Offline Reinforcement Learning With Implicit Q-Learning

Author Yu-Chi Chen
Department of Computer Science
National Yang Ming Chiao Tung University
baumin12.cs10@nycu.edu.tw

1 Introduction

In most of current offline Reinforce learning methods, they are suffer from the problem of OOD (out of distribution). There are many other paper try to overcome the problem by constraining the policy to limit or by regularizing the learned value function. This paper first introduces concept of Multi-step DP and Single-step DP. Most of offline RL algorithm use the method they mention before called Multi-step DP because they use dynamic programming in iteration. Theoretically, if it can provide most of environment data, we can learn an optimal policy. Compare to Multi-step DP, Single-step DP approximate value or Q function and avoid to query unseen action. The auther propose the method that learn an optimal policy with in-sample learning without any querying the value of any unseen actions. The main contribution is implicit Q-Learning, a new offline RL algorithm that being able to perform multi-step dynamic programming updates and not suffer from OOD problem.

2 Problem Formulation

They first formulate the problem in the context of a MDP where S is a state space, A is an action space, $p_0(s)$ is a distribution of initial states $p(s'|s,a)$ is the environment dynamics, $r(s,a)$ is reward function and γ is a discount factor. Following the policy $\pi(a|s)$, the goal is to maximaize the cumulative discounted returns:

$$\pi^* = \arg\max_{\pi} \mathrm{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 \sim p_0(\cdot), a_t \sim \pi(\cdot|s_t), s_{t+1} \sim p(\cdot|s_t, a_t)\right]$$

Similar to most of recent offline RL method, they build on approximate dynamic programming method that minimize TD error.

$$L_{TD}(\theta) = \mathrm{E}_{(s,a,s')\sim D}\left[(r(s,a) + \gamma \max_{a'} Q_{\hat{\theta}}(s', a') - Q_{(\theta)}(s,a))^2\right]$$

Where D is the dataset. $Q_{(\theta)}(s,a)$ is parameterized Q-function. $Q_{\hat{\theta}}(s', a')$ is a target network. Base on the TD error above, they want to avoid querying out-of-sample action in TD error. They consider fitting Q evaluation with SARSA-style objective to avoid state of overestimation of action which is out-of-sample and accumulate error to crash the learned policy. They first refer to prior work to construct a term of loss that never queries value of out-of-sample action as below.

$$L(\theta) = \mathrm{E}_{(s,a,s',a')\sim D}\left[(r(s,a) + \gamma Q_{\hat{\theta}}(s', a') - Q_{(\theta)}(s,a))^2\right]$$

If we assume unlimited capacity and no sampling error, the optimal parameter should satisfy:

$$Q_{\theta^*}(s,a) \approx r(s,a) + \gamma \mathrm{E}_{s'\sim p(\cdot|s,a) a' \sim \pi_\beta(\cdot|s)}\left[Q_{\hat{\theta}}(s', a')\right]$$

Base on prior work, they know this could work well on simple task, but it performs poorly on more complex tasks. They want to retain the benefit of SARSA-like objective, they reconstruct the objective aim to learned like below:

$$L(\theta) = \mathrm{E}_{(s,a,s')\sim D}\left[(r(s,a) + \gamma \max_{a' \in A, s.t. \pi_\beta(a'|s') > 0} Q_{\hat{\theta}}(s', a') - Q_{(\theta)}(s,a))^2\right]$$

First, they use Expectile Regression to construct loss function.

$$L(\theta) = \mathrm{E}_{(s,a,s',a')\sim D}\left[L_2^{\tau}(r(s,a) + \gamma Q_{\hat{\theta}}(s',a') - Q_{(\theta)}(s,a))\right]$$

Then the author reconstruct the value and Q-value function:

$$L_V(\psi) = \mathrm{E}_{(s,a)\sim D}\left[L_2^{\tau}(Q_{\hat{\theta}}(s,a) - V_{\psi}(s))\right] \tag{1}$$

$$L_Q(\theta) = \mathrm{E}_{(s,a,s')\sim D}\left[(r(s,a) + \gamma V_{\psi}(s') - Q_{\theta}(s,a))\right] \tag{2}$$

Finally, they use the method of policy extraction proposed in AWR:

$$L_{\pi}(\phi) = \mathrm{E}_{(s,a)\sim D}\left[exp(\beta(Q_{\hat{\theta}}(s,a) - V_{\psi}(s)))log\pi_{\phi}(a|s)\right] \tag{3}$$

Below is the algorithm for Implicit Q-learning:

```
$Initialize \quad parameter \quad \psi, \theta, \hat{\theta}, \phi.$
$TD \quad learning \quad (IQL):$
$for \quad each \quad gradient \quad step \quad do$
  $\psi \leftarrow \psi - \lambda_V \nabla_{\psi} L_V(\psi) \Rightarrow consider \ eq(1)$
  $\theta \leftarrow \theta - \lambda_Q \nabla_{\theta} L_Q(\theta) \Rightarrow consider \ eq(2)$
  $\hat{\theta} \leftarrow (1-\alpha)\hat{\theta} + \alpha\theta$
$end \quad for$
$Policy \quad extraction \quad (AWR):$
$for \quad each \quad gradient \quad step \quad do$
  $\phi \leftarrow \phi - \lambda_{\pi} \nabla_{\phi} L_{\pi}(\phi) \Rightarrow consider \ eq(3)$
$end \quad for$
```

3 Theoretical Analysis

First, prove a simple lemma that will be used.

Lemma 1 *Let X be a real-valued random variable with a bounded support and supremum of the support is x^*. Then,*

$$lim_{\tau \leftarrow 1} m_{\tau} = x^*$$

In the following theorems, they will show that under certain assumption, the method indeed approximates the optimal state-action value Q and performs multi-step dynamical programming. In the beginning, define $V_{\tau}(s)$ and $Q_{\tau}(s,a)$ which correspond to optimal solution of Eq1 and Eq2.

$$V_{\tau}(s) = \mathrm{E}^{\tau}_{a\sim\mu(\cdot|s)}\left[Q_{\tau}(s,a)\right]$$

$$Q_{\tau}(s,a) = r(s,a) + \gamma \mathrm{E}_{s'\sim p(\cdot|s,a)}\left[V_{\tau}(s')\right]$$

And then prove a technical lemma relating different expectiles of Q-function.

Lemma 2 *For all s, τ_1 and τ_2 such that τ_1 < τ_2 we get*

$$V_{\tau 1}(s) \leq V_{\tau 2}(s)$$

Corollary 1 *For any τ and s we have*

$$V_{\tau}(s) \leq \max_{a \in A s.t. \pi_{\beta}(a|s)>0} Q^*(s,a)$$

Where $V_{\tau}(s)$ is define as above and $Q^*(s,a)$ is an optimal state-action value function constrained to the dataset and define as

$$Q^*(s,a) = r(s,a) + \gamma E_{s'\sim p(\cdot|s,a)}\left[\max_{a' \in A s.t. \pi_{\beta}(a'|s')>0} Q^*(s',a')\right]$$

Proof 1 *The proof follows the policy improvement. We can rewrite $V_{\tau 1}(s)$ as*

$$\begin{aligned}
V_{\tau 1}(s) &= \mathrm{E}^{\tau 1}_{a\sim\mu(\cdot|s)}\left[r(s,a)+\gamma\mathrm{E}_{s'\sim p(\cdot|s,a)}\left[V_{\tau 1}(s')\right]\right] \\
&\leq \mathrm{E}^{\tau 2}_{a\sim\mu(\cdot|s)}\left[r(s,a)+\gamma\mathrm{E}_{s'\sim p(\cdot|s,a)}\left[V_{\tau 1}(s')\right]\right] \\
&= \mathrm{E}^{\tau 2}_{a\sim\mu(\cdot|s)}\left[r(s,a)+\gamma\mathrm{E}_{s'\sim p(\cdot|s,a)}\mathrm{E}^{\tau 1}_{a'\sim\mu(\cdot|s')}\left[r(s',a')+\gamma\mathrm{E}_{s''\sim p(\cdot|s',a')}\left[V_{\tau 1}(s'')\right]\right]\right] \\
&\leq \mathrm{E}^{\tau 2}_{a\sim\mu(\cdot|s)}\left[r(s,a)+\gamma\mathrm{E}_{s'\sim p(\cdot|s,a)}\mathrm{E}^{\tau 2}_{a'\sim\mu(\cdot|s')}\left[r(s',a')+\gamma\mathrm{E}_{s''\sim p(\cdot|s',a')}\left[V_{\tau 1}(s'')\right]\right]\right] \\
&= \mathrm{E}^{\tau 2}_{a\sim\mu(\cdot|s)}\left[r(s,a)+\gamma\mathrm{E}_{s'\sim p(\cdot|s,a)}\mathrm{E}^{\tau 2}_{a'\sim\mu(\cdot|s')}\left[r(s',a')+\gamma\mathrm{E}_{s''\sim p(\cdot|s',a')}\mathrm{E}^{\tau 1}_{a''\sim\mu(\cdot|s'')}\left[r(s'',a'')+\cdots\right]\right]\right] \\
&\;\cdot \\
&\;\cdot \\
&\;\cdot \\
&\leq V_{\tau 2}(s)
\end{aligned}$$

Finally, they derive the main result regarding the optimality of the method.

Theorem 1

$$lim_{\tau\leftarrow 1}V_{\tau}(s) = \max_{a\in A s.t.\pi_{\beta}(a|s)>0} Q^{*}(s,a)$$

Proof 2 *Combining Lemma 1 and Corollary 1*

As a result, for a larger $\tau < 1$, we get a better approximation of the maximum. They treat τ as a hyperparameter and it make the learning specturm of methods between SARSA($\tau = 0.5$) and Q-Learning ($\tau \rightarrow 1$)

4 Conclusion

In this report, I simple introduce the idea of implicit Q-Learning and the algorithm. In the experiment, it perform excellent over all of the tasks in D4RL benchmark.Besides, the algorithm is computationally efficient. It combines both of the feature of multi-step and single-step dynamic programming. Something interesting is that still some of other RL algorithm could perform better than IQL in some dataset. Perhaps it is the problem of collected data or still some conditions they didn't notice. Thus, it still has some probability to improve in the future.