Deterministic Uncertainty Propagation for Improved Model-Based Offline Reinforcement Learning

Abdullah Akgül, Manuel Haußmann, Melih Kandemir

University of Southern Denmark

NeurIPS 2024

The Scope

- Model-based Offline reinforcement learning

- Problems
  - Distributional shift
    - Limited coverage on state-action space
  - Overestimation bias
    - Errors due to policy search algorithms
    - Yields suboptimal policies
  - Sampling and function approximation errors
    - Further noise on training
    - Decrease on learning speed

- PEssimistic Value Iteration (PEVI)[1]

[1] Jin et al., 2021. Is pessimism provably efficient for offline RL?

Pessimistic Value Iteration

PEVI penalizes Bellman target estimation with the uncertainty on the predicted next state to minimize the suboptimality of a policy π:

$$\texttt{SubOpt}(\pi; s) \triangleq Q_{\pi^*}(s, \pi^*(s)) - Q_\pi(s, \pi(s))$$

for an initial state s.

Pessimistic Value Iteration

PEVI penalizes Bellman target estimation with the uncertainty on the predicted next state to minimize the suboptimality of a policy π:

$$\texttt{SubOpt}(\pi; s) \triangleq Q_{\pi^*}(s, \pi^*(s)) - Q_\pi(s, \pi(s))$$

for an initial state s.

Theorem (*Suboptimality of PEVI*)

For any π derived with PEVI that satisfies

$$\Big| \underbrace{\mathbb{B}_\pi Q(s,a) - \widehat{\mathbb{B}}_\pi Q(s,a,s')}_{\textit{Bellman approximation error}} \Big| \leq \underbrace{\Gamma^{Q}_{\widehat{\mathrm{P}}}(s,a)}_{\textit{Uncertainty quantifier}}, \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}$$

with probability at least $1-\delta$ for some error tolerance $\delta \in (0,1)$, the following inequality holds:

$$\mathit{SubOpt}(\pi; s) \leq f(\Gamma^{Q}_{\widehat{\mathrm{P}}}, s, \pi^*).$$

PEVI Approaches

- MOPO[2] penalizes via uncertainty on the next state
- MOBILE[3] penalizes via uncertainty on the Bellman target

Both approximate the Bellman target by evaluating with a sample s'.

[2] Yu et al., 2020. MOPO: Model-based offline policy optimization

[3] Sun et al., 2023. Model-Bellman inconsistency for model-based offline reinforcement learning

PEVI Approaches

- MOPO[2] penalizes via uncertainty on the next state
- MOBILE[3] penalizes via uncertainty on the Bellman target

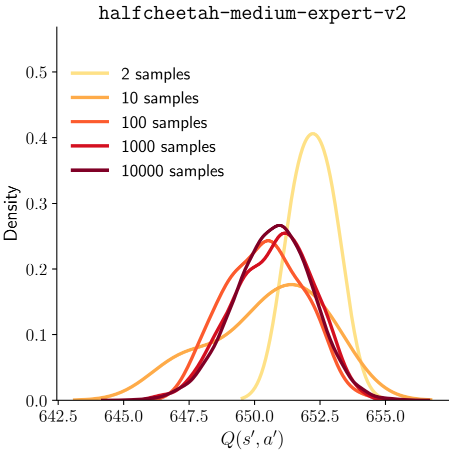
Both approximate the Bellman target by evaluating with a sample s'.

1. Contribution: We provide a suboptimality guarantee for sampling-based PEVI approaches.

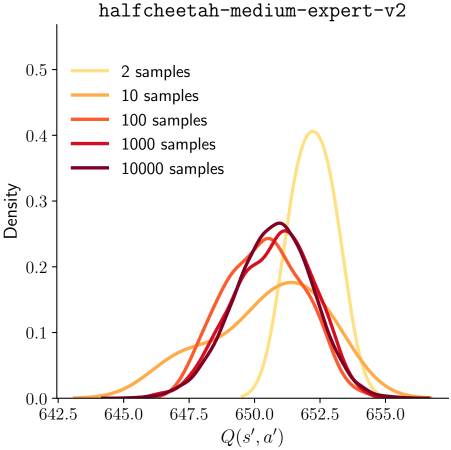
[2] Yu et al., 2020. MOPO: Model-based offline policy optimization

[3] Sun et al., 2023. Model-Bellman inconsistency for model-based offline reinforcement learning

The Challenge: High Variance

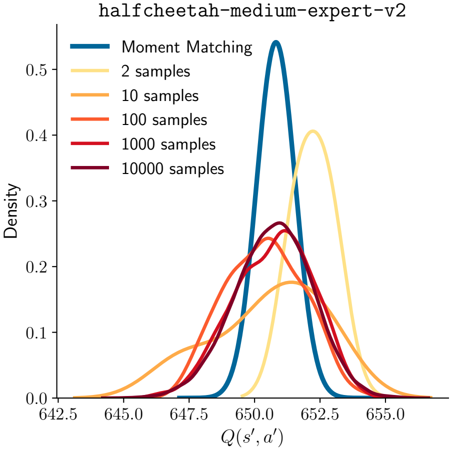


The Challenge: High Variance



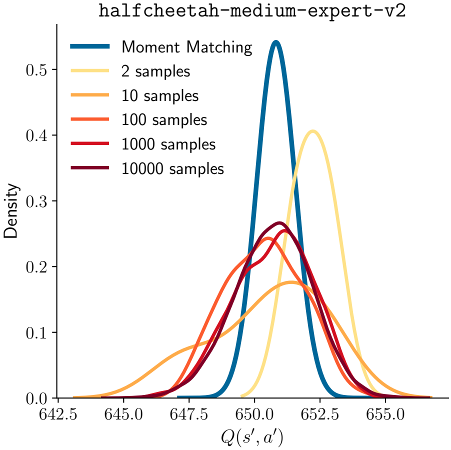
- Distorted gradient signals, delayed convergence
- Poor approximation of the first two moments of Bellman target
- Requirement of larger confidence radii

The Challenge: High Variance



- Distorted gradient signals, delayed convergence
- Poor approximation of the first two moments of Bellman target
- Requirement of larger confidence radii

The Challenge: High Variance



- Distorted gradient signals, delayed convergence
- Poor approximation of the first two moments of Bellman target
- Requirement of larger confidence radii

2. Contribution: Moment matching.

The Solution

MOMBO: Moment Matching Offline Model-Based Policy Optimization

- Deterministic uncertainty propagation
  - Propagating first two moments of uncertain input through a value function
    - Neural network
- Lower confidence bound on the estimation of Bellman target

The Solution

MOMBO: Moment Matching Offline Model-Based Policy Optimization

- Deterministic uncertainty propagation
  - Propagating first two moments of uncertain input through a value function
    - Neural network
- Lower confidence bound on the estimation of Bellman target

- **3. Contribution:** Suboptimality bound for moment matching
  - Tighter bound
  - Provably more efficient

The Experiments

Performance Evaluation

Dataset Type	Environment	Normalized Reward (↑)			AULC (↑)		
		MOPO	MOBILE	MOMBO	MOPO	MOBILE	MOMBO
random	halfcheetah	37.2±1.6	41.2±1.1	**43.6±1.1**	36.3±1.0	39.5±1.2	**41.4±1.0**
	hopper	**31.7±0.1**	31.3±0.1	25.4±10.2†	**28.6±1.4**	23.6±3.7	17.3±1.3
	walker2d	8.2±5.6	**22.1±0.9**	21.5±0.1	5.4±3.2	18.0±0.4	**19.2±0.5**
	Average	25.7	**31.5**	30.2	23.4	**27.1**	26.0
medium	halfcheetah	72.4±4.2	75.8±0.8	**76.1±0.8**	70.9±2.0	72.1±1.0	**73.0±0.9**
	hopper	62.8±38.1	103.6±1.0	**104.2±0.5**	37.0±15.3	82.2±7.3	**95.9±2.5**
	walker2d	85.4±2.9	**88.3±2.5**	86.4±1.2	77.6±1.3	79.0±1.3	**84.0±1.1**
	Average	73.6	**89.3**	88.9	61.8	77.8	**84.3**
medium-replay	halfcheetah	**72.1±3.8**	71.9±3.2	72.0±4.3	68.4±4.7	67.9±2.8	**68.7±3.9**
	hopper	92.7±20.7	**105.1±1.3**	104.8±1.0	81.7±4.6	78.7±4.0	**87.3±2.0**
	walker2d	85.9±5.3	**90.5±1.7**	89.6±3.8	65.3±12.7	79.9±4.3	**80.8±5.6**
	Average	83.4	**89.2**	88.8	72.4	75.5	**78.9**
medium-expert	halfcheetah	83.6±12.5	100.9±1.5	**103.3±0.8**	77.1±4.0	94.5±1.8	**95.2±0.7**
	hopper	74.9±44.2	112.5±0.2	**112.6±0.3**	55.6±17.3	82.7±7.3	**84.3±4.7**
	walker2d	108.2±4.3	**114.5±2.2**	113.9±0.9	88.3±6.3	94.3±0.9	**98.9±3.3**
	Average	88.9	109.3	**109.9**	73.6	90.5	**92.8**
	Average Score	67.6	**79.8**	79.5	57.5	67.7	**70.5**
	Average Ranking	2.7	**1.7**	**1.7**	2.7	2.2	**1.2**

† *High standard deviation due to failure in one repetition, which can be mitigated by increasing β. Median result:* 31.3

Conclusion

We introduce MOMBO

- has faster convergence and more stable training
- provides a competitive final performance
- estimates Bellman target more precisely

Conclusion

We introduce MOMBO

- has faster convergence and more stable training
- provides a competitive final performance
- estimates Bellman target more precisely

ADIN LAB Web Page

ADIN LAB

Paper Link

Code Link