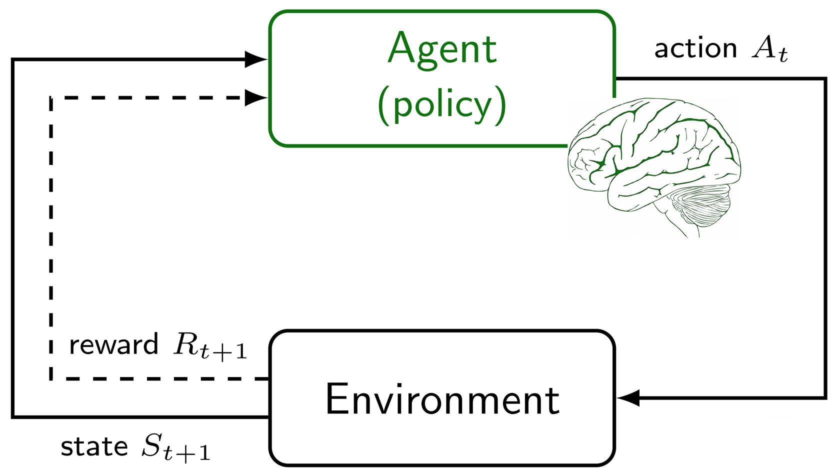
Reinforcement Learning in Configurable Continuous Environments

Alberto Maria Metelli, Emanuele Ghelfi and Marcello Restelli
{albertomaria.metelli, marcello.restelli}@polimi.it, emanuele.ghelfi@mail.polimi.it
POLITECNICO MILANO 1863

Problem

Markov Decision Process
(MDP, Puterman, 2014)

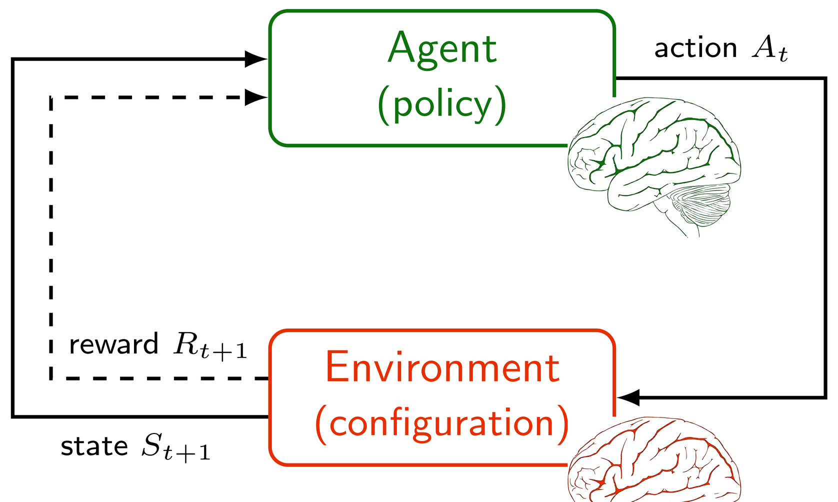


$$S_0 \sim \mu,\ A_t \sim \pi_{\theta}(\cdot|S_t),\ S_{t+1} \sim p(\cdot|S_t, A_t)$$

- Learn the policy parameters θ under the fixed environment p:

$$\theta^* = \arg\max_{\theta \in \Theta} J(\theta) = \mathbb{E}\left[\sum_{t=0}^{+\infty} \gamma^t R_{t+1}\right]$$

Configurable Markov Decision Process
(Conf-MDP, Metelli et al., 2018)



$$S_0 \sim \mu,\ A_t \sim \pi_{\theta}(\cdot|S_t),\ S_{t+1} \sim p_{\omega}(\cdot|S_t, A_t)$$

- Learn the policy parameters θ together with the environment configuration ω:

$$\theta^*, \omega^* = \arg\max_{\theta \in \Theta, \omega \in \Omega} J(\theta, \omega) = \mathbb{E}\left[\sum_{t=0}^{+\infty} \gamma^t R_{t+1}\right]$$

Motivation and Contributions

- Existing approaches (SPMI, Metelli et al., 2018):
  - work in **finite** state-actions spaces only
  - require the **full knowledge** of the environment dynamics

How to solve continuous Conf-MDPs without the knowledge of the exact dynamics?

- **REMPS** is able to learn in continuous Conf-MDPs and can be equipped with an approximate transition model
  - REMPS alternates **optimization** and **projection** phases
  - Inspired to *trust region* methods (Peters et al., 2010)
  - We provide a *finite-sample analysis* for the single step
  - We *empirically* evaluate REMPS on Conf-MDPs

Relative Entropy Model Policy Search (REMPS)

Optimization

- π_{θ} and p_{ω} induce a *stationary distribution* d_{π_θ, p_ω}
- **Goal**: find a new stationary distribution d' in a *trust region* centered in d_{π_θ, p_ω} (PRIMAL$_\kappa$):

$$\max_{d'} J_{d'} = \mathop{\mathbb{E}}_{S,A,S' \sim d'}[r(S,A,S')] \quad \text{s.t. } D_{\mathrm{KL}}(d' \| d_{\pi_\theta, p_\omega}) \le \kappa$$

- $\kappa > 0$ defines the trust region in KL-divergence
- We solve the dual problem (DUAL$_\kappa$) in the Lagrange multiplier η:

$$\min_{\eta \in [0, +\infty)} \eta \log \mathop{\mathbb{E}}_{S,A,S' \sim d_{\pi_\theta, p_\omega}}\left[\exp\left(\frac{1}{\eta} r(S,A,S') + \kappa\right)\right]$$

- d' *exponentially reweighs* the probability of each (s,a,s') by the reward $r(s,a,s')$

$$d'(s,a,s') \propto d_{\pi_\theta, p_\omega}(s,a,s') \exp\left(\frac{1}{\eta} r(s,a,s')\right)$$

- When using *samples* $\{(s_i, a_i, s_i', r_i)\}_{i=1}^N$ we minimize the empirical version of DUAL$_\kappa$

Projection

- d' might fall outside the space of representable stationary distribution, given $\Theta \times \Omega$
- **Goal:** perform a *moment projection* onto $\Theta \times \Omega$:
  - project the stationary distribution (PROJ$_d$)

$$\min_{\theta' \in \Theta, \omega' \in \Omega} D_{\mathrm{KL}}\left(d' \| d_{\pi_{\theta'}, p_{\omega'}}\right)$$

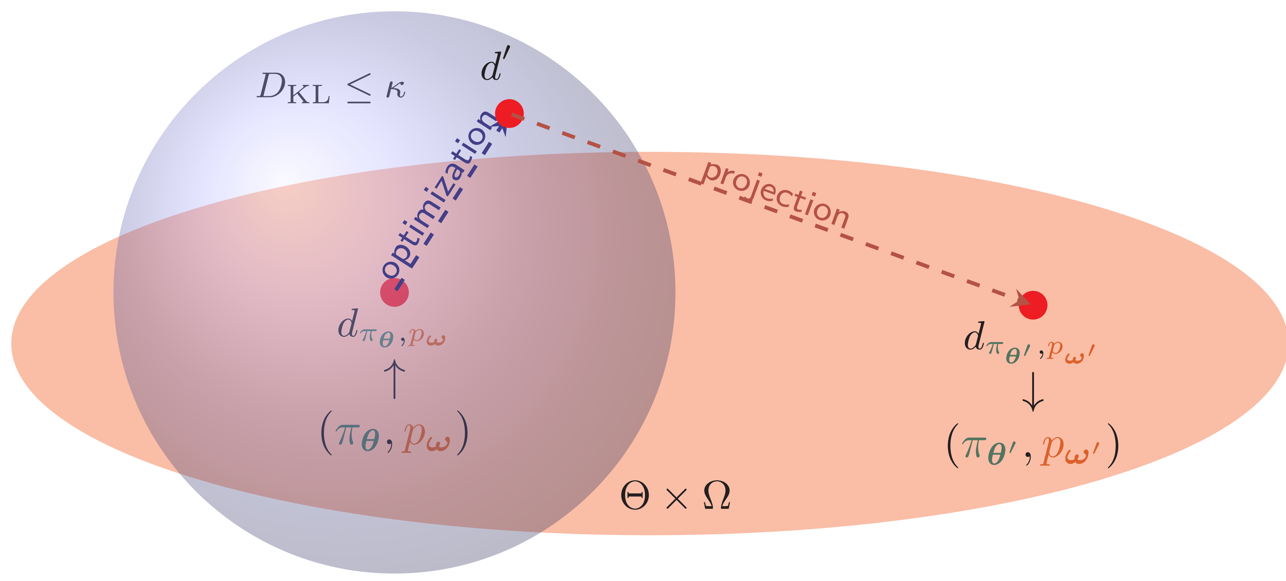
  - project the state kernel (PROJ$_{p^\pi}$)

$$\min_{\theta' \in \Theta, \omega' \in \Omega} \mathop{\mathbb{E}}_{S \sim d'}\left[D_{\mathrm{KL}}\left(p'^{\pi'}(\cdot|S) \| p_{\omega'}^{\pi_{\theta'}}(\cdot|S)\right)\right]$$

  - project the policy and model separately (PROJ$_{\pi,p}$)

$$\min_{\theta \in \Theta} \mathop{\mathbb{E}}_{S \sim d'}\left[D_{\mathrm{KL}}(\pi'(\cdot|S) \| \pi_{\theta'}(\cdot|S))\right]$$
$$\min_{\omega \in \Omega} \mathop{\mathbb{E}}_{S,A \sim d''}\left[D_{\mathrm{KL}}(p'(\cdot|S,A) \| p_{\omega'}(\cdot|S,A))\right]$$

- When using *samples* $\{(s_i, a_i, s_i', r_i)\}_{i=1}^N$ we minimize the log-likelihood employing *importance sampling* (Owen, 2013)



Approximate Transition Model

- In the projection phase, we can use an *approximate transition model* $\widehat{p}$ learned from samples $\{(s_i, a_i, s_i', \omega_i)\}_{i=1}^N$ instead of the real one

$$\max_{\widehat{p} \in \widehat{\mathcal{P}}_\Omega} \frac{1}{N} \sum_{i=1}^N \log \widehat{p}(s_i'|s_i, a_i, \omega_i),$$

- $\widehat{p}$ can be any model approximator (e.g., neural network, Gaussian process)

Theoretical Analysis

- **Goal**: bound the performance of the stationary distribution d' obtained with infinite samples and the performance of $(\pi_{\widetilde{\theta}'}, p_{\widetilde{\omega}'})$ obtained after projection using N i.i.d. samples

$$J_{d'} - J(\widetilde{\theta}', \widetilde{\omega}') \le \sqrt{2} r_{\max} \sup_{d : D_{\mathrm{KL}}(d' \| d_{\pi_\theta, p_\omega}) \le \kappa} \inf_{\overline{\theta} \in \Theta, \overline{\omega} \in \Omega} \sqrt{D_{\mathrm{KL}}(d \| d_{\pi_{\overline{\theta}}, p_{\overline{\omega}}})} + \widetilde{\mathcal{O}}\left(\sqrt[4]{\frac{v \log \frac{2eN}{v} + \log \frac{8}{\delta}}{N}}\right)$$

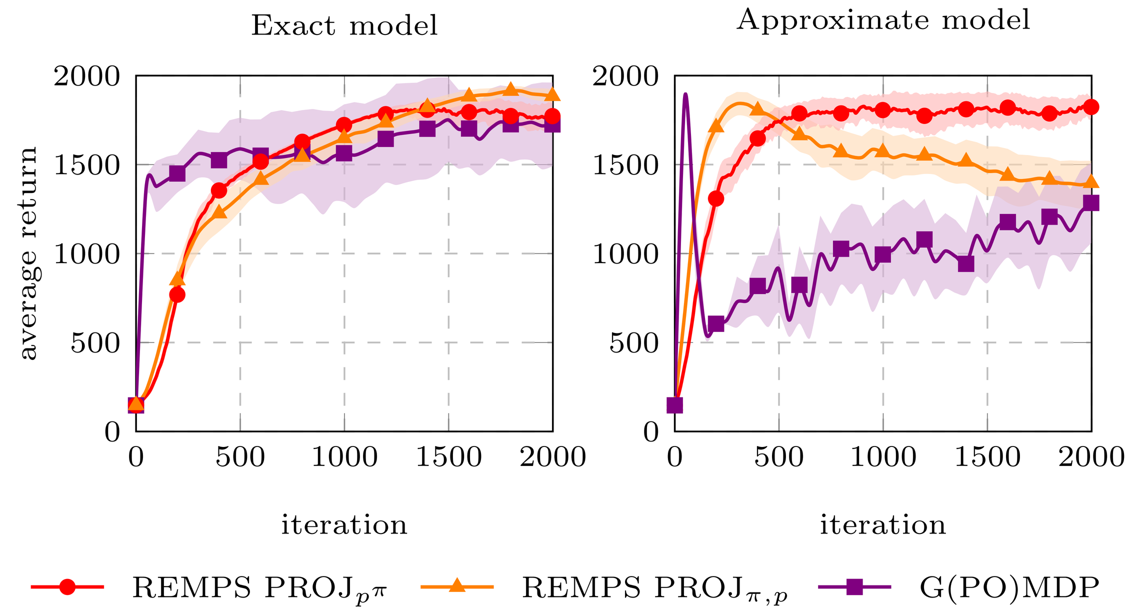
Approximation Error: due to the finite capacity of the parametric space $\Theta \times \Omega$, depends also on the threshold κ

Estimation Error: due to the finite samples N, depends also on the pseudo-dimension v (Cortes et al., 2013)

Experiments

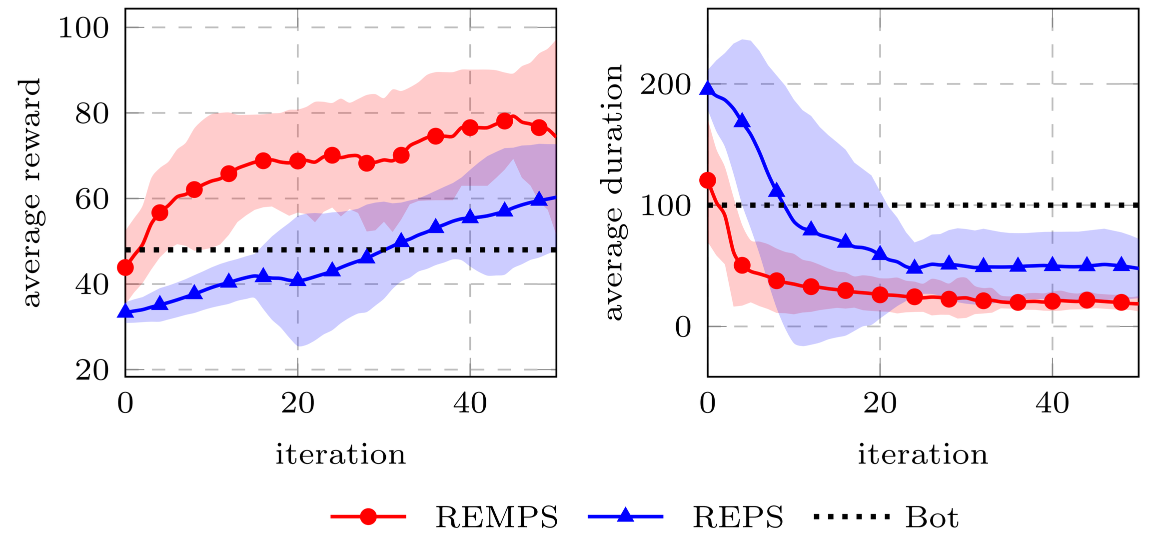
Cartpole

- Configure the *cart force*

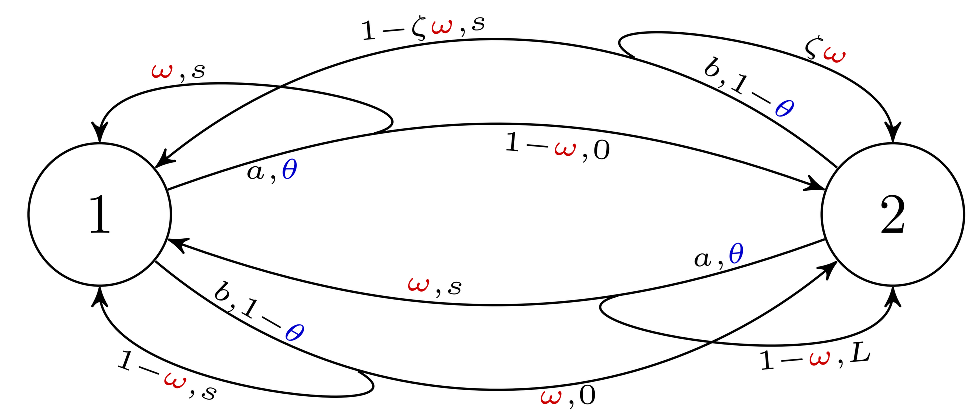


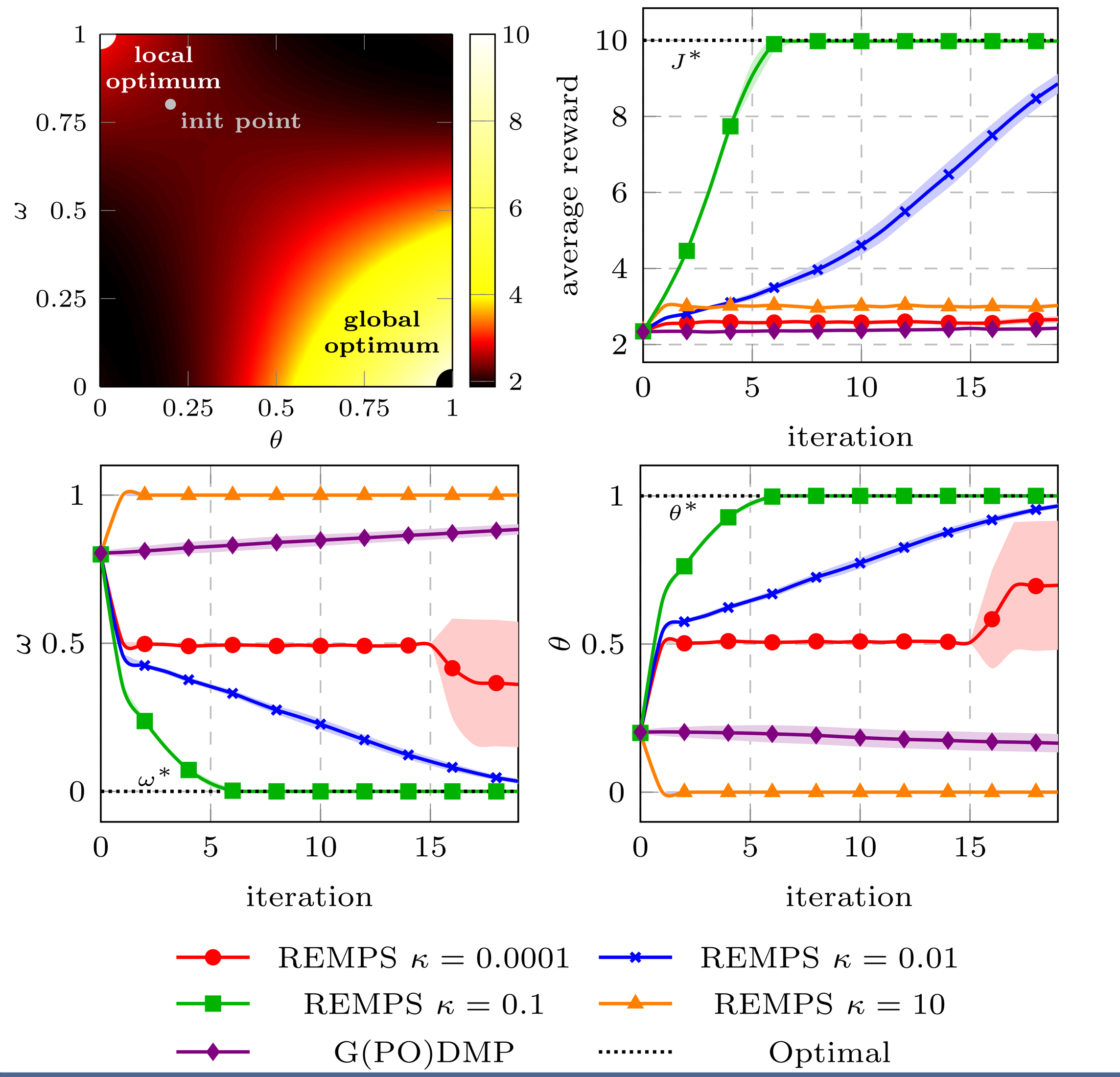
TORCS

- Configure the *front-rear wing orientation* and *brake repartition*



Chain Domain





References

C. Cortes, S. Greenberg, and M. Mohri. Relative deviation learning bounds and generalization with unbounded loss functions. *arXiv preprint arXiv:1310.5796*, 2013.

A. M. Metelli, M. Mutti, and M. Restelli. Configurable markov decision processes. In *35th International Conference on Machine Learning*, volume 80, pages 3491–3500. PMLR, 2018.

A. B. Owen. *Monte Carlo theory, methods and examples*. 2013.

J. Peters, K. Mülling, and Y. Altun. Relative entropy policy search. In *AAAI*, pages 1607–1612. Atlanta, 2010.

M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.