SimuDICE: Offline Policy Optimization Through Iterative World Model Updates and DICE Estimation

Cătălin-Emanuel Brița (C.E.Brita@student.tudelft.nl)

Supervisors:
Frans Oliehoek,
Stephan Bongers

In short

In **offline reinforcement learning**, some of the main problems are limited sample sizes and mismatched state-action distributions between the **target** and **behavioral policies** that collected the data. SimuDICE tackles these issues by using **Dual stationary DIstribution Correction (DICE)** to improve the **sampling of simulated experiences**. It **iteratively updates** the DICE estimations and the world model, aligning the model's *training objective* with its *usage objective*. This approach reduces both the need for pre-collected data and the number of simulated experiences, achieving results comparable to other algorithms with greater robustness to varying data quality.

Background

In Reinforcement Learning, an agent uses a policy to interact with the environment. Figure 1 shows this process simplified. The goal of the agent is to maximize the cumulative expected return.

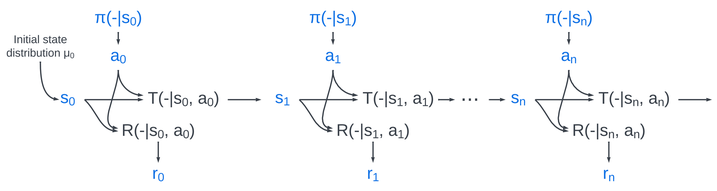

Figure 1. The policy starts at $s_0 \sim \mu_0$ and samples an action $a_t \sim \pi(s_t)$ at each step t from the policy. This action is applied to the environment, resulting in a reward $r_t \sim R(s_t, a_t)$ and the environment transitions to a new state $s_{t+1} \sim T(s_t, a_t)$.

This work lies within the context of **offline reinforcement learning**, where the main problem is addressing the limited experiences represented as $\mathcal{D} = \left\{\left(s_0^{(i)}, s^{(i)}, a^{(i)}, r^{(i)}, s'^{(i)}\right)\right\}_{i=1}^{N}$ and the use of different policies to collect them.

Planning: Planning is the process by which an agent uses a trained world model to simulate the environment, allowing it to predict possible future states without direct environment interactions. The primary goal of planning is to improve the performance of the agent's policy, especially in limited data scenarios. In this work we adapt the Dyna-Q [1] framework to the offline setting.

DualDICE estimation: DualDICE [2] achieved impressive results for DICE estimation by reducing the problem to density estimation and using a change-of-variable technique. Eq. (1) shows how policy evaluation is transformed into density ratio estimation, while Eq. (2) the weights estimated by the algorithm.

$$\rho(\pi) = \mathbb{E}_{(s,a)\sim d^{\pi}}[r(s,a)] = \mathbb{E}_{(s,a)\sim d^{\mathcal{D}}}\left[\frac{d^{\pi}(s,a)}{d^{\mathcal{D}}(s,a)} r(s,a)\right] \quad (1)$$

$$w_{\pi/\mathcal{D}}(s,a) := \frac{d^{\pi}(s,a)}{d^{\mathcal{D}}(s,a)} \quad (2)$$

SimuDICE

SimuDICE uses a dynamic interplay between a learned world model and DICE estimation to **generate higher-quality synthetic experiences**, which in turn are used to enhance policy learning. We align the generation of synthetic experiences with the policy optimization objective by continuously updating both the world model and the DICE estimation. The code is available on GitHub.

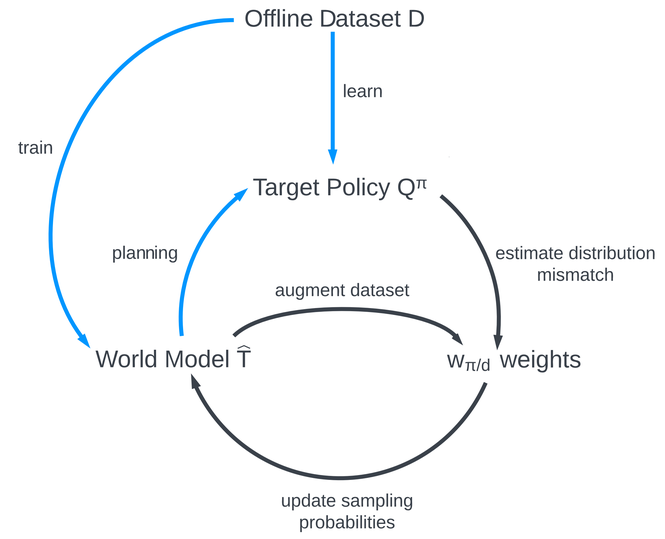

Figure 2. The components of SimuDICE and their interactions. Transitions adapted from Dyna-Q [1] are in blue, while those unique to SimuDICE are depicted in black.

Figure 2 shows how we have **extended** the Dyna-Q [1] framework to iterate until convergence. Initially, offline data is utilized to pre-train a one-step forward dynamic model of the environment and to learn an initial target policy. This policy is **iteratively improved** by sampling experiences that are likely to be encountered by the *target policy*, as determined by the $w_{\pi/\mathcal{D}}$ weights, and the world model's confidence in sampling those experiences. The $w_{\pi/\mathcal{D}}$ estimates are improved using the synthetically generated experiences.

Distribution mismatch estimation is achieved through DICE estimation, for which we reimplemented certain components of DualDICE [2].

Sampling probability updates: are done by the normalized sum of the model confidence of a prediction $\mathcal{C}(s,a)$ and the λ regularized softmax of the $w_{\pi/\mathcal{D}}$ distribution mismatch weights, as in Eq. (3), and normalized as in Eq. (4).

$$\tilde{\mathcal{P}}(s,a) = \mathcal{C}(s,a) + \frac{e^{w_{\pi/\mathcal{D}}(s,a)\cdot\lambda}}{\sum_{(s',a')} e^{w_{\pi/\mathcal{D}}(s',a')\cdot\lambda}}/\lambda \quad (3)$$

$$\mathcal{P}(s,a) = \frac{\tilde{\mathcal{P}}(s,a)}{\sum_{(s',a')} \tilde{\mathcal{P}}(s',a')} \quad (4)$$

Experiments

We compare SimuDICE with two other algorithms: Implicit Q-learning [3] and offline adapted Dyna-Q [1]. The Implicit Q-learning algorithm is part of SimuDICE that learns the target policy from the offline data, without further improvement. The offline adapted Dyna-Q (part of SimuDICE) improves the policy in one iteration using equal probabilities for each state-action pair.

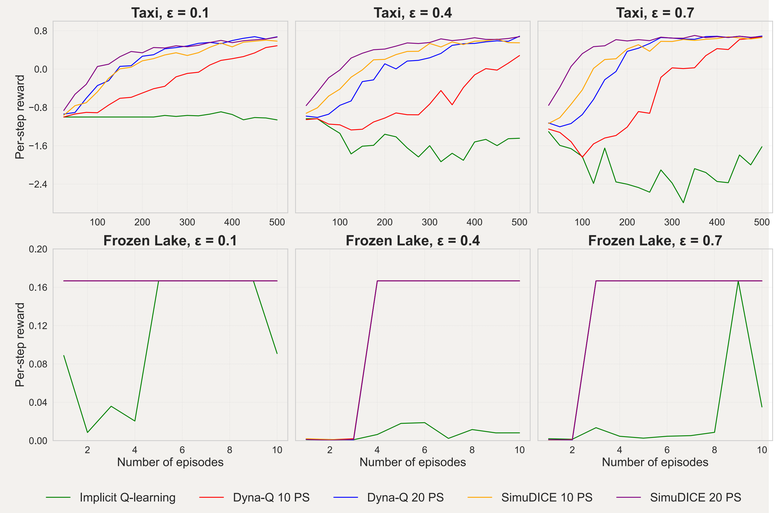

Figure 3. Comparison of algorithm performance in two discrete environments, with data collected under different ϵ-greedy policies. PS represents the planning steps.

Takeaways

Our SimuDICE algorithm achieved comparable results to similar offline RL algorithms with both **less data** and **less simulated experiences**. It improves sample efficiency and reduces distribution mismatch by combining DICE estimation with world models. However, **testing was limited** to few, simple environments. **Future work** should explore more complex settings, sophisticated world models, stable DICE estimation, and improved policies.

References

[1] R. S. Sutton, "Dyna, an integrated architecture for learning, planning, and reacting," *ACM SIGART Bulletin*, vol. 2, no. 4, pp. 160–163, 1991.

[2] O. e. a. Nachum, "DualDICE: Behavior-agnostic estimation of discounted stationary distribution corrections," in *Advances in Neural Information Processing Systems*, 2019, pp. 2315–2325.

[3] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit q-learning," *arXiv preprint arXiv:2110.06169*, 2021.