Action Improvement Policy

Shuo

July 14th, 2021

1 Problem Definition

Under-actuated adaptive hands with compliant fingers are appealing due to their ability to passively adapt to objects of uncertain size and shape. They are low cost, have many degrees of freedom and can provide a stable and robust grasp. Controlling hands with learned transition models have been investigated by Avishai, however, with limited performance of reaching some challenging positions, especially in the case of existence of obstacles, due to model inaccuracy. There always exist some errors when we try to execute the actions coming from the planning in dynamics model to the real environment, sometimes even leading to the collision with obstacles. Thus, we want to develop a high-performance yet fast approach of securely moving the hand to the goal position. In this project, we mainly focused on **Tasks of Goal Reaching**.

1.1 "Reacher-v2" on Mujoco

- Relatively simple dynamics model
- Fast simulation (approximately 1000 steps per second)
- Easy planning and/or learning without obstacles

1.2 "Two-Finger Adaptive Hand" on Gazebo Simulator

- Complex dynamics model with limited accuracy
- Slow simulation (approximately 2 steps per second)
- Difficult planning and/or learning with obstacles

Another 2 extra settings may be out of the scope of this project.

1.3 Extra 1: Real "Two-Finger Adaptive Hand"

- More complex and inaccurate dynamics model
- Slow simulation (approximately 2.5 steps per second)
- Without obstacles

1.4 Extra 2: "Acrobot" on Mujoco

- Inaccurate dynamics model
- Somewhat different type of task with goal of reaching a height ≥ 1.0
- Discretized actions (0, -1, 1)

2 Proposed Method

Model free reinforcement learning(MFRL) methods usually take long time and need much data while model based methods are more sample-efficient, however, with limited performance due to the limitation of accuracy of the dynamics model we learned, and due to the incapability of naively executing actions/policy from the model-based planning/learning methods to the real environment. Therefore, we want to propose a hybrid method (model-based method + model-free method).

2.1 Baselines

- A^* planning in dynamics model + naive action rollouts on real environment
- PPO reinforcement learning in dynamics model + naive action rollouts on real environment
- Iterative LQR policy on real environment based on aiming at following the desired trajectory from A^* planning in dynamics model
- Another baseline method might be added in future(before deadline): PPO reinforcement learning in dynamics model + policy rollout on real environment

2.2 Our New Method: Action Improvement Method (AIP)

We first introduce the flow of the algorithm, then the explanation follows.

2.2.1 Flow of AIP

- Input:
  Current state in real environment$s$;
  Reference policy $a_{ref}(s)$ from the model-based method;
  Dynamics model $s'_{ref} = f(s, a)$;
  A heuristic function $\alpha(d_\phi(s, a_{ref}(s))) \in [0, 1]$ for computing the confidence/trust value α of executing $a_{ref}(s)$ under state s, where $d_\phi(s, a)$ is a learnable metric for measuring the discrepancy between executing $a_{ref}(s)$ under s in the real environment and in the learned dynamics model.

- Output: The desired policy $a(s)$ in real environment

1. We initialize all the confidence $\alpha = 1$.

2. We compute $a_{ref}(s)$ and collect data tuples (s,a,s') in real environment using the AIP policy

$$\pi_\theta(a|s) = \mathcal{N}(a|\mu_\theta(s, a_{ref}(s)), \sigma_\theta^2) \tag{1}$$

where

$$\mu_\theta(s, a_{ref}(s)) = \alpha a_{ref}(s) + (1-\alpha) a_\theta^{\text{AIP}}(s) \tag{2}$$

3. We update ϕ of $d_\phi(s, a)$ in a supervised way based on two approximate action-value functions $\tilde{Q}(s, a, s')$ of executing a under s in the real environment and $\tilde{Q}(s, a, s'_{ref})$ in the learned dynamics model. We define the ground truth of $d^{\text{label}}(s, a)$, using the collected data tuples (s,a,s') and the reference data tuples (s,a,s'_{ref}):

$$d^{\text{label}}(s, a) = \text{clip}(\frac{\tilde{Q}(s, a, s') - \tilde{Q}(s, a, s'_{ref})}{|\tilde{Q}(s, a, s'_{ref})|}, -1, 0) \tag{3}$$

where the definition of $\tilde{Q}(s, a, s')$ could depend on tasks. In the tasks of goal reaching with g as goal state, could be simply as follows:

$$\tilde{Q}(s, a, s') = -||s' - g|| - ||a||^2 \tag{4}$$

4. Accordingly, we also acquire the updated trust value α, e.g.:

$$\alpha(d) = 1 - \sqrt{-d} \tag{5}$$

which essentially measures how good/trustworthy it is to execute a under s in the real environment compared to in the dynamics model.

5. We update θ in $a_\theta^{AIP}(s)$, accordingly $\pi_\theta(a|s)$, based on some model-free reinforcement learning method, e.g., TRPO, practically using conjugate gradient method and line search to solve the following equations:

$$\begin{aligned} &\max_\theta \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}, a\sim\pi_{\theta_{\text{old}}}}[\frac{\pi_{\theta_{\text{new}}}(a|s)}{\pi_{\theta_{\text{old}}}(a|s)}] A_{\theta_{\text{old}}}(s, a) \\ &\text{s.t.} \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}}[D_{KL}(\pi_{\theta_{\text{old}}}(\cdot|s)||\pi_\theta(\cdot|s))] \leq \delta \end{aligned} \tag{6}$$

6. We go back to Step 2 and iterate using new ϕ and θ, according new α and π.

2.2.2 Observation

- In equation (2), a_{ref} comes directly from a model based method on dynamics model, while a_θ^{AIP} is a neural network policy to be aggregated to the reference action a_{ref} with a weight of $1-\alpha$. This is also why the algorithm is called "Action Improvement Policy".
- In equation (3), we clip the value for d as our label because we want to keep the whole action a if it performs better in real environment than in dynamics model, which could save much unnecessary exploration at s. Also, there is no meaning to keep the action at all if it underperforms too much in the real environment than in the dynamics model.
- In equation (4), the approximate action-value function $\tilde{Q}(s,a,s')$ is exactly defined as the immediate reward in the case of goal reaching tasks. However, further investigation is necessary for other types of tasks.
- Equation (5) works for both Reacher-v2 and adaptive hand. However, for adaptive hand we need additional metrics to reduce the risk of colliding with an obstacle. We define α' as follows to better measure how good/trustworthy it is to execute a under s in the real environment compared to in the dynamics model:

$$\begin{aligned} \alpha' &= \alpha_1 \times \alpha_2 \\ \alpha_1(d) &= 1 - \sqrt{-d} \\ \alpha_2(l') &= min(l'/L, 1) \end{aligned} \tag{7}$$

  where l' is the minimum distance from state s' to any of the obstacles and the hyperparameter L is the threshold length, above which no effect of distance to the obstacles takes place. Also, α_2 needs to be learned separately or we can learn α' alone combining α_1 and α_2 together. Lastly, we also need to take into account if $a_{ref}(s)$ is trustworthy enough in the case of obstacles since $a_{ref}(s)$ itself may in a danger region of collision. Thus, the final confidence α for adaptive hand in the case of obstacles is computed as follows:

$$\begin{aligned} \alpha &= \alpha' \times \beta \\ \beta(l) &= min(l/L, 1) \end{aligned} \tag{8}$$

  where l is the minimum distance from state s'_{ref} to any of the obstacles and the hyperparameter L is the same threshold length. Other types of definition are also possible. Further investigation might be interesting.

3 Results So Far

3.1 Results except AIP algorithm

Please refer to the attached word-converted pdf.

3.2 AIP for Reacher-v2 on Mujoco

3.2.1 Ablation Study

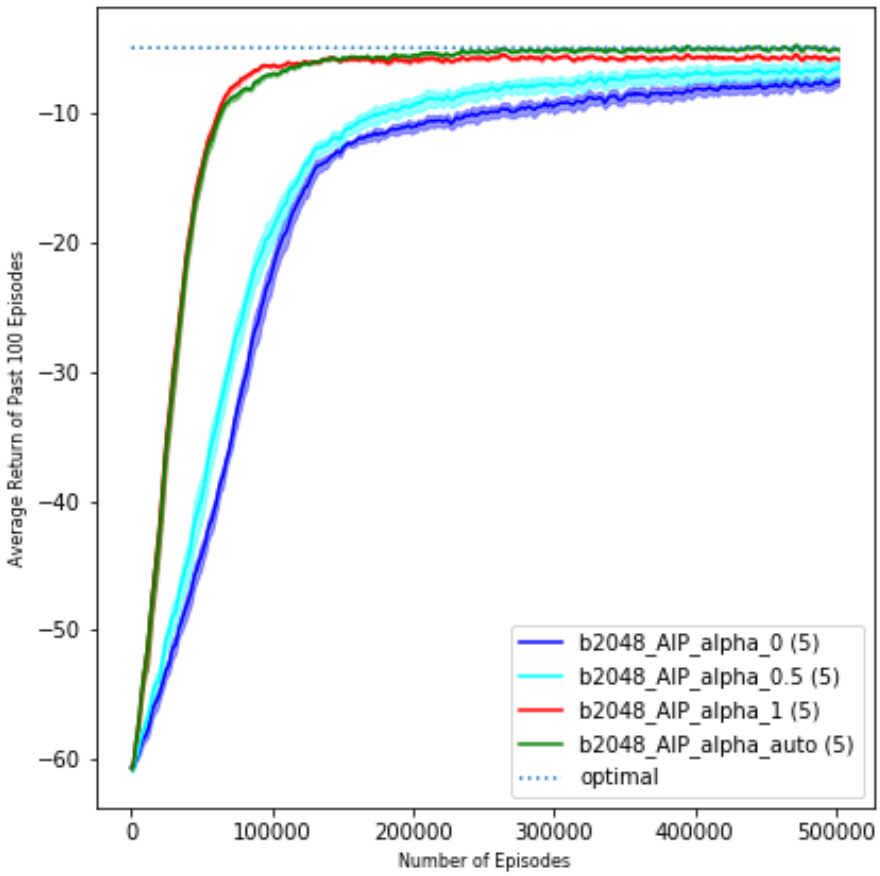

Figure 1: Ablation Study for Reacher-v2 on Mujoco

3.2.2 Trajectories Comparison

3.3 AIP for adaptive hand on Gazebo

To be done.

α	0(model-free)	0.5	1(model-based)	automatic
Final Performance	-7.49	-6.36	-5.74	-5.04
Improvement	0%	15.1%	23.4%	32.7%

Table 1: Ablation Study of Reacher-v2

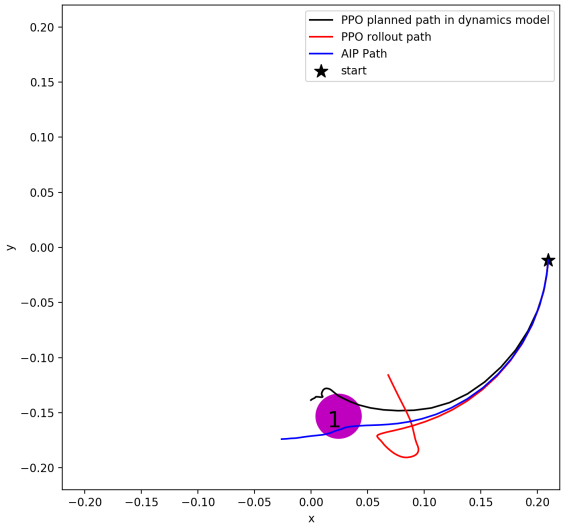

Figure 2: Reacher-v2 Trajectories Comparison: Goal 1

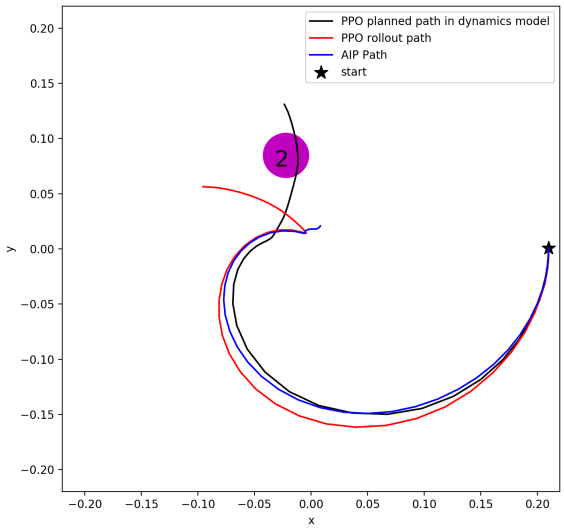

Figure 3: Reacher-v2 Trajectories Comparison: Goal 1

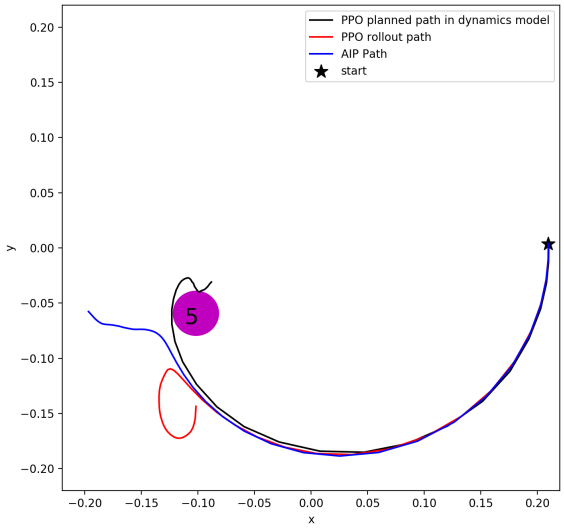

Figure 4: Reacher-v2 Trajectories Comparison: Goal 1

4 Related Works

Hybrid reinforcement learning has been investigated by [3], where imaginary data from dynamics model is incorporated into the training of model-free Q-learning. However, to the opposite our work do not trust the imaginary data since it is imperfect and we try to learn what kind of imaginary model data is worse or better in order to accelerate the training of model-free reinforcement learning. [2] and [1] utilize model-based rollouts to compute targets of value function training, thus accelerating the value function learning. However, again due to the inaccuracy of model dynamics, model-based data can not be fully trusted and naively be used to accelerate the value function estimation for model-free learning, which is exactly focued on in our work by introducing a confidence value.

References

[1] Jacob Buckman, Danijar Hafner, George Tucker, Eugene Brevdo, and Honglak Lee. Sample-efficient reinforcement learning with stochastic ensemble value expansion. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 8234–8244, Red Hook, NY, USA, 2018. Curran Associates Inc.

[2] Vladimir Feinberg, Alvin Wan, Ion Stoica, Michael I Jordan, Joseph E Gonzalez, and Sergey Levine. Model-based value estimation for efficient model-free reinforcement learning. *arXiv preprint arXiv:1803.00101*, 2018.

[3] Shixiang Gu, Timothy Lillicrap, Ilya Sutskever, and Sergey Levine. Continuous deep q-learning with model-based acceleration. In *International conference on machine learning*, pages 2829–2838. PMLR, 2016.