多智能体系统与强化学习

主讲人：高阳、杨林、杨天培

https://reinforcement-learning-2025.github.io/

第七讲：离线强化学习

从实际环境到虚拟环境

杨　林

大 纲

离线强化学习

批量限制Q学习（BCQ）

保守Q学习（CQL）

大 纲

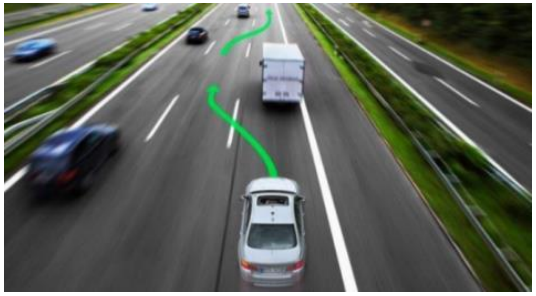
离线强化学习

批量限制Q学习（BCQ）

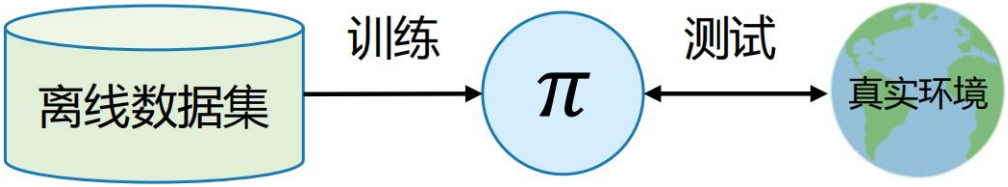
保守Q学习（CQL）

离线强化学习概念

- 动机：在真实环境中从零开始训练一个强化学习智能体往往不可取
  - ✓ 风险较高，例如无人驾驶归控、智能医疗等
  - ✓ 十分昂贵，例如机器人控制、推荐系统等



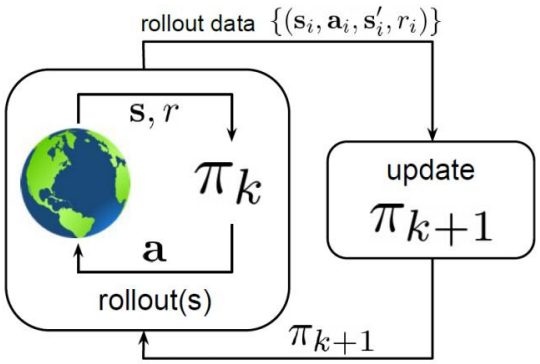
- 离线强化学习：在一个给定的离线数据集上直接训练出智能体策略，训练的过程中，智能体不得和环境做交互



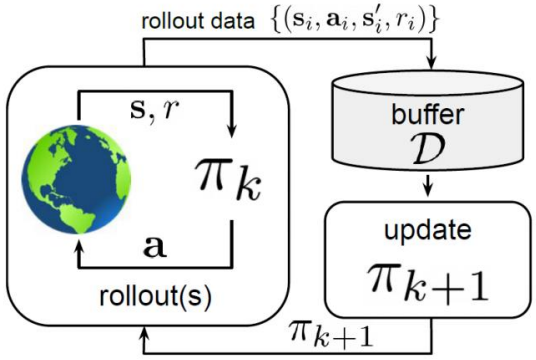
- 离线强化学习有潜力大大扩宽强化学习落地的范围

离线强化学习范式

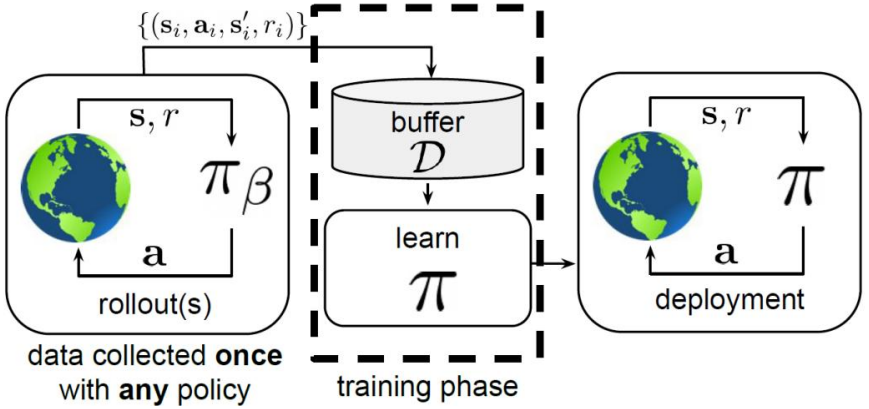
(a) On-policy



(b) off-policy



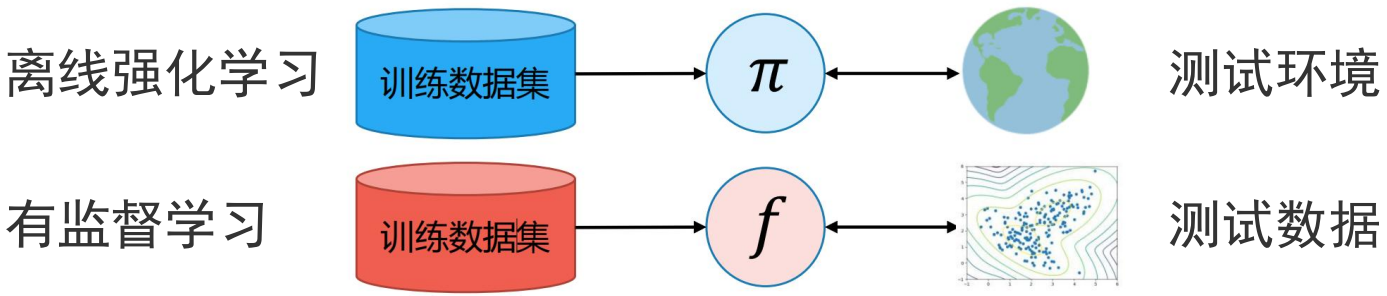
(c) 离线强化学习



- 训练的过程中与环境交互：
  - ✓ 在线策略学习与离线策略学习的智能体可以和环境交互
  - ✓ 离线强化学习的智能体不和环境做交互
- 训练数据是否来自别的策略交互经验：
  - ✓ Yes - 离线强化学习和离线策略学习
  - ✓ No - 在线强化学习

离线强化学习与模仿学习

- 离线强化学习让强化学习更像有监督学习



均依赖静态数据、避免在线学习的风险

- 不同点
  - ✓ 奖励信号需求
  - ✓ 数据需求（离线强化学习数据包含次优或随机行为）

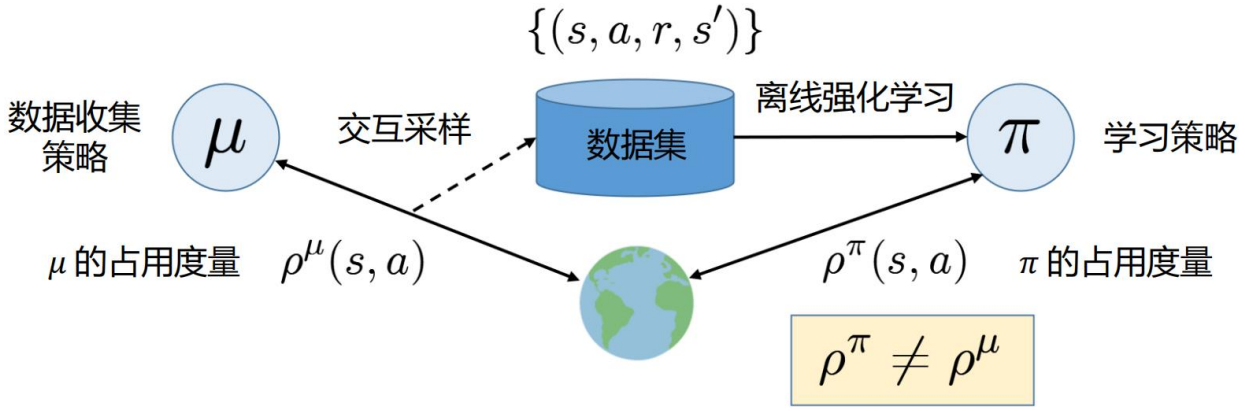
离线强化学习的主要问题和挑战

- 离线强化学习面临的最重要的挑战是分布偏移（Distribution Shift）
- 数据集分布和当前策略的分布不一致导致外推误差（Extrapolation Error）
  - ✓ 智能体如果涉足到了从没有见过的、远离数据集的状态动作对，怎么办？

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a'))$$

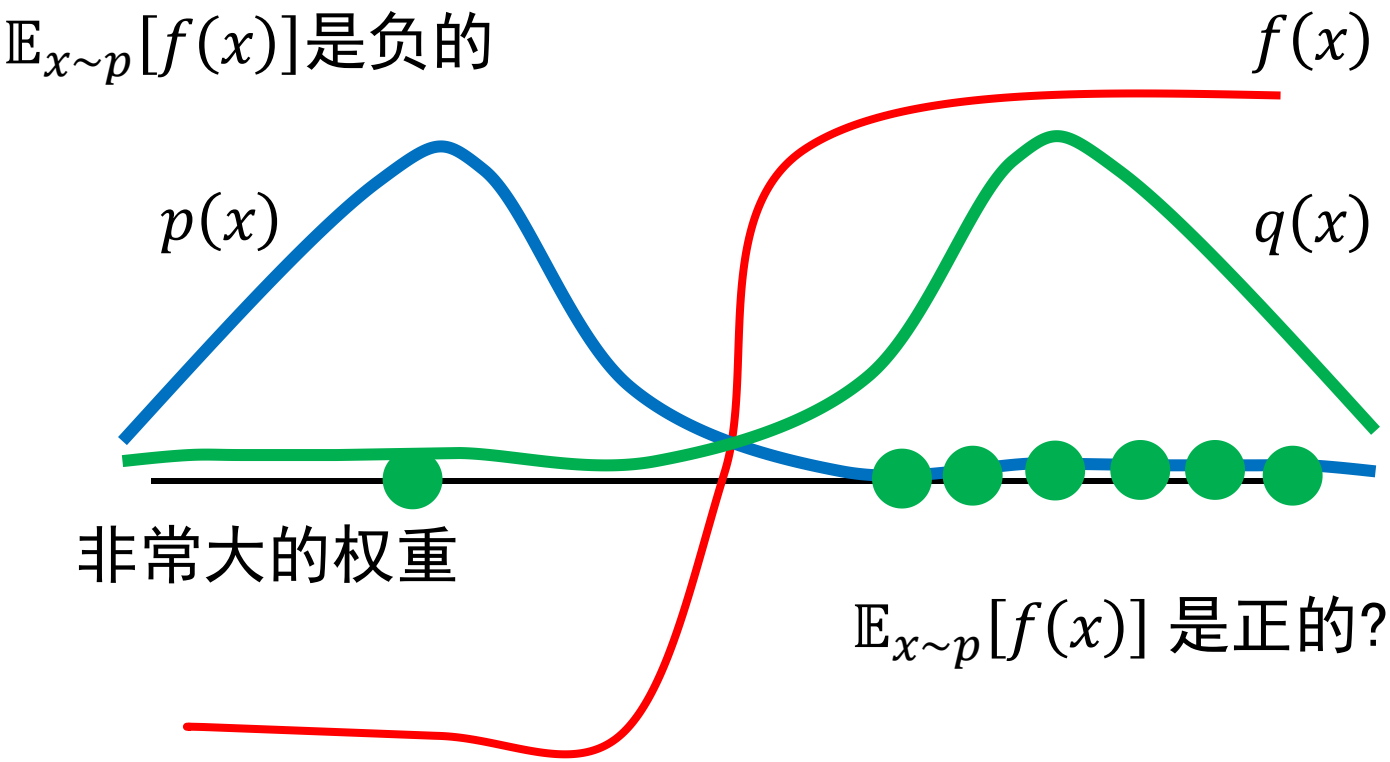
  - ✓ 外推误差：策略尝试数据中未见的动作， Q函数的预测会严重脱离真实值

如果a'是一个分布外的动作，怎么处理？



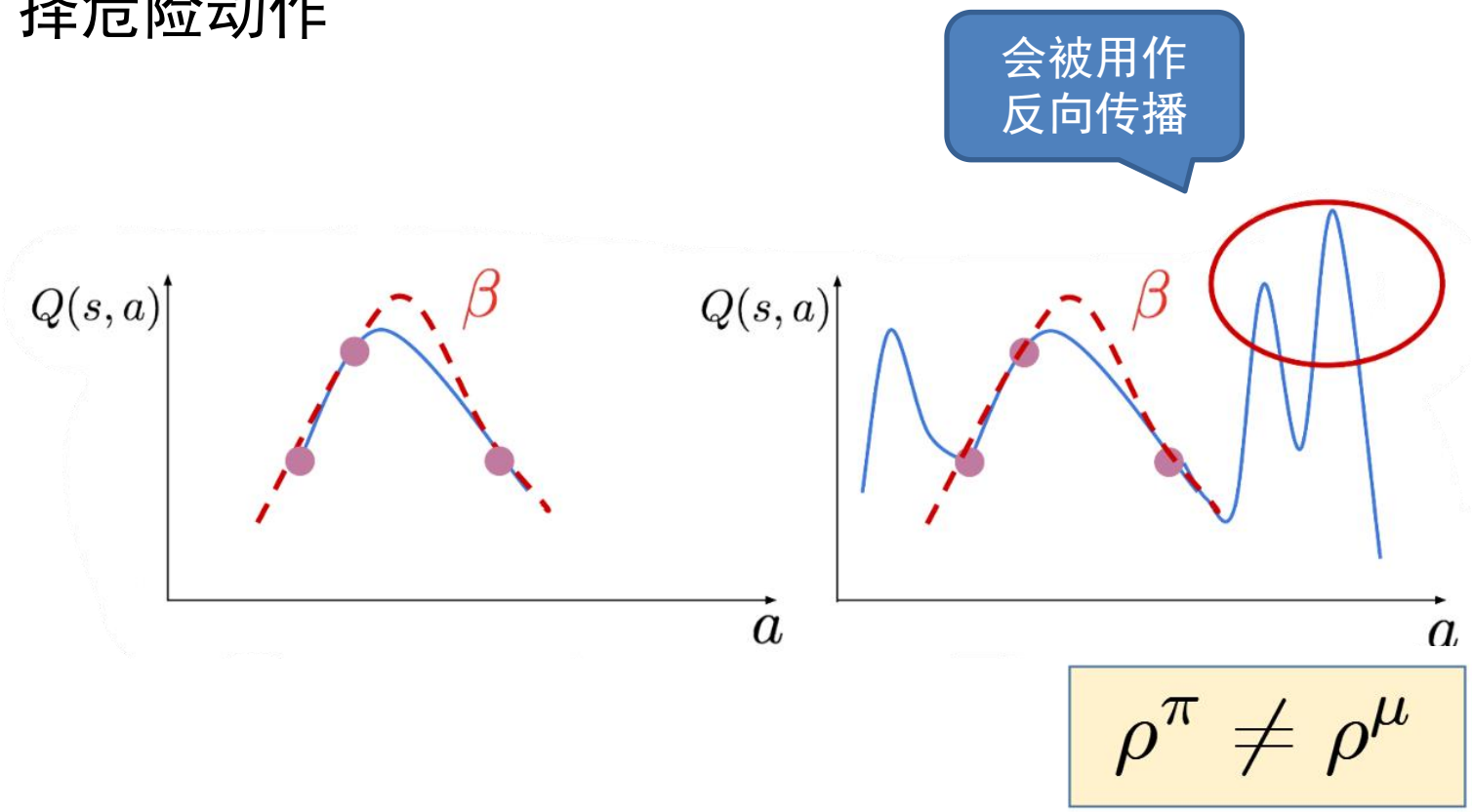
Q学习中的分布偏移

- 策略与数据分布偏离



Q学习中的外推误差

- OOD 区域无法正确推断奖励，为了追求虚假的高奖励而选择危险动作



大 纲

离线强化学习

批量限制Q学习（BCQ）

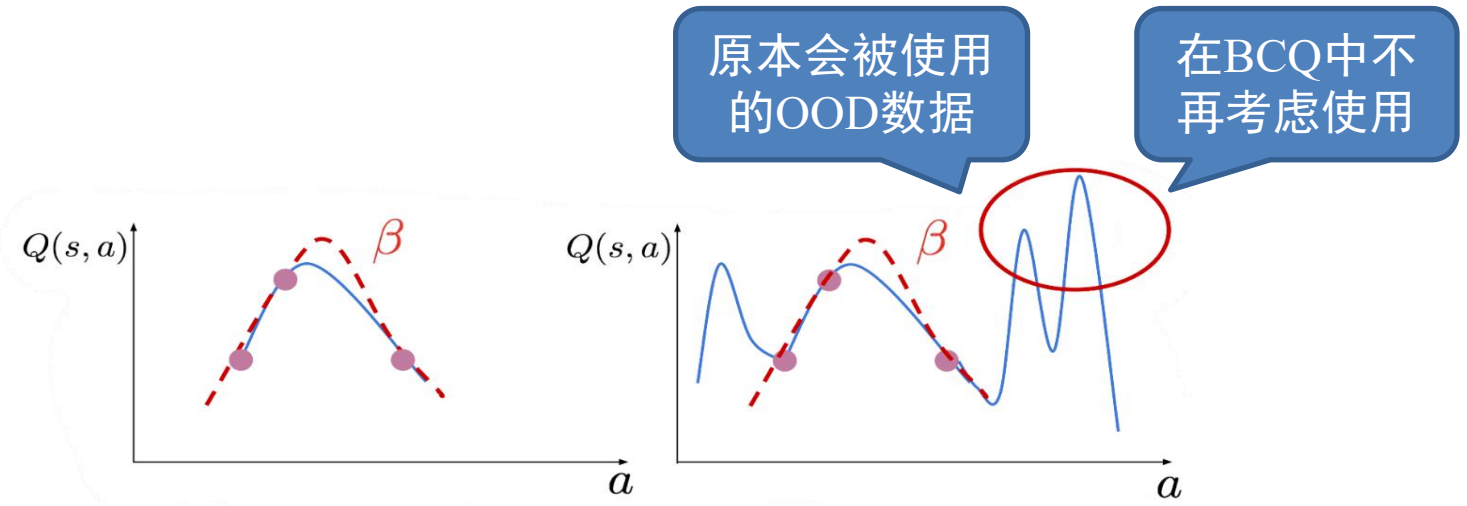
保守Q学习（CQL）

BCQ原理

- 对于经典表格型强化学习，BCQ的基本思路是仅仅使用在数据集支撑上的目标Q值做时序差分的计算

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha(r + \gamma \max_{a' \text{ s.t. } (s',a') \in \mathcal{B}} Q(s',a'))$$

仅仅考虑在数据集支撑上的(s', a')



BCQ：批量限制Q学习

- 对于更广泛的连续动作强化学习设置，BCQ的基本思路是“仅仅使用在数据集支撑上的目标Q值做时序差分的计算”可以如下实现：
  - ✓ 使用一个生成模型，如变分自动编码器VAE，来生成距离数据集较近的状态动作对

$$\pi(s) = \underset{a_i+\xi_\phi(s,a_i,\Phi)}{\arg\max} Q_\theta(s, a_i + \xi_\phi(s, a_i, \Phi))$$

在$[-\Phi, +\Phi]$的扰动

其中$\{a_i \sim G_\omega(s)\}_{i=1}^{n}$，生成模型，如变分自动编码器VAE

- 对于 n 和 Φ 的选择，形成模仿学习和强化学习之间的一个权衡
  - ✓ n 和 Φ 越小，越接近模仿学习，策略性能可能不好
  - ✓ n 和 Φ 越大，越接近强化学习，但容易出OOD问题

BCQ伪代码

Algorithm 1 基于批量约束的 Q 学习（Batch-Constrained Q-learning, BCQ）

1: **输入:** 经验批量数据 $\mathcal{B}$, 训练轮数 T, 目标网络更新率 τ, 小批量大小 N, 最大扰动量 Φ, 采样动作数 n, 最小加权系数 λ

2: 初始化 Q 网络 $Q_{\theta_1}, Q_{\theta_2}$，扰动网络 ξ_ϕ，以及 VAE $G_\omega = \{E_{\omega_1}, D_{\omega_2}\}$，参数随机初始化为 $\theta_1, \theta_2, \phi, \omega$，目标网络 $Q_{\theta'_1}, Q_{\theta'_2}, \xi_{\phi'}$ 设为 $\theta'_1 \leftarrow \theta_1, \theta'_2 \leftarrow \theta_2, \phi' \leftarrow \phi$

3: **for** $t = 1$ to T **do**

4: 从批量数据 $\mathcal{B}$ 中随机采样 N 个转换样本 (s, a, r, s')

5: 计算 VAE 编码:

$$\mu, \sigma = E_{\omega_1}(s, a), \quad \tilde{a} = D_{\omega_2}(s, z), \quad z \sim \mathcal{N}(\mu, \sigma)$$

6: 更新 VAE 参数:

$$\omega \leftarrow \text{argmin}_\omega \sum (a - \tilde{a})^2 + D_{\text{KL}}(\mathcal{N}(\mu, \sigma) \| \mathcal{N}(0, 1))$$

VAE做模仿学习

7: 采样 n 个候选动作:

$$\{a_i \sim G_\omega(s')\}_{i=1}^n$$

BCQ伪代码

8: 对每个采样动作进行扰动：

$$\{a_i = a_i + \xi_\phi(s', a_i, \Phi)\}_{i=1}^{n}$$

9: 计算值目标 y：

$$y = r + \gamma \max_{a_i} \left[\lambda \min_{j=1,2} Q_{\theta'_j}(s', a_i) + (1-\lambda) \max_{j=1,2} Q_{\theta'_j}(s', a_i) \right]$$

乐观与保守估计之间的平衡

10: 更新 Q 网络：

$$\theta \leftarrow \text{argmin}_\theta \sum (y - Q_\theta(s,a))^2$$

11: 更新扰动网络：

$$\phi \leftarrow \text{argmax}_\phi \sum Q_{\theta_1}(s, a + \xi_\phi(s, a, \Phi)), \quad a \sim G_\omega(s)$$

扰动函数ξ像是actor

12: 更新目标网络：

$$\theta'_i \leftarrow \tau\theta + (1-\tau)\theta'_i, \quad \phi' \leftarrow \tau\phi + (1-\tau)\phi'$$

13: **end for**

大 纲

离线强化学习

批量限制Q学习（BCQ）

保守Q学习（CQL）

CQL原理

- 学习一个保守的、可作为价值下界的 Q 函数，以避免在OOD数据上的过高估计
  - ✓ 对于一个新的学习策略 μ ，增加一个其遇见数据上的 Q 函数的惩罚

$$\widehat{Q}^{k+1} \leftarrow \arg\min_{Q} \alpha(\mathbb{E}_{s\sim\mathcal{D},a\sim\mu(a|s)}[Q(s,a)] - \mathbb{E}_{s\sim\mathcal{D},a\sim\widehat{\pi}(a|s)}[Q(s,a)]) + \frac{1}{2}\mathbb{E}_{s,a,s'\sim\mathcal{D}}[(Q(s,a) - \widehat{\mathcal{B}}^{\pi}\widehat{Q}^{k}(s,a))^2]$$

最大化分布内动作的估值

最小化分布外动作的估值

最小化MSE

CQL：保守Q学习

- 使用max μ操作来估计当前的学习策略 π，为了增加覆盖度，加上正则

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

$$\min_Q \alpha\mathbb{E}_{s\sim\mathcal{D}}[\log\sum_a \exp(Q(s,a)) - \mathbb{E}_{a\sim\hat{\pi}(a|s)}[Q(s,a)]] + \frac{1}{2}\mathbb{E}_{s,a,s'\sim\mathcal{D}}[(Q(s,a) - \hat{\mathcal{B}}^\pi\hat{Q}^k(s,a))^2] \quad (\mathrm{CQL}(\mathcal{H}))$$

最大值逼近

- 经验上使用μ 和均匀分布的KL散度作为正则项的实现

$$\mathcal{R}(\mu) = -\mathrm{D}_{\mathrm{KL}}(\mu, \mathrm{Unif(a)})$$

CQL伪代码

Algorithm 1 保守 Q 学习（两种变体）

1: 初始化 Q 函数 Q_θ，可选初始化策略 π_ϕ

2: **for** 步骤 t 从 1 到 N **do**

3: 使用函数 $\mathrm{CQL}(\mathcal{H})$，通过 G_Q 次梯度步训练 Q 函数（Q 学习用 $\mathcal{B}^*$，行动者-评论家用 $\mathcal{B}^{\pi_{\phi_t}}$）：

4: $\theta_t := \theta_{t-1} - \eta_Q \nabla_\theta \mathrm{CQL}(\mathcal{R})(\theta)$

5: **if** 使用行动者-评论家架构 **then**

6: 通过 G_π 次梯度步改进策略 π_ϕ，采用 SAC 风格的熵正则化：

7: $\phi_t := \phi_{t-1} + \eta_\pi \nabla_\phi \mathbb{E}_{\mathrm{s}\sim\mathcal{D},\mathrm{a}\sim\pi_\phi(\cdot|\mathrm{s})} \left[Q_\theta(\mathrm{s},\mathrm{a}) - \log \pi_\phi(\mathrm{a} \mid \mathrm{s}) \right]$

8: **end if**

9: **end for**

使用策略π采样

熵项：控制动作的收敛性

求期望得到价值函数

- CQL 可以直接做基于价值函数的训练
- 如果需要做策略训练，则在训练价值函数 Q 的同时，使用Soft AC算法训练出策略 π

CQL：保守Q学习的实验

- 在多个 Gym 环境和不同的数据集采样设置下，CQL 几乎都能取得最好的策略性能

Task Name	SAC	BC	BEAR	BRAC-p	BRAC-v	CQL($\mathcal{H}$)
halfcheetah-random	30.5	2.1	25.5	23.5	28.1	**35.4**
hopper-random	**11.3**	9.8	9.5	**11.1**	**12.0**	**10.8**
walker2d-random	4.1	1.6	**6.7**	0.8	0.5	**7.0**
halfcheetah-medium	-4.3	36.1	38.6	**44.0**	**45.5**	**44.4**
walker2d-medium	0.9	6.6	33.2	72.7	**81.3**	79.2
hopper-medium	0.8	29.0	47.6	31.2	32.3	**58.0**
halfcheetah-expert	-1.9	**107.0**	**108.2**	3.8	-1.1	104.8
hopper-expert	0.7	**109.0**	**110.3**	6.6	3.7	**109.9**
walker2d-expert	-0.3	125.7	106.1	-0.2	-0.0	**153.9**
halfcheetah-medium-expert	1.8	35.8	51.7	43.8	45.3	**62.4**
walker2d-medium-expert	1.9	11.3	10.8	-0.3	0.9	**98.7**
hopper-medium-expert	1.6	**111.9**	4.0	1.1	0.8	**111.0**
halfcheetah-random-expert	53.0	1.3	24.6	30.2	2.2	**92.5**
walker2d-random-expert	0.8	0.7	1.9	0.2	2.7	**91.1**
hopper-random-expert	5.6	10.1	10.1	5.8	11.1	**110.5**
halfcheetah-mixed	-2.4	38.4	36.2	**45.6**	**45.9**	**46.2**
hopper-mixed	3.5	11.8	25.3	0.7	0.8	**48.6**
walker2d-mixed	1.9	11.3	10.8	-0.3	0.9	**26.7**

谢 谢 ！