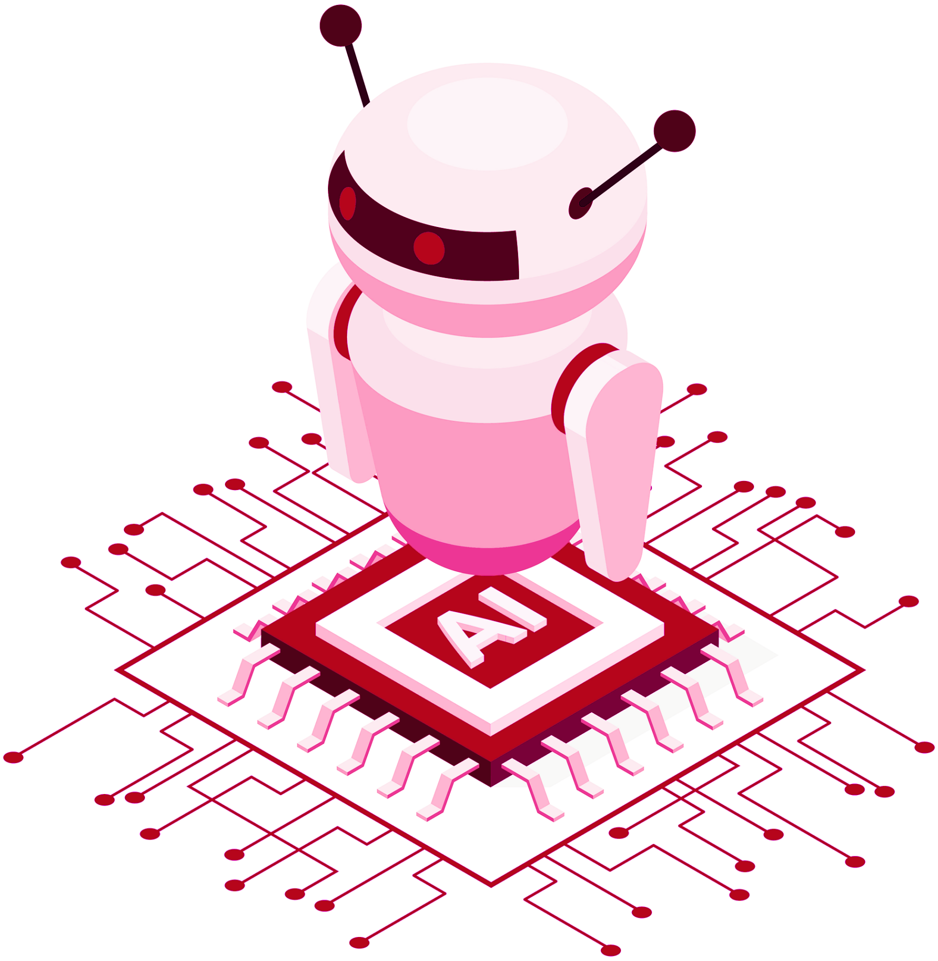

Deep Reinforcement Learning

Professor Mohammad Hossein Rohban

Solution for Homework 12:

Offline Methods

By:

Amir Kooshan Fattah Hesari

401102191

Spring 2025

Contents

1 Part 1 [60-points]

1. Considering the Bellman update, explain with reasoning why value estimation suffers from overestimation in the offline framework. [10-points]

$$Q(s,a) \leftarrow r(s,a) + \mathbb{E}_{a' \sim r_{new}}[Q(s',a')]$$

Answer

Offline RL algorithms based on the given Bellman backup formula suffer from action distribution shift during training, because the target values for Bellman backups in policy evaluation use actions sampled from learned policy , $\hat{\pi}^k$ but the Q-function is trained only on actions sampled from the behavior policy that produced the dataset $\mathcal{D}, \pi_\beta$. Since π is trained to maximize Q-values, it may be biased towards out-of-distribution (OOD) actions with high Q-values.

2. One of the solutions to address the overestimation problem in the offline framework is CQL, whose objective function for computing the value is given below. Explain the role of each of the four terms in this objective function. [20-points]

$$\begin{aligned}\hat{Q}^T = \arg\min_Q \max_\mu & \alpha \mathbb{E}_{s\sim D, a\sim\mu(a|s)}[Q(s,a)] \\ & - \alpha\mathbb{E}_{(s,a)\sim D}[Q(s,a)] \\ & - \mathbb{E}_{s\sim D}[\mathcal{H}(\mu(\cdot|s))] \\ & + \mathbb{E}_{(s,a,s')\sim D}\left[(Q(s,a) - (r(s,a) + \mathbb{E}[Q(s',a')]))^2\right]\end{aligned}$$

Answer

From the original paper we have that when we want to maximize the current Q-function iterate, we could choose $\mu(\mathbf{a}|\mathbf{s})$ to approximate the policy.
This gives rise to the following family of optimization problems over $\mu(\mathbf{a}|\mathbf{s})$:

$$\min_Q \max_\mu \alpha\left(\mathbb{E}_{s\sim\mathcal{D}, \mathbf{a}\sim\mu(\mathbf{a}|\mathbf{s})}\left[Q(\mathbf{s},\mathbf{a})\right] - \mathbb{E}_{s\sim\mathcal{D}, \mathbf{a}\sim\hat{\pi}_\beta(\mathbf{a}|\mathbf{s})}\left[Q(\mathbf{s},\mathbf{a})\right]\right) + \frac{1}{2}\mathbb{E}_{\mathbf{s},\mathbf{a},\mathbf{s}'\sim\mathcal{D}}\left[\left(Q(\mathbf{s},\mathbf{a}) - \hat{\mathcal{B}}^{\pi_k}\hat{Q}^k(\mathbf{s},\mathbf{a})\right)^2\right] + \mathcal{R}(\mu)$$

The roles of each of the four terms are as follows:

(a) $\alpha\mathbb{E}_{s\sim D,\, a\sim\mu(a|s)}[Q(s,a)]$ **(Conservatism Penalty):**
This term penalizes the Q-function for having high values on actions a that might not have been seen in the dataset D (i.e., out-of-distribution actions). By maximizing over $\mu(a|s)$, this term finds the distribution over actions that would most overestimate $Q(s,a)$ and penalizes it, encouraging the learned Q-values to be conservative, especially for OOD actions.

(b) $-\alpha\mathbb{E}_{(s,a)\sim D}[Q(s,a)]$ **(In-Dataset Correction):**
This term rewards high Q-values on actions that are *in* the dataset, offsetting the penalty from the previous term for those actions that have been observed. It ensures the Q-function is not overly pessimistic for in-distribution actions, maintaining accurate value estimation for what has actually been seen.

(a) $-\mathbb{E}_{s\sim D}[\mathcal{H}(\mu(\cdot|s))]$ **(Entropy Regularization):** This regularizes the auxiliary distribution $\mu(a|s)$, encouraging it to have high entropy (i.e., to be broad and exploratory). This prevents μ from collapsing onto a single action and instead ensures the conservatism penalty (first term) robustly explores the action space, making the Q-learning process more stable and reliable.

(b) $+\mathbb{E}_{(s,a,s')\sim D}\left[(Q(s,a)-(r(s,a)+\mathbb{E}_{a'\sim\pi(\cdot|s')}[Q(s',a')]))^2\right]$ **(Bellman Error):** This is the standard Bellman error loss, which ensures that the Q-function remains consistent with the expected return dictated by the reward and transition dynamics. It is the typical temporal-difference regression used in Q-learning, anchoring the Q-values to valid, learned returns from the environment.

3. Rewrite the optimization problem from part 2 as a minimization-only problem. [20-points]

Answer

(a) Write the maximization:

$$\max_{\mu(\cdot|s)}\sum_a \mu(a|s)Q(s,a)+\mathcal{H}(\mu(\cdot|s))$$

(b) Write the Lagrangian (with multiplier λ):

$$\mathcal{L}=\sum_a \mu(a|s)\big(Q(s,a)-\log\mu(a|s)\big)+\lambda\left(1-\sum_a \mu(a|s)\right)$$

(c) Take derivative w.r.t. $\mu(a|s)$ and set to zero:

$$Q(s,a)-\log\mu(a|s)-1-\lambda=0 \implies \mu(a|s)=\exp(Q(s,a)-1-\lambda)$$

(d) Enforce normalization:

$$\sum_a \mu(a|s)=1 \implies \mu^*(a|s)=\frac{\exp(Q(s,a))}{\sum_{a'}\exp(Q(s,a'))}$$

(e) Plug back to get:

$$\sum_a \mu^*(a|s)Q(s,a)+\mathcal{H}(\mu^*(\cdot|s))=\log\sum_a\exp(Q(s,a))$$

(f) Thus, the minimization-only CQL objective is:

$$\hat{Q}^T=\arg\min_Q\Bigg[\alpha\,\mathbb{E}_{s\sim D}\left[\log\sum_a\exp(Q(s,a))\right]-\alpha\mathbb{E}_{(s,a)\sim D}[Q(s,a)] \\ +\mathbb{E}_{(s,a,s')\sim D}\left(Q(s,a)-\left(r(s,a)+\mathbb{E}_{a'\sim\pi(\cdot|s')}[Q(s',a')]\right)\right)^2\Bigg]$$

4. To apply this method in model-based reinforcement learning, what changes are needed in the objective function? Rewrite the new objective function. [10-points]

Answer

To apply CQL in model-based reinforcement learning (MBRL), we need to modify the objective function to use samples generated by the learned dynamics model, instead of (or in addition to) samples from the real environment. This means that for the Bellman error term, the next state s' and reward r are sampled or predicted from the learned models $\hat{T}$ and $\hat{r}$.
The new objective function becomes:

$$\begin{aligned}\hat{Q}^T = \arg\min_Q \Big[\, \alpha\, \mathbb{E}_{s\sim D}\big[\log\sum_a \exp(Q(s,a))\big] - \alpha\, \mathbb{E}_{(s,a)\sim D}[Q(s,a)] \\ + \mathbb{E}_{(s,a)\sim D,\, s'\sim\hat{T}(\cdot|s,a)}\Big(Q(s,a) - \big(\hat{r}(s,a) + \mathbb{E}_{a'\sim\pi(\cdot|s')}[Q(s',a')]\big)\Big)^2 \,\Big]\end{aligned}$$

In this new objective:

- s' is sampled from the learned transition model $\hat{T}(\cdot|s,a)$.
- $\hat{r}(s,a)$ is the reward predicted by the learned reward model.
- All other terms remain the same as in the original CQL objective, but now the Bellman backup uses the model-generated samples.

References

[1] Cover image designed by freepik