Theory Paper–When to Trust Your Model: Model-Based Policy Optimization

Cheng-Yi XU, 0811510
Department of Computer Science
National Yang Ming Chiao Tung University
cm967413@gmail.com

1 Introduction

1.1 The main research challenges tackled by the paper

Sometimes, it is really hard to collect data from the real-world physical systems, and hence, model-based RL algorithms are appealing due to the idea of producing simulated data by a model learning from the real environment, and their training time is also relatively shorter. However, model accuracy acts as a bottleneck to policy quality, that is, model-based approaches often perform worse than model-free approaches. And this issue is exactly what this paper tries to fix.

1.2 The high-level technical insights into the problem of interest

The reason why model-based RL approaches can't outperform, or even catch up with model-free approaches, is generally because we cannot guarantee policy improvement. So, if we can derive a method that can make sure the policy can always improve monotonically in the simulated environment provided by model, then we can design a model-based algorithm that works effectively in performance. Hence, this paper provides a method called "Model-Based Policy Optimization".

1.3 The main contributions of the paper (compared to the prior works)

When it comes to designing model-based RL algorithms, there are already plenty of researches. For example, PETS (Chua et al., 2018) tries to directly use the model for planning, instead of performing explicit policy learning. STEVE (Buckman et al., 2018), which also uses short-horizon model-based rollouts, but incorporates data from these rollouts into value estimation rather than policy learning. Finally, SLBO (Luo et al., 2019), a model-based algorithm with performance guarantees that performs model rollouts from the initial state distribution. This method is really similar to MBPO (algorithm provided by this paper), but the main difference is that MBPO do not need to perform model rollouts from the initial state distribution, instead it can perform rollouts from any intermediate state distribution. So, explicitly, the main contributions of this paper, provides an model-based algorithm that can perform model rollouts shortly and begins from any intermediate state distribution, and this method is quite effectively, since it can keep the advantage of fast training, and also catch up with the performance of state-of-the-art model-free methods, such as SAC (Haarnoja et al., 2018) and PPO (Schulman et al., 2017).

1.4 Your personal perspective on the proposed method

From my perspective, deriving a well-performed model-based approach is really necessary. Since there are too many tasks that are hard to collect data in the real environments. For example, my bachelor project is about to train an RL agent that can perform anesthesia on different patients. But it is really an arduous process to collect real patient data from operations. Hence, if we can apply a model-based approach to this scenario, we can save a lot of efforts of collecting real patient data. Besides, intuitively, I think it is possible to find an inequality that guarantees monotonic policy improvement in model-based method by solving some optimization problem, which is actually similar to what we do in model-free approaches.

2 Problem Formulation

2.1 Preliminaries and notations

Just like other RL researches do, we consider a Markov decision process(MDP), defined by the tuple $(S, A, p, r, \gamma, \rho_0)$, S and A are the state and action spaces respectively,and the discount factor $\gamma \in (0, 1)$. The transition distribution is denoted as $p(s'|s, a)$, and the initial state distribution is $\rho_0(s)$, and the reward function is $r(s, a)$. The goal of RL is to find the optimal policy π^* that is in form of

$$\pi^* = \arg\max_{\pi} \eta[\pi] = \arg\max_{\pi} E_{\pi}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)].$$

where the expected sum of discounted rewards is denoted by η. Typically, the dynamics $p(s'|s.a)$ are assumed to be unknown. And we aim to construct a model of the transition distribution, $p_\theta(s'|s, a)$, using data collected from the real environments of MDP.

2.2 The optimization problem of interest

So now, we try to achieve monotonic model-based improvement by introducing the following optimization problem:

$$\eta[\pi] \geq \hat{\eta}[\pi] - C$$

where $\eta[\pi]$ denotes the returns of the policy in the true MDP, and $\hat{\eta}[\pi]$ denotes the returns of the policy under our model. And C is the gap between true returns and model returns, where it can be expressed in terms of two error quantities of the model: Firstly, generalization error due to sampling, which can be denoted as $\epsilon_m = max_t E_{s\sim\pi_{D,t}}[D_{TV}(p(s', r|s, a)||p_\theta(s', r|s, a))]$, where π_D is the data-collecting policy from the real environment, and D_{TV} denotes total-variation distance. Secondly, the other error is the distribution shift error, which is denoted as $\epsilon_\pi \geq max_s D_{TV}(\pi||\pi_D)$.

2.3 The technical assumptions

As making my all-out effort to try to fully understand this paper, i only see two specific assumptions: the MDP process assumption and the assumption mentioned in the appendices such that $\epsilon_\pi \geq max_s D_{TV}(\pi||\pi_D)$. Besides, they have emphasized that they do not make any assumptions about the generalization capacity or smoothness properties of the model.

3 Theoretical Analysis

So, in section 2.2, we mentioned that C comes from two error quantities: generalization error by ϵ_m and distribution shift error by ϵ_π. With these two sources of error controlled, we can now present our bound:

Theorem 4.1. *Let the expected TV-distance between two transition distributions be bounded at each timestep by ϵ_m and the policy divergence be bounded by ϵ_π. Then the true returns and model returns of the policy are bounded as:*

$$\eta[\pi] \geq \hat{\eta}[\pi] - [\frac{2\gamma r_{max}(\epsilon_m + 2\epsilon_\pi)}{(1-\gamma)^2} + \frac{4r_{max}\epsilon_\pi}{(1-\gamma)}]$$

This bound implies that as long as we improve the returns under the model $\hat{\eta}[\pi]$ by more than $C(\epsilon_m, \epsilon_\pi)$, we can guarantee improvement under the true returns.

Although Theorem 4.1 gives us a pretty relationship between model returns and true returns, but if we meet the cases that the model error ϵ_m is high, there may not exist such a policy that $\hat{\eta}[\pi] - \eta[\pi] > C(\epsilon_m, \epsilon_\pi)$, implying that improvement is not guaranteed.

Hence, in order to allow for dynamic adjustment between model-based and model-free rollouts, here we introduce the idea of **a branched rollout**, where we begin a rollout from a state under the previous policy's state distribution $d_{\pi D}(s)$ and run k steps according to π under the learned model p_θ. Therefore, we reach

Theorem 4.2. *Given returns $\eta^{branch}[\pi]$ from the k-branched rollout method,*

$$\eta[\pi] \geq \eta^{branch}[\pi] - 2r_{max}[\frac{\gamma^{k+1}\epsilon_\pi}{(1-\gamma)^2} + \frac{\gamma^k + 2}{(1-\gamma)}\epsilon_\pi + \frac{k}{1-\gamma}(\epsilon_m + 2\epsilon_\pi)].$$

Finally, in practical model generalization, if we can approximate the model error on the distribution of the current policy π, which we denote as $\epsilon_{m'}$, we can use this directly. Here we can approximate $\epsilon_{m'}$ with a linear function of the policy divergence:

$$\hat{\epsilon}_{m'}(\epsilon_\pi) \approx \epsilon_m + \epsilon_\pi \frac{d\epsilon_{m'}}{d\epsilon_\pi}$$

where $\frac{d\epsilon_{m'}}{d\epsilon_\pi}$ can be empirically estimated. Therefore, we can modify Theorem 4.2 and reach:

Theorem 4.3. *Under the k-branched rollout method, using model error under the updated policy* $\epsilon_{m'} = max_t E_{s\sim\pi_{D,t}}[D_{TV}(p(s'|s,a)||p_\theta(s'|s,a))]$, we have

$$\eta[\pi] \geq \eta^{branch}[\pi] - 2r_{max}[\frac{\gamma^{k+1}\epsilon_\pi}{(1-\gamma)^2} + \frac{\gamma^k + 2}{(1-\gamma)}\epsilon_\pi + \frac{k}{1-\gamma}(\epsilon_{m'})].$$

From the above three theorems, it is enough for us to design brand new model-based algorithm **MBPO**

```
Algorithm 2 Model-Based Policy Optimization with Deep Reinforcement Learning
Initialize policy $\pi_\phi$, predictive model $p_\theta$, environment dataset $\mathcal{D}_{env}$, model dataset $\mathcal{D}_{model}$
for $N$ epochs do
    Train model $p_\theta$ on $\mathcal{D}_{env}$ via maximum likelihood
    for $E$ steps do
        Take action in environment according to $\pi_\phi$; add to $\mathcal{D}_{env}$
        for $M$ model rollouts do
            Sample $s_t$ uniformly from $\mathcal{D}_{env}$
            Perform $k$-step model rollout starting from $s_t$ using policy $\pi_\phi$; add to $\mathcal{D}_{model}$
        for $G$ gradient updates do
            Update policy parameters on model data: $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi, \mathcal{D}_{model})$
```

Both empirically and theoretically, the smaller k(even $k = 1$), the better policy quality we get when applying this algorithm.

When it comes to the errors I found in this paper, I saw a typesetting error. Specifically, they put Figure 3 before their description of that experiment settings, so it is hard to understand Figure 3 at first glance, since we don't even know what is going to be talked about in Figure 3.

4 Conclusion

4.1 The potential future research directions

In this paper, the authors provided us both theoretical and empirical results proving that model-based approaches have the ability to catch up with the performance of model-free approaches. However, we know that training a model that simulates the real environment well is not easy, and this paper lacks about how to train the models. So, in my opinion, the potential future research directions could be studying about deriving a general idea about how to trade off between exploration and exploitation, so that the behavior policy for collecting data can collect more diverse data, and this can really help training a great environment model!

4.2 Any technical limitations

The authors said that empirically, the model trained by MBPO would underestimate the returns, however, they didn't emphasize this problem and didn't provide any idea about how to fix it.

5 References

[1] Chua, K., Calandra, R., McAllister, R., and Levine, S. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In Advances in Neural Information Processing Systems. 2018.

[2] Buckman, J., Hafner, D., Tucker, G., Brevdo, E., and Lee, H. Sample-efficient reinforcement learning with stochastic ensemble value expansion. In Advances in Neural Information Processing Systems, 2018.

[3] Luo, Y., Xu, H., Li, Y., Tian, Y., Darrell, T., and Ma, T. Algorithmic framework for model-based deep reinforcement learning with theoretical guarantees. In International Conference on Learning Representations, 2019.

[4] Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In International Conference on Machine Learning, 2018.

[5] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.