Active Learning for Iterative Offline Reinforcement Learning

Lan Zhang EMMAZL@AMAZON.COM

Luigi Tedesco LTEDESC@AMAZON.COM

Pankaj Rajak RAJAKPAN@AMAZON.COM

Youcef Zemmouri YOUCZ@AMAZON.COM

Hakan Brunzell BRUNZELL@AMAZON.COM
Amazon Inc.

Abstract

Offline Reinforcement Learning (RL) has emerged as a promising approach to address real-world challenges where online interactions with the environment are limited, risky, or costly. Although, recent advancements produce high quality policies from offline data, currently, there is no systematic methodology to continue to improve them without resorting to online fine-tuning. This paper proposes to repurpose Offline RL to produce a sequence of improving policies, namely, Iterative Offline Reinforcement Learning (IORL). To produce such sequence, IORL has to cope with imbalanced offline datasets and to perform controlled environment exploration. Specifically, we introduce "Return-based Sampling" as means to selectively prioritize experience from high-return trajectories and active learning driven "Dataset Uncertainty Sampling" to probe state-actions inversely proportional to density in the dataset.We demonstrate that our proposed approach produces policies that achieve monotonically increasing average returns, from 65.4 to 140.2, in the Atari environment.

Keywords: Offline Reinforcement Learning, Active Learning

1. Introduction

Tabula rasa policy learning has prevented Reinforcement Learning (RL) from being used in many real-world applications - as starting with a random policy is prohibitive. In recent years, offline RL has gained significant attention as means to learn policies from logged data, and, therefore, became a promising approach to high-risk domains, such as healthcare, finance, robotics, and autonomous driving - Tang and Wiens (2021); Liu et al. (2022).

Unlike online RL, offline RL does not interact with the environment, and thus faces unique hurdles by exclusively relying on pre-collected data Levine et al. (2020); Yarats et al. (2022) to learn robust policies. However, real-world datasets often consist of diverse experiences from multiple suboptimal policies or human experts, leading to imbalanced datasets where valuable and suboptimal experiences coexist. Even if the dataset includes optimal trajectories, extracting valuable insights from a large, noisy dataset can be challenging for state-of-the-art offline RL algorithms, potentially limiting generalization in practical applications.

A common solution to continuous policy improvement is to fine-tune the offline policy using online algorithms. Not only offline RL algorithm performance degrades after fine-tuning

Nair et al. (2020), but switching to off-policy learning is impractical in many applications due to potential risks associated with policy shifts. Alternatively, the policy learned through offline data can be held fix after deployment. This ensures that the policy behaves predictably in high risk environments. However, a fixed policy will continue to make the same mistakes even if actions have been observed to lead to bad returns Ghosh et al. (2022b).

Considering the trade-offs between policy stability and improvement, we proposed Iterative Offline Reinforcement Learning (IORL). During policy learning, the agent doesn't interact with the environment, guaranteeing that no adverse effects will arise from the learning process itself. The policy learned through offline data is held fix after deployment to ensure that it behaves predictably in high risk environments. IORL attempts to continue to improve policies by introducing active learning based controlled exploration to collect novel experience. This process seeks to bridge the gap between off-policy and offline settings by allowing the policy to improve iteratively without drifting.

This work addresses both (1) handling imbalanced experience in offline datasets and (2) improving dataset coverage with controlled exploration. Section 3.1 covers the challenge of transforming imbalanced datasets to facilitate offline RL learning. Through experiments, we demonstrate the importance of selectively weighting high-return trajectories (Return-based Sampling), resulting in a notable performance improvement - the average return increased from 269.4 to 350.78 in the Atari Breakout environment with half the data. The section 3.2 delves into the active learning based exploration (Dataset Uncertainty Sampling) which prioritizes state-action spaces that were unseen in the logged data. Our experimental findings (Section 4) presents empirical evidence of the efficacy of the Iterative Offline Reinforcement Learning (IORL) framework - doubling of the offline policy's performance after 5 iterations.

2. Background

2.1 Offline Reinforcement Learning

Given states, $s \in S$, actions, $a \in A$, and rewards $r \in R$ spaces, together with environment transition dynamics, $P : S \times A \times S \times R \rightarrow [0, 1]$, and a discounting factor, γ, we define a Markov Decision Process, $M = (S, A, R, P, \gamma)$. The sequence of interactions over the MDP produces a trajectory, $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \ldots)$. A transition or step, $\delta_t^{\tau_i} = (s_t, a_t, r_t, s_{t+1})$, is defined as the sub-trajectory between s_t and s_{t+1} from τ_i. Let $\mathcal{L}_{\text{TD}}(\mathcal{D}) = \mathbb{E}_{\delta \sim \mathcal{D}} \left[(r_i + \gamma \max_a Q_{\hat{\theta}}(s_{t+1}, a) - Q_\theta(s_t, a_t))^2 \right]$ be the Temporal Difference (TD) error using a target $Q_{\hat{\theta}}$. The goal of Offline RL is to learn a parameterized policy, π_θ, where $\theta = \arg\min_\theta \mathcal{L}_{\text{TD}}(\mathcal{D}_{\pi_\beta})$, over a fixed dataset of episodic experience, $\mathcal{D}_{\tau_i \sim \pi_\beta} = (\tau_1, \tau_2, \ldots, \tau_N)$, collected according to a behavioral policy, $\pi_\beta(a|s)$. Note that π_β is possibly unknown and it can represent a single policy or the result of a set of policies acting over the environment.

The learning agent, π_θ, is expected to extract knowledge from the offline dataset to performs well across a wide range of scenarios. However, if $\mathcal{D}_{\pi_\beta}$ does not adequately cover the full distribution of possible states and actions, the trained policy may struggle to generalize effectively to unseen or out-of-distribution (OOD) states and actions Fujimoto et al. (2018, 2019); Kumar et al. (2020). Offline RL algorithms are carefully designed to prevent policies from choosing OOD actions. For instance, *Batch-Constrained Deep Q-learning* (BCQ), Fujimoto et al. (2019, 2018), tackles the OOD issue by constraining

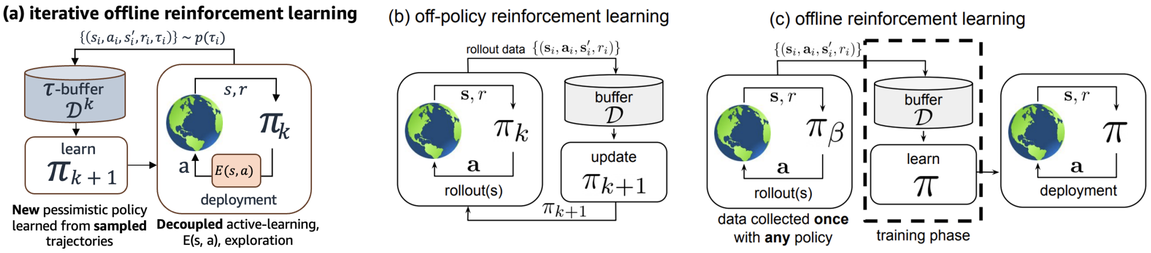

(a) iterative offline reinforcement learning
$\{(s_i, a_i, s'_i, r_i, \tau_i)\} \sim p(\tau_i)$
τ-buffer $\mathcal{D}^k$
learn π_{k+1}
s, r
π_k
a
$E(s,a)$
deployment
New pessimistic policy learned from sampled trajectories
Decoupled active-learning, E(s, a), exploration
(b) off-policy reinforcement learning
rollout data $\{(\mathbf{s}_i, \mathbf{a}_i, \mathbf{s}'_i, r_i)\}$
$\mathbf{s}, r$
π_k
$\mathbf{a}$
rollout(s)
buffer $\mathcal{D}$
update π_{k+1}
π_{k+1}
(c) offline reinforcement learning
$\{(\mathbf{s}_i, \mathbf{a}_i, \mathbf{s}'_i, r_i)\}$
$\mathbf{s}, r$
π_β
$\mathbf{a}$
rollout(s)
data collected once with any policy
buffer $\mathcal{D}$
learn π
training phase
$\mathbf{s}, r$
π
$\mathbf{a}$
deployment

Figure 1: General comparison of (a) IORL with other RL algorithms: (b) off-policy RL and (c) offline RL. For detailed comparison see sub-sections 3.1 to 3.3, and Levine et al. (2020).

next action in the Q-learning backup combining a behavioral cloning model G_w and a perturbation model ξ_ϕ. On the other hand, *Critic Regularized Regression (CRR)* Wang et al. (2020) discourages the Q-value estimates from taking actions that lie outside the training distribution by filtering unpromising actions during training where $Q(s_t, a_t) \leq Q(s_t, \pi(s_t))$. Conversely, *Conservative Q-learning(CQL)* Kumar et al. (2020) introduces a loss term, $\mathbb{E}_{s\sim\mathcal{D}}\left[\log \sum_a \exp(Q(s,a)) - \mathbb{E}_{a\sim\pi_\beta}\left[Q(s,a)\right]\right]$, to penalize Q-values for unseen state-actions.

2.2 Exploration in Offline RL

All the aforementioned algorithms effectively learn to avoid regions of the state-action space unexplored by π_β, which also makes these algorithms in general anti-exploratory once deployed in test environments. Taking CQL as an example, the Q-values for unseen state-actions are distorted by the pessimistic term and even fine-tuning it with new experience may take time to produce optimistic Q-values that promote policy-driven exploration. In the limit, offline policies solely coupled with ϵ-greedy exploration have an exponential sample complexity to find promising state-actions. Here, some research work has to be done to either construct naturally adaptive offline RL algorithm Ghosh et al. (2022a) or exploration incentives in them in test environment Rezaeifar et al. (2022).

3. Iterative Offline Reinforcement Learning

The primary goal of IORL is to produce a progressive sequence of policies, denoted as $(\pi_0, \pi_1, \ldots, \pi_k)$, where each subsequent policy π_{t+1} is an enhancement over its predecessor π_t, $(\pi_t \leq \pi_{t+1})$. The iterative process unfolds as follows:

1. **Initial Policy Learning**: Train the first policy π_1 over the pre-collected data $\mathcal{D}^1_{a\sim\pi_\beta}$.
2. **Policy Deployment**: Policy π_t is held fix after deployed in the environment and exploratory actions are taken on a fraction of trajectories, collecting a new dataset $\mathcal{D}^{t+1}_{a\sim\pi_t,\pi_{t-1},\ldots,\pi_\beta} = \mathcal{D}^t \cup \{\tau_{k+1}, \tau_{k+2}, \ldots\}, \tau_j \sim \pi^E_t$.
3. **Policy Improvement**: Whenever the agent reaches a predefined policy update criteria the next iteration of the offline policy, π_{t+1}, is trained using offline RL algorithms and the augmented dataset $\mathcal{D}^{t+1}$ and, subsequently, deployed.

By construction, the next policy π_{t+1} is learned on a super set of the previous dataset D^t used to trained π_t. Therefore, we guarantee the next policy have access to data to be, at least, as good as the existing policy. However, policy improvement will only happen if we overcome the absence of built-in exploration mechanisms in Offline RL. This demands to enrich the offline dataset $\mathcal{D}^t$ with uncharted state-action spaces.

3.1 Offline Learning on Imbalanced Data

Offline RL is particularly sensitive to the composition of the offline dataset. We argue that training an effective policy offline is challenging when datasets encompasses (1) experiences from a broad range of performance levels or (2) incomplete trajectories. In practice, CQL have been reported to struggle to learn when sub-optimal trajectories are added to an expert dataset (c.f. D4RL's mixture vs expert) Kumar et al. (2020). Moreover, we have observed the importance of keeping complete trajectories in the experience buffer for the agent to learn how to reach intermediate states - further explored in section 4.1.

We propose to use a trajectory-level buffer (τ-buffer) and employ a weighted sampling mechanism that trajectory returns, $w_{\tau_i} \propto \sum_{t=0}^{\infty} \gamma^t r_t^{\tau_i}$, to prioritize trajectories according to their overall return - where the probability of sampling a trajectory is $p(\tau_i) = w_{\tau_i} / \sum_j w_{\tau_j}$. This approach concentrates learning on the trajectories exhibiting the highest returns. It's worth noting that omitting a specific experience from the dataset instructs the policy not to follow that particular action, as pessimistic penalties are added. This mechanism allows the policy to learn from experiences which are not included in the training dataset and to reduce the training size through removing under-performing experiences.

3.2 Active Learning Environment Exploration

We introduce a novel method for exploring and sampling new experiences from the environment, termed "Dataset Uncertainty" exploration. As outlined in Algorithm 1, we initiate by training the offline policy π_t using the dataset $\mathcal{D}^t$ and the uncertainty model $E_{\omega_t}(a|s) : S \times A \to [0, 1]$, such that $\sum_a E_\omega(a|s_i) = 1, \forall s_i \in S$. For MDPs with long trajectories (i.e. 1000+ transitions), traditional ϵ-greedy will, very likely, take at least one exploratory action in every trajectory. Therefore, during deployment, we first determine whether to engage exploration in a certain trajectory. Then, we use the parameter ϵ to determine when to explore within a given episode. The new experiences are then appended to the existing offline dataset weighted by their return.

The action probability model is trained minimizing the expected Negative Log-Likelihood together with a L2-regularization component - weighted by the scalar: α. Exploratory action, a^E, is draw from the probability model:

$$\mathcal{L}_E^t(\omega) = \mathbb{E}_{(a_i, s_i) \sim D^t} \left[-\log(E_\omega(a_i|s_i) \right] + \alpha \|\omega\|^2$$

$$a^E(s_i) = \arg\min_a E_\omega(a|s_i)$$

The Dataset Uncertainty exploration method aims to uncover unexplored areas in the state-action space. The uncertainty model is trained minimizing the loss L_E^t over $\mathcal{D}^t$, predicting the probability of each action, a, been seen on the dataset D^t, given a state s. The objective is to select the action that is least likely to have been encountered in the dataset $\mathcal{D}^t$ - $a^E = \arg\min_a E_\omega(s)$. This process ensures that the action selected for exploration is the one associated with the highest uncertainty value under the prevailing offline policy.

3.3 Off-Policy RL Comparison

Off-Policy RL is intrinsically optimistic, meaning, it promotes policy-driven exploration via overestimation of unseen state-action value function. In contrast, IORL relies on offline RL algorithms to learn, by construction, anti-exploratory policies. Thus, it requires additional mechanism such as active learning, as proposed in this paper, to explore unseen state-actions.

4. Evaluation

4.1 Return-based Sampling

To evaluate the effectiveness of the Return-based Sampling technique, we conducted experiments using intentionally imbalanced datasets that captures various levels of experience. At certain epochs of training Rainbow, Hessel et al. (2017), we would stop training and use the current policy to collect environment experience. The datasets were constructed by concatenating data obtained from distinct epochs . This compilation is consistent to the ”mixed” dataset found in the D4RL dataset Fu et al. (2021). This deliberate imbalance simulates real-world complexities where experience comes from different policies.

In our experimental evaluation, we scrutinized the performance of offline RL methods on both the accumulated dataset and the sampled datasets within the Breakout and Pong environments (refer to Table 1). A comparative analysis was conducted between the Return-based sampling method and uniform sampling, a technique involving the uniform selection of complete trajectories from the accumulated dataset.

Our findings, illustrated in Table 1, underscore that the proposed sampling strategy consistently produces better policies in imbalanced datasets. Particularly noteworthy is the performance enhancement in the Breakout environment, where the CRR algorithm exhibited a remarkable 300% increase, followed by BCQ with a 40% increase, and CQL with a 30% increase. A parallel trend was observed in the Pong environment. In Off-Policy RL, transition weighting and sampling is a common approach to extract information from the experience replay buffer. Nevertheless, we have observed the importance of maintaining complete trajectories, as all transition-based sampling methods failed to improve performance - including Prioritized Experience Replay. Schaul et al. (2016) and reward-based sampling. Intuitively, incomplete trajectories may prevent the agent to learn to reach critical states.

Table 1: Trajectory based Data Sampling

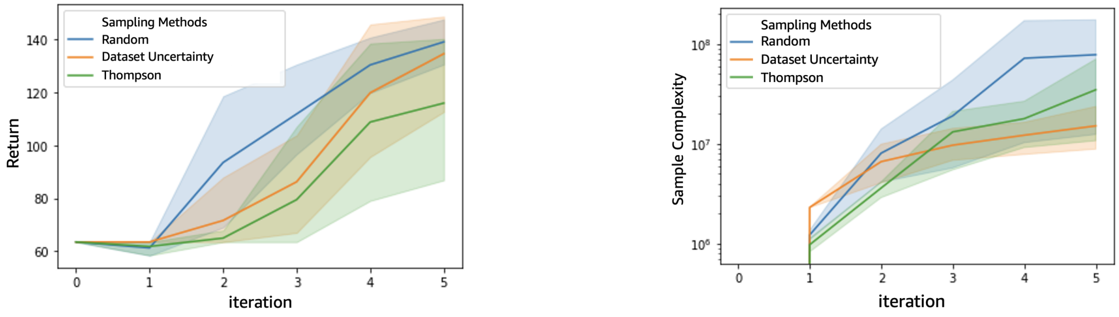
Environment	Dataset Transformation	CRR	BCQ	CQL
Breakout	Accumulated Dataset	66.9	222.14	269.4
	Trajectory based Uniform Sampling	28.21	254.4	165.59
	Return-based Sampling	**295.23**	**311.3**	**350.78**
Pong	Accumulated Dataset	-0.09	10.92	11.27
	Trajectory based Uniform Sampling	-0.62	11.92	15.35
	Return-based Sampling	**12.73**	**16.56**	**20.21**

4.2 Environment Exploration

To gauge the effectiveness of IORL, we initiate the process by utilizing a sub-optimal dataset generated by Rainbow in the early stages of learning, $\mathcal{D}^1_{\pi^{on}_\beta}$. Then, we ran IORL for five iterations, following the 2-step process of: Policy Deployment and Policy Improvement (3). Based on the size of our initial training dataset of 100 trajectories, we came with a policy update rule of 20 new trajectories (c.f. $collectExperience$ in algorithm 1) that exhibit higher returns than the preceding dataset.

Figure 2a depicts the average return of Dataset Uncertainty based exploration and its comparison with other exploration methods (Random and Thompson sampling) in the Breakout environment, where each experiment is repeated five times. Random ($a_t \sim uniform(A)$) and Dataset Uncertainty based exploration exhibited good performance increase compared to the initial 65.4 points, averaging a return around 140.2 points. Thompson sampling increase was not as expressive, achieving only 120 points in average. Better prior distribution selection could improve the performance of Thompson sampling, however, determining the appropriate prior reward distribution can be challenging in real-world applications and, thus, we used the default beta distribution.

In Figure 2b, we quantitatively assessed the sample complexity, a crucial metric that measures the number of transitions needed to discover enhanced trajectories within each iteration. Random exploration displays exponential sample complexity, requiring 100 million data samples. We limited our experimentation to five iterations, since searching for optimal trajectories would demand a substantial time investment. The Dataset Uncertainty approaches require only around 5 to 7 million data samples, significantly fewer than Random exploration and Thompson sampling.



(a) IORL performance

(b) Sample Complexity

(

Figure 2: Left is the performance improvement experienced by applying five iterations of IORL over a initial policy in the Breakout environment. Right is the sample complexity required by each exploration method.

5. Conclusion

In this paper, we introduced the concept of Iterative Offline Reinforcement Learning (IORL), a methodology that combines learning from evolving datasets and controlled environment exploration through active learning to continue to improve policies after environment deployment. The conducted empirical experiments underscore the efficacy of IORL in the Atari environment - summarized in figures 2a and 2b. In conclusion, IORL offers a well-defined route for policy enhancement - even in high risk domains.

Acknowledgments

We would like to thank Wojciech Kowalinski for discussions and guidance

Appendix A. - Iterative Offline Reinforcement Algorithm

Below we present the outline of the algorithm used during our experiementation:

Algorithm 1 Iterative Offline RL

```
Require: $\mathcal{D}^0, \epsilon$, env
  while  do
    $\mathcal{D}^{t+1} \leftarrow TrajectorySampling(\mathcal{D}^t)$
    $\theta_{t+1} \leftarrow \theta_t - \nabla_\theta \mathcal{L}_{\text{CQL}}(\mathcal{D}^{t+1})$
    $\omega_{t+1} \leftarrow \omega_t - \nabla_\omega \mathcal{L}_E(\mathcal{D}^{t+1})$
    while $collectExperience(\mathcal{D}^{t+1})$ do
      for $s_t$ in env.$notDone()$ do
        if $explore(\tau_i)$ and $Rand() < \epsilon$ then
          $a_t \leftarrow \arg\min_a E_\omega(a|s_t)$
        else
          $a_t \sim \pi(a|s_t)$
        end if
        $r_t \leftarrow$ env.$act(a_t)$
        $\tau_i \leftarrow \tau_i \cup (s_t, a_t, r_t)$
      end for
      $\mathcal{D}^{t+1} \leftarrow \mathcal{D}^{t+1} \cup \tau_i$
    end while
  end while
```

Note that in our experiments we have updated the uncertainty model at every iteration, however E_w can be updated online in order to provide better uncertainty estimates. Also, the *explore* is basically an indicator enabling exploration to happen in that episode.

References

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning, 2021.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. *CoRR*, abs/1812.02900, 2018. URL `http://arxiv.org/abs/1812.02900`.

Scott Fujimoto, Edoardo Conti, Mohammad Ghavamzadeh, and Joelle Pineau. Benchmarking batch deep reinforcement learning algorithms. *arXiv preprint arXiv:1910.01708*, 2019.

Dibya Ghosh, Anurag Ajay, Pulkit Agrawal, and Sergey Levine. Offline rl policies should be trained to be adaptive, 2022a.

Dibya Ghosh, Anurag Ajay, Pulkit Agrawal, and Sergey Levine. Offline rl policies should be trained to be adaptive, 2022b.

Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning, 2017.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 1179–1191, 2020.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Xiao-Yang Liu, Ziyi Xia, Jingyang Rui, Jiechao Gao, Hongyang Yang, Ming Zhu, Christina Wang, Zhaoran Wang, and Jian Guo. Finrl-meta: Market environments and benchmarks for data-driven financial reinforcement learning. *Advances in Neural Information Processing Systems*, 35:1835–1849, 2022.

Ashvin Nair, Murtaza Dalal, Abhishek Gupta, and Sergey Levine. Accelerating online reinforcement learning with offline datasets. *CoRR*, abs/2006.09359, 2020. URL `https://arxiv.org/abs/2006.09359`.

Shideh Rezaeifar, Robert Dadashi, Nino Vieillard, Léonard Hussenot, Olivier Bachem, Olivier Pietquin, and Matthieu Geist. Offline reinforcement learning as anti-exploration, 2022.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay, 2016.

Shengpu Tang and Jenna Wiens. Model selection for offline reinforcement learning: Practical considerations for healthcare settings. In *Machine Learning for Healthcare Conference*, pages 2–35. PMLR, 2021.

Ziyu Wang, Alexander Novikov, Konrad Zolna, Josh S Merel, Jost Tobias Springenberg, Scott E Reed, Bobak Shahriari, Noah Siegel, Caglar Gulcehre, Nicolas Heess, et al. Critic regularized regression. *Advances in Neural Information Processing Systems*, 33:7768–7778, 2020.

Denis Yarats, David Brandfonbrener, Hao Liu, Michael Laskin, Pieter Abbeel, Alessandro Lazaric, and Lerrel Pinto. Don't change the algorithm, change the data: Exploratory data for offline reinforcement learning. *CoRR*, abs/2201.13425, 2022. URL `https://arxiv.org/abs/2201.13425`.