ADVANCEMENTS IN OFFLINE REINFORCEMENT LEARNING

AUTHORS

Aylin Rasteh

Pouya Lahabi

Adversarial Model for Offline Reinforcement Learning
Model-based Offline Policy Optimization
Conservative Offline Model-Based Policy Optimization

INTRODUCTION

Offline Reinforcement Learning (RL) uses pre-collected datasets to train policies without new interactions with the environment, which is crucial for applications where gathering data is costly or risky. This approach is vital in fields like autonomous driving and healthcare, where safety and reliability are essential. Offline RL enables robust policy deployment in real-world scenarios but faces challenges like distributional shift, uncertainty estimation, and ensuring policy robustness.

OBJECTIVE

The objective of this poster is to present and compare the methodologies, theoretical insights, and empirical results from three key papers in offline reinforcement learning: Adversarial Model for Offline Reinforcement Learning (ARMOR), Model-based Offline Policy Optimization (MOPO), and Conservative Offline Model-Based Policy Optimization (COMBO)

RELATED LITERATURE

1. Bhardwaj, M., Xie, T., Boots, B., Jiang, N., & Cheng, C.-A. (2023). Adversarial Model for Offline Reinforcement Learning.
2. Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J., Levine, S., Finn, C., & Ma, T. (2020). Model-based Offline Policy Optimization.
3. Yu, T., Kumar, A., Rafailov, R., Rajeswaran, A., Levine, S., & Finn, C. (2020). Conservative Offline Model-Based Policy Optimization.

METHODOLOGY

Adversarial Model for Offline Reinforcement Learning (ARMOR)

- Adversarial training for robust policy improvement.
- Ensures performance relative to a reference policy under uncertain data conditions.
- Achieves state-of-the-art results with a single model.

Model-based Offline Policy Optimization (MOPO)

- Incorporates model uncertainty into the reward function.
- Penalizes rewards based on dynamics model uncertainty.
- Provides a theoretical guarantee for a lower bound on policy performance.
- Outperforms traditional model-free and other model-based offline RL algorithms.

Conservative Offline Model-Based Policy Optimization (COMBO)

- Integrates conservative planning into the learning process.
- Enhances robustness and reliability of policy optimization.
- Achieves superior performance in complex dynamic environments.

ARMOR

We propose to directly optimize **relative performance** to the reference policy

$$\max_{\pi \in \Pi} J(\pi) - J(\pi_{\text{ref}})$$

to obtain a lower bound that is tight at π_{ref}

ARMOR constructs a two-player game using hypothesis class $\mathcal{M}_\alpha$

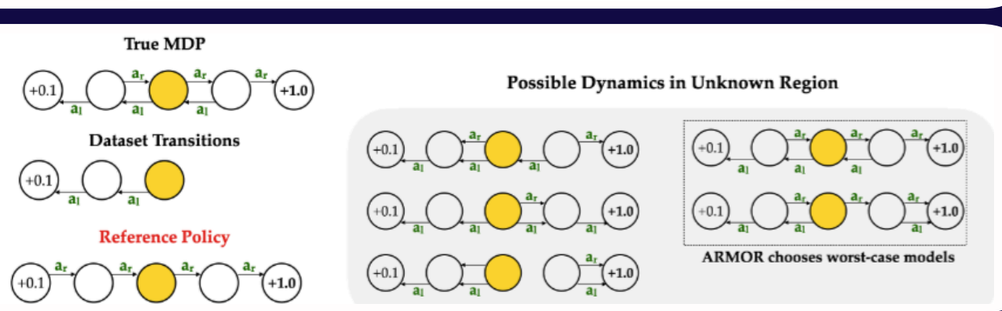
$$\hat{\pi} \in \max_{\pi \in \Pi} \min_{M \in \mathcal{M}_\alpha} J_M(\pi) - J_M(\pi_{\text{ref}})$$

where inner optimization minimizes lower bound on relative performance if $\mathcal{M}_\alpha$ contains true MDP model $M^\star$

$$\mathcal{M}_\alpha := \{M \in \mathcal{M} : \mathcal{E}_\mathcal{D}(M) - \min_{M' \in \mathcal{M}} \mathcal{E}_\mathcal{D}(M') \le \alpha\}$$
$$\mathcal{E}_\mathcal{D}(M) := \textstyle\sum_\mathcal{D} -\log P_M(s'|s,a) + {}^{(R_M(s,a)-r)^2}/_{V_{\max}^2}$$

Choose α s.t. $M^\star \in \mathcal{M}_\alpha$

ARMOR ensures policy improvement even when π_{ref} is not covered by the dataset



Algorithm 1 ARMOR (Adversarial Model for Offline Reinforcement Learning)

Input: Batch data $\mathcal{D}_{\text{real}}$, policy π, MDP model M, critics f_1, f_2, horizon H, constants $\beta, \lambda \ge 0$, $\tau \in [0,1]$, $w \in [0,1]$,

1: Initialize target networks $\bar{f}_1 \leftarrow f_1$, $\bar{f}_2 \leftarrow f_2$ and $\mathcal{D}_{\text{model}} = \emptyset$
2: **for** $k = 0, \ldots, K-1$ **do**
3: Sample minibatch $\mathcal{D}^{\text{mini}}_{\text{real}}$ from dataset $\mathcal{D}_{\text{real}}$ and minibatch $\mathcal{D}^{\text{mini}}_{\text{model}}$ from dataset $\mathcal{D}_{\text{model}}$.
4: Construct transition tuples using model predictions

$$\mathcal{D}_M := \left\{(s, a, r_M, s'_M) : r_M = R_M(s,a), s'_M \sim P_M(\cdot \mid s,a), (s,a) \in \mathcal{D}^{\text{mini}}_{\text{real}} \cup \mathcal{D}^{\text{mini}}_{\text{model}}\right\}$$

5: Update the adversary networks; for $i = 1, 2$,

$$l^{\text{adversary}}(f, M) := \mathcal{L}_{\mathcal{D}_M}(f, \pi, \pi_{\text{ref}}) + \beta\left(\mathcal{E}^w_{\mathcal{D}_M}(f, M, \pi) + \lambda \mathcal{E}_{\mathcal{D}^{\text{mini}}_{\text{real}}}(M)\right) \quad (4)$$
$$M \leftarrow M - \eta_{\text{fast}}\left(\nabla_M l^{\text{adversary}}(f_1, M) + \nabla_M l^{\text{adversary}}(f_2, M)\right)$$
$$f_i \leftarrow \text{Proj}_\mathcal{F}(f_i - \eta_{\text{fast}} \nabla_{f_i} l^{\text{adversary}}(f_i, M)) \quad \text{and} \quad \bar{f}_i \leftarrow (1-\tau)\bar{f}_i + \tau f_i$$

6: Update actor network with respect to the first critic network and the reference policy

$$l^{\text{actor}}(\pi) := -\mathcal{L}_{\mathcal{D}_M}(f_1, \pi, \pi_{\text{ref}}) \quad (5)$$
$$\pi \leftarrow \text{Proj}_\Pi(\pi - \eta_{\text{slow}} \nabla_\pi l^{\text{actor}}(\pi))$$

7: If $k \% H = 0$, then reset model state: $\bar{S}_\pi \leftarrow \{s \in \mathcal{D}^{\text{mini}}_{\text{real}}\}$ and $\bar{S}_{\pi_{\text{ref}}} \leftarrow \{s \in \mathcal{D}^{\text{mini}}_{\text{real}}\}$
8: Query the MDP model to expand $\mathcal{D}_{\text{model}}$ and update model state

$$\bar{A}_\pi := \{a : a \sim \pi(s), s \in \bar{S}_\pi\} \quad \text{and} \quad \bar{A}_{\pi_{\text{ref}}} := \{a : a \sim \pi_{\text{ref}}(s), s \in \bar{S}_{\pi_{\text{ref}}}\}$$
$$\mathcal{D}_{\text{model}} := \mathcal{D}_{\text{model}} \cup \{\bar{S}_\pi, \bar{A}_\pi\} \cup \{\bar{S}_{\pi_{\text{ref}}}, \bar{A}_{\pi_{\text{ref}}}\}$$
$$\bar{S}_\pi \leftarrow \{s' \mid s' \sim \texttt{detach}(P_M(\cdot \mid s,a)), s \in \bar{S}_\pi, a \in \bar{A}_\pi\}$$
$$\bar{S}_{\pi_{\text{ref}}} \leftarrow \{s' \mid s' \sim \texttt{detach}(P_M(\cdot \mid s,a)), s \in \bar{S}_{\pi_{\text{ref}}}, a \in \bar{A}_{\pi_{\text{ref}}}\}$$

9: **end for**

COMBO

Key Idea: Augment the offline dataset with model rollouts while addressing distribution shift by learning lower-bounded Q-values without uncertainty quantification

Policy evaluation

$$\hat{Q}^{k+1} \leftarrow \arg\min_Q \beta\left(\mathbb{E}_{\mathbf{s},\mathbf{a} \sim \rho(\mathbf{s},\mathbf{a})}[Q(\mathbf{s},\mathbf{a})] - \mathbb{E}_{\mathbf{s},\mathbf{a} \sim \mathcal{D}}[Q(\mathbf{s},\mathbf{a})]\right)$$
$$+\frac{1}{2}\mathbb{E}_{\mathbf{s},\mathbf{a},\mathbf{s}' \sim d_f}\left[\left(Q(\mathbf{s},\mathbf{a}) - \widehat{\mathcal{B}}^\pi \hat{Q}^k(\mathbf{s},\mathbf{a})\right)^2\right].$$

Minimize big Q-values on model-generated data

Maximize Q-values under the dataset

Bellman Error with mixture of offline data and model-generated data

Policy improvement

$$\pi' \leftarrow \arg\max_\pi \mathbb{E}_{\mathbf{s} \sim \rho, \mathbf{a} \sim \pi(\cdot|\mathbf{s})}\left[\hat{Q}^\pi(\mathbf{s},\mathbf{a})\right].$$

Theoretical Guarantees of COMBO

CQL lower-bound

$$\hat{V}^\pi_{\text{CQL}}(s) := \mathbb{E}_{a \sim \pi_k}[\hat{Q}^\pi_{\text{CQL}}(s,a)] \le V^\pi(s) \quad \forall\, s \in \mathcal{D}$$

Conservative Q-values learned by CQL

Ground-Truth Q-values

Lower-bounded under all states in the dataset

COMBO lower-bound

$$\mathbb{E}_{\mathbf{s} \sim \mu_0, \mathbf{a} \sim \pi(\cdot|\mathbf{s})}[\hat{Q}^\pi(\mathbf{s},\mathbf{a})] \le \mathbb{E}_{\mathbf{s} \sim \mu_0, \mathbf{a} \sim \pi(\cdot|\mathbf{s})}[Q^\pi(\mathbf{s},\mathbf{a})]$$

Conservative Q-values learned by COMBO

Ground-Truth Q-values

Lower-bounded under initial state distribution and learned policy

COMBO lower-bound vs CQL lower bound

$$\hat{V}^\pi_{\text{COMBO}}(\mathbf{s}) \ge \hat{V}^\pi_{\text{CQL}}(\mathbf{s}) \quad \forall \mathbf{s} \in \mathcal{D}$$

If model-generated rollouts mostly consist of out-of-distribution states

COMBO is less conservative than CQL since COMBO does not

Algorithm 1 COMBO: Conservative Model Based Offline Policy Optimization

Require: Offline dataset $\mathcal{D}$, rollout distribution $\mu(\cdot|\mathbf{s})$, learned dynamics model $\widehat{T}_\theta$, initialized policy and critic π_ϕ and Q_ψ.

1: Train the probabilistic dynamics model $\widehat{T}_\theta(\mathbf{s}', r|\mathbf{s},\mathbf{a}) = \mathcal{N}(\mu_\theta(\mathbf{s},\mathbf{a}), \Sigma_\theta(\mathbf{s},\mathbf{a}))$ on $\mathcal{D}$.
2: Initialize the replay buffer $\mathcal{D}_{\text{model}} \leftarrow \varnothing$.
3: **for** $i = 1, 2, 3, \cdots$, **do**
4: Collect model rollouts by sampling from μ and $\widehat{T}_\theta$ starting from states in $\mathcal{D}$. Add model rollouts to $\mathcal{D}_{\text{model}}$.
5: Conservatively evaluate π^i_ϕ by repeatedly solving eq. 2 to obtain $\hat{Q}^{\pi^i_\phi}_\psi$ using samples from $\mathcal{D} \cup \mathcal{D}_{\text{model}}$.
6: Improve policy under state marginal of d_f by solving eq. 3 to obtain π^{i+1}_ϕ.
7: **end for**

MOPO

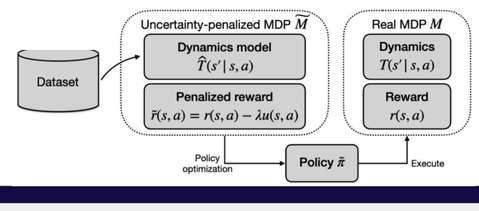
We optimize π in an *uncertainty-penalized* MDP $\tilde{\mathcal{M}} = (\mathcal{S}, \mathcal{A}, \hat{T}, \tilde{r}, \gamma)$, where

$$\tilde{r}(s,a) = r(s,a) - \lambda u(s,a)$$

Key property: if $u(s,a)$ is an *admissible error estimator*, i.e., upper bounds the error of $\hat{T}(s'|s,a)$, then for a particular choice of λ,

$$\eta_\mathcal{M}(\pi) \ge \eta_{\tilde{\mathcal{M}}}(\pi)$$

where $\eta_\mathcal{M}(\pi)$ and $\eta_{\tilde{\mathcal{M}}}(\pi)$ denote the expected return of π in the MDPs $\mathcal{M}$ and $\tilde{\mathcal{M}}$, respectively.



ANALYSIS

Standard Offline RL
(REF as behavior cloned policy on given dataset)

Algo	ARMOR	MoReL	MOPO	RAMBO	COMBO	ATAC	CQL	IQL
mujoco score	**87.6**	72.9	42.1	83.7	82.0	88.0	67.2	76.9

ARMOR is the only model-based method that doesn't rely on model ensembles!