POIS

Policy Optimization via Importance Sampling

Alberto M. Metelli, Matteo Papini, Francesco Faccio and Marcello Restelli
{albertomaria.metelli, matteo.papini, marcello.restelli}@polimi.it, francesco.faccio@mail.polimi.it
POLITECNICO MILANO 1863

Problem and Motivation

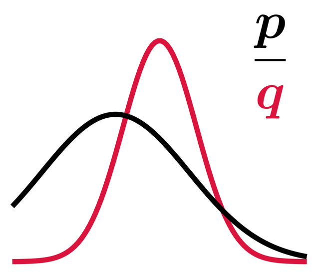
- **Reinforcement Learning (RL):** find optimal policy
- **Policy Search**: search over a class of policies π
- Every policy induces a distribution $p(\tau|\pi)$ over **trajectories** of the Markov Decision Process (MDP)
- Every trajectory τ has a **return** $R(\tau)$
- **Goal:** find π^* maximizing $J(\pi) = \mathbb{E}_\pi[R(\tau)]$
- **Using data collected with some policy $\widetilde{\pi}$:**
  - How can I evaluate proposals $\pi \neq \widetilde{\pi}$?
  - How can I trust conterfactual evaluations?
  - How can I best use my data for optimization?

Importance Sampling

How can I evaluate proposals?
With **Importance Sampling (IS):**

- Given a **behavioral** (data-sampling) distribution $p(x)$, a **target** distribution $p(x)$, and a function $f(x)$, **estimate** $\boldsymbol{\mu} = \mathbb{E}_{\boldsymbol{x \sim p}}[f(x)]$ with data **from** q:

$$\widehat{\mu}_{\text{IS}} = \frac{1}{N} \sum_{i=1}^{N} \underbrace{\frac{\boldsymbol{p(x_i)}}{\boldsymbol{q(x_i)}}}_{\boldsymbol{w(x_i)}} f(x_i)$$



- $x_i \sim q$ for $i = 1, 2, \dots, N$
- $\boldsymbol{w(x)} = p(x)/q(x)$ is the **importance weight**
- The estimate is **unbiased**: $\mathbb{E}_q[\widehat{\mu}_{\text{IS}}] = \mu$
- The variance can be very high!
- **Rényi divergence** measures the distance between p and q:

$$D_2(p\|q) = \log \mathop{\mathbb{E}}_{x \sim q}\left[\left(\frac{p(x)}{q(x)}\right)^2\right]$$

$$d_2(p\|q) = \exp\{D_2(p\|q)\} \quad \textbf{exponentiated Rényi}$$

- Variance of the weight depends **exponentially** on the distributional divergence (Cortes et al., 2010)

$$\mathbb{V}\text{ar}[w] = d_2(p\|q) - 1$$

- **Effective Sample Size (ESS)**: number of equivalent samples in plain Monte Carlo estimation ($x_i \sim p$)

$$\text{ESS} = \frac{N}{d_2(p\|q)} \approx \frac{\|w\|_1^2}{\|w\|_2^2} = \widehat{ESS}$$

Off-Distribution Learning

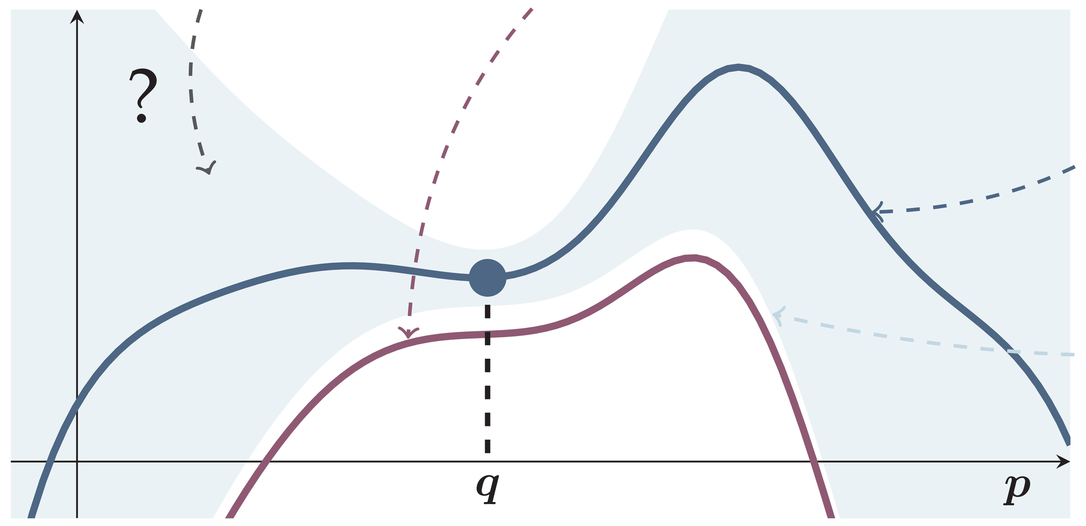
How (far) can I trust conterfactual evaluations?

- Evaluate only close solutions: REPS (Peters et al., 2010), TRPO (Schulman et al., 2015)
- **Use a lower bound**: EM (Dayan and Hinton, 1997; Kober et al., 2011), PPO (Schulman et al., 2017), **POIS**

Given a behavioral $q(x)$, a function $f(x)$ and a proposal $p(x)$, with probability at least $1-\delta$:

True function — Lower Bound — IS Estimator — Variance Bound (Cantelli)

$$\mathop{\mathbb{E}}_{x \sim p}[f(x)] \geq \mathcal{L}_\delta^{\text{POIS}}(p/q) = \frac{1}{N}\sum_{i=1}^{N} w(x_i) f(x_i) - \|f\|_\infty^2 \sqrt{\frac{(1-\delta) d_2(p\|q)}{\delta N}}$$



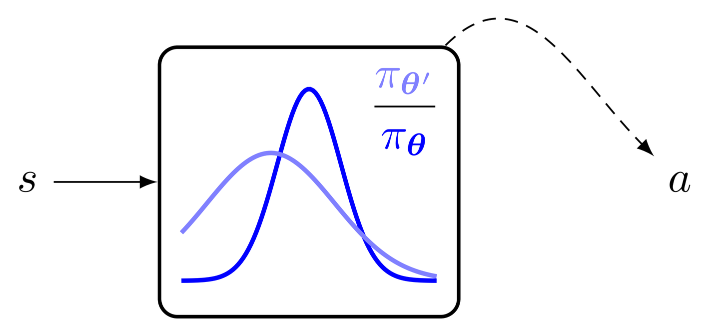
How can I best use my data for optimization?
Given the behavioral q, find p **maximizing** $\mathbb{E}_{x\sim p}[f(x)]$:

- Collect data with q (expensive in RL)
- Find p maximizing $\mathcal{L}(p/q)$ (offline optimization)
- Set new behavioral $q \leftarrow p$
- Repeat until convergence

Action-Based POIS

- Find the **policy** parameters $\boldsymbol{\theta}^*$ that maximize $J(\boldsymbol{\theta}')$

$$J(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\tau \sim p(\cdot|\boldsymbol{\theta})}[R(\tau)]$$



- Given a **behavioral policy** $\pi_{\boldsymbol{\theta}}$ we compute a target policy $\pi_{\boldsymbol{\theta}}$ by optimizing:

$$\mathcal{L}_\lambda^{\text{A-POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}\prod_{t=0}^{H-1}\frac{\pi_{\boldsymbol{\theta}'}(a_{\tau_i,t}|s_{\tau_i,t})}{\pi_{\boldsymbol{\theta}}(a_{\tau_i,t}|s_{\tau_i,t})} R(\tau_i) - \lambda\sqrt{\frac{\widehat{d}_2\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right)}{N}}$$

- The term $d_2\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right)$ is estimated from samples
- The d_2 grows exponentially with the task horizon H
- λ is a regularization hyperparameter

$$\lambda = \frac{R_{\max}}{1-\gamma}\sqrt{\frac{1-\delta}{\delta}}$$

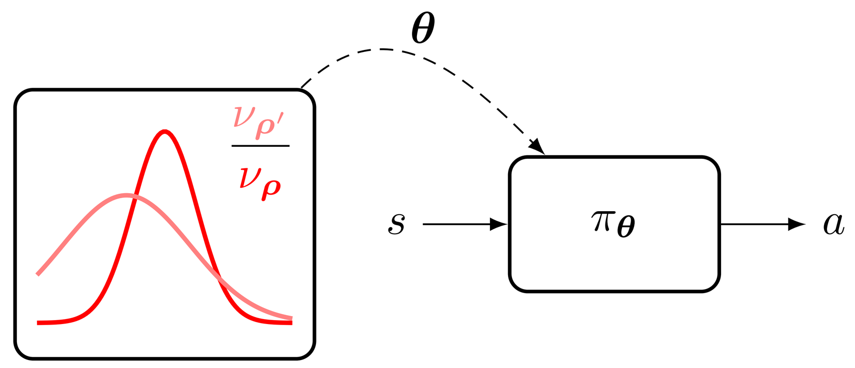
- We consider diagonal Gaussian policies $\pi_{\boldsymbol{\theta}}$

$$a \sim \pi_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\cdot|s) = \mathcal{N}\left(u_{\boldsymbol{\mu}}(s), \text{diag}(\boldsymbol{\sigma}^2)\right)$$

Parameter-Based POIS

- Find the **hyperpolicy** parameters $\boldsymbol{\rho}^*$ that maximize $J(\boldsymbol{\rho})$

$$J(\boldsymbol{\rho}) = \mathop{\mathbb{E}}_{\boldsymbol{\theta} \sim \nu_{\boldsymbol{\rho}}} \mathop{\mathbb{E}}_{\tau \sim p(\cdot|\boldsymbol{\theta})}[R(\tau)]$$



- Given a **behavioral hyperpolicy** $\nu_{\boldsymbol{\rho}}$ we compute a target hyperpolicy $\nu_{\boldsymbol{\rho}'}$ by optimizing:

$$\mathcal{L}_\lambda^{\text{P-POIS}}(\boldsymbol{\rho}'/\boldsymbol{\rho}) = \frac{1}{N}\sum_{i=1}^{N}\frac{\nu_{\boldsymbol{\rho}'}(\boldsymbol{\theta}_i)}{\nu_{\boldsymbol{\rho}}(\boldsymbol{\theta}_i)} R(\tau_i) - \lambda\sqrt{\frac{d_2\left(\nu_{\boldsymbol{\rho}'}\|\nu_{\boldsymbol{\rho}}\right)}{N}}$$

- The term $d_2\left(\nu_{\boldsymbol{\rho}'}\|\nu_{\boldsymbol{\rho}}\right)$ can be computed exactly
- Affected by the parameter space dimension $\dim(\boldsymbol{\theta})$
- λ is a regularization hyperparameter

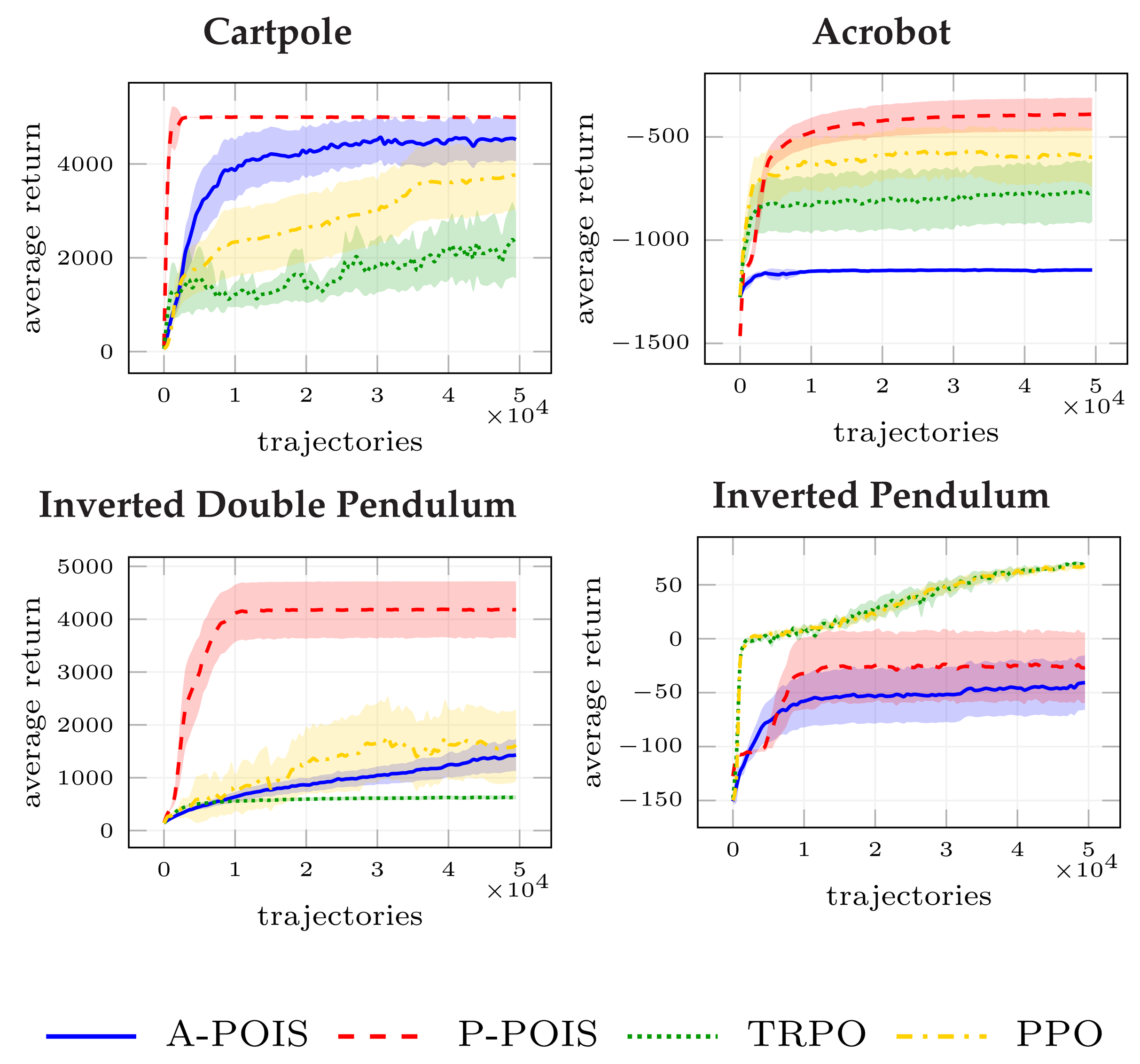
$$\lambda = \frac{R_{\max}}{1-\gamma}\sqrt{\frac{1-\delta}{\delta}}$$

- We consider diagonal Gaussian hyperpolicies $\nu_{\boldsymbol{\rho}}$

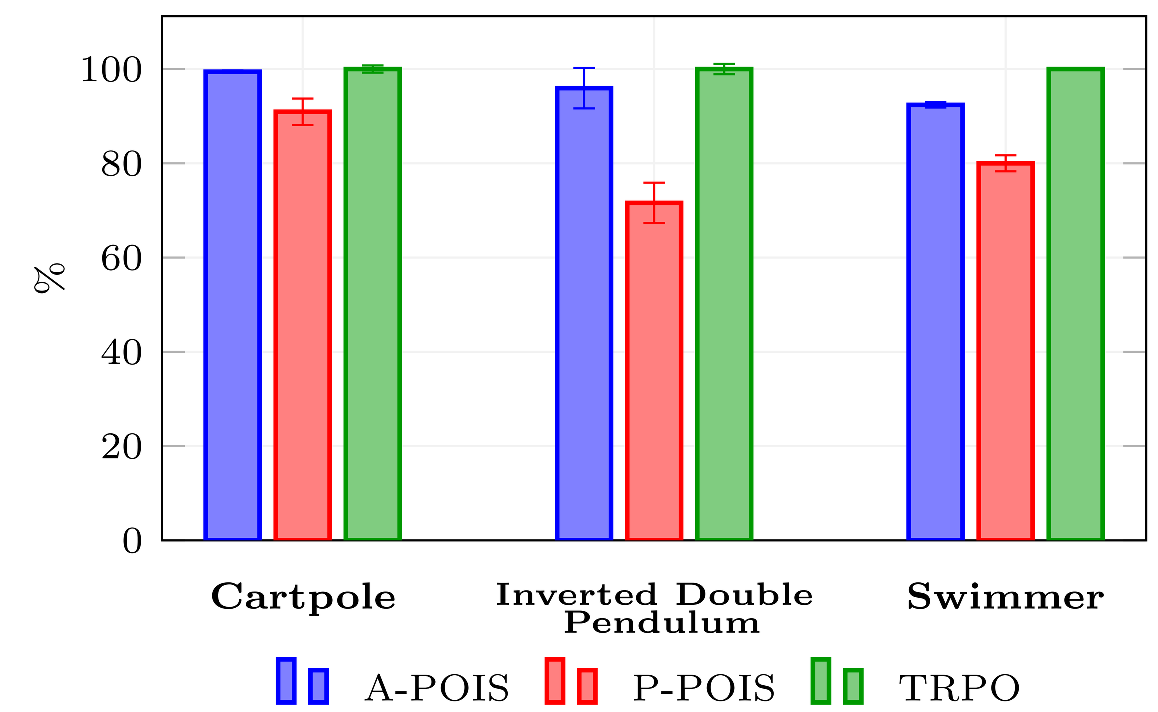
$$\boldsymbol{\theta} \sim \nu_{\boldsymbol{\mu},\boldsymbol{\sigma}} = \mathcal{N}\left(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}^2)\right)$$

Experiments

Linear Policies



Deep Policies



Practical Tricks

- *Self–normalized* importance sampling (Owen, 2013)

$$\widetilde{\mu}_{P/Q} = \frac{\sum_{i=1}^{N} w_{P/Q}(x_i) f(x_i)}{\sum_{i=1}^{N} w_{P/Q}(x_i)} \qquad x_i \sim Q$$

- Effective Sample Size vs d_2
- Gradient optimization of the bound using *line search*
- Natural gradient for P-POIS

References

C. Cortes, Y. Mansour, and M. Mohri. Learning bounds for importance weighting. In *NeurIPS*, 2010.
P. Dayan and G. E. Hinton. Using expectation-maximization for reinforcement learning. *Neural Computation*, 1997.
J. Kober, E. Öztop, and J. Peters. Reinforcement learning to adjust robot movements to new situations. In *IJCAI*, 2011.
A. B. Owen. *Monte Carlo theory, methods and examples*. 2013.
J. Peters, K. Mülling, and Y. Altun. Relative entropy policy search. In *AAAI*, 2010.
J. Schulman, S. Levine, P. Abbeel, et al. Trust region policy optimization. In *ICML*, 2015.
J. Schulman, F. Wolski, P. Dhariwal, et al. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.