Data-efficient Policy Evaluation Through Behavior Policy Search

Josiah Hanna[1], Phil Thomas[2,3], Peter Stone[1], and Scott Niekum[1]

1. The University of Texas at Austin, {jphanna,pstone,sniekum}@cs.utexas.edu
2. The University of Massachusetts Amherst, pthomas@cs.umass.edu
3. Carnegie Mellon University

Personal Autonomous Robotics Lab

Learning Agents Research Group
The University of Texas at Austin

Abstract

- In reinforcement learning, policy evaluation falls into one of two categories:
  1. On-policy: Collect data with the policy to be evaluated.
  2. Off-policy: Collect data with a **different** policy than the one to be evaluated.
- On-policy is generally assumed to be more data-efficient than off-policy.
- We show that off-policy policy evaluation can be **more data-efficient** than on-policy policy evaluation.
- We introduce a method for learning the data collection policy and demonstrate it leads to **more accurate** value estimates with **less data**.

Background

Environment modeled as Markov Decision Process
In state S_t at time step t:

1. Agent selects action $A_t \sim \pi(\cdot|S_t)$
2. Environment responds with $S_{t+1}, R_t \sim P(\cdot|S_t, A_t)$

The policy and environment determine a distribution over trajectories, $H : S_0, A_0, R_0, S_1, A_1, R_1, ..., S_L, A_L, R_L$

Policy performance measured by its expected sum of rewards:

- $\rho(\pi) = \mathbb{E}\left[\sum_{t=0}^{L}\gamma^t R_t \middle| H \sim \pi\right]$

Policy Evaluation

Given a policy to be evaluated, π_e, estimate $\rho(\pi_e)$ with minimal **mean squared error**.

On-policy Evaluation: Monte Carlo

Given a dataset $\mathcal{D}$ of trajectories where $\forall H_i \in \mathcal{D}$, H_i is sampled from π_e.

$$\rho(\pi_e) \approx \mathrm{MC}(\mathcal{D}) := \frac{1}{|\mathcal{D}|}\sum_{H_i \in \mathcal{D}} g(H_i)$$

Off-policy Evaluation: Importance Sampling

Given a dataset $\mathcal{D}$ of trajectories where $\forall H_i \in \mathcal{D}$, H_i is sampled from a **behavior** policy π_i:

$$\rho(\pi_e) \approx \mathrm{IS}(\mathcal{D}) := \frac{1}{|\mathcal{D}|}\sum_{H_i \in \mathcal{D}} g(H_i)\frac{w_{\pi_e}(H_i)}{w_{\pi_i}(H_i)}.$$

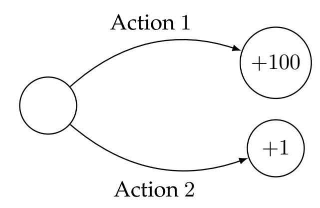
where $w_\pi(H) := \prod_{t=0}^{L}\pi(A_t|S_t)$

Acknowledgments

This work has taken place in the Personal Autonomous Robotics Lab (PeARL) and Learning Agents Research Group (LARG) at the Artificial Intelligence Laboratory, The University of Texas at Austin. PeARL research is supported in part by the National Science Foundation under grant IIS1208497. LARG research is supported in part by NSF (CNS-1330072, CNS-1305287, IIS-1637736, IIS-1651089), ONR (21C184-01), and AFOSR (FA9550-14-1-0087). Josiah Hanna is supported by an NSF Graduate Research Fellowship. Peter Stone serves on the Board of Directors of, Cogitai, Inc. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research.

Policy Evaluation Example

Consider the following simple MDP:



- Policy π_e selects the high-rewarding first action with probability 0.01.
- Monte Carlo evaluation of π_e has **high variance**.
- Importance-sampling with a behavior policy that samples either action with approximately equal probability gives a **low variance** evaluation.

The Optimal Behavior Policy

There exists an optimal behavior policy, π_b for an importance-sampling evaluation of π_e that gives **zero mean squared error** with only a single trajectory:

$$\rho(\pi_e) = g(H)\frac{w_\pi(H)}{w_{\pi_b}(H)} = IS(\{H, \pi_b\})$$

$$w_{\pi_b}(H) = \frac{g(H)}{\rho(\pi_e)}w_\pi(H)$$

Unfortunately, cannot be analytically computed:

- Depends on unknown $\rho(\pi_e)$.
- Depends on unknown reward function.
- Transition function must be deterministic.

Behavior Policy Search

Search for a behavior policy that leads to a low variance importance-sampling policy evaluation.

- Assume behavior policy is parameterized by $\boldsymbol{\theta}$.
- Choose initial behavior policy parameters $\boldsymbol{\theta}_0$.
- Repeat for $i = 0, ..., \infty$:
  1. Sample m trajectories, $H \sim \boldsymbol{\theta}_i$ and add to a data set $\mathcal{D}$.
  2. Estimate $\rho(\pi_e)$ as $\mathrm{IS}(\mathcal{D})$.
  3. Select $\boldsymbol{\theta}_{i+1}$ using trajectories in $\mathcal{D}$.

Behavior Policy Gradient Algorithm

Key idea: adapt the behavior policy with gradient descent on the mean squared error.

$$\boldsymbol{\theta}_0 = \boldsymbol{\theta}_e$$

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \alpha\frac{\partial}{\partial\boldsymbol{\theta}}\mathrm{MSE}(\mathrm{IS}(H_i, \boldsymbol{\theta}))\bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}_i}$$

Theorem 1.

$$\frac{\partial}{\partial\boldsymbol{\theta}}\mathrm{MSE}(\mathrm{IS}(H, \boldsymbol{\theta})) = \mathbf{E}\left[-\mathrm{IS}(H, \boldsymbol{\theta})^2\sum_{t=0}^{L-1}\frac{\partial}{\partial\boldsymbol{\theta}}\log\pi_{\boldsymbol{\theta}}(A_t|S_t)\right]$$

Empirical Results

We compare behavior policy search with **B**ehavior **P**olicy **G**radient (BPG) to Monte Carlo policy evaluation across different policy evaluation problems. For each domain:

- BPG adapts the behavior policy for n iterations and estimates $\rho(\pi_e)$ using importance-sampling with all trajectories.
- Monte Carlo uses an equal number of trajectories to estimate $\rho(\pi_e)$ but always samples actions according to π_e.

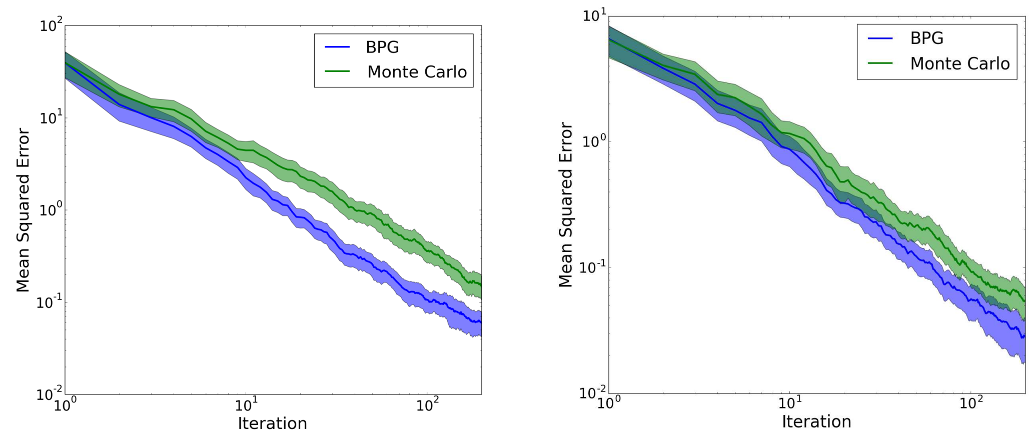


Figure 1: Empirical results on the cartpole swing-up (left) and acrobot (right) domains show that Behavior Policy Search with BPG leads to **more accurate** policy evaluation for any amount of data.

Contributions

1. Demonstrated behavior policy search **lowers the variance** of off-policy policy evaluation over on-policy evaluation!
2. **Behavior Policy Gradient** is an effective behavior policy search method.
3. Additional results, extensions, and analysis in paper!

Open Questions

1. Can behavior policy search lead to lower variance **policy improvement**?
2. Are there better measures of a good behavior policy?