POLITECNICO
MILANO 1863

Exploration in Policy Search via Multiple Importance Sampling

Lorenzo Lupo
Supervisor: Marcello Restelli
Co-supervisors: Matteo Papini, Alberto Maria Metelli

April 16th, 2019

Reinforcement Learning

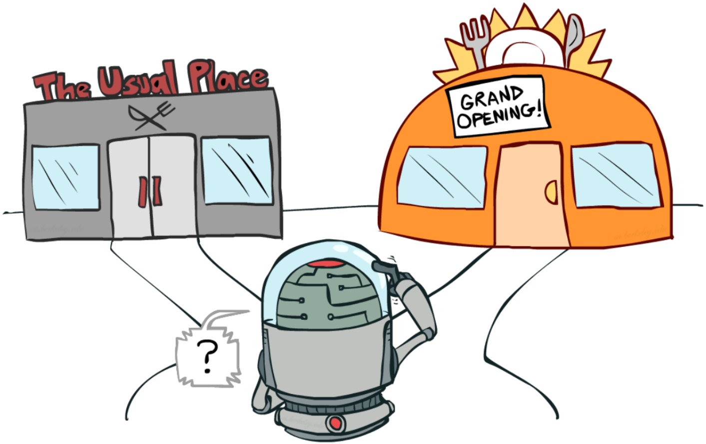
Reinforcement Learning

Reinforcement Learning

Reinforcement Learning

Exploitation VS Exploration

Exploitation VS Exploration



Plan

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

The Reinforcement Learning Framework

Environment

$\mathcal{P}$

The Reinforcement Learning Framework

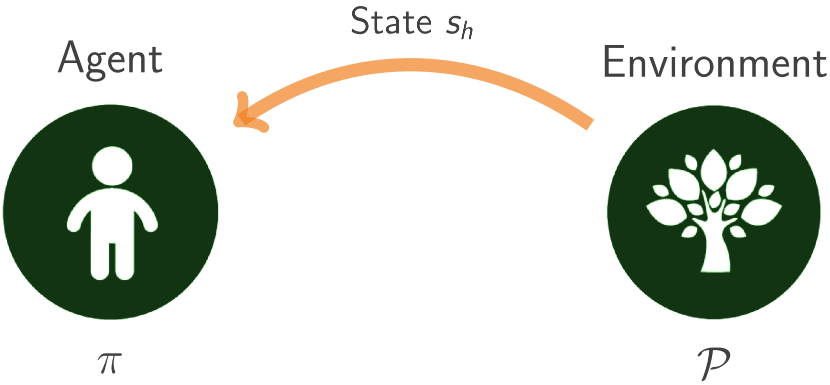
Agent

π

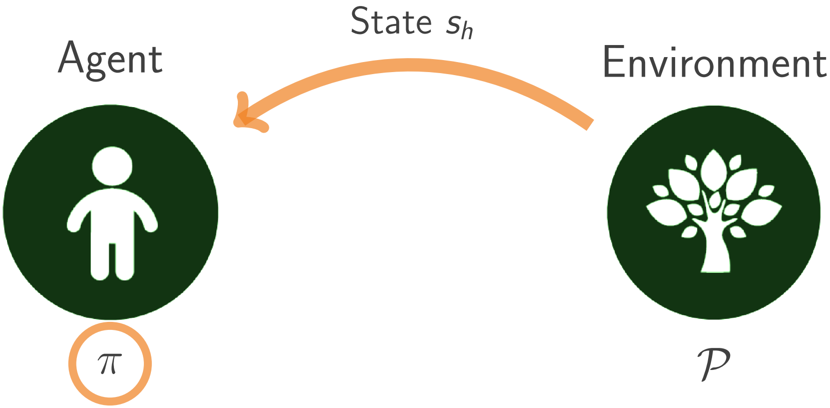
Environment

$\mathcal{P}$

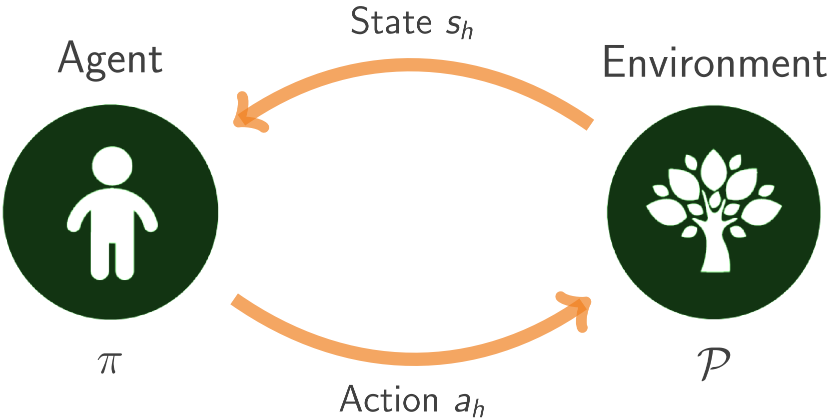
The Reinforcement Learning Framework



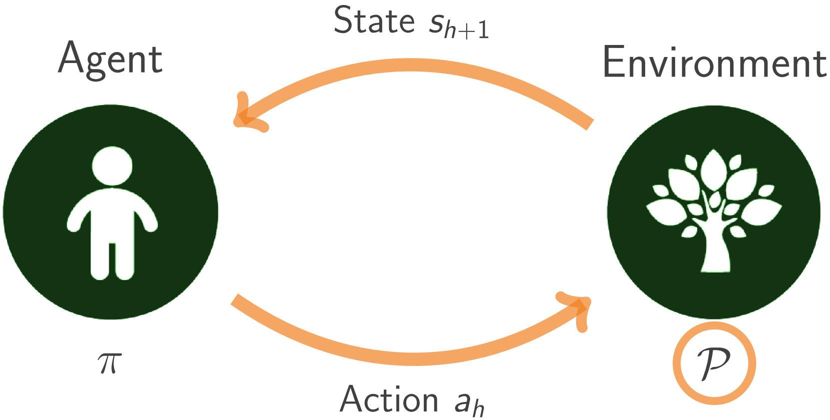
The Reinforcement Learning Framework



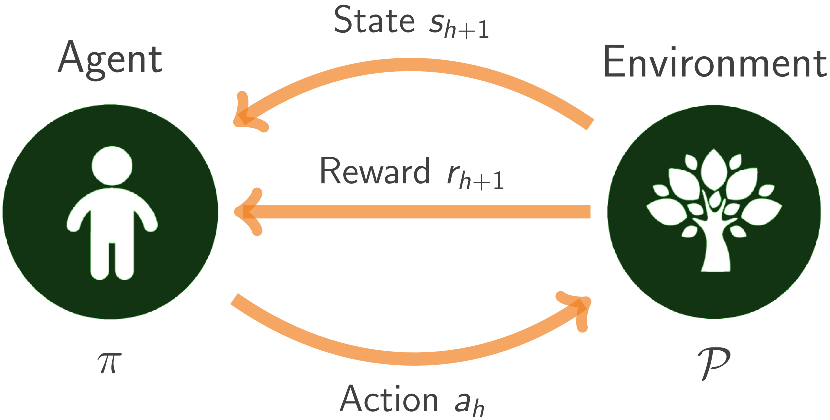
The Reinforcement Learning Framework



The Reinforcement Learning Framework



The Reinforcement Learning Framework



Policy Search Formulation

Parametric policy:

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

Policy Search Formulation

Parametric policy:

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

Return of a trajectory τ:

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

Policy Search Formulation

Parametric policy:

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

Return of a trajectory τ:

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

Performance:

$$\mu(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\tau \sim \pi_{\boldsymbol{\theta}}} [\mathcal{R}(\tau)]$$

Policy Search Formulation

Parametric policy:

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

Return of a trajectory τ:

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

Performance:

$$\mu(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\tau \sim \pi_{\boldsymbol{\theta}}} [\mathcal{R}(\tau)]$$

Objective:

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

Plan

Exploration in Policy Search

Undirected exploration

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

Exploration in Policy Search

Undirected exploration

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

- E.g.1: by adopting stochastic policies [Deisenroth et al., 2013].

Exploration in Policy Search

Undirected exploration

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

- E.g.1: by adopting stochastic policies [Deisenroth et al., 2013].
- E.g.2: by augmenting rewards with the entropy of the policy [Haarnoja et al., 2018]:

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

Exploration in Policy Search

Undirected exploration

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

- E.g.1: by adopting stochastic policies [Deisenroth et al., 2013].
- E.g.2: by augmenting rewards with the entropy of the policy [Haarnoja et al., 2018]:

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

Directed exploration

Leverage on the knowledge acquired during learning.

Plan

Problem Formalization

Decision Set

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

Problem Formalization

Decision 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

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

Procedure

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

1. **Select** a parametrization $\boldsymbol{\theta}_t \in \Theta$;

Problem Formalization

Decision Set

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

Procedure

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

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

Problem Formalization

Decision Set

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

Procedure

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

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

Problem Formalization

Decision Set

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

Procedure

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

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

Problem Formalization

Decision Set

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

Procedure

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

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

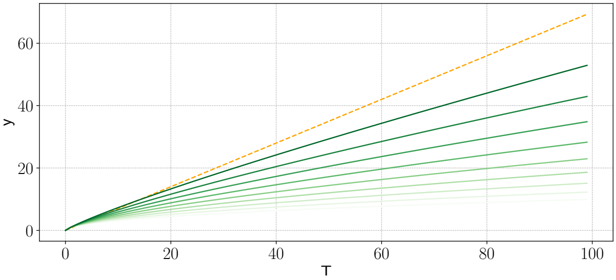
Goal

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

Desideratum

Sub-linear Regret

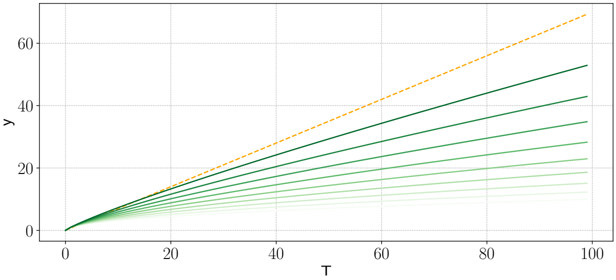
1. for any linear function $f(T)$, for sufficiently large input T, $Regret(T)$ grows slower than $f(T)$;
2. Alternatively, $\lim_{T \to \infty} Regret(T)/T = 0$.



Desideratum

Sub-linear Regret

1. for any linear function $f(T)$, for sufficiently large input T, $Regret(T)$ grows slower than $f(T)$;
2. Alternatively, $\lim_{T \to \infty} Regret(T)/T = 0$.
3. **Meaning**: after a certain number of iterations, the policy keeps improving.



Plan

Optimistic Policy opTImization via Multiple Importance Sampling with Truncation

$$\boldsymbol{\theta}_t = \arg\max_{\boldsymbol{\theta}\in\Theta} B_t(\boldsymbol{\theta}, \delta_t)$$

Optimistic Policy opTImization via Multiple Importance Sampling with Truncation

$$\boldsymbol{\theta}_t = \arg\max_{\boldsymbol{\theta}\in\Theta} B_t(\boldsymbol{\theta}, \delta_t)$$

$$\mu(\boldsymbol{\theta}) \leqslant B_t(\boldsymbol{\theta}, \delta_t)$$

OPTIMIST

Optimistic Policy opTImization via Multiple Importance Sampling with Truncation

$$\boldsymbol{\theta}_t = \arg\max_{\boldsymbol{\theta}\in\Theta} B_t(\boldsymbol{\theta}, \delta_t)$$

$$\mu(\boldsymbol{\theta}) \leqslant B_t(\boldsymbol{\theta}, \delta_t)$$

$$B_t(\boldsymbol{\theta}, \delta_t) = \ \check{\mu}_t^{MIS}(\boldsymbol{\theta}) \ + \ C_b\sqrt{\frac{d_2(\pi_{\boldsymbol{\theta}_t}\|\Phi_t) \ \log\frac{1}{\delta_t}}{t}}$$

Optimistic Policy opTImization via Multiple Importance Sampling with Truncation

$$\boldsymbol{\theta}_t = \arg\max_{\boldsymbol{\theta}\in\Theta} B_t(\boldsymbol{\theta}, \delta_t)$$

$$\mu(\boldsymbol{\theta}) \leqslant B_t(\boldsymbol{\theta}, \delta_t)$$

Truncated Multiple Importance Sampling Estimator

$$B_t(\boldsymbol{\theta}, \delta_t) = \boxed{\check{\mu}_t^{MIS}(\boldsymbol{\theta})} + C_b \sqrt{\frac{d_2(\pi_{\boldsymbol{\theta}_t} \| \Phi_t) \ \log \frac{1}{\delta_t}}{t}}$$

OPTIMIST

Optimistic Policy opTImization via Multiple Importance Sampling with Truncation

$$\boldsymbol{\theta}_t = \arg\max_{\boldsymbol{\theta}\in\Theta} B_t(\boldsymbol{\theta}, \delta_t)$$

$$\mu(\boldsymbol{\theta}) \leqslant B_t(\boldsymbol{\theta}, \delta_t)$$

Exploration Bonus

$$B_t(\boldsymbol{\theta}, \delta_t) = \check{\mu}_t^{MIS}(\boldsymbol{\theta}) + \boxed{C_b \sqrt{\frac{d_2(\pi_{\boldsymbol{\theta}_t} \| \Phi_t) \ \log \frac{1}{\delta_t}}{t}}}$$

Optimistic Policy opTImization via Multiple Importance Sampling with Truncation

$$\boldsymbol{\theta}_t = \arg\max_{\boldsymbol{\theta}\in\Theta} B_t(\boldsymbol{\theta}, \delta_t)$$

$$\mu(\boldsymbol{\theta}) \leqslant B_t(\boldsymbol{\theta}, \delta_t)$$

Exploration Bonus

$$B_t(\boldsymbol{\theta}, \delta_t) = \check{\mu}_t^{MIS}(\boldsymbol{\theta}) + \boxed{C_b \sqrt{\frac{\boxed{d_2(\pi_{\boldsymbol{\theta}_t} \| \Phi_t)} \log \frac{1}{\delta_t}}{t}}}$$

Regret Analysis - Discrete Arms Set

Theorem 1 - Discrete Parameter Space:

$$Regret(T) \leqslant \Delta_0 + C_1 \sqrt{T \left[v_1 \left(2 \log T + \log \frac{2\pi^2}{3\delta} \right) \right]}$$

Regret Analysis - Discrete Arms Set

Theorem 1 - Discrete Parameter Space:

$$Regret(T) \leqslant \Delta_0 + C_1 \sqrt{T \left[v_1 \left(2 \log T + \log \frac{2\pi^2}{3\delta} \right) \right]} = \mathcal{O}(\sqrt{T \log T})$$

Regret Analysis - Discrete Arms Set

Theorem 1 - Discrete Parameter Space:

$$Regret(T) \leqslant \Delta_0 + C_1 \sqrt{T \left[v_1 \left(2 \log T + \log \frac{2\pi^2}{3\delta} \right) \right]} = \mathcal{O}(\sqrt{T \log T})$$

Theorem 2 - Compact Parameter Space:

$$Regret(T) \leqslant C_2 + C_3 \sqrt{T \left[v_1 \left(2(d+1) \log T + d \log d + \log \frac{\pi^2}{3\delta} \right) \right]}$$

Regret Analysis - Discrete Arms Set

Theorem 1 - Discrete Parameter Space:

$$Regret(T) \leqslant \Delta_0 + C_1\sqrt{T\left[v_1\left(2\log T + \log\frac{2\pi^2}{3\delta}\right)\right]} = \mathcal{O}(\sqrt{T\log T})$$

Theorem 2 - Compact Parameter Space:

$$Regret(T) \leqslant C_2 + C_3\sqrt{T\left[v_1\left(2(d+1)\log T + d\log d + \log\frac{\pi^2}{3\delta}\right)\right]} = \mathcal{O}(\sqrt{dT\log T})$$

Regret Analysis - Discrete Arms Set

Theorem 1 - Discrete Parameter Space:

$$Regret(T) \leqslant \Delta_0 + C_1 \sqrt{T \left[v_1 \left(2 \log T + \log \frac{2\pi^2}{3\delta} \right) \right]} = \mathcal{O}(\sqrt{T \log T})$$

Theorem 2 - Compact Parameter Space:

$$Regret(T) \leqslant C_2 + C_3 \sqrt{T \left[v_1 \left(2(d+1) \log T + d \log d + \log \frac{\pi^2}{3\delta} \right) \right]} = \mathcal{O}(\sqrt{dT \log T})$$

Theorem 3 - Discretized Parameter Space:

$$Reg(T) \leqslant \Delta_0 + C_4 \sqrt{T} d + C_5 \sqrt{T \left[v_1 \left((2 + {}^{d}\!/_{2}) \log T + d \log 2 + \log \frac{\pi^2}{3\delta} \right) \right]}$$

Regret Analysis - Discrete Arms Set

Theorem 1 - Discrete Parameter Space:

$$Regret(T) \leqslant \Delta_0 + C_1 \sqrt{T \left[v_1 \left(2 \log T + \log \frac{2\pi^2}{3\delta} \right) \right]} = \mathcal{O}(\sqrt{T \log T})$$

Theorem 2 - Compact Parameter Space:

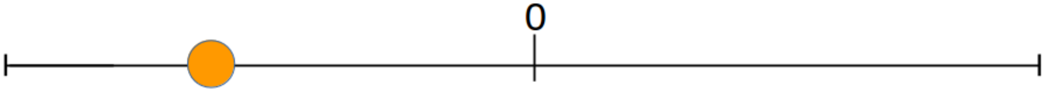
$$Regret(T) \leqslant C_2 + C_3 \sqrt{T \left[v_1 \left(2(d+1) \log T + d \log d + \log \frac{\pi^2}{3\delta} \right) \right]} = \mathcal{O}(\sqrt{dT \log T})$$

Theorem 3 - Discretized Parameter Space:

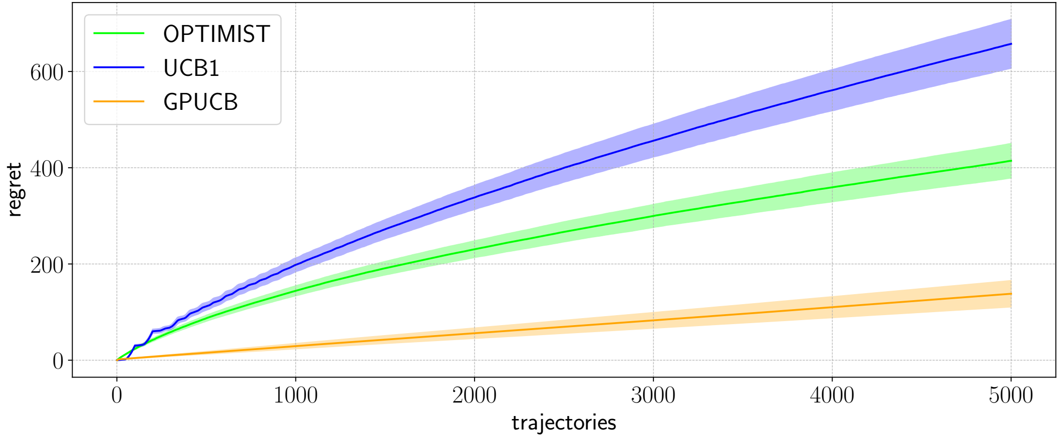
$$Reg(T) \leqslant \Delta_0 + C_4 \sqrt{T} d + C_5 \sqrt{T \left[v_1 \left((2 + d/2) \log T + d \log 2 + \log \frac{\pi^2}{3\delta} \right) \right]} = \mathcal{O}(\sqrt{dT \log T})$$

Plan

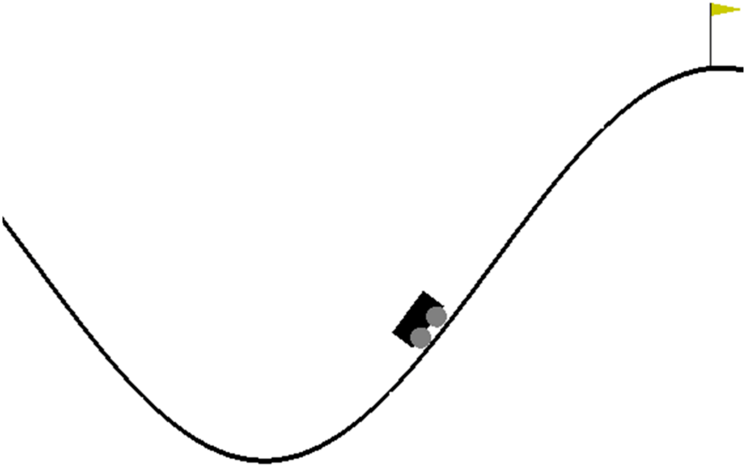
Linear Quadratic Gaussian Regulator



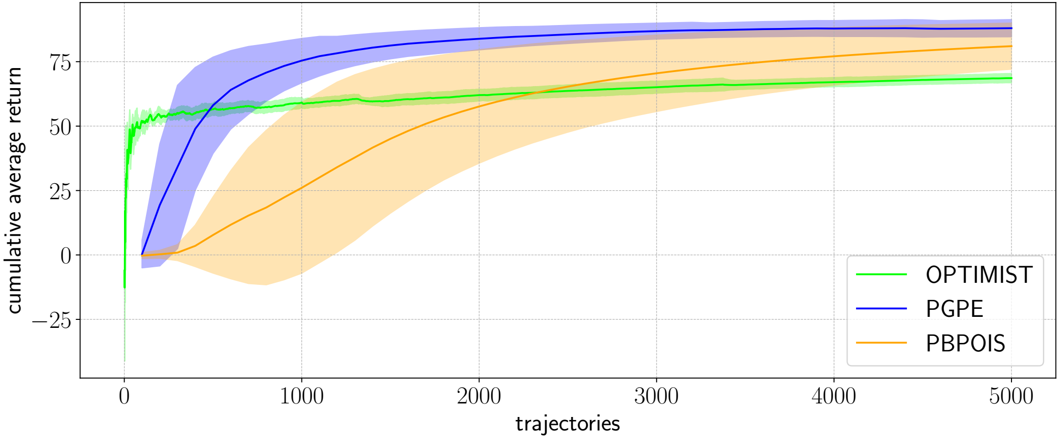
Linear Quadratic Gaussian Regulator - Regret



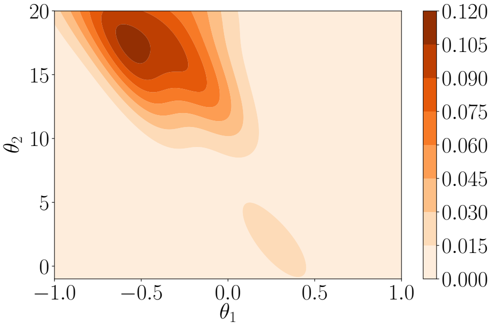
Mountain Car



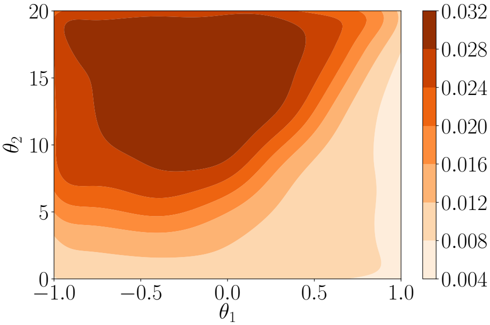
Mountain Car - Performance



Mountain Car - Parameter Space Exploration



PGPE



OPTIMIST

Plan

Conclusions

Original Contributions

1. Algorithmic contributions: OPTIMIST

Conclusions

Original Contributions

1. Algorithmic contributions: OPTIMIST
2. Theoretical contributions:

Conclusions

Original Contributions

1. Algorithmic contributions: OPTIMIST
2. Theoretical contributions:
   2.1 novel problem formalization of Policy Search;

Conclusions

Original Contributions

1. Algorithmic contributions: OPTIMIST
2. Theoretical contributions:
    2.1 novel problem formalization of Policy Search;
    2.2 proved sub-linear regret for both algorithms.

Conclusions

Original Contributions

1. Algorithmic contributions: OPTIMIST
2. Theoretical contributions:
   2.1 novel problem formalization of Policy Search;
   2.2 proved sub-linear regret for both algorithms.
3. Experimental contributions.

Conclusions

Original Contributions

1. Algorithmic contributions: OPTIMIST
2. Theoretical contributions:
   2.1 novel problem formalization of Policy Search;
   2.2 proved sub-linear regret for both algorithms.
3. Experimental contributions.

Paper submission at **ICML2019** (International Conference on Machine Learning)

Conclusions

Original Contributions

1. Algorithmic contributions: OPTIMIST
2. Theoretical contributions:
   2.1 novel problem formalization of Policy Search;
   2.2 proved sub-linear regret for both algorithms.
3. Experimental contributions.

Paper submission at **ICML2019** (International Conference on Machine Learning)

Future Works

Conclusions

Original Contributions

1. Algorithmic contributions: OPTIMIST
2. Theoretical contributions:
    2.1 novel problem formalization of Policy Search;
    2.2 proved sub-linear regret for both algorithms.
3. Experimental contributions.

Paper submission at **ICML2019** (International Conference on Machine Learning)

Future Works

1. Study new solutions to the optimization problem arising for compact decision sets;

Conclusions

Original Contributions

1. Algorithmic contributions: OPTIMIST
2. Theoretical contributions:
   2.1 novel problem formalization of Policy Search;
   2.2 proved sub-linear regret for both algorithms.
3. Experimental contributions.

Paper submission at **ICML2019** (International Conference on Machine Learning)

Future Works

1. Study new solutions to the optimization problem arising for compact decision sets;
2. Leverage on posterior sampling techniques instead of upper confidence bound optimization.

Thank you for your attention!

Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. (2016).
Unifying count-based exploration and intrinsic motivation.
In *Advances in Neural Information Processing Systems*, pages 1471–1479.

Bubeck, S., Cesa-Bianchi, N., and Lugosi, G. (2013).
Bandits with heavy tail.
IEEE Transactions on Information Theory, 59(11):7711–7717.

Deisenroth, M. P., Neumann, G., Peters, J., et al. (2013).
A survey on policy search for robotics.
Foundations and Trends® in Robotics, 2(1–2):1–142.

References II

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018).
Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor.
In *Proceedings of the 35th International Conference on Machine Learning*, pages 1856–1865.

Lattimore, T. and Szepesvári, C. (2019).
Bandit Algorithms.
Cambridge University Press (preprint).

Srinivas, N., Krause, A., Kakade, S., and Seeger, M. (2010).
Gaussian process optimization in the bandit setting: No regret and experimental design.

References III

In Fürnkranz, J. and Joachims, T., editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 1015–1022, Haifa, Israel. Omnipress.

Sutton, R. S. and Barto, A. G. (2018).
Reinforcement learning: An introduction.
MIT press.

Markov Decision Processes

Reinforcement Learning

General class of algorithms that allow an agent to learn how to behave in a stochastic and possibly unknown environment by trial-and-error.

Markov Decision Process (MDP)

stochastic dynamical system specified by $< \mathbb{S}, \mathbb{A}, \mathcal{P}, \mathcal{R}, \gamma >$

1. $(\mathbb{S}, \mathcal{S})$ is a measurable state space
2. $(\mathbb{A}, \mathcal{A})$ is a measurable action space
3. $\mathcal{P} : \mathbb{S} \times \mathbb{A} \times \mathcal{S} \to \mathbb{R}$ is a Markov transition kernel
4. $\mathcal{R} : \mathbb{S} \times \mathbb{A} \to \mathbb{R}$ is a reward function
5. $0 < \gamma < 1$ is the discount factor.

Monte-Carlo Policy Gradient: Pseudocode

Input: Stochastic policy π_θ, Initial parameters θ_0, learning rate $\{\alpha_k\}$
Output: Approximation of the optimal policy $\pi_{\theta*} \approx \pi_*$

1: **repeat**
2: Sample M trajectories $h^{(m)} = \{(s_t^{(m)}, a_t^{(m)}, r_{t+1}^{(m)})\}_{t=0}^{T^{(m)}}$ under policy π_{θ_k}
3: Approximate policy gradient

$$\nabla_\theta J(\theta_k) \approx \frac{1}{M} \sum_{m=0}^{M} \sum_{u=0}^{T^{(m)}-1} \nabla_\theta \log \pi_{\theta_k} \left(s_u^{(m)}, a_u^{(m)}\right) \sum_{v \geqslant u}^{T^{(m)}-1} \gamma^{v-u} r_{v+1}^{(m)}$$

4: Update parameters using gradient ascent $\theta_{k+1} = \theta_k + \alpha_k \nabla_\theta J(\theta_k)$
5: $k \leftarrow k + 1$
6: **until** converged

Episodic PGPE Algorithm: Pseudocode

Input: Controller F_θ, hyper-distribution p_ξ, initial guess ξ_0, learning rate $\{\alpha_k\}$
Output: Approximation of the optimal policy $F_{\xi*} \approx \pi_*$

1: **repeat**
2: **for** $m = 1, \dots, M$ **do**
3: Sample controller parameters $\theta^{(m)} \sim p_{\xi_k}$
4: Sample trajectory $h^{(m)} = \{(s_t^{(m)}, a_t^{(m)}, r_{t+1}^{(m)})\}_{t=0}^{T^{(m)}}$ under policy $F_{\theta^{(m)}}$
5: **end for**
6: Approximate policy gradient

$$\nabla_\xi J(\xi_k) \approx \frac{1}{M} \sum_{m=1}^{M} \nabla_\xi \log p_\xi \left(\theta^{(m)}\right) \left[G\left(h^{(m)}\right) - b\right]$$

7: Update hyperparameters using gradient ascent $\xi_{k+1} = \xi_k + \alpha_k \nabla_\xi J(\xi_k)$
8: $k \leftarrow k + 1$
9: **until** converged

Truncated Multiple Importance Sampling Estimator

Importance Sampling

Given a bounded function $f : \mathcal{Z} \to \mathbb{R}$, and a set of i.i.d. outcomes $z_1, \ldots, z_N$ sampled from Q, the importance sampling estimator of $\mu := \mathop{\mathbb{E}}_{z \sim P} [f(z)]$ is:

$$\widehat{\mu}_{\text{IS}} = \frac{1}{N} \sum_{i=1}^{N} f(z_i) w_{P/Q}(z_i), \tag{1}$$

which is an unbiased estimator, i.e., $\mathop{\mathbb{E}}_{z_i \overset{\text{iid}}{\sim} Q} [\widehat{\mu}_{IS}] = \mu$.

Truncated Estimator With Balance Heuristic

$$\check{\mu}_{\text{BH}} = \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{N_k} \min \left\{ M, \frac{p(z_{ik})}{\sum_{j=1}^{K} \frac{N_j}{N} q_j(z_{ik})} \right\} f(z_{ik}). \tag{2}$$

OPTIMIST2

Theorem

regretdiscretized Let $\mathcal{X}$ be a d-dimensional compact arm set with $\mathcal{X} \subseteq [-D, D]^d$. For any $\kappa \geqslant 2$, under Assumptions 1 and 2, OPTIMIST2 with confidence schedule $\delta_t = \dfrac{6\delta}{\pi^2 t^2 \left(1 + \lceil t^{1/\kappa} \rceil^d\right)}$ and discretization schedule $\tau_t = \lceil t^{\frac{1}{\kappa}} \rceil$ guarantees, with probability at least $1 - \delta$:

$$Regret(T) \leqslant \Delta_0 + C_1 T^{\left(1 - \frac{1}{\kappa}\right)d} + C_2 T^{\frac{1}{1+\epsilon}} \cdot \left[v_\epsilon \left((2 + {}^d\!/\!_\kappa) \log T + d \log 2 + \log \frac{\pi^2}{3\delta} \right) \right]^{\frac{\epsilon}{1+\epsilon}},$$

where $C_1 = \dfrac{\kappa}{\kappa - 1} LD$, $C_2 = (1 + \epsilon) \left(2\sqrt{2} + \dfrac{5}{3} \right) \|f\|_\infty$, and Δ_0 is the instantaneous regret of the initial arm $\boldsymbol{x}_0$.