OFFLINE REINFORCEMENT LEARNING WITH IMPLICIT Q-LEARNING

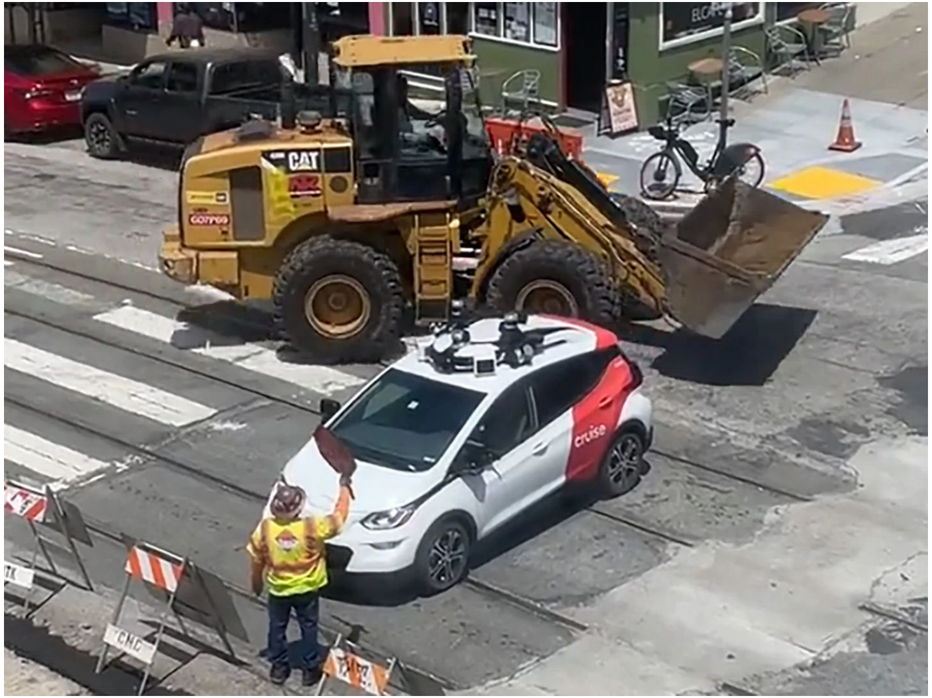
I. Kostrikov, A. Nair, & S. Levine

Presenter: Marlan McInnes-Taylor

Oct. 5, 2023

Motivation

- Environment exploration during training can be impractical or dangerous
    - Train policies using data collected by a behavior policy (Offline RL)
- Improvement over a behavior policy requires deviation
    - Estimate values for actions not present in the dataset



Main Problem

- Values of actions too different from those in a dataset are unlikely to be estimated accurately
- Prior methods:
  - Constrain resulting policy to limit deviation from behavior policy
  - Regularize learned value function
    - Assign low values to out-of-distribution actions
- Such methods trade policy improvement for limited misestimation
- Proposed work: approximate an upper expectile of the distribution over values w.r.t the distribution of dataset actions for each state

Context - Reinforcement Learning

- Formulated as a Markov decision process (**S**, **A**, $p_0(s)$, $p(s_0|s, a)$, $r(s, a)$, γ)
- **S**: space
- **A**: action space
- $p_0(s)$: distribution of initial states
- $p(s_0|s, a)$: environment dynamics
- $r(s, a)$: reward function
- γ: discount factor

Context - Reinforcement Learning

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

- Agent interacts with a MDP using a policy $\pi(a|s)$
- **Goal**: obtain a policy that maximizes the cumulative discounted returns

Problem Setting

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

- Modify the Temporal Difference loss $L_{TD}(\theta)$ to avoid out-of-dataset (unseen) action estimations
- D: a dataset
- r(s, a): reward function
- γ: discount factor
- Q_{θ_hat}(s', a'): target network
- Q_θ(s, a): parameterized Q-function
- policy $\pi(s) = \arg\max_a Q_\theta(s, a)$

Prior Work - “multi-step” approaches

Offline RL methods based on approximate dynamic programming.

- Constraints implemented as explicit density model
  - Wu et al., 2019; Fujimoto et al., 2019; Kumar et al., 2019
- Implicit divergence constraints
  - Nair et al., 2020; Wang et al., 2020; Peters & Schaal, 2007; Peng et al., 2019
- Supervised learning term in policy improvement objective
  - Fujimoto & Gu, 2021
- Direct Q-function regularization
  - Kostrikov et al., 2021; Kumar et al., 2020

Prior Work - “single-step” approaches

Methods which don’t use a value function, or learn that of the behaviour policy.

- Single policy iteration step + greedy policy extraction
  - Peng et al., 2019; Brandfonbrener et al., 2021
- Behavorial cloning objectives
  - Chen et al., 2021

Advantages:

- Simple to implement
- Effective on some benchmark tasks (MuJoCo locomotion in D4RL)

Disadvantages:

- Perform poorly on complex D4RL benchmarks requiring combination of suboptimal trajectories

Implicit Q-Learning

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

- Learn the value function given by L(θ) objective
- Evaluate the Q-function only on the state-action pairs in the dataset
  - Estimate maximum Q-value using actions in support of the data distribution
  - Reformulate L(θ) to use upper expectile prediction

Implicit Q-Learning

$$L_V(\psi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}[L_2^\tau(Q_{\hat{\theta}}(s,a) - V_\psi(s))]$$

- Introduce a separate value function that approximates an expectile only with respect to the action distribution

$$L_Q(\theta) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}}[(r(s,a) + \gamma V_\psi(s') - Q_\theta(s,a))^2]$$

Implicit Q-Learning

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

- The updated TD learning procedure estimates the optimal Q-function, but does not represent the corresponding policy
- Policy extraction performed by advantage weighted regression
- β: an inverse temperature
  - small values causes behavior similar to behavioral cloning
  - larger values attempt to recover the maximum of the Q-function

Algorithm Summary

Stage 1:

- Fit the value function and Q-function
- Gradient steps on $L_V(\psi)$ & $L_Q(\theta)$

Stage 2:

- Perform SGD on the policy extraction objective

Algorithm 1 Implicit Q-learning

Initialize parameters ψ, θ, $\hat{\theta}$, ϕ.
TD learning (IQL):
for each gradient step **do**
  $\psi \leftarrow \psi - \lambda_V \nabla_\psi L_V(\psi)$
  $\theta \leftarrow \theta - \lambda_Q \nabla_\theta L_Q(\theta)$
  $\hat{\theta} \leftarrow (1-\alpha)\hat{\theta} + \alpha\theta$
end for
Policy extraction (AWR):
for each gradient step **do**
  $\phi \leftarrow \phi - \lambda_\pi \nabla_\phi L_\pi(\phi)$
end for

Implicit Q-Learning - Theory

Section 4.4 and corresponding appendices present a series of lemmas and theorems which show that the IQL procedure correctly recovers the optimal value function under the given sampling constraints.

- General idea: apply and prove an upper bound on value expectation

- The τ hyperparameter results from introducing expectile regression
  - τ = 0.5 (SARSA, on-policy)
  - $\tau \rightarrow 1$ (Q-learning, off-policy)

Experimental Setup

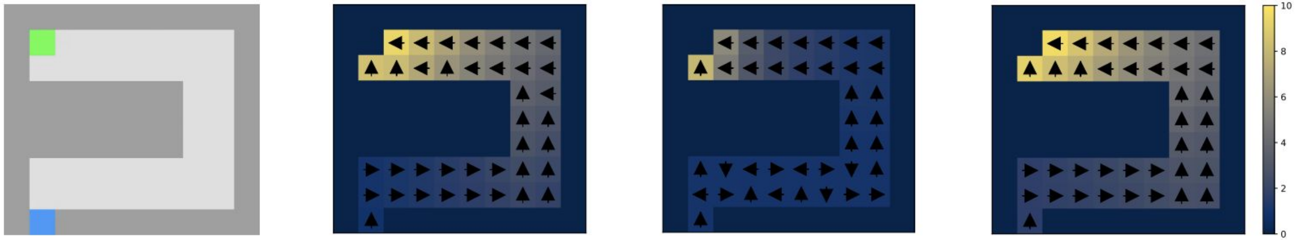
Perform comparative analysis between IQL, single-step methods, and multi-step methods.

1. Demonstrate benefits of multi-step methods over single-step methods
2. Compare IQL to state of the art single & multi-step methods on D4RL benchmark tasks
3. Compare IQL to other methods during online finetuning

Experimental Setup: One-step vs IQL

- U shaped maze w/ one start and one goal state
- Reward of 10 for entering the goal state and zero otherwise
- **Dataset**: 1 optimal trajectory and 99 trajectories with uniform random actions
- **Baseline**: Onepstep RL (Brandfonbrener et al., 2021; Wang et al., 2018)

Results: One-step vs IQL



(a) toy maze MDP (b) true optimal $V^\star$ (c) One-step Policy Eval. (d) IQL

- **One-step**
  - state rewards decay faster than true value function
  - resulting policy dominated by noise
- **IQL**
  - better propagates reward signal
  - closely approximates *V**

Experimental Setup: Offline RL Benchmarks

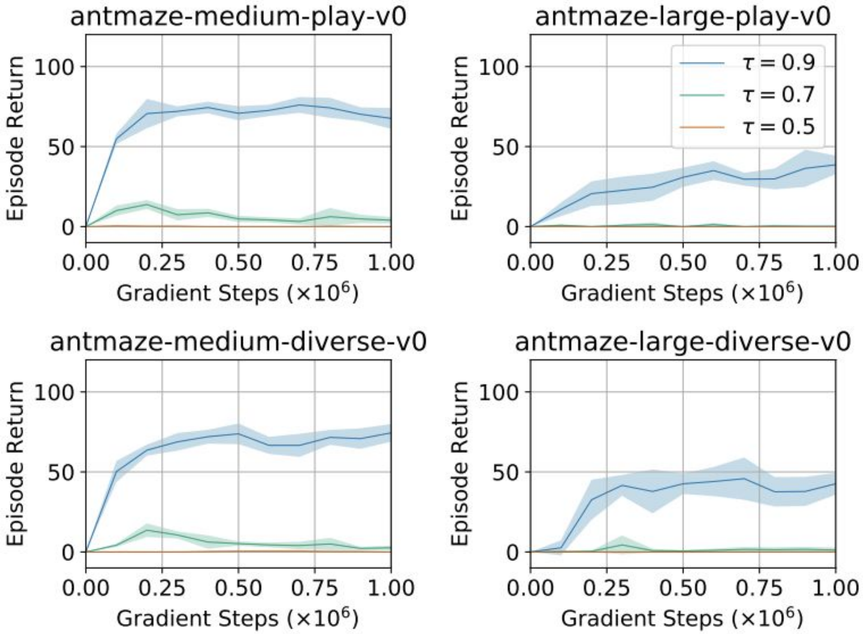
- **MuJoCo simulator**: Gym locomotion, Ant Maze, Adroit & Kitchen manipulation environments
- **Dataset**: D4RL
- **Baselines**:
  - One-step: Onestep RL (Brandfonbrener et al., 2021), Decision Transformers (Chen et al., 2021)
  - Multi-step: CQL (Kumar et al., 2020), TD3+BC (Fujimoto & Gu, 2021), and AWAC (Nair et al., 2020)
- **Metrics:** averaged normalized scores on MuJoCo tasks

Experimental Results: D4RL

Dataset	BC	10%BC	DT	AWAC	Onestep RL	TD3+BC	CQL	IQL (Ours)
halfcheetah-medium-v2	42.6	42.5	42.6	43.5	**48.4**	**48.3**	44.0	**47.4**
hopper-medium-v2	52.9	56.9	**67.6**	57.0	59.6	59.3	58.5	**66.3**
walker2d-medium-v2	75.3	75.0	74.0	72.4	**81.8**	83.7	72.5	78.3
halfcheetah-medium-replay-v2	36.6	40.6	36.6	40.5	38.1	**44.6**	**45.5**	**44.2**
hopper-medium-replay-v2	18.1	75.9	82.7	37.2	**97.5**	60.9	**95.0**	**94.7**
walker2d-medium-replay-v2	26.0	62.5	66.6	27.0	49.5	**81.8**	77.2	73.9
halfcheetah-medium-expert-v2	55.2	**92.9**	86.8	42.8	**93.4**	**90.7**	**91.6**	86.7
hopper-medium-expert-v2	52.5	**110.9**	**107.6**	55.8	103.3	98.0	**105.4**	91.5
walker2d-medium-expert-v2	**107.5**	**109.0**	**108.1**	74.5	**113.0**	**110.1**	**108.8**	**109.6**
locomotion-v2 total	466.7	**666.2**	**672.6**	450.7	**684.6**	**677.4**	**698.5**	**692.4**
antmaze-umaze-v0	54.6	62.8	59.2	56.7	64.3	78.6	74.0	**87.5**
antmaze-umaze-diverse-v0	45.6	50.2	53.0	49.3	60.7	71.4	**84.0**	62.2
antmaze-medium-play-v0	0.0	5.4	0.0	0.0	0.3	10.6	61.2	**71.2**
antmaze-medium-diverse-v0	0.0	9.8	0.0	0.7	0.0	3.0	53.7	**70.0**
antmaze-large-play-v0	0.0	0.0	0.0	0.0	0.0	0.2	15.8	**39.6**
antmaze-large-diverse-v0	0.0	6.0	0.0	1.0	0.0	0.0	14.9	**47.5**
antmaze-v0 total	100.2	134.2	112.2	107.7	125.3	163.8	303.6	**378.0**
total	566.9	800.4	784.8	558.4	809.9	841.2	1002.1	**1070.4**
kitchen-v0 total	**154.5**	-	-	-	-	-	144.6	**159.8**
adroit-v0 total	104.5	-	-	-	-	-	93.6	**118.1**
total+kitchen+adroit	825.9	-	-	-	-	-	1240.3	**1348.3**
runtime	10m	10m	960m	20m	$\approx 20\text{m}^*$	20m	80m	20m

Results Analysis

- The τ hyper parameter is crucial to effective performance on complex tasks
- Baseline and IQL methods have similar performance on easier tasks
- IQL is computationally faster than baseline methods



Critique

- The importance of the τ hyperparameter results in IQL's effectiveness being coupled to hyperparameter tuning procedures.

Extended Readings

- **Kostrikov, Ilya, Ashvin Nair, and Sergey Levine**, "*IDQL: Implicit Q-Learning as an Actor-Critic Method with Diffusion Policies*." arXiv preprint arXiv:2304.10573 (2023).
- **Snell, Charlie, et al.** "*Offline rl for natural language generation with implicit language q learning.*" arXiv preprint arXiv:2206.11871 (2022).
- **Chitnis, Rohan, et al.** "*IQL-TD-MPC: Implicit Q-Learning for Hierarchical Model Predictive Control.*" arXiv preprint arXiv:2306.00867 (2023).

Summary

- **Problem**: Developing an offline RL algorithm which avoids out-of-dataset action value estimation while still performing multi-step dynamic programming
    - Value estimation of out-of-dataset actions is frequently inaccurate
- **Prior work** primarily focuses on constraining distributional drift, regularizing out-of-distribution sample estimates, or avoids value estimates entirely

Summary

- **Insight**: fitting the Q-function to estimate state conditional expectiles correctly represents the maximum Q-value over actions within the data distribution
- **Results**: The modified optimization objective can avoid out-of-dataset action estimation, improve upon a behavior policy, outperform or match existing offline RL algorithms, while being computationally more efficient.

Discussion

- How might we procedurally estimate a ‘good’ value for the τ hyperparameter?