Data-Efficient Policy Evaluation Through Behavior Policy Search

ICML 2017

Po-Chun Huang, Zhi-Xuan Tai

Project Type: Replication
GitHub Link

June 13, 2023

NATIONAL
YANG MING CHIAO TUNG
UNIVERSITY

Table of contents

Table of contents

Introduction

In RL,we usually introduce behavior policy and importance sampling for better policy evaluation, and can be used to produce unbiased estimates with lower mean squared error.

But we also find out that if behavior policy isn't chosen carefully, importance sampling often result in high variance.

So in this paper they try to resolve the behavior policy search problem, which proposed a method called **Behavior Policy Gradient** and demonstrated its effectiveness in lowering the mean squared error of policy performance estimates.

Table of contents

Monte-Carlo Estimates

In this paper, they address the Monte-Carlo Estimator,which with a given π_e, the estimate of $\rho(\pi_e)$ at iteration i is the average return:

$$MC(D_i) := \frac{1}{i+1}\sum_{j=0}^{i}\sum_{t=0}^{L}\gamma^t R_t = \frac{1}{i+1}\sum_{j=0}^{i} g(H_j)$$

This estimator is unbiased and strongly consistent given mild assumptions. However, this method can have high variance.

Importance Sampling

Importance Sampling is a method for reweighting returns from a behavior policy, θ, such that they are unbiased returns from the evaluation policy.

In RL, the re-weighted IS return of a trajectory, H, sampled from π_θ is:

$$IS(H;\theta) := g(H) \prod_{t=0}^{L} \frac{\pi_e(s_t|A_t)}{\pi_\theta(S_t|A_t)}.$$

IS off-policy estimator is the Monte Carlo estimates of $E[IS(H,\theta)|H \sim \pi_\theta]$:

$$IS(D_i) := \frac{1}{i+1} \sum_{j=0}^{i} IS(H_j, \theta_j)$$

Table of contents

The Optimal Behavior Policy

An appropriately selected behavior policy can lower variance to zero.

In general MDPs settings, all returns are positive and the domain is deterministic. if we let $w_\pi(H) := \prod_{t=0}^{L} \pi\left(A_t \mid S_t\right)$

Consider the optimal behavior policy π_b^* for any trajectory, H, we have:

$$\rho\left(\pi_e\right) = \mathrm{IS}\left(H, \pi_b^*\right) = g(H)\frac{w_{\pi_e}(H)}{w_{\pi_b^*}(H)}$$

which can be rewrited as :

$$w_{\pi_b^*}(H) = g(H)\frac{w_{\pi_e}(H)}{\rho\left(\pi_e\right)}$$

The Optimal Behavior Policy

Then, if we can express any evaluation policy as a mixture of these deterministic policies. The optimal behavior policy π_b^* can be expressed similarly and thus the optimal behavior policy exists.

But Unfortunately, the optimal behavior policy depends on the unknown value $\rho(\pi_e)$ as well as the unknown reward function $R := g(H)$

Thus, while there exists an optimal behavior policy for IS, In practice we can't analytically determine π_b^*.

The Optimal Behavior Policy

So in this paper, they propose the behavior policy search (BPS) problem instead to estimate to the optimal behavior policy π_b^*, which is defined by the inputs:

1. An evaluation policy π_e with policy parameters θ_e
2. An off-policy policy evaluation algorithm, $OPE(H;\theta)$, that takes a trajectory, $H \sim \theta$, and returns an estimate of $\rho(\pi_e)$.

To address the problem, this paper propose **B**ehavior **P**olicy **G**radient method, which adapts θ to minimize the MSE Loss between the IS estimate and $\rho(\pi_e)$ with stochastic gradient descent.

Behavior Policy Gradient Theorem

Begins with on-policy estimates and adapts the behavior policy with gradient descent on the MSE with respect to θ. The gradient of the MSE with respect to the policy parameters is given by the following theorem:

Theorem 1.

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

where the expectation is taken over $H \sim \pi_\theta$

Behavior Policy Search Problem

BPG uses stochastic gradient descent and replacing the intractable expectation in Theorem 1 with an unbiased estimate of the true gradient,then we have the algo as below:

Algorithm 1 Behavior Policy Gradient

Input: Evaluation policy parameters, $\boldsymbol{\theta}_e$, batch size k, a step-size for each iteration, α_i, and number of iterations n.

Output: Final behavior policy parameters $\boldsymbol{\theta}_n$ and the IS estimate of $\rho(\pi_e)$ using all sampled trajectories.

$\boldsymbol{\theta}_0 \leftarrow \boldsymbol{\theta}_e$
$\mathcal{D}_0 = \{\}$
for all $i \in 0...n$ **do**
  $\mathcal{B}_i =$ Sample k trajectories $H \sim \pi_{\theta_i}$
  $\mathcal{D}_{i+1} = \mathcal{D}_i \cup \mathcal{B}_i$
  $\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i + \frac{\alpha_i}{k} \sum_{H \in \mathcal{B}} \mathrm{IS}(H, \boldsymbol{\theta})^2 \sum_{t=0}^{L} \frac{\partial}{\partial \boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}_i}(A_t|S_t)$
end for
Return $\boldsymbol{\theta}_n$, $\mathrm{IS}(\mathcal{D}_n)$

Figure: Behavior Policy Gradient algorithm

Connection to REINFORCE

In original, REINFORCE is trying to maximize $\rho(\pi_e)$ with gradient ascent and increase the probability of all actions taken in H with $g(H)$:

$$\frac{\partial}{\partial\theta}\rho(\pi_e) = E[g(H)\sum_{t=0}^{L}\frac{\partial}{\partial\theta}\log\pi_\theta(A_t|S_t)|H\sim\pi_\theta]$$

BPG replace the $g(H)$ to $IS(H,\theta)^2$ instead, and the $IS(H,\theta)^2$ magnitude depends on the below condition:

1. $g(H)^2$ is large
2. H is rare relative to its probability under the evaluation policy.

That is, for a selected trajectory, BPG will enhance the probability if the condition hold.

Table of contents

Experiment Setting- Cartpole

In experiment setting , we apply BPG on two different Tasks of OpenAI Gym Cartpole and Acrobot

Task 1: Cartpole

- random seed : 10
- Learning Rate lr :3e-5
- Discount factor γ : 0.95
- optimizer : Adam (lr)
- Scheduler : StepLR (Step size=100 ,$\gamma = 0.9$)
- Behavior update step n : 64

Experiment Setting- Acrobot

Task 2: Acrobot

- random seed : 10
- Learning Rate lr :3e-5
- Discount factor γ : 0.95
- optimizer : Adam (lr)
- Scheduler : StepLR (Step size=100 ,$\gamma = 0.9$)
- Behavior update step n : 64

Experiment Setting- Alogrithm

Algorithm 1 Replication of BPG Algprithm with REINFORCE

Randomly initialize θ_0
Define Step$_e$ =64, Step$_b$ =64, k=8 for update counting
Discounted rewards $G_t = \sum_{i=t}^{T} \gamma^{i-t} r_i$
for i:= 0,...,T **do**
  $b_i := \theta_i$, $D = \{\}$
  for $0, ..., k$ iterations **do**
    $\mathcal{B} :=$ Sample k trajectories $\sim \pi_{b_i}$
    $\mathcal{D} := \mathcal{D} \cup \mathcal{B}$
    for 0, ... ,Step$_b$ **do**
      $b_i := b_i + \frac{\alpha}{k} \sum_{\mathcal{H} \in \mathcal{B}} \sum_{t=0}^{L} [G_t \frac{\pi_{\theta_i}(a_t|s_t)}{\pi_{b_i}(a_t|s_t)}]^2 \frac{\partial}{\partial b_i} \log \pi_{b_i}(a_t|s_t)$
    end for
  end for
  for 0, ... ,Step$_e$ **do**
    $\theta_i := \theta_i + \frac{\alpha}{k} \sum_{\mathcal{H} \in \mathcal{D}} \sum_{t=0}^{L} [G_t \frac{\pi_{\theta_i}(a_t|s_t)}{\pi_{b_i}(a_t|s_t)}] \frac{\partial}{\partial \theta_i} \log \pi_{\theta_i}(a_t|s_t)$
  end for
  $\theta_{i+1} := \theta_i$
end for

Experiment Setting

Base on algorithm, we update the behavior policy with $Step_b$:

$$b_i := b_i + \frac{\alpha}{k} \sum_{\mathcal{H} \in \mathcal{B}} \sum_{t=0}^{L} [G_t \frac{\pi_{\theta_i}(a_t|s_t)}{\pi_{b_i}(a_t|s_t)}]^2 \frac{\partial}{\partial b_i} \log \pi_{b_i}(a_t|s_t)$$

So in experiment setting we apply two setting for BPG and off-policy-evaluation respectively.

$$Step_b = 64 \qquad \rightarrow \text{BPG} \qquad (1)$$

$$Step_b = 0 \qquad \rightarrow \text{Off-policy REINFORCE} \qquad (2)$$

Finally, BPG will converge to a behavior policy that locally minimizes the variance and ideally converges to the globally optimal behavior policy within the parametrization of π_e

Experiment Result- Cartpole

The result of Cartpole has demonstrate that with BPG the Reward could converge more faster and stable than the Standard Off-policy evaluation.

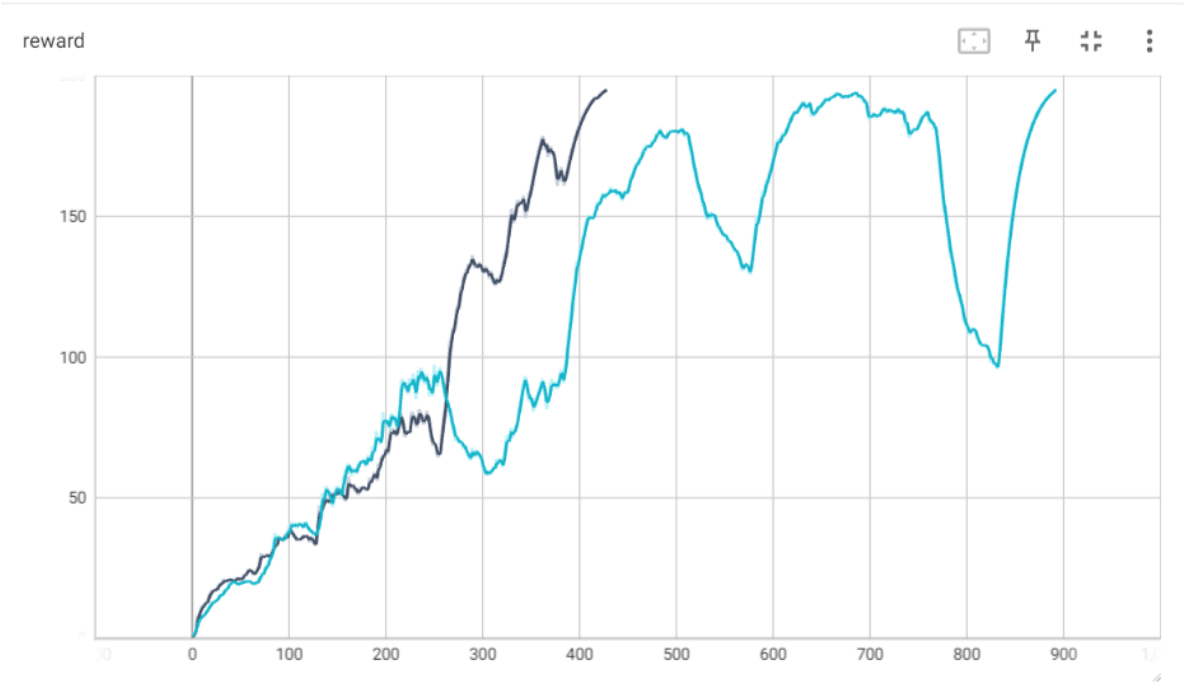

Figure: Cartpole rewards

Experiment Result- Acrobot

In the result, the step of BPG and Standard Off-policy evaluation is almost the same, but the reward of BPG reach to the high point much faster, it means that with BPG, the agent can see higher reward event much more quicker and has better chance to converge.

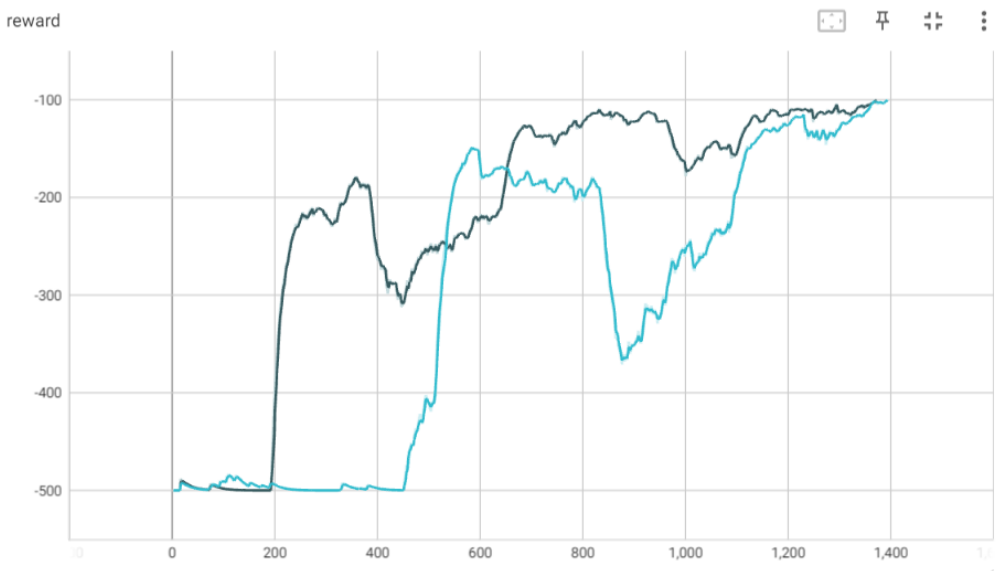

Figure: Acrobot rewards

Table of contents

Conclusion and Insights

- In this paper they introduce a sub-optimal problem Behavior Policy Search Problem, and present a solution Called Behavior Policy Gradient.and has demonstrated that behavior policy search can lower the variance of policy evaluation,
- But in the practice, it may only offer marginal improvement with much more complicated environment.
- Because BPG will increase the probability of exploration to rare events, but if the evaluation policy never see such events, the benefit would be very little.
- BPG also limited to the proper select of the good step size, if chosen poorly may lead to a worse behavior policy and further increase the variance of the gradient estimate.

Table of contents

References

[1] Mehrdad Farajtabar, Yinlam Chow, and Mohammad Ghavamzadeh. *More Robust Doubly Robust Off-policy Evaluation*. 2018. arXiv: 1802.03493 [cs.AI].

[2] Josiah P. Hanna et al. *Data-Efficient Policy Evaluation Through Behavior Policy Search*. 2017. arXiv: 1706.03469 [cs.AI].

[3] Nan Jiang and Lihong Li. *Doubly Robust Off-policy Value Evaluation for Reinforcement Learning*. 2016. arXiv: 1511.03722 [cs.LG].

[4] John Schulman et al. *Proximal Policy Optimization Algorithms*. 2017. arXiv: 1707.06347 [cs.LG].

[5] Zifeng Zhuang et al. *Behavior Proximal Policy Optimization*. 2023. arXiv: 2302.11312 [cs.LG].

[2] [1] [3] [4] [5]