Reinforcement Learning vs Optimal Control on the Mountain Car Problem

Dewan Chowdhury [1], Jacob Mantooth [2], Arjun Sethi-Olowin [3]

[1]Rutgers University, [2] East Central University, [3] Rice University

EMORY

Abstract

In recent years, reinforcement learning (RL) methods have shown success in solving optimal control (OC) problems. RL approaches differ drastically from traditional optimal control methods. Where optimal control methods rely on the ODE models, RL tries to learn the optimal control simply by observations (data). There is tremendous activity in the sciences and engineering utilizing both methods. However, RL and optimal control theory is rarely compared on the same problem. We wish to fill that gap and tries to investigate the trade-off between data-driven RL methods and model-driven optimal control methods on the continuous mountain car problem.

Model

The continuous mountain car problem [1] involves getting an under-powered car out of a sinusoidal valley and has been well-studied with reinforcement learning. It also has a 2-D state space, which allows for easy visualizations. More importantly, while being a RL problem, it can be modeled as a finite-horizon optimal control problem with continuous state and motion.

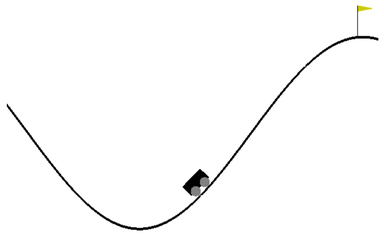

Fig. 1: Continuous mountain car scenario where the car (black) must reach the goal (flag)

In particular, we model the scenario as the following system:

$$\begin{cases} \partial_s \boldsymbol{z}(s) &= \boldsymbol{f}(\boldsymbol{z}(s)) + \mathbf{B}u(s) \quad \text{for } s \in (t, T] \\ \boldsymbol{z}(0) &= z_t, \end{cases} \tag{1}$$

where

$$\boldsymbol{f}(s, \boldsymbol{z}) = \begin{pmatrix} z_2 \\ -\mu\cos(3z_1) \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0 \\ \gamma \end{pmatrix}$$

with the constants $\mu = 0.0025$ and $\gamma = 0.0015$ and $\mathbf{z}(s) = (z_1, z_2)$. Here, the *control* $u : [t, T] \to [-1, 1]$ is the acceleration of the car, $\boldsymbol{z} : [t, T] \mapsto \mathbb{R}^2$ encapsulates the position and velocity of the car, $\boldsymbol{f} : [t, T] \times \mathbb{R}^2 \mapsto \mathbb{R}^2$ describes the evolution of the state, and $\mathbf{B}$ describes the how the control impacts the dynamics. Our goal is to minimize the objective function

$$J_{s,\boldsymbol{z}}[u] = \int_t^T L(s, \boldsymbol{z}, u)\, ds + g(\boldsymbol{z}(T)), \tag{2}$$

where $L(s, \boldsymbol{z}, u) = 0.01\|u(s)\|^2$ penalizes excessive acceleration and $g(z) = \text{relu}(0.45 - z_1)$ gives feedback for not reaching the goal. We also define the *value function* $\Phi : [t, T] \times \mathbb{R}^2 \mapsto \mathbb{R}$ with

$$\Phi(s, z_s) = \min_u J_{s,\boldsymbol{z}}[u] \quad \text{subject to} \quad (1). \tag{3}$$

Local method (LM) with numerical solvers

We first obtain a local solution $\mathbf{u}_h$ to serve as a baseline to determine the optimality of our global solutions. We formulate an optimization problem by first discretizing the control, state and the Lagrangian using a forward Euler scheme $\begin{pmatrix} \mathbf{z}_h^{(i+1)} \\ \ell_h^{(i+1)} \end{pmatrix}$. We now can approximate our objective function (2) as

$$J_{s,\boldsymbol{z}}[u] \approx J_{\mathbf{z}_h}(\mathbf{u}_h) = \ell_N + g(\mathbf{z}_h^{(N)})$$

Then we have the optimization problem

$$\min_{\mathbf{u}_h, \mathbf{z}_h} \ell_N + g(\mathbf{z}_h^{(N)}),$$

which we solve using gradient descent.

Global method with Reinforcement Learning

RL utilizes only observations and rewards to learn the optimal control. We adapt the continuous mountain car problem for RL exploration by making the initial position and action space stochastic. We estimate a stochastic optimal control policy in the form of a normal distribution $\psi(u \mid \mu_u, \sigma) = N(u \mid \mu_u, \sigma)$ via

$$\min_\psi \mathbb{E}_{x\sim\rho}\left(J_{s,\boldsymbol{z}}[\psi]\right) \quad \text{subject to} \quad (1).$$

where $\rho = U(-0.6, -0.4)$. We use the actor-critic architecture and more specifically the TD-advantage algorithm to train our model. Our actor $\pi_\psi^W \approx \psi(u \mid \mu_u, \sigma)$ with weights W is a neural network (NN) that estimates the optimal policy. The critic $V_\psi^\theta \approx \mathbb{E}_{x\sim\rho}\left(J_{s,\boldsymbol{z}}[\psi]\right)$ with weights θ is an NN estimate analogous to our value function (3). More specifically, we use the TD-advantage actor-critic algorithm, where actor and critic train based on the TD-error

$$\delta_{s_k} = c_{s_k} + V_\psi^\theta(s_{k+1}, \mathbf{z}_{s_{k+1}}) - V_\psi^\theta(s_k, \mathbf{z}_{s_k})$$

where c_{s_k} is the cost of following policy π_ψ^W at time s_k.

Global method with Optimal Control

We adapt the method from [2] and approximate the value function Φ using NNs. We first compute the *Hamiltonian*

$$\begin{aligned} H(s, z, p) &= \sup_u -p^\top\left(\boldsymbol{f}(s, \boldsymbol{z}) + \mathbf{B}u\right) - L(s, \boldsymbol{z}, u) \\ &= \sup_u \mathcal{H}(s, \boldsymbol{z}, p, u) \\ &= -p^\top \boldsymbol{f}(s, \boldsymbol{z}, u^*(s, \boldsymbol{z}, p)) \end{aligned}$$

where $u^*(s, \boldsymbol{z}, p) = 50 \cdot p^\top \mathbf{B}$. Note that Φ satisfies the Hamilton-Jacobi-Bellman (HJB) equation:

$$-\partial_s \Phi(s, z) + H(s, z, p) = 0, \qquad \Phi(T, z) = g(z). \tag{4}$$

Then global optimal policy $\psi : [t, T] \times \mathbb{R}^2 \mapsto \mathbb{R}$ can be obtained through Pontryagin's maximum principle (PMP):

$$\psi(s, \boldsymbol{z}) = \arg\max_u \mathcal{H}\left(s, \boldsymbol{z}, \nabla_z \Phi(s, z), u\right). \tag{5}$$

Hence, we need only find Φ to obtain the optimal solution. So we parameterize Φ using a residual neural network and train using the feedback form (4) and (5).

Results

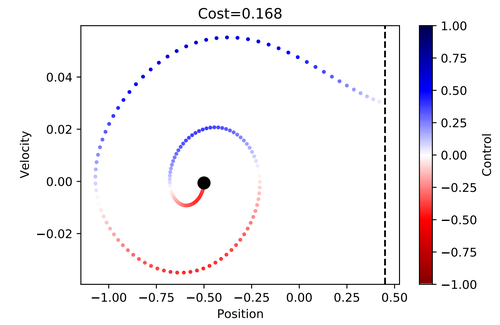

Fig. 2: Local

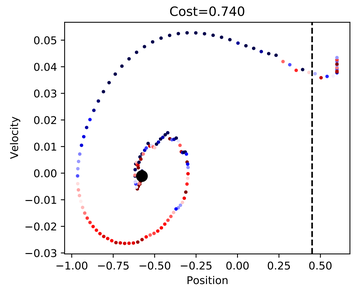

Fig. 3: Reinforcement Learning

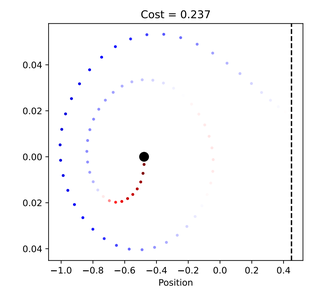

Fig. 4: Optimal Control

Conclusion

We found that there is a substantial trade-off for not using the model in solving the continuous mountain car problem. For one, RL struggles to find an optimal solution when the control problem is finite horizon. In addition, the solutions found are very suboptimal. Small changes in the tradition continuous mountain car problem resulted in substantial differences in results using RL. Thus, we find RL to be a fragile method for optimal control problems. In contrast, the hybrid approach incorporating the model with OC theory resulted in more consistent and optimal solutions.

This work is supported in part by the US NSF awards DMS-2051019.

References

[1] Andrew William Moore. *Efficient Memory-based Learning for Robot Control*. Tech. rep. University of Cambridge, 1990.

[2] Derek Onken et al. "A Neural Network Approach for High-Dimensional Optimal Control". In: *arXiv* (2021). eprint: 2104.03270.