy of the policy:

$$\mathcal{R}(\tau) = \sum_{h=0}^{H-1} \gamma^h r_{h+1} + \mathcal{H}(\pi_{\boldsymbol{\theta}}(\cdot|s_h)).$$

Directed exploration

Leverage on the knowledge acquired during learning.

Exploration in Policy Search

Undirected exploration

Generate actions based on randomness, without any knowledge of the learning process.

- Ex1: by adopting stochastic policies.
- Ex2: by augmenting rewards with the entropy of the policy:

$$\mathcal{R}(\tau) = \sum_{h=0}^{H-1} \gamma^h r_{h+1} + \mathcal{H}(\pi_{\boldsymbol{\theta}}(\cdot|s_h)).$$

Directed exploration

Leverage on the knowledge acquired during learning.

- Count-based techniques.

Plan

Problem Formalization

Decision Set or Arms Set

The parameter space $\Theta \subseteq \mathbb{R}^d$.

Problem Formalization

Decision Set or Arms Set

The parameter space $\Theta \subseteq \mathbb{R}^d$.

Procedure

At every decision step $t \in [0, 1, 2, \ldots, T]$:

Problem Formalization

Decision Set or Arms Set

The parameter space $\Theta \subseteq \mathbb{R}^d$.

Procedure

At every decision step $t \in [0, 1, 2, \ldots, T]$:

1. **Select** an arm $\boldsymbol{\theta}_t \in \Theta$;

Problem Formalization

Decision Set or Arms Set

The parameter space $\Theta \subseteq \mathbb{R}^d$.

Procedure

At every decision step $t \in [0, 1, 2, \ldots, T]$:

1. **Select** an arm $\boldsymbol{\theta}_t \in \Theta$;
2. **Sample** a trajectory $\tau_t \in \mathcal{T}$ by following $\pi_{\boldsymbol{\theta}_t}$;

Problem Formalization

Decision Set or Arms Set

The parameter space $\Theta \subseteq \mathbb{R}^d$.

Procedure

At every decision step $t \in [0, 1, 2, \ldots, T]$:

1. **Select** an arm $\boldsymbol{\theta}_t \in \Theta$;
2. **Sample** a trajectory $\tau_t \in \mathcal{T}$ by following $\pi_{\boldsymbol{\theta}_t}$;
3. **Observe** the cumulative return $\mathcal{R}(\tau_t)$.

Problem Formalization

Decision Set or Arms Set

The parameter space $\Theta \subseteq \mathbb{R}^d$.

Procedure

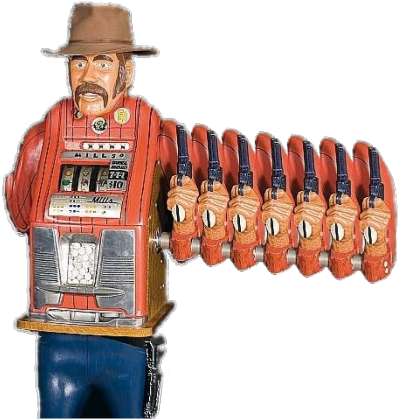
At every decision step $t \in [0, 1, 2, \ldots, T]$:

1. **Select** an arm $\boldsymbol{\theta}_t \in \Theta$;
2. **Sample** a trajectory $\tau_t \in \mathcal{T}$ by following $\pi_{\boldsymbol{\theta}_t}$;
3. **Observe** the cumulative return $\mathcal{R}(\tau_t)$.

Goal

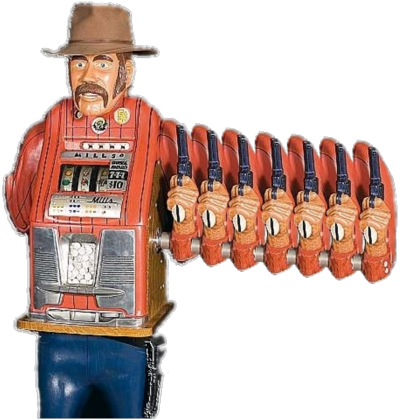
Minimize $\mathit{Regret}(T) = \sum_{t=0}^{T} \mu(\boldsymbol{\theta}^*) - \mu(\boldsymbol{\theta}_t)$, where $\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta} \in \Theta} \mu(\boldsymbol{\theta})$

Problem Formulation



Multi Armed Bandits

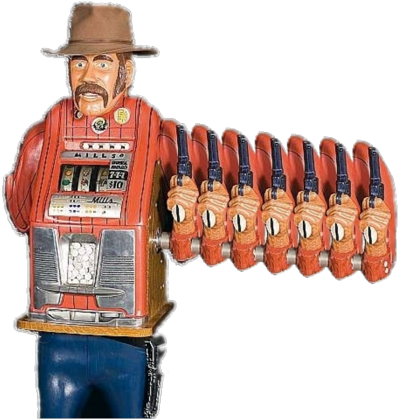
Problem Formulation



Multi Armed Bandits

- Simpler framework;

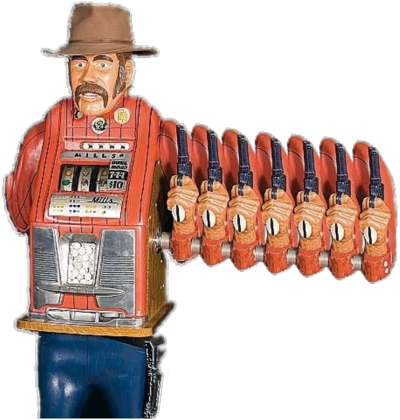
Problem Formulation



Multi Armed Bandits

- Simpler framework;
- Share the exploration-exploitation tradeoff;

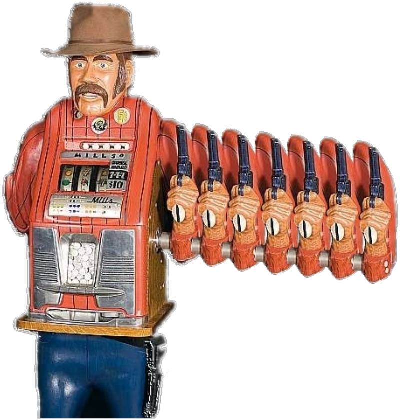
Problem Formulation



Multi Armed Bandits

- Simpler framework;
- Share the exploration-exploitation tradeoff;
- Ample literature available;

Problem Formulation



Multi Armed Bandits

- Simpler framework;
- Share the exploration-exploitation tradeoff;
- Ample literature available;

Desideratum

sub-linear $Regret(T) \Leftrightarrow \lim_{T\to\infty} Regret(T)/T = 0$

E.g. $Regret(T) = \mathcal{O}(\log T)$

Plan

1. Basics of Reinforcement Learning
2. Exploration in Policy Search
3. Problem Formalization
4. OPTIMIST

Optimistic Policy opTImization via MIS with Truncation

Algorithm 1 OPTIMIST

1: **Input:** initial parameters $\boldsymbol{\theta}_0$, confidence schedule $(\delta_t)_{t=1}^{T}$, order $\epsilon \in (0, 1]$

Optimistic Policy opTImization via MIS with Truncation

Algorithm 2 OPTIMIST

1: **Input:** initial parameters $\boldsymbol{\theta}_0$, confidence schedule $(\delta_t)_{t=1}^{T}$, order $\epsilon \in (0, 1]$

2: Draw trajectory $\tau_0 \sim p_{\boldsymbol{\theta}_0}$ and observe return $\mathcal{R}(\tau_0)$

Optimistic Policy opTImization via MIS with Truncation

Algorithm 3 OPTIMIST

Input: initial parameters $\boldsymbol{\theta}_0$, confidence schedule $(\delta_t)_{t=1}^{T}$, order $\epsilon \in (0, 1]$
Draw trajectory $\tau_0 \sim p_{\boldsymbol{\theta}_0}$ and observe return $\mathcal{R}(\tau_0)$
for $t = 1, \dots, T$ **do**

Optimistic Policy opTImization via MIS with Truncation

Algorithm 4 OPTIMIST

1: **Input:** initial parameters $\boldsymbol{\theta}_0$, confidence schedule $(\delta_t)_{t=1}^{T}$, order $\epsilon \in (0, 1]$
2: Draw trajectory $\tau_0 \sim p_{\boldsymbol{\theta}_0}$ and observe return $\mathcal{R}(\tau_0)$
3: **for** $t = 1, \dots, T$ **do**
4: Select arm $\boldsymbol{\theta}_t = \arg\max_{\boldsymbol{\theta} \in \Theta} B_t^{\epsilon}(\boldsymbol{\theta}, \delta_t)$

Optimistic Policy opTImization via MIS with Truncation

Algorithm 5 OPTIMIST

1: **Input:** initial parameters $\boldsymbol{\theta}_0$, confidence schedule $(\delta_t)_{t=1}^{T}$, order $\epsilon \in (0, 1]$
2: Draw trajectory $\tau_0 \sim p_{\boldsymbol{\theta}_0}$ and observe return $\mathcal{R}(\tau_0)$
3: **for** $t = 1, \dots, T$ **do**
4: Select arm $\boldsymbol{\theta}_t = \arg\max_{\boldsymbol{\theta} \in \Theta} B_t^{\epsilon}(\boldsymbol{\theta}, \delta_t)$
5: Draw trajectory $\tau_t \sim p_{\boldsymbol{\theta}_t}$ and observe return $\mathcal{R}(\tau_t)$
6: **end for**

Optimistic Policy opTImization via MIS with Truncation

Algorithm 6 OPTIMIST

1: **Input:** initial parameters $\boldsymbol{\theta}_0$, confidence schedule $(\delta_t)_{t=1}^{T}$, order $\epsilon \in (0, 1]$
2: Draw trajectory $\tau_0 \sim p_{\boldsymbol{\theta}_0}$ and observe return $\mathcal{R}(\tau_0)$
3: **for** $t = 1, \dots, T$ **do**
4: Select arm $\boldsymbol{\theta}_t = \arg\max_{\boldsymbol{\theta} \in \Theta} B_t^{\epsilon}(\boldsymbol{\theta}, \delta_t)$
5: Draw trajectory $\tau_t \sim p_{\boldsymbol{\theta}_t}$ and observe return $\mathcal{R}(\tau_t)$
6: **end for**

$$\mu(\boldsymbol{\theta}) \leqslant B_t^{\epsilon}(\boldsymbol{\theta}, \delta_t) \;=\; \check{\mu}_t^{MIS}(\boldsymbol{\theta}) \;+\; \|f\|_{\infty} \left(\sqrt{2} + \frac{4}{3}\right) \left(\frac{d_{1+\epsilon}(p_{\boldsymbol{\theta}_t} \| \Phi_t) \;\log \frac{1}{\delta_t}}{t} \right)^{\frac{\epsilon}{1+\epsilon}}$$

Optimistic Policy opTImization via MIS with Truncation

Algorithm 7 OPTIMIST

1: **Input:** initial parameters $\boldsymbol{\theta}_0$, confidence schedule $(\delta_t)_{t=1}^{T}$, order $\epsilon \in (0,1]$
2: Draw trajectory $\tau_0 \sim p_{\boldsymbol{\theta}_0}$ and observe return $\mathcal{R}(\tau_0)$
3: **for** $t = 1, \dots, T$ **do**
4: Select arm $\boldsymbol{\theta}_t = \arg\max_{\boldsymbol{\theta} \in \Theta} B_t^{\epsilon}(\boldsymbol{\theta}, \delta_t)$
5: Draw trajectory $\tau_t \sim p_{\boldsymbol{\theta}_t}$ and observe return $\mathcal{R}(\tau_t)$
6: **end for**

Upper Confidence Bound

$$\boxed{\mu(\boldsymbol{\theta}) \leqslant B_t^{\epsilon}(\boldsymbol{\theta}, \delta_t)} = \check{\mu}_t^{MIS}(\boldsymbol{\theta}) + \|f\|_{\infty} \left(\sqrt{2} + \frac{4}{3}\right) \left(\frac{d_{1+\epsilon}(p_{\boldsymbol{\theta}_t} \| \Phi_t) \ \log \frac{1}{\delta_t}}{t} \right)^{\frac{\epsilon}{1+\epsilon}}$$

Optimistic Policy opTImization via MIS with Truncation

Algorithm 8 OPTIMIST

Input: initial parameters $\boldsymbol{\theta}_0$, confidence schedule $(\delta_t)_{t=1}^{T}$, order $\epsilon \in (0,1]$
Draw trajectory $\tau_0 \sim p_{\boldsymbol{\theta}_0}$ and observe return $\mathcal{R}(\tau_0)$
for $t = 1, \dots, T$ **do**
    Select arm $\boldsymbol{\theta}_t = \arg\max_{\boldsymbol{\theta} \in \Theta} B_t^{\epsilon}(\boldsymbol{\theta}, \delta_t)$
    Draw trajectory $\tau_t \sim p_{\boldsymbol{\theta}_t}$ and observe return $\mathcal{R}(\tau_t)$
end for

Truncated Multiple Importance Sampling Estimator

$$\mu(\boldsymbol{\theta}) \leqslant B_t^{\epsilon}(\boldsymbol{\theta}, \delta_t) = \boxed{\check{\mu}_t^{MIS}(\boldsymbol{\theta})} + \|f\|_\infty \left(\sqrt{2} + \frac{4}{3}\right) \left(\frac{d_{1+\epsilon}(p_{\boldsymbol{\theta}_t} \| \Phi_t) \log \frac{1}{\delta_t}}{t}\right)^{\frac{\epsilon}{1+\epsilon}}$$

Optimistic Policy opTImization via MIS with Truncation

Algorithm 9 OPTIMIST

1: **Input:** initial parameters $\boldsymbol{\theta}_0$, confidence schedule $(\delta_t)_{t=1}^{T}$, order $\epsilon \in (0, 1]$
2: Draw trajectory $\tau_0 \sim p_{\boldsymbol{\theta}_0}$ and observe return $\mathcal{R}(\tau_0)$
3: **for** $t = 1, \dots, T$ **do**
4: Select arm $\boldsymbol{\theta}_t = \arg\max_{\boldsymbol{\theta} \in \Theta} B_t^{\epsilon}(\boldsymbol{\theta}, \delta_t)$
5: Draw trajectory $\tau_t \sim p_{\boldsymbol{\theta}_t}$ and observe return $\mathcal{R}(\tau_t)$
6: **end for**

Exploration Bonus

$$\mu(\boldsymbol{\theta}) \leqslant B_t^{\epsilon}(\boldsymbol{\theta}, \delta_t) \;=\; \check{\mu}_t^{MIS}(\boldsymbol{\theta}) \;+\; \boxed{\|f\|_\infty \left(\sqrt{2} + \frac{4}{3}\right) \left(\frac{d_{1+\epsilon}(p_{\boldsymbol{\theta}_t} \| \Phi_t) \;\log \frac{1}{\delta_t}}{t} \right)^{\frac{\epsilon}{1+\epsilon}}}$$

Optimistic Policy opTImization via MIS with Truncation

Algorithm 10 OPTIMIST

Input: initial parameters $\boldsymbol{\theta}_0$, confidence schedule $(\delta_t)_{t=1}^{T}$, order $\epsilon \in (0, 1]$
Draw trajectory $\tau_0 \sim p_{\boldsymbol{\theta}_0}$ and observe return $\mathcal{R}(\tau_0)$
for $t = 1, \dots, T$ **do**
  Select arm $\boldsymbol{\theta}_t = \arg\max_{\boldsymbol{\theta} \in \Theta} B_t^{\epsilon}(\boldsymbol{\theta}, \delta_t)$
  Draw trajectory $\tau_t \sim p_{\boldsymbol{\theta}_t}$ and observe return $\mathcal{R}(\tau_t)$
end for

Exploration Bonus

$$\mu(\boldsymbol{\theta}) \leqslant B_t^{\epsilon}(\boldsymbol{\theta}, \delta_t) \;=\; \check{\mu}_t^{MIS}(\boldsymbol{\theta}) \;+\; \boxed{\|f\|_\infty \left(\sqrt{2} + \frac{4}{3}\right) \left(\frac{\boxed{d_{1+\epsilon}(p_{\boldsymbol{\theta}_t} \| \Phi_t)} \log \frac{1}{\delta_t}}{t} \right)^{\frac{\epsilon}{1+\epsilon}}}$$

Thank you for your attention!

References

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011).
Improved algorithms for linear stochastic bandits.
In *Advances in Neural Information Processing Systems*, pages 2312–2320.

Agrawal, R. (1995a).
The continuum-armed bandit problem.
SIAM Journal on Control and Optimization, 33(6):1926–1951.

Agrawal, R. (1995b).
Sample mean based index policies by o (log n) regret for the multi-armed bandit problem.
Advances in Applied Probability, 27(4):1054–1078.

Agrawal, S. and Goyal, N. (2013).
Further optimal regret bounds for thompson sampling.
In *Artificial intelligence and statistics*, pages 99–107.

Amari, S.-I. (1998).