计算机科学
COMPUTER SCIENCE

基于不确定性权重的保守Q学习离线强化学习算法

王天久, 刘全, 乌兰

引用本文

王天久, 刘全, 乌兰. 基于不确定性权重的保守Q学习离线强化学习算法[J]. 计算机科学, 2024, 51(9): 265-272.

WANG Tianjiu, LIU Quan, WU Lan. Offline Reinforcement Learning Algorithm for Conservative Q-learning Based on Uncertainty Weight [J]. Computer Science, 2024, 51(9): 265-272.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.230700151

基于不确定性权重的保守Q学习离线强化学习算法

王天久[1]　刘　全[1,2]　乌　兰[1]
1 苏州大学计算机科学与技术学院　江苏 苏州 215006
2 苏州大学江苏省计算机信息处理技术重点实验室　江苏 苏州 215006
(20214227063@stu.suda.edu.cn)

摘　要　离线强化学习(Offline RL)中,智能体不与环境交互而是从一个固定的数据集中获得数据进行学习,这是强化学习领域研究的一个热点。目前多数离线强化学习算法对策略训练过程进行保守正则化处理,训练策略倾向于选择存在于数据集中的动作,从而解决离线强化学习中对数据集分布外(OOD)的状态-动作价值估值错误的问题。保守Q学习算法(CQL)通过值函数正则赋予分布外状态-动作较低的价值来避免该问题。然而,由于该算法正则化过于保守,数据集内的分布内状态-动作也被赋予了较低的价值,难以达到训练策略选择数据集中动作的目的,因此很难学习到最优策略。针对该问题,提出了一种基于不确定性权重的保守Q学习算法(UWCQL)。该方法引入不确定性计算,在保守Q学习算法的基础上添加不确定性权重,对不确定性高的动作给予更高的保守权重,使得策略能更合理地选择数据集分布内的状态-动作。将UWCQL算法应用于D4RL的MuJoCo数据集中进行了实验,实验结果表明,UWCQL算法具有更好的性能表现,从而验证了算法的有效性。

关键词:离线强化学习;深度强化学习;强化学习;保守Q学习;不确定性
中图分类号　TP181

Offline Reinforcement Learning Algorithm for Conservative Q-learning Based on Uncertainty Weight

WANG Tianjiu[1],LIU Quan[1,2] and WU Lan[1]
1 School of Computer and Technology,Soochow University,Suzhou,Jiangsu 215006,China
2 Provincial Key Laboratory for Computer Information Processing Technology,Soochow University,Suzhou,Jiangsu 215006,China

Abstract　Offline reinforcement learning,in which the agent learns from a fixed dataset without interacting with the environment,is a current hot spot in the field of reinforcement learning. Many offline reinforcement learning algorithms try to regularize value function to force the agent choose actions in the given dataset. The conservative Q-learning(CQL) algorithm avoids this problem by assigning a lower value to the OOD(out of distribution) state-action pairs through the value function regularization. However,the algorithm is too conservative to recognize the state-action pairs outside the distribution precisely,and therefore it is difficult to learn the optimal policy. To address this problem,the uncertainty-weighted conservative Q-learning algorithm(UWCQL) is proposed by introducing an uncertainty mechanism during training. The UWCQL adds uncertainty weight to the CQL regularization term,assigns higher conservative weight to actions with high uncertainty to ensure that the algorithm can more effectively train the agent to choose proper state-action pairs in the dataset. The effectiveness of UWCQL is verified by applying it to the D4RL MuJoCo dataset,along with the best offline reinforcement learning algorithms,and the experimental results show that the UWCQL algorithm has better performance.

Keywords　Offline reinforcement learning,Deep reinforcement learning,Reinforcement learning,Conservative Q-learning,Uncertainty

到稿日期:2023-07-20　返修日期:2023-11-23
基金项目:国家自然科学基金(61772355,61702055,61876217,62176175);新疆维吾尔自治区自然科学基金(2022D01A238);江苏高校优势学科建设工程资助项目
This work was supported by the National Natural Science Foundation of China(61772355,61702055,61876217,62176175),National Natural Science Foundation of Xinjiang Uygur Autonomous Region(2022D01A238) and Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions(PAPD).
通信作者:刘全(quanliu@suda.edu.cn)

1 引言

强化学习(Reinforcement Learning,RL)[1]的基本思想是智能体与环境交互,获得数据以进行学习。其利用价值函数对动作和状态进行评估,不断调整策略反复迭代,直到智能体能够最大化从环境中得到的累积奖赏值,学习到最优策略。强化学习是目前机器学习领域的研究热点之一,已被广泛应用在机器人控制、优化调度,游戏博弈等领域[2]。

然而,强化学习要求智能体通过在线的方式直接与环境进行交互,这一特点给强化学习的广泛应用带来了阻碍。例如,在无人驾驶、机器人控制和医疗等领域的应用实践中,智能体直接与环境交互获取经验的方式通常代价很高,且存在危险性[3]。此外,在很多复杂任务中更倾向于使用已收集到的数据进行学习。如果强化学习能和监督学习一样使用已经大量收集到的数据对智能体进行训练,将大大拓展强化学习的应用领域。离线强化学习(Offline Reinforcement Learning,Offline RL)是强化学习领域的一个新兴研究方向,在离线强化学习中,智能体不需要与环境直接进行交互,而是从一个静态的数据集中获得数据进行学习。离线强化学习直接利用已有的数据进行学习,能够避免传统强化学习中智能体需要不断与环境交互带来的风险。

另一方面,因为智能体不与环境进行交互,离线强化学习也存在一些弊端。使用离线的方式训练智能体会带来过拟合和外推误差(Extrapolation Error)[4]的问题:在进行策略评估时,智能体会错误地高估不存在于数据集中的分布外(Out of Distribution,OOD)状态-动作的价值,从而影响策略改进,导致智能体倾向于选择执行价值被过高估计的动作,使得训练效果很差。因此,离线强化学习相关的工作大多专注于解决高估分布外状态-动作的问题。目前已有的离线强化学习算法可分为以下几类:

1)通过正则化对学习策略进行限制,训练智能体仅选择限制在数据集内的动作的。Fujimoto 等受到变分自编码器[5](Variational Auto-Encoder,VAE)的启发,将变分自编码器与离线强化学习相结合,隐式地对策略进行限制,提出了数据集限制 Q 学习算法(Batch Constraint Q-Learning,BCQ)。Kumar 等[6]使用最大均值差异(Maximum Mean Discrepancy,MMD)作为策略更新公式的正则项,提出了自举减少错误累积算法(Bootstrapping Error Accumulation Reduction,BEAR)算法。Fujimoto 等[7]通过对双延迟深度确定性策略梯度(Twin Delayed Deep Deterministic Policy Gradient,TD3)添加行为克隆正则项的方式,提出了双延迟深度确定性策略梯度+行为克隆(Twin Delayed Deep Deterministic Policy Gradient + Behavior Cloning,TD3+BC)算法[8]限制策略。

2)限制价值函数,给予分布外状态-动作较低的价值。Kumar 等[9]提出的保守 Q 学习算法(Conservative Q-Learning,CQL)通过某先验策略分布限制分布外状态-动作的价值函数值大小。Lyu 等[10]在 CQL 算法的基础上对分布外状态-动作价值进行了泛化,提出了柔和保守 Q 学习算法(Mildly Conservative Q-Learning,MCQ)。

3)使用基于不确定性的方法。Agarwal 等[11]采用与 Osband 等[12]提出的自举深度 Q 网络算法(Bootstrapped-DQN)类似的方式,使用多个价值网络加权平均的方式来得到最优的动作价值,提出了随机整合混合价值网络算法(Random Ensemble Mixture,REM)。Wu 等[13]提出了不确定性权重演员评论家算法(Uncertainty Weighted Actor Critic,UWAC),在行动者-评论家框架下通过计算不确定性并将其作为权重,来缓解分布外状态-动作带来的分布偏移问题。

4)使用基于模型的方法,学习与真实实验环境相似的状态转移,再使用动态规划方法训练智能体。Kindambi 等[14]提出的基于模型的离线强化学习算法(Model Based Offline Reinforcement Learning,MOReL)首先建立一个未知状态-动作探测函数来模拟一个“悲观”的状态转移过程,再进行动态规划计算。Yu 等[15]进一步将限制价值函数的思想与基于模型的方法相结合,提出了基于模型的保守离线策略最优化算法(Conservative Offline Model-Based Policy Optimization,COMBO)。

事实上,CQL 算法虽然通过对价值网络的损失函数添加正则项的方式赋予分布外状态-动作较低的价值函数,缓解了分布外状态-动作价值估计过高的问题;但同时,算法存在训练出的策略过于保守的问题。CQL 算法训练得到的价值函数值整体偏低,区分度不足,难以达到预期中对分布外状态-动作赋予更低价值、对分布内状态-动作赋予更高价值的目的,最终导致算法无法训练得到最优策略。

针对上述问题,本文提出了一种基于不确定性权重的保守 Q 学习算法(Uncertainty Weighted Conservative Q-Learning,UWCQL),该算法能够有效避免上述问题,提高算法性能。UWCQL 在 CQL 算法的基础上额外计算不确定性,将不确定性作为权重赋予 CQL 算法中值函数的正则化项来更精确地对价值函数进行估计。通过不确定性判断状态-动作的分布,赋予不确定性较高的状态-动作更低的价值函数,反之对不确定性较低的状态-动作则不进行保守估计,从而解决 CQL 算法过于保守的问题,使智能体能够学习到最优策略。

本文的主要贡献总结如下:

1)提出了一种基于不确定性权重的保守 Q 学习算法,该方法使用不确定性作为保守 Q 学习正则项的权重,通过计算不确定性来判断训练的状态-动作是否处于分布内,从而更精确地对分布外状态-动作价值进行保守估计。

2)在 D4RL 中 MuJoCo 任务数据集上进行实验,将 UWCQL 方法与现有的离线强化学习算法进行对比,从而验证了本文方法的优越性。

2 背景知识

2.1 强化学习

在强化学习中,智能体与环境进行交互以期望得到最大的累积奖赏。通常考虑将环境表示为马尔可夫决策过程(Markov Decision Process,MDP)[16]进行建模。一个 MDP 问题可以由一个五元组(S,A,R,P,γ)表示,其中 S 为有限的状态集,$s_t\in S$ 表示时间步为 t 的环境状态;A 为有限的动作集,$a_t\in A$ 表示智能体在时间步 t 执行的动作;R 为奖赏函数,通常表示为 $r_{t+1}=R(s_t,a_t,s_{t+1})$,即智能体在状态 s_t 执行动作

a_t 转移到下一状态 s_{t+1} 时所获得的立即奖赏 r_{t+1}；P 为状态转移函数，通常表示为 $P(s_{t+1}|s_t,a_t)$，即智能体在状态 s_t 执行动作 a_t 到达下一时间步状态 s_{t+1} 的概率分布，且满足 $\sum_{s_{t+1}\in S} P(s_{t+1}|s_t,a_t)=1$；$\gamma\in[0,1)$ 为折扣因子。在马尔可夫决策过程建模下，智能体在时间步 t 处于状态 s_t 并根据策略函数 $\pi(a_t|s_t)$ 选择执行动作 a_t，进入下一状态 s_{t+1} 并获得奖赏 r_{t+1}。

在强化学习中，从时间步 t 到时间步 T 时一个情节(Episode)结束的累积折扣奖赏定义为：

$$G_t=\sum_{k=t}^{T}\gamma^{k-t}r_k \tag{1}$$

强化学习的目标是使智能体学习到一个能够最大化累积折扣奖赏期望 $E_\pi[G_t]$ 的策略 $\pi(a|s)$。在标准的强化学习中，通常假设智能体与环境交互收集经验进行学习。

对于每一个不同的策略 π，通常使用状态-动作价值函数 $Q_\pi(s,a)$ 来对智能体的当前策略进行评估。其中，状态-动作价值函数 $Q_\pi(s,a)$ 指智能体在当前策略 π 下，在状态 s 执行动作 a，直到情节结束所能得到的累积奖赏和的期望，表示为：

$$Q_\pi(s,a)=E_\pi[G_t|s_t=s,a_t=a] \tag{2}$$

由于状态-动作值函数 $Q_\pi(s,a)$ 满足贝尔曼方程，故可以通过贝尔曼算子 B 对 $Q_\pi(s,a)$ 进行计算，即：

$$BQ_\pi(s,a)=r(s,a)+\gamma E_{s'\sim P(s'|s,a)}[Q_\pi(s',a')] \tag{3}$$

在训练智能体的过程中，贝尔曼方程不断迭代，Q_π 最终将收敛，从而得到最优策略。

在实际任务中，状态空间都是庞大且复杂的，同时动作空间也是连续的，通过迭代贝尔曼方程的方式求解最优策略可行性不高。因此，为了解决大型连续状态-动作空间的任务，很多算法使用函数逼近器，如神经网络拟合等方法，通过参数化的方式来表达状态-动作价值函数。

2.2 离线强化学习

在离线强化学习中，智能体不再与环境进行交互而是从一个静态的数据集 D 中抽取数据进行学习，经过训练后再在真实环境中对策略进行评估。其中，数据由一个或多个遵循不同策略的智能体与环境交互收集得到。生成数据集数据的智能体策略被称为行为策略(Behavior Policy)。离线强化学习过程如图 1 所示。

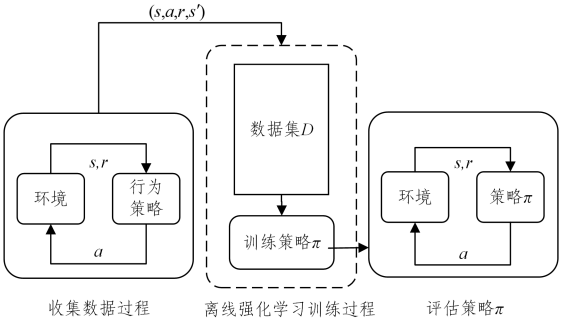


图 1 离线强化学习

Fig. 1 Offline reinforcement learning

然而，当标准在线强化学习算法直接离线使用数据集 D 进行训练时，由于外推误差的存在，无法得到很好的训练效果。外推误差主要由以下几个原因造成：

1)缺乏数据。一般离线强化学习中智能体用于学习的数据集中并不包含完整的经验四元组 (s,a,r,s') 分布，即状态-动作集合 $\{(s,a)|(s,a,r,s')\in D\}$ 是环境完整状态-动作空间 $S\times A$ 的一个子集，智能体无法通过迭代的方式正确学习到数据集中不存在的状态-动作的价值。

2)模型偏差。当基于一个固定的数据集 D 使用强化学习算法进行学习时，由于无法和环境交互访问全部状态空间，通过式(3)计算的期望是基于数据集中的状态分布，而不是真实环境的状态分布，这对状态-动作价值函数的近似产生了偏差，即基于数据集建模的马尔可夫决策过程与基于真实环境建模的马尔可夫决策过程存在偏差。

3)分布偏移。在 SAC，TD3 等算法的训练过程中，采取均匀随机的方式从经验池中抽取智能体收集到的数据进行训练。然而数据集中的数据并不是由当前训练的智能体收集得到，同样可能造成价值函数估值的不准确。

综上，由于无法与环境交互从而对数据集中的数据进行补充，标准的强化学习算法会由于外推误差的存在而无法得到较好的训练效果，算法往往会过分高估分布外状态-动作的价值，导致智能体倾向于选择分布外的动作，无法学习到最优策略。

经典的离线强化学习算法专注于使用策略限制或是值函数惩罚等方式来解决上述离线强化学习中存在的问题。在策略网络中添加正则项来限制学习策略是离线强化学习中常用的方式之一。在经典的 BCQ 算法中对策略网络添加正则项来限制策略只能选择数据集附近的动作：

$$\pi(s)=\arg\max Q_\theta(s,a_i+\xi_\varphi(s,a_i,\varphi)),\{a_i\sim G_\omega(s)\}_{i=1}^{n} \tag{4}$$

其中，G_ω 表示生成模型，用于生成动作 a；ξ 表示扰动模型，对生成的动作 a 在 $[-\varphi,\varphi]$ 范围进行扰动调整。训练得到的生成模型生成的动作被限制在行为策略附近，从而达到策略限制的效果。

Wu 等[17]提出的 BRAC 框架使用学习策略与行为策略散度作为正则项。

$$\max \mathbb{E}_{(s,a,r,s')\in D}[\mathbb{E}_{a'\sim\pi_\theta(s)}Q(s,a')]-\alpha\widehat{D}(\pi_\theta(s),\pi_b(s)) \tag{5}$$

其中，π_b 表示行为策略，$\widehat{D}$ 表示行为策略分布与学习策略分布之间的分布距离。例如当 $\widehat{D}$ 为 MMD 距离时，式(5)可以表示为 BEAR 算法中策略网络的损失函数。

3 基于不确定性的保守 Q 学习算法

本章详细介绍了 UWCQL 算法，该算法由保守 Q 学习和不确定性权重的计算两个方面组成。首先，保守 Q 学习在标准的强化学习值函数更新公式的基础之上构造了一个新的正则项，对值函数的更新进行了保守限制，可限制离线强化学习中的外推误差问题。其次，通过计算不确定性使算法对分布外的状态-动作识别更精确，解决了算法过于保守的问题。

3.1 问题分析

为了解决离线强化学习中对分布外状态-动作值函数评估过高的问题，CQL 算法对 Q 值函数更新添加惩罚项。

$$\arg\min_{Q} \alpha \cdot (\mathbb{E}_{s\sim D,a\sim\mu(a|s)}[Q_\theta(s,a)]-\mathbb{E}_{s\sim D,a\sim\pi_\beta}[Q_\theta(s,a)])+\mathbb{E}_{(s,a,r,s')\sim D}[(Q_\theta(s,a)-\hat{B}Q_{\hat{\theta}}(s',a'))^2] \tag{6}$$

其中，α 表示调整正则项的超参数，μ 表示某特定策略，π_β 表示行为策略。由于数据集 D 中不包含所有的状态-动作，因此使用经验贝尔曼算子 $\hat{B}$ 来代替贝尔曼算子 B，二者之间存在误差。通过定理证明[17-19]可以得到，在当前策略 π 下，式(6)更新得到的状态-动作函数 Q 值是真实状态函数 V 值的下界，因此 Q 值的错误高估问题得以解决。通过式(6)推导可以得到 CQL 算法更新公式：

$$\min_{Q}\max_{\mu} \alpha \cdot (\mathbb{E}_{s\sim D,a\sim\mu(a|s)}[Q_\theta(s,a)]-\mathbb{E}_{s\sim D,a\sim\pi_\beta}[Q_\theta(s,a)])+\mathbb{E}_{(s,a,r,s')\sim D}[(Q_\theta(s,a)-\hat{B}Q_{\hat{\theta}}(s',a'))^2]+R(\mu) \tag{7}$$

其中，$R(\mu)$ 是策略 μ 与先验策略 ρ 之间的 KL 散度，即 $R(\mu)=-D_{\mathrm{KL}}(\mu,\rho)$。式(7)可以表示为求解该问题：

$$\max_{\mu} \mathbb{E}_{a\sim\mu}[Q(s,a)]+D_{\mathrm{KL}}(\mu,\rho) \quad \text{s.t.} \sum_a u(a)=1 \tag{8}$$

通过式(8)可以得到 CQL 算法价值函数更新公式：

$$\min_{Q} \alpha \cdot (\mathbb{E}_{s}\sim D[\log\sum_a \exp(Q(s,a))-\mathbb{E}_{a\sim D}[Q(s,a)]])+\mathbb{E}_{(s,a,r,s')\sim D}[(Q_\theta(s,a)-\hat{B}Q_{\hat{\theta}}(s',a'))^2] \tag{9}$$

CQL 算法虽然能够避免分布外状态-动作价值过大的问题，但同时，算法存在对分布内的状态-动作值的估计过于“悲观”的问题。主要原因是 CQL 算法在区分分布外状态-动作时不够精确，仅通过最小化策略 μ 与策略 ρ 之间 KL 散度的方式来实现降低数据集中未出现的状态-动作价值的目的，导致算法对分布内的动作也赋予了较低的价值。因此 CQL 算法在进行策略优化时，智能体难以学习到最优策略。

3.2 不确定性权重计算

在深度学习领域，很多算法模型只能得到一个输出，而无法得到模型对该输出结果的置信度，这在很多情况下会出现严重问题。例如，在分类任务中，对于输入的正常图片，模型能够很好地完成任务，但如果输入的是其他不包含在模型识别范围内的图片，模型依然会输出分布内的结果，然而这样的输出结果的置信度非常低。因此，通过计算结果的不确定性，让模型在输出结果的同时输出对于该结果的置信度，能够有效解决该问题。

将不确定性机制引入离线强化学习，能够有效解决价值网络难以区分分布外状态-动作的问题。不确定性可通过多种方式进行计算，例如使用蒙特卡洛丢弃方法(Monte Carlo Dropout，MC Dropout)[18]或集成方法(Ensemble)[19]。

如图 2 所示，蒙特卡洛丢弃方法在神经网络中开启 Dropout 层，在向前通过网络的过程中，某些神经元有一定的概率权重为 0，计算多次通过网络得到的值的方差作为不确定性。状态-动作多次通过开启 Dropout 层的网络得到的结果方差较小，说明该状态-动作来自数据集内，反之，方差较大则表明为分布外状态-动作。集成方法与蒙特卡洛丢弃类似，但其使用多个网络模型来代替单一网络的 Dropout 层。多个模型随机初始化不同的权重进行训练，多个网络得到各自的输出 Q 值来计算不确定性。

本文选择使用蒙特卡洛丢弃方法计算不确定性，该方法消耗的算力较少，且便于在 CQL 算法的基础上实现。

价值网络对于输入价值网络的动作状态对 (s,a)，在输出价值 $Q(s,a)$ 的同时，计算网络对于该值的不确定性 $Var(s,a)$。对于不确定性高的结果，认为该状态-动作在数据集中不存在或数量较少，价值估计的置信度较低；反之则表明该状态-动作存在于数据集中且经过了充分训练。

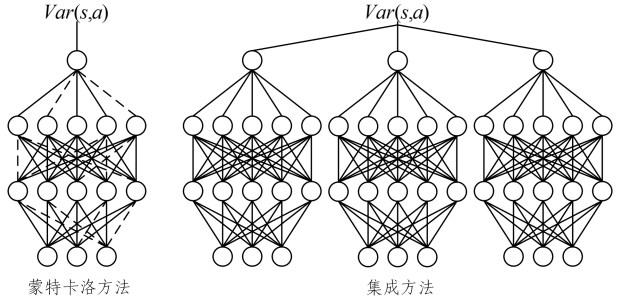


图 2 不确定性计算

Fig. 2 Uncertainty calculation

令 X 表示数据集 D 中所有的状态-动作 (s,a)，即 $X=\{(s,a)|(s,a)\in D\}$；Y 表示当前策略 π 下的真实 Q 值。由贝叶斯公式有：

$$P(\theta|X,Y)=\frac{P(Y|X,\theta)P(\theta)}{P(Y|X)} \tag{10}$$

然而由于无法得到真实数据分布，因此在网络的每一层添加 Dropout，同时在测试阶段也开启 Dropout，通过有限次的采样来对分布进行估计即可得到分布的近似值。由式(12)计算方差得到输出 $Q(s,a)$ 的不确定性：

$$E(\hat{Q}_p(s,a))=\frac{1}{T}\sum_t^T \hat{Q}_p(s,a)^{\mathrm{T}} \tag{11}$$

$$Var(s,a)=\sigma^2+\frac{1}{T}\sum_t^T \hat{Q}_p(s,a)^{\mathrm{T}}\cdot\hat{Q}_p(s,a)-E(\hat{Q}_p(s,a))^{\mathrm{T}}E(\hat{Q}_p(s,a)) \tag{12}$$

其中，σ^2 是数据内的噪声，本文忽略其影响而只选取式(11)中的后两项计算不确定性。在代码实现中，只需要在开启价值网络 Dropout 层的情况下将数据输入网络 T 次，并计算 T 次输出的均值与方差，得到的方差即输出价值的不确定性。

3.3 UWCQL 算法描述及分析

UWCQL 算法使用不确定性 $Var(s,a)$ 作为权重，取代了式(9)中 CQL 算法限制 Q 值过估计的正则项的权重 α，如式(13)所示：

$$\min_{Q} Var(s,a)\cdot(\mathbb{E}_{s\sim D}[\log\sum_a \exp(Q(s,a))-\mathbb{E}_{s,a\sim D}[Q(s,a)]])+\mathbb{E}_{(s,a,r,s')\sim D}\left[\frac{\beta}{Var(s',a')}(Q_\theta(s,a)-\hat{B}Q_{\hat{\theta}}(s',a'))^2\right] \tag{13}$$

其中，β 是调整 UWCQL 算法的超参数。UWCQL 算法易于实现，无需做额外的复杂运算，而仅需在 CQL 算法的基础上做不确定性计算即可。在 UWCQL 算法中，价值网络的输入是数据集 D 的数据 (s,a)，在正常进入网络输出其价值 $Q(s,a)$ 的同时，网络开启 Dropout 层，用于计算不确定性 $Var(s,$

a)。得到更新公式(13)中的 $Var(s,a)$ 和 $Var(s',a')$ 后，使用梯度下降方法最小化损失函数。此外，UWCQL 算法采取与 TD3 算法相同的延迟更新方式对动作网络的参数进行更新。

UWCQL 算法伪代码如算法 1 所示，在算法 1 中，第 9 行通过蒙特卡洛丢弃方法计算不确定性，第 10 行得到基于不确定性权重的保守 Q 网络损失函数后在第 11 行计算梯度，其中不确定性值的大小决定了算法对当前状态-动作的评价。若不确定性大，则表明应当对当前状态-动作进行保守估计，继而产生较小的 Q 值。第 12－16 行采取与 TD3 算法类似的延迟更新方式对行动网络进行参数更新。

由于本文方法仅对价值网络更新过程进行修改，因此其同样适用于仅包含价值函数的经典强化学习算法。UWCQL 算法能够快速计算得到不确定性，帮助智能体更精确地分辨状态-动作分布；对于高不确定性的结果，分配更大的权重进行保守价值估计。

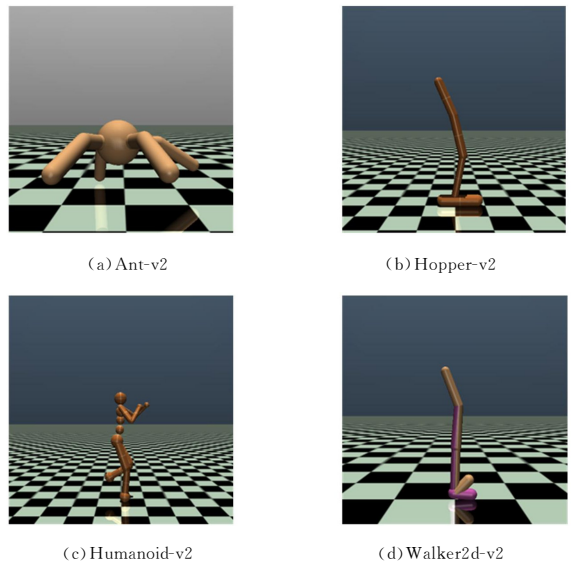
算法 1 UWCQL 算法

1. 参数初始化
2. 初始化 Actor/Critic 网络参数
3. 将 Critic 网络参数赋值给目标网络
   $\theta' \leftarrow \theta$
4. 将 Actor 网络参数赋值给目标网络
   $\varphi' \leftarrow \varphi$
5. for 每次迭代 do
6. 抽取样本数据(s,a,r,s′)～D
7. 从目标动作网络 $\hat{\pi}$ 采样动作 $a'=\hat{\pi}(s')$
8. 计算 TD-Error：$y=r+Q(s',a')$
9. 通用式(12)计算 Var(s,a)，Var(s′,a′)
10. 通过式(13)得到价值网络 Q 损失函数 $L(\theta)$
11. 对 $L(\theta)$ 梯度下降更新 θ
12. if t mod d 步 then
13. 对 $L(\varphi)=-\mathbb{E}a\sim\pi[Q(s,a)]$ 梯度下降更新 φ
14. 更新目标网络参数：
15. $\theta'=\tau\theta+(1-\tau)\theta'$
16. $\varphi'=\tau\varphi+(1-\tau)\varphi'$
17. end if
18. end for

4 实验及分析

为了验证 UWCQL 算法的有效性，本文在 Kumar 等[20]提出的面向离线强化学习算法评估的基准数据集 D4RL 上进行了实验。D4RL 数据集包含 Gym-MuJoCo，AntMaze 以及 Offline CARLA 等多种经典强化学习环境的数据集，其中每一项任务由多种不同等级的数据集组成。

Gym[21]作为 OpenAI 的仿真平台，是强化学习中重要的开源工具包。其提供了丰富的实验环境，包含 Atari 游戏、MuJoCo、经典控制和 Box2D 等。其中 MuJoCo 是一个免费的开源物理引擎，不仅能用于实现基于模型的计算，还可以用作传统的模拟器包括游戏和交互式虚拟环境。采用 Gym 下的经典 MuJoCo 环境来解决连续状态-动作空间中的实验任务，如图 3 所示。



(a) Ant-v2　(b) Hopper-v2

(c) Humanoid-v2　(d) Walker2d-v2

图 3 MuJoCo 环境

Fig. 3 MuJoCo environments

实验使用 D4RL 中的 MuJoCo 数据集进行训练并在标准 MuJoCo 环境进行评估，其中每个任务包含 5 种不同等级的数据集以供对比。

1) Random：数据由一个随机初始化策略的智能体收集得到。

2) Medium：数据由使用 SAC 方法训练并提前结束训练的智能体收集得到。

3) Medium-Replay：包括从训练开始直到满足 Medium 等级表现的智能体训练过程中收集的全部数据。

4) Medium-Expert：数据由训练到一定程度的智能体和专家智能体混合收集得到。

5) Expert：数据由专家样本组成。

针对一系列连续控制任务，本文选取了 D4RL 中 HalfCheetah，Hopper 和 Walker2d 这 3 个任务的数据集进行离线强化学习算法的实现，以下为各环境对应的任务介绍：

1) HalfCheetah：智能体控制猎豹型双腿机器人。机器人只能向前或向后移动，任务目标是让机器人学习到奔跑的姿势。向前移动可以得到正值奖赏而向后移动或是采取的动作过大则得到负值奖赏。当达到预定时间则一个情节结束。

2) Hopper：智能体学习控制二维单腿机器人通过调整关节进行向前跳跃的动作。当机器人保持姿势良好或是向前跳跃时可以得到正值奖赏，而向后或动作过大则得到负值奖赏。当机器人因姿势不佳而摔倒或是达到预定时间步时则一个情节结束。

3) Walker2d：智能体控制双腿机器人。任务目标是学习控制关节使得机器人能够向前行走。而奖赏与 Hopper 任务相似，当机器人保持姿势良好和向前行走时得到正值奖赏，向后或是动作过大得到负值奖赏。当机器人因姿势不佳而摔倒或达到预定时间步时则一个情节结束。

4.1 实验设置

本文实验使用 AMD 5600X 处理器、RTX 3070 图形处理器对深度学习运算进行加速计算。使用 Python3.8 以及 Py-

torch 框架进行编程，操作系统为 Ubuntu18.04。

对比实验使用了 CQL，TD3 + BC，Fisher-BRC 以及 UWCQL 算法。实验中，前 3 种方法均基于原作者在 Github 上公布的代码实现，实验中的超参数设置如表 1 所列。在上述算法中，批量选取的样本容量为 $n=256$，训练次数为 10^6 次，每经过 5000 次训练后在实际任务环境中对当前训练效果进行评估。

表 1 算法超参数设置

Table 1 Algorithm hyperparameters

参数描述	参数值
优化器	Adam
价值网络学习率	0.0003
动作网络学习率	0.0003
衰退因子	0.99
目标网络更新率	0.005
网络中间层维数	256
网络中间层数	2
激活函数	ReLU

4.2 实验结果分析

图 4—图 6 分别给出了 UWCQL 算法与对比算法在各任务训练过程中的标准化性能分数，为了区分不同的算法，使用不同颜色表示不同算法的平均累积奖赏。图中横坐标表示智能体的训练次数，纵坐标表示标准化性能分数。其中图 4 为各算法在 HalfCheetah 任务数据集上训练过程中的标准化平均累积奖赏；图 5 为各算法在 Hopper 任务数据集上训练过程中的标准化平均累积奖赏；图 6 为各算法在 Walker2d 任务数据集上训练过程中的标准化平均累积奖赏。

分析可得，在大部分的任务中，UWCQL 算法得到的性能分数都要优于其他对比算法，但在稳定性方面，例如 Medium 等级和 Expert 等级数据集的各任务中，UWCQL 算法在训练后期的性能曲线上出现了较大幅度的震荡，表明 UWCQL 算法存在稳定性不足的问题。这可能是由于数据集中的数据缺少多样性，算法对其他状态-动作的训练不足，使策略陷入次优解。而当数据相对较为丰富时，如使用 Medium-Expert 以及 Medium-Replay 等级的数据集进行训练时算法则相对稳定。在收敛速率方面，在使用 Expert 等级数据集进行训练时，UWCQL 算法能够更快收敛到较高的性能分数，在 3 个任务中分别大约在 30 万步训练次数中就收敛到了 100 分以上的性能分数。而在其他等级的数据集收敛速度方面，各算法之间仅有较小的差距，这可能是由于离线强化学习中不涉及强化学习中的探索问题，而使用固定数据集进行训练，因此各算法之间得到的用于训练的数据并无差异。

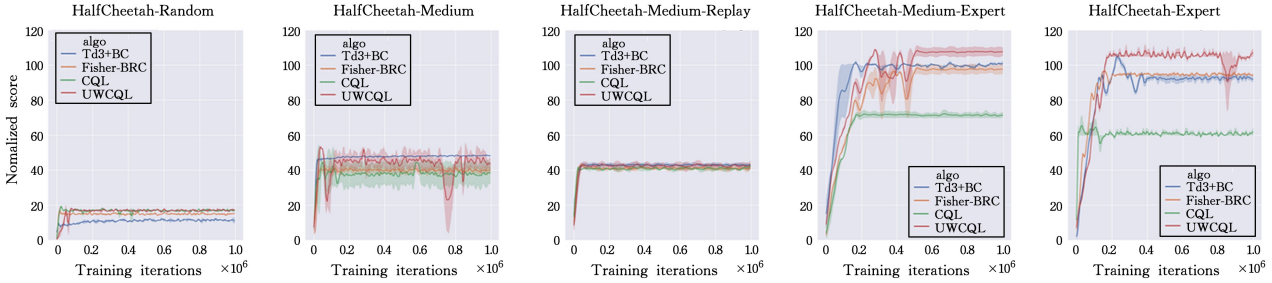


图 4 HalfCheetah 任务训练曲线

Fig. 4 Training curves of HalfCheetah task

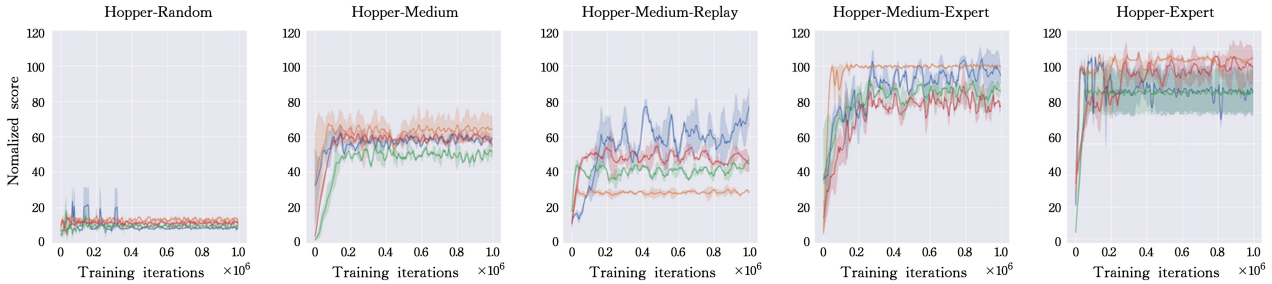


图 5 Hopper 任务训练曲线

Fig. 5 Training curves of Hopper task

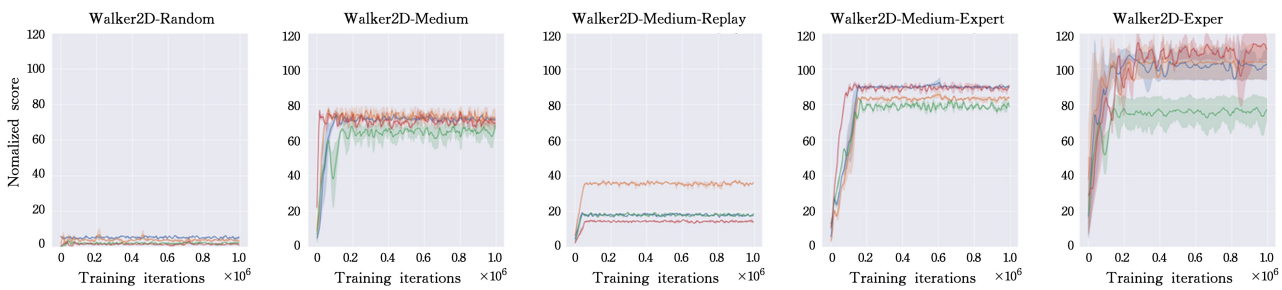


图 6 Walker2D 任务训练曲线

Fig. 6 Training curves of Walker2D task

表 2 列出了各算法在各任务 5 种等级数据集下获得的标准化平均奖赏，其中每种算法随机取 4 个不同的种子训练得到的均值作为平均奖赏。进行算法效果评估时，在实际任务环境中运行 10 个情节并对标准化累积奖赏取均值作为当前训练效果参考。其中各任务最高分数使用加粗标记。

表 2 实验算法的最终性能

Table 2 Final performance of experimental algorithms

数据集	环境	TD3+BC	Fisher-BRC	CQL	UWCQL
Random	HalfCheetah	13.3	15.3	15.6	**16.6**
	Hopper	8.3	**14.7**	9.2	**12.3**
	Walker2d	5.5	3.7	2.5	1.3
Medium	HalfCheetah	47.3	41.9	37.3	**51.6**
	Hopper	58.0	60.2	52.0	**65.2**
	Walker2d	72.0	**76.4**	63.6	**72.4**
Medium-Replay	HalfCheetah	43.6	**45.2**	42.7	43.1
	Hopper	35.0	28.9	31.2	**42.5**
	Walker2d	20.8	**37.2**	20.3	15.7
Medium-Expert	HalfCheetah	103.2	100.7	73.4	**113.7**
	Hopper	91.4	**96.5**	85.8	80.4
	Walker2d	92.3	88.5	81.5	**95.1**
Expert	Hopper-v2	95.1	97.7	63.0	**103.4**
	Humanoid-v2	105.3	103.7	74.1	**107.2**
	Walker2d-v2	103.7	95.3	83.1	**106.7**

分析可得，在使用 Random 等级数据集进行训练时，各算法的性能表现都较差，这主要是因为数据本身的质量不高，算法很难学习到最优策略。而使用 Medium 等级数据集进行训练时，UWCQL 算法性能相比其他对比算法更好，在 HalfCheetah 任务、Hopper 任务以及 Walker2D 任务中分别得到了 51.6，65.2 和 72.4 的性能分数，相比 CQL 算法性能分别提升了 38.3%，25%和 13.8%。使用 Medium-Replay 等级数据集训练时，UWCQL 算法相比其他算法没有显著优势。使用 Medium-Expert 等级数据集进行训练时，UWCQL 算法在 HalfCheetah 以及 Walker2D 任务上优于其他对比算法，分别得到了 113.7 和 95.1 的性能分数，相比 CQL 算法性能提升了 54%和 16%。使用 Expert 等级数据集进行训练时，UWCQL 算法的性能分数最高，在 HalfCheetah，Hopper，Walker2D 这 3 个任务上分别得到了 103.4，107.2 和 106.7 的性能分数，相比 CQL 算法性能分别提升了 64%，44%以及 30.2%。通过对比实验，验证了 UWCQL 算法相较于其他对比算法在大多数任务中能够更有效地学习到最优策略。

结束语 CQL 算法在离线强化学习任务中有较好的表现，但仍然存在学习不到最优解以及算法稳定性较差的问题。主要原因在于算法希望遏制数据集分布外状态-动作价值的估计，但实际上出现了无法有效分辨分布内外状态-动作的问题，导致学习到的策略过于保守。针对该问题，本文提出了一种基于不确定性权重的保守 Q 学习算法，该方法通过蒙特卡洛丢弃方法计算输入状态-动作的不确定性，利用不确定性作为权重来判断保守估计的程度，可以很大程度上避免难以分辨输入状态-动作分布的问题。将 UWCQL 算法运用于 D4RL 基准的 MuJoCo 任务数据集中进行对比实验，验证了所提算法的有效性。实验结果表明，UWCQL 算法的总体性能要优于其他经典离线强化学习算法。

本文所提出的算法易于理解，便于在 CQL 算法的基础上实现，且对 CQL 算法的性能有较大的提升。然而，UWCQL 算法的稳定性问题没有得到很好解决。因此，如何提高算法的稳定性，将是下一步工作的研究重点。

参 考 文 献

[1] LIU Q，ZHAI J W，ZHANG Z Z，et al. A survey on deep reinforcement learning [J]. Chinese Journal of Computers，2018，41(1)：1-27.

[2] MNIH V，KAVUKCUOGLU K，SILVER D，et al. Human-level control through deep reinforcement learning [J]. Nature，2015，518(7540)：529-533.

[3] LEVINE S，KUMAR A，TUCKER G，et al. Offline reinforcement learning：Tutorial，review，and perspectives on open problems [J]. arXiv：2005.01643，2020.

[4] FUJIMOTO S，MEGER D，PRECUP D. Off-policy deep reinforcement learning without explora-tion[C]// International Conference on Machine Learning. PMLR，2019：2052-2062.

[5] KINGMA D P，WELLING M. Auto-Encoding Variational Bayes [J]. arXiv：1312.6114，2014.

[6] KUMAR A，FU J，SOH M，et al. Stabilizing off-policy q-learning via bootstrapping error reduction [J]. arXiv：1906.00949，2019.

[7] FUJIMOTO S，HOOF H，MEGER D. Addressing function approximation error in actor-critic methods [C]// International Conference on Machine Learning. PMLR，2018：1587-1596.

[8] FUJIMOTO S，GU S S. A minimalist approach to offline reinforcement learning [J]. Advances in Neural Information Processing Systems，2021，34：20132-20145.

[9] KUMAR A，ZHOU A，TUCKER G，et al. Conservative Q-learning for offline reinforcement learning [J]. Advances in Neural Information Processing Systems，2020，33：1179-1191.

[10] LYU J，MA X，LI X，et al. Mildly conservative Q-learning for offline reinforcement learning [J]. Advances in Neural Information Processing Systems，2022，35：1711-1724.

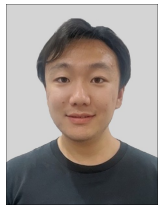
[11] AGARWAL R，SCHUURMANS D，NOROUZI M. An optimistic perspective on offline reinforcement learning[C]// Proceedings of the 37th International Conference on Machine Learning. 2020：104-114.

[12] OSBAND I，BLUNDELL C，PRITZEL A，et al. Deep exploration via bootstrapped DQN [C]// Proceedings of the 30th International Conference on Neural Information Processing Systems. 2016：4033-4041.

[13] WU Y，ZHAI S，SRIVASTAVA N，et al. Uncertainty Weighted Actor-Critic for Offline Reinforcement Learning[C]// International Conference on Machine Learning. PMLR，2021：11319-11328.

[14] KIDAMBI R，RAJESWARAN A，NETRA-PALLI P，et al. Morel：Model-based offline rein-forcement learning [J]. Advances

in Neural Information Processing Systems, 2020, 33: 21810-21823.
[15] YU T, KUMAR A, RAFAILOV R, et al. Combo: Conservative offline model-based policy optimization [J]. Advances in Neural Information Processing Systems, 2021, 34: 28954-28967.
[16] SUTTON R S, BARTO A G. Reinforcement learning: An introduction [M]. MIT press, 2018.
[17] WU Y, TUCKER G, NACHUM O. Behavior regularized offline reinforcement learning [J]. arXiv: 1911. 11361, 2019.
[18] GAL Y, GHAHRAMANI Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning[C]// International Conference on Machine Learning. PMLR, 2016: 1050-1059.
[19] LAKSHMINARAYANAN B, PRITZEL A, BLUNDELL C. Simple and scalable predictive uncertainty estimation using deep ensembles [C]// Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017: 6405-6416.
[20] FU J, KUMAR A, NACHUM O, et al. D4rl: Datasets for deep data-driven reinforcement learning [J]. arXiv: 2004. 07219, 2020.
[21] BROCKMAN G, CHEUNG V, PETTERSSON L, et al. Openai gym[J]. arXiv: 1606. 01540, 2016.



WANG Tianjiu, born in 1999, postgraduate. His main research interests include reinforcement learning and offline reinforcement learning.

LIU Quan, born in 1969, Ph.D, professor, Ph.D supervisor, is a member of CCF(No. 15231S). His main research interests include deep reinforcement learning and automated reasoning.

(责任编辑:何杨)

2024“科创中国”科技服务团项目,CCF 入选两项

“科创中国”科技服务团项目由中国科协以加快建设现代化产业体系、因地制宜发展新质生产力为牵引,主动对接国家重大战略落地和重点产业集群发展需要,在创新驱动示范市(区)、“科创中国”试点城市(园区)等探索科创有效机制,支持全国学会牵头组建“科创中国”科技服务团,协同省、市两级科协征集遴选重点产业链发展难题和重大产业共性技术问题,坚持“有限目标、提升实效”原则,跨界跨域组织动员科技工作者开展多元化精准科技服务,通过建立学会服务站、长期“蹲点”服务等深化长效合作机制,推动问题难题解决取得实质性进展,助力改造提升传统产业、培育壮大新兴产业、布局建设未来产业。

“科创中国”工业互联网产业科技服务团由 CCF 牵头,CCF 会士、前理事长、梅宏院士领衔,重点服务长三角一体化发展区域,以发展创新示范市苏州为试点城市,聚焦以工业互联网为牵引推动苏州市打造未来制造产业等重点产业链发展难题和重大产业共性技术问题,提出解决方案,组织科技服务团提供针对性、多元化、组合式的科技服务,形式包括但不限于技术交流对接、产业规划咨询、企业技术诊断、难题联合攻关、成果转化推广、团体标准研制、项目签约落地等,促进问题难题的解决取得实质性进展。服务团以破解制约工业互联网发展和应用的痛、难点为导向,面向制造业企业遇到的数字化转型、工业数据安全、人工智能算法等问题,通过难题研判、组建专家服务团及产学对接等工作,为企业提供专业服务。

“科创中国”新质生产力科技服务团由 CCF 牵头,上海区块链技术协会、CCF 上海会员活动中心等相关机构共同组成。该服务团汇聚了人工智能、区块链、大数据等多个领域的院士、专家学者和科技工作者,将对接国家重大战略落地和重点产业集群发展需要,发挥全国学会优势与省级学会协会联合联动,聚焦上海市创新驱动示范区和“科创中国”试点及培育区建设,并辐射长三角等相关区域,通过建立学会服务站、学企联合体,下沉专家资源长期“蹲点”服务等深化长效合作机制,助力上海科创中心建设和长三角高质量一体化发展,推动科技创新与产业融合,促进新质生产力发展。

【技术需求对接合作】

咨询时间:周一至周五(法定节假日除外)

上午 8:30—12:00 下午 13:00—17:30

座机:0512—6590 0856(分机号 34)

邮箱:kaihou@ccf.org.cn

据 CCF 微信公众号