Iteratively Refined Behavior Regularization for Offline Reinforcement Learning

Yi Ma[1,2], Jianye Hao[3,4*], Xiaohan Hu[3], Yan Zheng[3*], Chenjun Xiao[5]

1) School of Computer and Information Technology, Shanxi University 3) College of Intelligence and Computing, Tianjin University

2) Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education

4) Noah's Ark Lab, Huawei 5) The Chinese University of Hongkong, Shenzhen. * indicates corresponding authors.

Abstract

One of the fundamental challenges for offline reinforcement learning (RL) is ensuring robustness to data distribution. Whether the data originates from a near-optimal policy or not, we anticipate that an algorithm should demonstrate its ability to learn an effective control policy that seamlessly aligns with the inherent distribution of offline data. Unfortunately, behavior regularization, a simple yet effective offline RL algorithm, tends to struggle in this regard. In this paper, we propose a new algorithm that substantially enhances behavior-regularization based on conservative policy iteration. Our key observation is that by iteratively refining the reference policy used for behavior regularization, conservative policy update guarantees gradually improvement, while also implicitly avoiding querying out-of-sample actions to prevent catastrophic learning failures. We prove that in the tabular setting this algorithm is capable of learning the optimal policy covered by the offline dataset, commonly referred to as the in-sample optimal policy. We then explore several implementation details of the algorithm when function approximations are applied. The resulting algorithm is easy to implement, requiring only a few lines of code modification to existing methods. Experimental results on the D4RL benchmark indicate that our method outperforms previous state-of-the-art baselines in most tasks, clearly demonstrate its superiority over behavior regularization.

Conservative Policy Iteration (CPI)

Definition of CPI

Let $\tau > 0$, the conservative policy optimization (CPO) update the policy for any $s \in \mathcal{S}$ with the following policy optimization problem

$$\max_{\pi} \mathbb{E}_{a\sim\pi}[Q^{\pi}(s,a)] - \tau D_{\mathrm{KL}}(\pi(s) \parallel \pi(s)),$$

which implicitly guarantees policy improvement constrained on the support of the reference policy π. By extending this key observation in an iteratively manner, we obtain the following Conservative Policy Iteration (CPI) algorithm for offline RL. It starts with the behavior policy $\pi_0 = \pi_{\mathcal{D}}$. Then in each iteration $t = 0,1,2,\ldots$, the following computations are done:

- Policy evaluation: compute Q^{π_t} and;
- Policy improvement: $\forall s \in \mathcal{S}, \pi_{t+1} = \mathrm{argmax}_{\pi} \mathbb{E}_{a\sim\pi}[Q^{\pi_t}(s,a)] - \tau D_{\mathrm{KL}}(\pi \parallel \pi_t)$.

In this approach, the algorithm commences with the behavior policy and proceeds to iteratively refine the reference policy used for behavior regularization. Thanks to the conservative policy update, CPI ensures policy improvement while mitigating the risk of querying any OOD actions that could potentially introduce instability to the learning process.

Convergence of CPI

We consider the tabular setting with finite state and action space. In this setting, CPI converges to the optimal policy that are well-covered by the dataset, commonly referred to as the in-sample optimal policy.

Theorem 1. *We consider tabular MDPs with finite $\mathcal{S}$ and $\mathcal{A}$. Let π_t be the produced policy of CPI at iteration t. There exists a parameter $\tau > 0$ such that for any $s \in \mathcal{S}$*

$$V^{*}_{\pi_{\mathcal{D}}}(s) - V^{\pi_t}(s) \leq \frac{1}{(1-\gamma)^2}\sqrt{\frac{2\log|\mathcal{A}|}{t}}$$

Practical Implementations

CPI consists of two steps at each iteration. The first step involves policy evaluation, which is carried out using standard TD learning: we learn the critic by

$$\min_{\theta} \mathbb{E}_{s,a,r,s'\sim\mathcal{D},a'\sim\pi_{\omega}(s')}\left[\frac{1}{2}\left(r + \gamma Q_{\theta}(s',a') - Q_{\theta}(s,a)\right)^2\right]$$

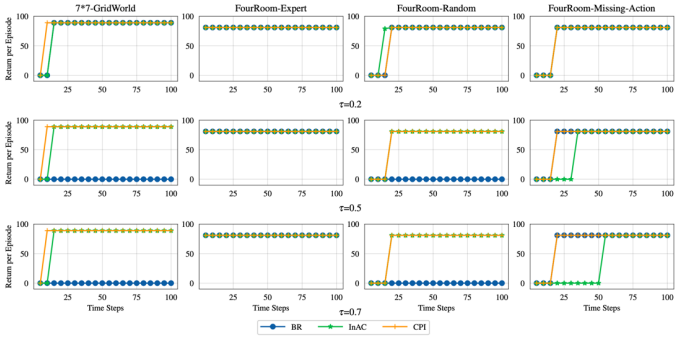
where Q_{θ} is a target network. We also apply the double-Q trick to stabilize training. The policy improvement step requires more careful algorithmic design. Practical implementations often rely on a limited number of gradient descent steps for optimizing, thus could suffer from our-of-support samples. This leads to policy optimization errors which are further exacerbated iteratively. We find it is useful to add the original behavior regularization,

$$\max_{\omega} \mathbb{E}_{s\sim\mathcal{D}}\left[\mathbb{E}_{a\sim\pi_{\omega}}[Q_{\theta}(s,a)] - \tau\lambda D_{\mathrm{KL}}(\pi_{\omega}(s) \parallel \pi(s)) - \tau(1-\lambda) D_{\mathrm{KL}}(\pi_{\omega}(s) \parallel \pi_{\mathcal{D}}(s))\right]$$

Ensembles of Reference Policy. One limitation of CPI lies in the potential for a negligible difference between the learning policy and the reference policy, due to the limited gradient steps when optimizing. To improve the efficiency of policy improvement, we explore the idea of using an ensembles of reference policies. In particular, we apply two policies with independently initialized parameters ω^1 and ω^2. Let Q_{θ^1} and Q_{θ^2} be the value functions of these two policies respectively. When updating the parameters ω^i for $i \in \{1,2\}$, we choose the current best policy as the reference policy, where the superiority is decided according to the current value estimate. We call this algorithm Conservative Policy iteration with Reference Ensembles (CPI-RE).

Optimality in the Tabular Setting

We use an inferior behavior policy to collect 10k transitions, of which the action probability is {up:0.1, down:0.4, right:0.1, left:0.4} at every state in the $7 * 7$ grid environment. For the FourRoom environment, we use three types of behavior policy to collect data: (1) Expert dataset: collect 10k transitions with the optimal policy; (2) Random dataset: collect 10k transitions with a random restart and equal probability of taking each action; (3) Missing-Action dataset: remove all down actions in transitions of the upper-left room from the Mixed dataset. Although some behavior policies are suboptimal, the optimal path is ensured to exist in the offline data, in which case a clear algorithm should still be able to identify the optimal path.

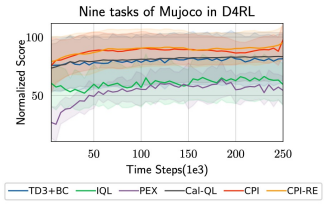


Results on Continuous Control Problems

Dataset	DT	TD3+BC	CQL	IQL	POR	EDAC	InAC	STR	D-QL	CPI	CPI-RE
halfcheetah-random	2.2	11.0	**31.3**	13.7	29.0	28.4	19.6	20.6	22.0	**29.7±1.1**	**30.7±0.4**
hopper-random	5.4	8.5	5.3	8.4	12.0	25.3	**32.4**	**31.3**	18.3	29.5±3.7	**30.4±2.9**
waker2d-random	2.2	1.6	5.4	5.9	**6.3**	**16.6**	6.3	4.7	5.5	**5.9±1.7**	5.5±0.9
halfcheetah-medium	42.6	48.3	46.9	47.4	48.8	**65.9**	48.3	51.8	51.5	**64.4±1.3**	**65.9±1.6**
hopper-medium	67.6	59.3	61.9	66.3	98.2	**101.6**	60.3	**101.3**	96.6	**98.5±3.0**	97.9±4.4
waker2d-medium	74.0	83.7	79.5	78.3	81.1	**92.5**	82.7	85.9	**87.3**	85.8±0.8	**86.3±1.0**
halfcheetah-medium-replay	36.6	44.6	45.3	44.2	43.5	**61.3**	44.3	47.5	48.3	**54.6±1.3**	**55.9±1.5**
hopper-medium-replay	82.7	60.9	86.3	94.7	98.9	101.0	92.1	100.0	**102.0**	**101.7±1.6**	**103.2±1.4**
waker2d-medium-replay	66.6	81.8	76.8	73.9	76.6	87.1	69.8	85.7	**98.0**	**91.8±2.9**	**93.8±2.2**
halfcheetah-medium-expert	86.8	90.7	95.0	86.7	94.7	**106.3**	83.5	94.9	**97.2**	94.7±1.1	**95.6±0.9**
hopper-medium-expert	107.6	98.0	96.9	91.5	90.0	**110.7**	93.8	**111.9**	**112.3**	106.4±4.3	110.1±4.1
waker2d-medium-expert	108.1	110.1	109.1	109.6	109.1	**114.7**	109.0	110.2	**111.2**	110.9±0.4	**111.2±0.5**
halfcheetah-expert	87.7	96.7	**97.3**	94.9	93.2	**106.8**	93.6	95.2	96.3	96.5±0.2	**97.4±0.4**
hopper-expert	94.2	107.8	106.5	108.8	110.4	110.1	103.4	**111.2**	102.6	**112.2±0.5**	**112.3±0.5**
walker2d-expert	108.3	110.2	109.3	109.7	102.9	**115.1**	110.6	110.1	109.5	**110.6±0.1**	**111.2±0.2**
Gym-MuJoCo Total	972.6	1013.2	1052.8	1034.0	1094.7	**1243.4**	1049.7	1162.2	1158.6	**1193.2**	**1207.4**
antmaze-umaze	59.2	78.6	74.0	87.5	76.8	16.7	84.8	93.6	**96.0**	**98.8±1.1**	**99.2±0.5**
antmaze-umaze-diverse	53.0	71.4	84.0	62.2	64.8	0.0	82.4	77.4	**84.0**	**88.6±5.7**	**92.6±10.0**
antmaze-medium-play	0.0	3.0	61.2	71.2	**87.2**	0.0	-	**82.6**	79.8	82.4±5.8	**84.8±5.0**
antmaze-medium-diverse	0.0	10.6	53.7	70.0	75.2	0.0	-	**87.0**	**82.0**	80.4±8.9	**80.6±11.3**
antmaze-large-play	0.0	0.0	15.8	**39.6**	24.4	0.0	-	**42.8**	**49.0**	20.6±16.3	33.6±8.1
antmaze-large-diverse	0.0	0.2	14.9	47.5	**59.2**	0.0	-	46.8	**61.7**	45.2±6.9	**48.0±6.2**
Antmaze Total	112.2	163.8	303.6	378.0	387.6	16.7	-	**430.2**	**452.5**	416.0	**438.8**
pen-human	73.9	-1.9	35.2	71.5	**76.9**	52.1	52.3	-	75.7	**80.1±16.9**	**87.0±25.3**
pen-cloned	67.3	9.6	27.2	37.3	67.6	**68.2**	-8.0	-	60.8	**71.8±35.2**	**70.7±15.8**
Adroit Total	141.2	7.7	62.4	108.8	**144.5**	120.3	44.3	-	136.5	**151.9**	**157.7**
Total	1226.0	1184.7	1418.8	1520.8	**1626.8**	1380.4	-	-	**1747.6**	**1761.2**	**1803.9**
Runtime (s/epoch)	-	**7.4**	-	-	-	19.6	-	-	39.8	**8.5**	19.1
GPU Memory (GB)	-	**1.4**	-	-	-	1.9	-	-	1.5	**1.4**	**1.4**

Online Fine-tuning

The distinctive feature of CPI lies in its policy iteration training, which makes it particularly well-suited for the online fine-tuning process. In an online setting, as the agent can interact with the environment, there is a need to progressively enhance the level of exploration during the training process.



To achieve this, in the online training process, the weight of $D_{\mathrm{KL}}(\pi_{\omega}(s) \parallel \pi(s))$ remains constant, while the weight of $D_{\mathrm{KL}}(\pi_{\omega}(s) \parallel \pi_{\mathcal{D}}(s))$ decreases exponentially, eventually reducing to 0.1.

Contact

WeChat:

Personal Page: