Optimistic Policy Optimization via Multiple Importance Sampling

Matteo Papini  Alberto Maria Metelli
Lorenzo Lupo  Marcello Restelli

11th June 2019
Thirty-sixth International Conference on Machine Learning, Long Beach, CA, USA
Policy Optimization

- **Parameter space** $\Theta \subseteq \mathbb{R}^d$

- A parametric policy for each $\theta \in \Theta$

- Each inducing a distribution $p_\theta$ over trajectories

- A return $R(\tau)$ for every trajectory $\tau$

- **Goal:** $\max_{\theta \in \Theta} J(\theta) = \mathbb{E}_{\tau \sim p_\theta} [R(\tau)]$

- Iterative optimization (e.g., gradient ascent)
Policy Optimization

- **Parameter space** $\Theta \subseteq \mathbb{R}^d$

- A parametric policy for each $\theta \in \Theta$

- Each inducing a distribution $p_\theta$ over trajectories

- A return $R(\tau)$ for every trajectory $\tau$

- **Goal:** $\max_{\theta \in \Theta} J(\theta) = \mathbb{E}_{\tau \sim p_\theta} [R(\tau)]$

- Iterative optimization (e.g., gradient ascent)
Policy Optimization

- **Parameter space** $\Theta \subseteq \mathbb{R}^d$

- A parametric **policy** for each $\theta \in \Theta$

- Each inducing a distribution $p_\theta$ over trajectories

- A **return** $R(\tau)$ for every trajectory $\tau$

- **Goal:** $\max_{\theta \in \Theta} J(\theta) = \mathbb{E}_{\tau \sim p_\theta} [R(\tau)]$

- Iterative optimization (e.g., gradient ascent)
Policy Optimization

- **Parameter space** $\Theta \subseteq \mathbb{R}^d$

- A parametric **policy** for each $\theta \in \Theta$

- Each inducing a distribution $p_\theta$ over **trajectories**

- A **return** $R(\tau)$ for every trajectory $\tau$

- **Goal:** $\max_{\theta \in \Theta} J(\theta) = \mathbb{E}_{\tau \sim p_\theta} [R(\tau)]$

- Iterative optimization (e.g., gradient ascent)
Policy Optimization

- **Parameter space** $\Theta \subseteq \mathbb{R}^d$

- A parametric policy for each $\theta \in \Theta$

- Each inducing a distribution $p_\theta$ over trajectories

- A return $R(\tau)$ for every trajectory $\tau$

- **Goal:** $\max_{\theta \in \Theta} J(\theta) = \mathbb{E}_{\tau \sim p_\theta} [R(\tau)]$

- Iterative optimization (e.g., gradient ascent)
Policy Optimization

- **Parameter space** $\Theta \subseteq \mathbb{R}^d$

- A parametric **policy** for each $\theta \in \Theta$

- Each inducing a distribution $p_\theta$ over trajectories

- A **return** $R(\tau)$ for every trajectory $\tau$

- **Goal:** $\max_{\theta \in \Theta} J(\theta) = \mathbb{E}_{\tau \sim p_\theta} [R(\tau)]$

- Iterative optimization (e.g., gradient ascent)
Exploration in Policy Optimization

- **Continuous** decision process $\implies$ difficult

- Policy gradient methods tend to be **greedy** (e.g., TRPO [6], PGPE [7])

- Mainly **undirected** (e.g., entropy bonus [2])

- Lack of theoretical guarantees
Exploration in Policy Optimization

- **Continuous** decision process $\implies$ difficult

- Policy gradient methods tend to be **greedy** (e.g., TRPO [6], PGPE [7])

- Mainly **undirected** (e.g., entropy bonus [2])

- **Lack of theoretical guarantees**
Exploration in Policy Optimization

- **Continuous** decision process $\implies$ difficult

- Policy gradient methods tend to be **greedy** (e.g., TRPO [6], PGPE [7])

- Mainly **undirected** (e.g., entropy bonus [2])

- Lack of theoretical guarantees
Exploration in Policy Optimization

- **Continuous** decision process $\implies$ difficult

- Policy gradient methods tend to be **greedy** (e.g., TRPO [6], PGPE [7])

- Mainly **undirected** (e.g., entropy bonus [2])

- **Lack of theoretical guarantees**
Exploration in Policy Optimization

- **Continuous** decision process $\implies$ difficult

- Policy gradient methods tend to be **greedy** (e.g., TRPO [6], PGPE [7])

- Mainly **undirected** (e.g., entropy bonus [2])

- Lack of theoretical guarantees

If only this were a Multi-Armed Bandit...
Exploration in Policy Optimization

- **Continuous** decision process → difficult

- Policy gradient methods tend to be **greedy** (e.g., TRPO [6], PGPE [7])

- Mainly **undirected** (e.g., entropy bonus [2])

- Lack of theoretical guarantees

If only this were a **Correlated Multi-Armed Bandit**...
Policy Optimization as a Correlated MAB

- **Arms**: parameters $\theta$

- **Payoff**: expected return $J(\theta)$

- **Continuous MAB** [3]: we need structure

- **Arm correlation** [5] through trajectory distributions

- **Importance Sampling (IS)**
Policy Optimization as a Correlated MAB

- **Arms**: parameters $\theta$
- **Payoff**: expected return $J(\theta)$
- **Continuous MAB** [3]
- **Arm correlation** [5] through trajectory distributions
- **Importance Sampling (IS)**
Policy Optimization as a Correlated MAB

- **Arms**: parameters $\theta$

- **Payoff**: expected return $J(\theta)$

- **Continuous MAB** [3]

- **Arm correlation** [5] through trajectory distributions

- **Importance Sampling (IS)**
Policy Optimization as a Correlated MAB

- **Arms:** parameters $\theta$

- **Payoff:** expected return $J(\theta)$

- **Continuous MAB** [3]

- **Arm correlation** [5] through trajectory distributions

- **Importance Sampling (IS)**
Policy Optimization as a Correlated MAB

- **Arms**: parameters $\theta$

- **Payoff**: expected return $J(\theta)$

- **Continuous MAB** [3]

- **Arm correlation** [5] through trajectory distributions

- **Importance Sampling (IS)**
A UCB-like index [4]:

\[ B_t(\theta) = \underbrace{\tilde{J}_t(\theta)}_{\text{ESTIMATE}} \]

a truncated multiple importance sampling estimator [8, 1]
A **UCB-like** index [4]:

\[
B_t(\theta) = \mathbb{E} \left[ J_t(\theta) \right] + \sqrt{C \sqrt{ \frac{d_2(p_{\theta} \| \Phi_t) \log \frac{1}{\delta_t}}{t} }}
\]

**ESTIMATE**

A **truncated multiple importance sampling estimator** [8, 1]

**EXPLORATION BONUS:**

distributional distance from previous solutions

\[
T
\]

\[
\Phi_t \quad d_2 \quad p_{\theta}
\]
A UCB-like index [4]:

\[ B_t(\theta) = \tilde{J}_t(\theta) + \sqrt{C \sum_{t} \frac{d_2(p_\theta \| \Phi_t) \log \frac{1}{\delta_t}}{t}} \]

- **ESTIMATE**
  - a truncated multiple importance sampling estimator [8, 1]

- **EXPLORATION BONUS:**
  - distributional distance from previous solutions

Select \( \theta_t = \arg \max_{\theta \in \Theta} B_t(\theta) \)
Sublinear Regret

- \( \text{Regret}(T) = \sum_{t=0}^{T} J(\theta^*) - J(\theta_t) \)

- **Compact**, \( d \)-dimensional parameter space \( \Theta \)

- Under **mild assumptions** on the policy class, with high probability:

\[
\text{Regret}(T) = \tilde{O} \left( \sqrt{dT} \right)
\]
Sublinear Regret

- \(\text{Regret}(T) = \sum_{t=0}^{T} J(\theta^*) - J(\theta_t)\)

- **Compact**, \(d\)-dimensional parameter space \(\Theta\)

- Under mild assumptions on the policy class, with high probability:

\[
\text{Regret}(T) = \tilde{O}\left(\sqrt{dT}\right)
\]
Sublinear Regret

- \( \text{Regret}(T) = \sum_{t=0}^{T} J(\theta^*) - J(\theta_t) \)

- **Compact**, \( d \)-dimensional parameter space \( \Theta \)

- Under **mild assumptions** on the policy class, with high probability:

\[
\text{Regret}(T) = \tilde{O} \left( \sqrt{dT} \right)
\]
Empirical Results

River Swim

<table>
<thead>
<tr>
<th>Episodes</th>
<th>Cumulative Return</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>1,000</td>
<td>1</td>
</tr>
<tr>
<td>2,000</td>
<td>1</td>
</tr>
<tr>
<td>3,000</td>
<td>1</td>
</tr>
<tr>
<td>4,000</td>
<td>1</td>
</tr>
<tr>
<td>5,000</td>
<td>1</td>
</tr>
</tbody>
</table>

- **OPTIMIST**
- **PGPE**

M. Papini Optimistic Policy Optimization via Multiple Importance Sampling

ICML 2019
Empirical Results

**River Swim**

![Graph showing Cumulative Return over Episodes for OPTIMIST and PGPE]

**Caveats**

- Easy implementation only for parameter-based exploration [7]
- Difficult optimization
  - Discretization
- ...

---

M. Papini
Optimistic Policy Optimization via Multiple Importance Sampling

ICML 2019
Thank You for Your Attention!

Poster #103
Code: github.com/WolfLo/optimist
Contact: matteo.papini@polimi.it
Web page: t3p.github.io/icml19
References


