基于不确定性估计的离线确定型 Actor-Critic

冯涣婷[1),2)] 程玉虎[1)] 王雪松[1)]

[1)](中国矿业大学信息与控制工程学院 江苏 徐州 221116)

[2)](江苏信息职业技术学院智能工程学院 江苏 无锡 214153)

摘 要 Actor-Critic是一种强化学习方法,通过与环境在线试错交互收集样本来学习策略,是求解序贯感知决策问题的有效手段.但是,这种在线交互的主动学习范式在一些复杂真实环境中收集样本时会带来成本和安全问题.离线强化学习作为一种基于数据驱动的强化学习范式,强调从静态样本数据集中学习策略,与环境无探索交互,为机器人、自动驾驶、健康护理等真实世界部署应用提供了可行的解决方案,是近年来的研究热点.目前,离线强化学习方法存在学习策略和行为策略之间的分布偏移挑战.针对这个挑战,通常采用策略约束或值函数正则化来限制访问数据集分布之外(Out-Of-Distribution, OOD)的动作,从而导致学习性能过于保守,阻碍了值函数网络的泛化和学习策略的性能提升.为此,本文利用不确定性估计和OOD采样来平衡值函数学习的泛化性和保守性,提出一种基于不确定性估计的离线确定型Actor-Critic方法(Offline Deterministic Actor-Critic based on Uncertainty Estimation, ODACUE).首先,针对确定型策略,给出一种Q值函数的不确定性估计算子定义,理论证明了该算子学到的Q值函数是最优Q值函数的一种悲观估计.然后,将不确定性估计算子应用于确定型Actor-Critic框架中,通过对不确定性估计算子进行凸组合构造Critic学习的目标函数.最后,D4RL基准数据集任务上的实验结果表明:相较于对比算法,ODACUE在11个不同质量等级数据集任务中的总体性能提升最低达9.56%,最高达64.92%.此外,参数分析和消融实验进一步验证了ODACUE的稳定性和泛化能力.

关键词 离线强化学习;不确定性估计;分布外采样;凸组合;Actor-Critic

中图法分类号 TP18 **DOI号** 10.11897/SP.J.1016.2024.00717

Offline Deterministic Actor-Critic Based on Uncertainty Estimation

FENG Huan-Ting[1),2)] CHENG Yu-Hu[1)] WANG Xue-Song[1)]

[1)](*School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, Jiangsu* 221116)

[2)](*School of Intelligent Engineering, Jiangsu Vocational College of Information Technology, Wuxi, Jiangsu* 214153)

Abstract Actor-critic is a reinforcement learning method that learns a policy by collecting samples through online trial-and-error interaction with the environment, which is an effective tool for solving sequential perceptual decision problems. However, the active learning paradigm of online interaction raises cost and security issues when collecting samples in some complex real-world environments. Offline reinforcement learning, as a data-driven reinforcement learning paradigm, emphasizes learning policy from a static sample dataset without exploratory interaction with the environment, which has been a research hotspot in recent years and provides a feasible solution for real-world deployment applications such as robotics, autonomous driving, healthcare, and so on. At present, offline reinforcement learning methods face the challenge of distribution shift between the learned and behavior policies, which generates extrapolation errors in the value function estimation for the out-of-distribution (OOD) actions of the static sample dataset. The extrapolation errors are accumulated

收稿日期:2023-01-16;在线发布日期:2023-12-25.本课题得到国家自然科学基金项目(62373364, 62176259)、江苏省重点研发计划项目(BE2022095)资助.**冯涣婷**,博士研究生,中国计算机学会(CCF)会员,主要研究领域为强化学习. E-mail: jhxmfht@163.com. **程玉虎**,博士,教授,中国计算机学会(CCF)会员,主要研究领域为机器学习和智能系统. **王雪松(通信作者)**,博士,教授,中国计算机学会(CCF)会员,主要研究领域为机器学习和模式识别. E-mail: wangxuesongcumt@163.com.

with the Bellman bootstrapping operation, which leads to the performance degradation or even non-convergence of offline reinforcement learning. In order to deal with the distribution shift problem, the policy constraint or value function regularization is usually used to restrict the agent access to OOD actions, which may result in overly conservative learning performance and hinder the generalization of value function network and performance improvement of policy. To this end, an offline deterministic actor-critic method based on uncertainty estimation (ODACUE) is proposed to balance the generalization and conservation of value function learning by utilizing the uncertainty estimation and OOD sampling. Firstly, for the deterministic policy, the definition of uncertainty estimation operator is given according to the different estimation methods of Q value function for the in-dataset and OOD actions. The in-dataset action value function is estimated according to the Bellman bootstrapping operation and ensemble uncertainty estimation. On the other hand, the OOD action value function is estimated based on a pseudo-target constructed by the ensemble uncertainty estimation and OOD sampling method. The pessimism of the uncertainty estimation operator is theoretically analyzed by ξ-uncertainty estimation theory. By choosing appropriate parameters, the Q value function learned according to the uncertainty estimation operator is a pessimistic estimation of the optimal Q value function. Then, by applying the uncertainty estimation operator to the deterministic actor-critic framework, the objective function of critic learning is constructed via a convex combination of the in-dataset and OOD action value functions, thus the conservative constraints and generalization of value function learning are balanced by using the convex combination coefficient. Moreover, the uncertainty estimation operator of value function is implemented by the critic target network during the in-dataset action value function learning process. During the OOD action value function learning process, the OOD sampling is implemented by the actor main network, and the uncertainty estimation operator of value function is implemented by the critic main network. Finally, ODACUE and some state-of-the-art baseline algorithms are evaluated on D4RL benchmark. Experimental results show that, in contrast to the comparative algorithms, the overall performance improvement of ODACUE on the 11 datasets with different quality levels is at least 9.56% and at most 64.92%. In addition, parameter analysis and ablation experiments further validate the stability and generalization ability of ODACUE.

Keywords offline reinforcement learning; uncertainty estimation; out-of-distribution sampling; convex combination; Actor-Critic

1 引 言

近年来,具有试错交互机制的在线强化学习在一些模拟任务[1-2]和游戏[3]领域取得了显著成绩.但是,这种在线交互的主动学习范式在一些真实世界收集样本时,会带来成本和安全问题[4-5],例如自动驾驶[6]、健康护理[7]、机器人[8]等领域.基于数据驱动的离线强化学习是利用以前收集的静态样本数据集训练策略,智能体在训练学习过程中与环境无任何探索交互.因此,离线强化学习为真实世界的部署应用提供了一种可行的解决方案.

目前,离线强化学习算法的主要挑战是学习策略和数据集行为策略之间的分布偏移[4],该分布偏移会对数据集分布之外(Out-Of-Distribution, OOD)的动作值函数估计产生外推误差.在基于 Q 学习的方法中,外推误差会随着贝尔曼自举操作不断累积,从而导致离线强化学习性能下降甚至不收敛[9-10].因此,为了解决离线强化学习的分布偏移问题,相关研究学者提出了很多有效方法,主要分为策略约束和值函数正则化两类.策略约束是采用 KL 散度[11-14]或最大均值差异(Maximum Mean Discrepancy, MMD)[10]等分布差异度量方法使学习策略尽可能地接近静态数据集的行为策略,从而避免在值函数估计中选择静态数据集之外的动作,以减小未知动作对策略学习的影响.以保守 Q 学习(Con-

servative Q Learning, CQL)为代表的 Q 值函数正则化是在 Q 值函数学习目标基础上对 OOD 动作值函数进行惩罚,学习一个保守型值函数[15-17].这两类方法虽然在一些相关任务上能够获得优越性能,但是它们悲观地认为 OOD 动作都是不好的行为,这种限制访问 OOD 动作的方法往往只能学到保守的次优策略,阻碍了 Q 值函数的学习泛化性能.

如何从有限的数据集中平衡保守和泛化呢?OOD 动作未必都是不好的,如果可以判别 OOD 动作值函数的置信区间,利用高置信区间 OOD 数据训练来提高 Q 值函数的泛化能力,则学到的策略可能会超越行为策略的次优范围.直观地,基于不确定性的离线强化学习方法[4]是根据对模型泛化能力的信任程度,对策略、值函数或者模型等进行不确定性估计,以此判断是对学习策略进行保守约束还是放松约束,为解决之前过度约束问题提供了一种有效方法.离线强化学习中采用的不确定性估计方法主要包括蒙特卡罗 dropout 和自举集成两种[18].蒙特卡洛 dropout 是监督学习中常用的估计模型认知不确定性方法,Wu 等人[19]利用蒙特卡洛 dropout 方法估计 OOD 动作值函数的不确定性,并根据 OOD 不确定性估计对值函数和策略的学习目标损失函数进行权重下调,从而对高认知不确定区域减小相应的更新.但是,该方法是在降低自举误差(Bootstrapping Error Accumulation Reduction, BEAR)的策略约束方法[10]基础上实施的,策略性能容易受策略约束的影响,对 OOD 动作约束过紧,仅仅在相关的专家任务数据集上表现较好.

自举集成是一种在线强化学习最常用的提升智能体探索能力[20]和解决值函数高估问题[9-10,21]的方法.Kidambi 等人[22]和 Yu 等人[23]利用自举集成神经网络学习环境的回报模型,并根据回报模型的不确定性来惩罚未见状态－动作的立即回报函数.但是,这种基于模型框架的方法引入了额外的环境模型训练,且利用有限的给定数据集往往很难训练一个好的模型.聚焦于无模型学习方法,Agarwal 等人[24]提出一种随机集成混合方法,利用自举集成方法对 Q 函数进行凸组合学习.但是,该方法偏向于乐观主义,不考虑分布偏移问题,在一些控制任务上表现极差.为了能够充分利用自举集成 Q 的优势来估计 OOD 数据的认知不确定性,An 等人[25-26]对软 Actor-Critic(Soft Actor-Critic, SAC)算法进行集成,利用自举集成 Q 网络的裁剪估计作为 OOD 动作的不确定性惩罚,并通过计算集成 Q 值函数两两之间的梯度来增加网络的多样性,以提高不确定性估计的准确度.Lee 等人[27]在 CQL 基础上对 Critic 和 Actor 分别进行集成,以解决离线到在线转移过程中的 OOD 数据.Bai 等人[28]利用自举先验集成 Q 网络的不确定性估计和 OOD 动作采样方法获得 Q 函数估计,其中数据集内的 Q 目标值是采用独立的目标 Q 减去标准差计算的(简称共享目标),这种 Q 学习方法在一些非专家数据集任务中能够表现出较好的性能.与 Bai 等人[28]基于共享目标的 Q 学习不同,Ghasemipour 等人[29]基于完全独立的 Q 网络计算 Q 目标值,并根据 CQL 值函数正则化方法估计 Q 值,利用集成 Q 网络的不确定性估计优化策略.可以看出:上述方法均是利用自举集成的思想来从不同角度估计 Q 的不确定性;大部分方法都是针对随机策略研究的,对于确定型策略涉及极少.

在离线强化学习方法中,最具代表性的确定型策略梯度算法是 TD3＋BC[30],其利用模仿学习方式将学习策略约束在数据集范围内.正如之前所述,这种策略约束方法依赖于数据集的质量,在非专家数据集任务上的性能很差.因此,为了提高确定型策略梯度算法在离线环境中的性能,本文将不确定性估计方法与确定型策略梯度相结合,利用类似于文献[28]的集成不确定性估计方法来监测 OOD 动作,以放松学习策略的保守约束,设计一种基于不确定性估计的离线确定型 Actor-Critic(Offline Deterministic Actor-Critic based on Uncertainty Estimation, ODACUE).本文主要贡献如下:

(1)针对确定型策略,根据数据集内和数据集外动作值函数的不同估计方式,给出一种 Q 值函数的不确定性估计算子定义.

(2)利用 ξ-不确定性估计理论证明了不确定性估计算子的悲观性,即根据不确定性估计算子学习得到的值函数是悲观值函数.

(3)将不确定性估计算子应用于确定型策略梯度方法中,通过在 Q 值函数学习目标上对数据集内动作值函数和 OOD 动作值函数进行凸组合设计,利用凸组合系数平衡值函数学习的保守约束和泛化,提出一种基于不确定性估计的离线确定型 Actor-Critic 算法.

2　背景知识

2.1　强化学习

强化学习问题通常被建模为一种马尔科夫决策

过程(Markov Decision Process, MDP)[31],描述为$(\mathcal{S},\mathcal{A},p,r,\rho_0,\gamma)$,其中,$\mathcal{S}$表示状态空间,$\mathcal{A}$表示动作空间,$p(\boldsymbol{s}'\mid\boldsymbol{s},\boldsymbol{a})$表示状态转移概率分布,$r:\mathcal{S}\times\mathcal{A}\rightarrow R$表示立即回报函数,$\rho_0$表示初始状态分布,$\gamma$为折扣因子.强化学习的目标是通过最大化期望累积折扣回报学习一个最优策略,此处考虑一个确定型策略$\pi:\mathcal{S}\rightarrow\mathcal{A}$,从状态$\boldsymbol{s}$和动作$\boldsymbol{a}$开始,根据策略$\pi$采样执行得到的期望累积折扣回报称为状态-动作值函数:

$$Q(\boldsymbol{s},\boldsymbol{a})=E_{\pi}\left[\sum_{k=0}^{\infty}\gamma^{k}r_{t+k}\mid\boldsymbol{s}_t=\boldsymbol{s},\boldsymbol{a}_t=\boldsymbol{a}\right] \tag{1}$$

状态-动作值函数$Q(\boldsymbol{s},\boldsymbol{a})$满足如下贝尔曼方程:

$$\mathrm{B}Q(\boldsymbol{s},\boldsymbol{a})=r(\boldsymbol{s},\boldsymbol{a})+\gamma E_{\boldsymbol{s}'\sim p(\boldsymbol{s}'\mid\boldsymbol{s},\boldsymbol{a})}[Q(\boldsymbol{s}',\pi(\boldsymbol{s}'))] \tag{2}$$

在基于近似动态规划的 Actor-Critic 强化学习方法中,状态-动作值函数和策略分别采用参数化函数逼近器如神经网络表示为$Q_{\varphi}(\boldsymbol{s},\boldsymbol{a})$和$\pi_{\theta}$.给定样本数据集$D=\{(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}',r(\boldsymbol{s},\boldsymbol{a}))\}$,Critic 通过最小化如下贝尔曼均方误差学习$Q$值函数:

$$L(\boldsymbol{\varphi})=E_{(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')\sim D}[[Q_{\varphi}(\boldsymbol{s},\boldsymbol{a})-(r(\boldsymbol{s},\boldsymbol{a})+\gamma E_{\boldsymbol{s}'}(Q_{\varphi'}(\boldsymbol{s}',\pi_{\theta}(\boldsymbol{s}'))))]^2] \tag{3}$$

其中,$Q_{\varphi'}$是根据Q_{φ}网络参数延时更新的目标网络,用于稳定学习算法[32].

Actor 通过最大化期望状态-动作值函数来更新参数化策略π_{θ}:

$$\max_{\boldsymbol{\theta}}J(\boldsymbol{\theta})=\max_{\boldsymbol{\theta}}E_{\boldsymbol{s}\sim D}[Q_{\varphi}(\boldsymbol{s},\pi_{\boldsymbol{\theta}}(\boldsymbol{s}))] \tag{4}$$

采用梯度方法优化式(4),其对应的梯度估计[30]为

$$\nabla_{\boldsymbol{\theta}}J(\boldsymbol{\theta})=E_{\boldsymbol{s}\sim D}[\nabla_{\boldsymbol{a}}Q_{\varphi}(\boldsymbol{s},\boldsymbol{a})\mid_{\boldsymbol{a}=\pi_{\boldsymbol{\theta}}(\boldsymbol{s})}\nabla_{\boldsymbol{\theta}}\pi_{\boldsymbol{\theta}}(\boldsymbol{s})] \tag{5}$$

在以上 Actor-Critic 强化学习框架下,离线强化学习的固定数据集是由任意行为策略$\mu_{\beta}(\boldsymbol{a}\mid\boldsymbol{s})$生成的经验样本构成.根据式(2)和(3)计算目标网络$Q_{\varphi'}$时,动作$\boldsymbol{a}'$是由学习策略π_{θ}决定的.由于学习策略π_{θ}和行为策略μ_{β}之间存在分布偏移,因此动作$\boldsymbol{a}'$可能位于离线数据集分布之外,该 OOD 动作对应的值函数估计将是错误的.假设 OOD 动作值函数被高估,通过贝尔曼自举操作计算的动作值函数也会随着高估,则学习策略将会沿着错误的值函数梯度方向更新.随着 Actor-Critic 算法的不断迭代更新,Q值函数估计误差将会不断累积增大,最终导致学习策略性能急剧下降.

2.2 不确定性估计

不确定性估计方法是估计Q值函数的认知不确定性,根据不确定性大小对学习策略进行不同程度的约束,在较高不确定性区域通过加大对策略学习目标的惩罚来实现学习策略的收紧约束,在较低不确定性区域通过减小对策略学习目标的惩罚来实现学习策略的放松约束[4,19].给定样本数据集D,假设$F_D(Q)$表示Q值函数分布,$U(F_D(Q))$表示不确定性度量,基于不确定性估计的策略学习目标如下:

$$J_U(\boldsymbol{\theta})=E_{\boldsymbol{s}\sim D}[E_{Q_{\varphi}\sim F_D(\cdot)}Q_{\varphi}(\boldsymbol{s},\pi_{\boldsymbol{\theta}}(\boldsymbol{s}))-\alpha U(F_D(Q))] \tag{6}$$

其中,α为不确定性惩罚系数.

式(6)中的不确定性度量主要依赖于不确定性估计方法的选择,自举集成[20]作为一种增强在线强化学习探索能力的不确定性估计方法,通过对多个随机初始化Q网络进行集成来学习值函数,目前在离线强化学习中得到了广泛应用.

3 基于不确定性估计的离线确定型 Actor-Critic

在离线强化学习中,式(6)中策略优化的不确定性度量一般是自举集成的下置信界估计.但是,经验表明这种不确定性度量方法对于学习策略约束往往过于松弛,不足以预防 OOD 动作对其性能的影响[15].为获得较好的性能,基于自举集成不确定性估计的策略学习通常与值函数正则化方法结合使用,在值函数学习中进一步惩罚 OOD 动作值函数[29].与基于自举集成不确定性估计的策略学习方法不同,本文利用自举集成不确定性估计方法学习Q值函数.考虑到 OOD 动作有可能位于离线数据集的分布内,如果能够利用Q神经网络的泛化特点准确地估计 OOD 动作值函数,很可能会学到一个超越次优行为策略的策略,而且离线强化学习对 OOD 动作值函数的不确定性也具有较高的保真度要求.因此,本文从提高Q值函数的泛化能力和增强 OOD 动作值函数估计的可靠性来考虑,将自举集成不确定性估计方法和 OOD 采样方法[28,33]相结合,在训练数据集内动作值函数的同时泛化学习 OOD 动作值函数,与直接惩罚 OOD 动作值函数的保守方法相比,本文方法可以平衡值函数的泛化与保守.

3.1 不确定性估计算子

定义 1. 假设有N个参数化神经网络用于拟合自举Q值函数,$\hat{\Gamma}_{\mathrm{UE}}Q(\boldsymbol{s},\boldsymbol{a})$表示$Q$值函数的不确

定性估计算子,基于不确定性估计方法计算的第 n 个值函数 Q_{φ_n} 的不确定性估计算子定义为

$$\hat{\Gamma}_{\mathrm{UE}}Q_{\varphi_n}(\boldsymbol{s},\boldsymbol{a})=\begin{cases}r(\boldsymbol{s},\boldsymbol{a})+\gamma E_{\boldsymbol{s}'\sim p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}),\boldsymbol{a}'=\pi_{\boldsymbol{\theta}}(\boldsymbol{s}')}[y(\boldsymbol{s}',\boldsymbol{a}')\\ -\alpha_{\mathrm{in}}U(Q_{\varphi'_n}(\boldsymbol{s}',\boldsymbol{a}'))], & (\boldsymbol{s},\boldsymbol{a})\in D\\ Q_{\varphi_n}(\boldsymbol{s},\boldsymbol{a})-\\ \alpha_{\mathrm{ood}}U(Q_{\varphi_n}(\boldsymbol{s},\boldsymbol{a})), & \boldsymbol{s}\in D,\boldsymbol{a}=\pi_{\boldsymbol{\theta}}(\boldsymbol{s})\notin D\end{cases} \tag{7}$$

$$y(\boldsymbol{s}',\boldsymbol{a}')=\eta\min_{n=1,\cdots,N}Q_{\varphi'_n}(\boldsymbol{s}',\boldsymbol{a}')+(1-\eta)\max_{n=1,\cdots,N}Q_{\varphi'_n}(\boldsymbol{s}',\boldsymbol{a}') \tag{8}$$

定义 1 将动作分为数据集内动作 $\boldsymbol{a}\in D$ 和数据集外动作 $\boldsymbol{a}=\pi_{\boldsymbol{\theta}}(\boldsymbol{s})$. 针对数据集内的状态-动作对 $(\boldsymbol{s},\boldsymbol{a})\in D$,$\alpha_{\mathrm{in}}$ 表示数据集内动作值函数的不确定性惩罚系数,$Q_{\varphi_n}(\boldsymbol{s},\boldsymbol{a})$ 的不确定性估计算子为贝尔曼自举操作形式. 相较于自举集成 DQN[20],不确定性估计算子有两处不同,一是自举操作中的 $\boldsymbol{a}'=\pi_{\boldsymbol{\theta}}(\boldsymbol{s}')$ 会引入 OOD 动作,此处采用集成不确定性估计对其进行惩罚. 二是为了减小贝尔曼自举操作中引入的 Q 值过估偏差,类似于 BCD 和 BEAR[9-10],目标值 $y(\boldsymbol{s}',\boldsymbol{a}')$ 采用最大最小的凸组合形式计算,如式(8)所示,其中 $0\leqslant\eta\leqslant1$,用于权衡 Q 值的最大化高估和最小化低估问题,一般最小化占比较大. 值得注意的是本文的 Q 目标值计算与文献[28—29]均不同,文献[28—29]中 $y(\boldsymbol{s}',\boldsymbol{a}')=Q_{\varphi'_n}(\boldsymbol{s}',\boldsymbol{a}')$,并且文献[29]中的目标值计算不引入不确定性估计项. 针对数据集外由学习策略采样的动作 $\boldsymbol{a}=\pi_{\boldsymbol{\theta}}(\boldsymbol{s})$,由于不能与环境交互获得立即回报进行自举操作,借鉴文献[28]方法,根据自举集成不确定性估计和 OOD 采样方法构造一个伪目标,将该伪目标作为 OOD 采样动作值函数的不确定性估计算子. α_{ood} 表示 OOD 采样动作值函数的不确定性惩罚系数,对于数据集分布附近的 OOD 采样动作,α_{ood} 设置较小;对于远离数据集分布的 OOD 采样动作,α_{ood} 设置较大. 相较于文献[33]的 OOD 采样方法,本文所用的 OOD 采样方法不需要额外引入行为模型的训练.

直观地,式(7)中的不确定性用自举集成 Q 值函数的标准差衡量:

$$U(Q_{\varphi_n}(\boldsymbol{s},\boldsymbol{a}))=\mathrm{std}(Q_{\varphi_n}(\boldsymbol{s},\boldsymbol{a}))=\sqrt{\frac{1}{n}\sum_{n=1}^{N}(Q_{\varphi_n}(\boldsymbol{s},\boldsymbol{a})-\bar{Q}_{\varphi}(\boldsymbol{s},\boldsymbol{a}))^2} \tag{9}$$

其中,$\bar{Q}_{\varphi}(\boldsymbol{s},\boldsymbol{a})$ 为均值.

理论上,离线强化学习的有效性依赖于不确定性估计的选取,Jin 等人[34]定义了一种 ξ-不确定性估计,并证明了在线性 MDP 下,基于 ξ-不确定性估计的悲观值迭代方法是信息理论最优. 以下将利用 ξ-不确定性估计理论分析不确定性估计算子的悲观性.

定义 2.(ξ-不确定性估计[34])对于所有 $(\boldsymbol{s},\boldsymbol{a})\in\mathcal{S}\times\mathcal{A}$,如果满足如下概率事件:

$$P_D\{|(\hat{\mathrm{B}}V)(\boldsymbol{s},\boldsymbol{a})-(\mathrm{B}V)(\boldsymbol{s},\boldsymbol{a})|\leqslant H(\boldsymbol{s},\boldsymbol{a})\}\geqslant1-\xi \tag{10}$$

则 $H(\boldsymbol{s},\boldsymbol{a})$ 是 ξ-不确定性估计.

$\hat{\mathrm{B}}V$ 是基于数据集的经验贝尔曼,$\mathrm{B}V$ 是真实贝尔曼,即

$$(\mathrm{B}V)(\boldsymbol{s},\boldsymbol{a})=E_{\boldsymbol{s}'}[r(\boldsymbol{s},\boldsymbol{a})+\gamma V(\boldsymbol{s}')] \tag{11}$$

由 ξ-不确定性估计可知,真实贝尔曼 $\mathrm{B}V$ 至少以 $1-\xi$ 的概率位于经验贝尔曼 $\hat{\mathrm{B}}V$ 的不确定性估计 $H(\boldsymbol{s},\boldsymbol{a})$ 的区间,即

$$(\hat{\mathrm{B}}V)(\boldsymbol{s},\boldsymbol{a})-H(\boldsymbol{s},\boldsymbol{a})\leqslant(\mathrm{B}V)(\boldsymbol{s},\boldsymbol{a})\leqslant(\hat{\mathrm{B}}V)(\boldsymbol{s},\boldsymbol{a})+H(\boldsymbol{s},\boldsymbol{a}) \tag{12}$$

因此,可以根据经验贝尔曼和 ξ-不确定性估计近似真实贝尔曼,当 Q 估计值为真实贝尔曼的下界时,即 $\hat{Q}(\boldsymbol{s},\boldsymbol{a})=(\hat{\mathrm{B}}V)(\boldsymbol{s},\boldsymbol{a})-H(\boldsymbol{s},\boldsymbol{a})$,通过不断迭代学习即可获得一种悲观 Q 值估计[34].

在确定型强化学习中,由于 $V(\boldsymbol{s}')=Q(\boldsymbol{s}',\pi(\boldsymbol{s}'))$,根据式(2)和式(11)可得

$$(\mathrm{B}V)(\boldsymbol{s},\boldsymbol{a})=\mathrm{B}Q(\boldsymbol{s},\boldsymbol{a}) \tag{13}$$

将式(13)代入式(10)中,得

$$P_D\{|\hat{\mathrm{B}}Q(\boldsymbol{s},\boldsymbol{a})-\mathrm{B}Q(\boldsymbol{s},\boldsymbol{a})|\leqslant H(\boldsymbol{s},\boldsymbol{a})\}\geqslant1-\xi \tag{14}$$

定理 1. 假设经验贝尔曼满足如下式:

$$\hat{\mathrm{B}}Q(\boldsymbol{s},\boldsymbol{a})=\hat{\Gamma}_{\mathrm{UE}}Q(\boldsymbol{s},\boldsymbol{a})+\beta U(Q(\boldsymbol{s},\boldsymbol{a}))$$

并且不确定性惩罚系数 α_{in} 满足如下条件:

$$\alpha_{\mathrm{in}}\geqslant\frac{y(\boldsymbol{s}',\pi(\boldsymbol{s}'))-Q(\boldsymbol{s}',\pi(\boldsymbol{s}'))}{U(Q(\boldsymbol{s}',\pi(\boldsymbol{s}')))}$$

则 $\beta U(Q(\boldsymbol{s},\boldsymbol{a}))$ 是一个 ξ-不确定性估计,其中 β 表示 α_{in} 或 α_{ood}.

证明:详见附录 A.

由定理 1 可得

$$\hat{\Gamma}_{\mathrm{UE}}Q(\boldsymbol{s},\boldsymbol{a})\leqslant\mathrm{B}Q(\boldsymbol{s},\boldsymbol{a})\leqslant\hat{\Gamma}_{\mathrm{UE}}Q(\boldsymbol{s},\boldsymbol{a})+2\beta U(Q(\boldsymbol{s},\boldsymbol{a}))$$

,即 $\hat{\Gamma}_{\mathrm{UE}}Q(\boldsymbol{s},\boldsymbol{a})$ 是真实贝尔曼的下

界. 因此,不确定性估计算子 $\hat{\Gamma}_{\mathrm{UE}}Q(\boldsymbol{s},\boldsymbol{a})$ 至少以 $1-\xi$ 概率获得悲观 Q 值估计. 此外,根据 $\hat{\Gamma}_{\mathrm{UE}}Q(\boldsymbol{s},\boldsymbol{a})$ 学习得到的 Q 值函数也是最优 Q^* 值函数的一种悲观估计,如定理 2 所示.

定理 2. 根据 $\hat{\Gamma}_{\mathrm{UE}}Q(\boldsymbol{s},\boldsymbol{a})$ 学习得到的 Q 值函数是最优 Q^* 值函数的一种悲观估计,即 $\hat{\Gamma}_{\mathrm{UE}}Q(\boldsymbol{s},\boldsymbol{a})<Q^*(\boldsymbol{s},\boldsymbol{a})$.

证明:详见附录 B.

3.2 基于不确定性估计的离线确定型 Actor-Critic

将定义 1 的不确定性估计算子应用于在线确定型 Actor-Critic 算法 TD3[35] 中,设计一种基于不确定性估计的离线确定型 Actor-Critic 算法,其整体框架如图 1 所示. 虚线上方部分表示数据集内动作值函数学习过程,其 Q 值函数不确定性估计算子由 Critic 目标网络实现. 虚线下方部分表示 OOD 采样动作值函数学习过程,其 OOD 采样由 Actor 主网络实现,Q 值函数不确定性估计算子由 Critic 主网络实现. Critic 学习损失由数据集内动作值函数损失 L_{in} 和 OOD 采样动作值函数损失 L_{ood} 加权构成,Critic 主网络参数根据 Critic 损失函数进行梯度更新,Actor 主网络参数根据 Critic 主网络计算的 Q 值进行梯度更新.

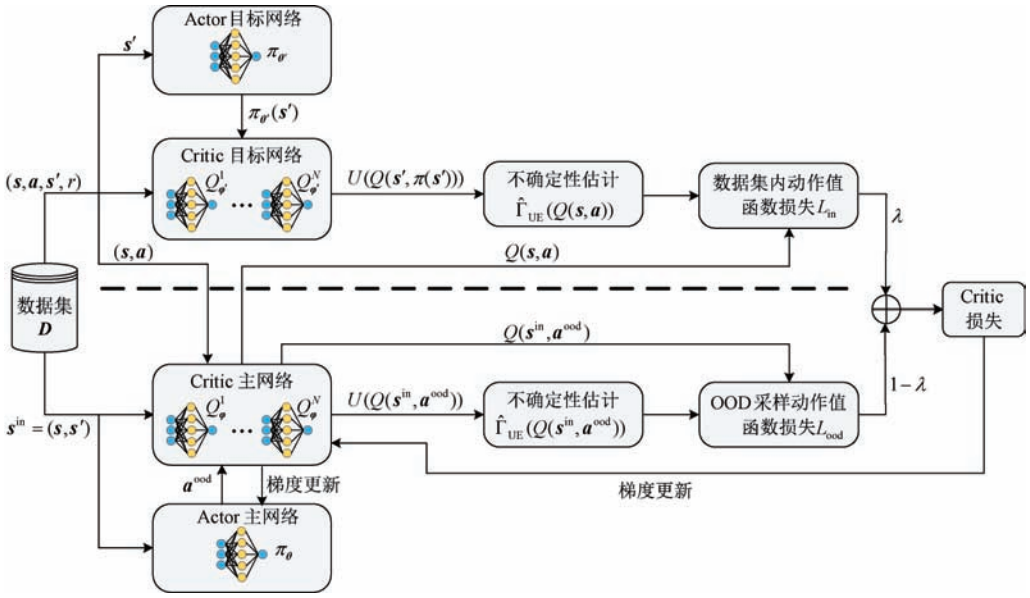


图 1 ODACUE 的整体框架

假设 $\boldsymbol{a}$ 表示数据集内动作,$\boldsymbol{a}^{\mathrm{ood}}$ 表示 OOD 采样动作. $\boldsymbol{a}^{\mathrm{ood}}$ 有可能位于数据集分布内,也有可能位于数据集分布外. 为了采用具有较高保真的不确定性估计方法惩罚 OOD 动作,同时提升 Q 值函数的学习泛化性能,Critic 对两类动作 $\boldsymbol{a}$ 和 $\boldsymbol{a}^{\mathrm{ood}}$ 值函数同时进行学习. 然后,利用不确定性估计算子估计 Q 目标值,采用凸组合形式对两类动作值函数对应的损失进行加权,则第 n 个 Q_{φ_n} 值函数对应的损失函数如下:

$$L_{\mathrm{critic}}(Q_{\varphi_n})=\lambda E_{(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}',r)\sim D}[(Q_{\varphi_n}(\boldsymbol{s},\boldsymbol{a})-\hat{\Gamma}_{\mathrm{UE}}(Q_{\varphi_n}(\boldsymbol{s},\boldsymbol{a})))^2]+(1-\lambda)E_{\boldsymbol{s}^{\mathrm{in}}\sim D}[(Q_{\varphi_n}(\boldsymbol{s}^{\mathrm{in}},\boldsymbol{a}^{\mathrm{ood}})-\hat{\Gamma}_{\mathrm{UE}}(Q_{\varphi_n}(\boldsymbol{s}^{\mathrm{in}},\boldsymbol{a}^{\mathrm{ood}})))^2] \tag{15}$$

其中,λ 用于平衡数据集内动作值函数和 OOD 动作值函数之间的贡献;$\boldsymbol{s}^{\mathrm{in}}=(\boldsymbol{s},\boldsymbol{s}')\sim D$ 来自于数据集内,用于增加 $\boldsymbol{a}^{\mathrm{ood}}$ 的样本数和多样性.

$$\hat{\Gamma}_{\mathrm{UE}}(Q_{\varphi_n}(\boldsymbol{s},\boldsymbol{a}))=r(\boldsymbol{s},\boldsymbol{a})+\gamma[y(\boldsymbol{s}',\boldsymbol{a}'_{\mathrm{ood}})-\alpha_{\mathrm{in}}U(Q_{\varphi'_n}(\boldsymbol{s}',\boldsymbol{a}'_{\mathrm{ood}}))]\big|_{\boldsymbol{a}'_{\mathrm{ood}}=\pi_{\theta'}(\boldsymbol{s}')} \tag{16}$$

为了稳定策略更新,式(16)中的 $\boldsymbol{a}'_{\mathrm{ood}}$ 是由策略目标网络 $\pi_{\theta'}$ 输出的,$\boldsymbol{\theta}'$ 为策略目标网络参数.

$$\hat{\Gamma}_{\mathrm{UE}}(Q_{\varphi_n}(\boldsymbol{s}^{\mathrm{in}},\boldsymbol{a}^{\mathrm{ood}}))=Q_{\varphi_n}(\boldsymbol{s}^{\mathrm{in}},\boldsymbol{a}^{\mathrm{ood}})-\alpha_{\mathrm{ood}}U(Q_{\varphi_n}(\boldsymbol{s}^{\mathrm{in}},\boldsymbol{a}^{\mathrm{ood}})) \tag{17}$$

在实际实施中,借鉴文献[28]做法,为了稳定训练初始阶段的性能,对 $\hat{\Gamma}_{\mathrm{UE}}(Q_{\varphi_n}(\boldsymbol{s}^{\mathrm{in}},\boldsymbol{a}^{\mathrm{ood}}))$ 进行了裁剪:当 $\hat{\Gamma}_{\mathrm{UE}}(Q_{\varphi_n}(\boldsymbol{s}^{\mathrm{in}},\boldsymbol{a}^{\mathrm{ood}}))\leqslant 0$ 时,将其裁剪为 0.

基于以上估计的 Q 值函数和式(4),Actor 通过最大化如下最小集成 Q 值函数形式更新策略[11]:

$$J(\boldsymbol{\theta})=E_{s\sim D}[\min_{n=1,\cdots,N}Q_{\varphi_n}(\boldsymbol{s},\pi_{\boldsymbol{\theta}}(\boldsymbol{s}))] \tag{18}$$

算法 1. 基于不确定性估计的离线确定型 Actor-Critic(ODACUE).

输入:离线数据集 D

输出:策略 $\pi_{\boldsymbol{\theta}}$

1. 初始化:集成 Q 网络个数 N,N 个 Critic 主网络 Q_{φ_n} 的参数 $\boldsymbol{\varphi}_n$,Actor 策略主网络 $\pi_{\boldsymbol{\theta}}$ 的参数 $\boldsymbol{\theta}$,N 个 Critic 目标

网络参数 $\boldsymbol{\varphi}'_n \leftarrow \boldsymbol{\varphi}_n$，Actor 策略目标网络参数 $\boldsymbol{\theta}' \leftarrow \boldsymbol{\theta}$，离线数据集 D，折扣因子 γ，小批次大小 $|B_T|$，权重系数 η、λ，不确定性惩罚系数 α_{in}、α_{ood}，目标网络延迟更新率 τ，OOD 动作裁剪范围 c，策略更新频率 f，迭代训练总次数 T.

2. FOR $t = 1$ TO T DO

3. 从数据集 D 中采样小批次样本 $B_T = \{\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}', d\}$，其中 d 表示一个回合结束标志；

4. 根据式(9)和 Critic 目标网络，计算不确定性 $U(Q_{\boldsymbol{\varphi}'_n}(\boldsymbol{s}', \pi_{\boldsymbol{\theta}'}(\boldsymbol{s}')))$；

5. 根据式(8)和式(16)，计算 $Q_{\boldsymbol{\varphi}_n}(\boldsymbol{s}, \boldsymbol{a})$ 的目标值；

6. 针对数据集内的状态 $\boldsymbol{s}^{\text{in}}$，采样 M 个 OOD 动作 $\boldsymbol{a}^{\text{ood}} \sim \pi_{\boldsymbol{\theta}}(\boldsymbol{s}^{\text{in}}) + \varepsilon$；

7. 根据式(9)和 Critic 主网络，计算不确定性 $U(Q_{\boldsymbol{\varphi}_n}(\boldsymbol{s}^{\text{in}}, \boldsymbol{a}^{\text{ood}}))$；

8. 根据式(17)，计算 $Q_{\boldsymbol{\varphi}_n}(\boldsymbol{s}^{\text{in}}, \boldsymbol{a}^{\text{ood}})$ 的伪目标值；

9. 最小化式(15)，采用梯度下降法更新 Critic 主网络；

10. IF t mod f THEN

11. 基于梯度更新策略参数 $\boldsymbol{\theta}$：

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \frac{\sum \nabla_{\boldsymbol{a}} \left[\min_{n=1,\cdots,N} Q_{\boldsymbol{\varphi}_n}(\boldsymbol{s}, \boldsymbol{a}) \right] |_{\boldsymbol{a}=\pi_{\boldsymbol{\theta}}(\boldsymbol{s})} \nabla_{\boldsymbol{\theta}} \pi_{\boldsymbol{\theta}}(\boldsymbol{s})}{|B_T|}$$

12. 更新目标网络参数：

$$\boldsymbol{\theta}' \leftarrow \tau\boldsymbol{\theta} + (1-\tau)\boldsymbol{\theta}'$$

$$\boldsymbol{\varphi}'_n \leftarrow \tau\boldsymbol{\varphi}_n + (1-\tau)\boldsymbol{\varphi}'_n, n = 1, \cdots, N$$

13. END IF

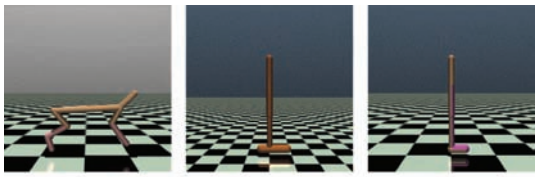
14. END FOR

综上所述，基于不确定性估计的离线确定型 Actor-Critic 的伪代码如算法 1 所示. 与 TD3 不同的是，为了减小下一个状态-动作对值函数的不确定性，在利用目标策略网络采样下一个状态动作时不引入随机噪声. 此外，为了增加 OOD 采样动作的多样性，ODACUE 在采样过程中引入裁剪噪声，即 $\boldsymbol{a}^{\text{ood}} \sim \pi_{\boldsymbol{\theta}}(\boldsymbol{s}^{\text{in}}) + \varepsilon, \varepsilon \sim \text{clip}(N(0, \sigma^2), -c, c)$. 其中，$\sigma$ 为方差，c 为裁剪范围.

4　实验结果及分析

4.1　实验设置

在 D4RL[36] 基准上比较了一些先进的算法：值函数惩罚算法 CQL[15]、策略约束的确定型策略梯度算法 TD3+BC[30]、基于 dropout 的不确定性估计算法 UWAC[19]、策略约束的随机策略梯度算法 Fisher-BRC[37] 和 BEAR[10]. 选用如图 2 所示的 3 个 MuJoCo 连续控制任务进行测试，分别包含 random、medium、medium-replay、medium-expert 和 expert 等 5 个不同质量等级的离线数据集. 为确保对比的公平性，采用最新发布的“-V2”版本数据集对所有算法进行测试对比，重新运行了所有对比算法的源代码. 实验采用的计算机硬件配置如下：处理器为 Intel Core i9-9900K，主频为 3.60 GHz，内存为 32 GB，GPU 为 Nvidia GeForce RTX 2060. 软件环境配置主要包括 Python 3.8.13、Pytorch 1.11.0、D4rl 1.1、Mujoco-py 2.0.2.8 和 Gym 0.17.0.



(a) halfcheetah-v2　(b) hopper-v2　(c) walker2d-v2

图 2　测试任务

实验中，集成 Q 网络个数为 $N = 10$；针对每个状态，OOD 采样动作数为 $M = 2$；通过试凑法选择不确定性惩罚系数 α_{in}、α_{ood} 和权重系数 λ；Actor 和 Critic 网络隐藏层数量、各隐藏层神经元数量及激活函数等结构参数，Actor 和 Critic 学习率、目标网络延迟更新率和策略网络更新频率等学习参数与 TD3 算法设置相同，所有任务的具体运行参数设置如表 1 所示；训练 100 万步，每 5000 步评估一次；每次评估运行 10 个回合，共运行 4 个随机种子；最后 10 次评估结果以标准平均分形式显示在表 2 中，其中标准平均分的计算公式如下：

$$\text{标准分} = 100 \times \frac{\text{得分} - \text{随机得分}}{\text{专家得分} - \text{随机得分}}.$$

表 1　ODACUE 的参数设置

参数	值
Critic 网络隐藏层数量	2
Critic 网络各隐藏层神经元数量	256
Critic 网络激活函数	ReLU
Actor 网络隐藏层数量	2
Actor 网络各隐藏层神经元数量	256
Actor 网络激活函数	ReLU
优化器	Adam
小批次大小	256
Critic 网络学习率	0.0003
Actor 网络学习率	0.0003
折扣因子	0.99
目标网络延迟更新率	0.005
策略网络更新频率	2
集成 Q 网络个数	10
OOD 采样动作数	2
η	0.75
c	0.2(非专家数据集)，0.05(专家数据集)
λ	0.4～0.85
α_{in}	0.01
α_{ood}	0.1(非专家数据集)，0.5(专家数据集)

标准分为 0 表示学习的性能类似于随机策略获得的平均回报,标准分为 100 表示学习的性能类似于专家策略获得的平均回报. ODACUE 相对于对比算法的性能提升率为

$$\text{性能提升率}=100\%\times\frac{\text{ODACUE 总得分}-\text{对比算法总得分}}{\text{对比算法总得分}}.$$

4.2 对比实验

图 3 给出了 ODACUE 和对比算法在 11 个数据集任务中的学习曲线,由表 2 和图 3 可以看出:(1)在 random、medium 和 medium-replay 三类不包含专家样本的数据集任务中,除了 Fisher-BRC 在 halfcheetah-random-v2 和 hopper-medium-v2 任务中得分最高外,ODACUE 的得分均高于其余对比算法. 这是由于尽管这些数据集任务的策略是随机策略或次优策略,但是 ODACUE 能够通过不确定性估计和 OOD 采样来避免值函数估计的过度保守,从而学到数据集之外的最优策略;(2)在 walker2d-medium-expert-v2 和 walker2d-expert-v2 两类含有专家样本的任务中,ODACUE 可以获得最高得分. 这是因为在这些含有专家样本的数据集任务中,通过对 OOD 采样动作值函数进行较大的不确定惩罚且选取较小的 λ,可以更好地利用数据集内的专家样本学习,与先进的基线算法类似,获得的学习性能均优于专家策略. 此外,相较于对比算法,ODACUE 的性能提升率最低为 9.56%,最高达 64.92%.

4.3 参数分析

为测试 ODACUE 的参数敏感性,选择 halfcheetah-medium-v2、hopper-medium-replay-v2 、walker2d-medium-expert-v2 和 walker2d-expert-v2 任务进行实验,实验运行 100 万步. 权重系数 λ 主要用于平衡数据集内动作值函数和 OOD 动作值函数的训练. 当 $\lambda=1$ 时,仅根据不确定性惩罚估计数据集内动作值函数,学到的策略与不确定性估计准确度有关,一般获得的不确定性惩罚过于松弛,其性能参见图 8 的 ODACUE-NOODS. 当 $\lambda=0$ 时,值函数估计主要由 OOD 动作主导,无法正确学习. 图 4 显示了权重系数 λ 对 ODACUE 性能的影响,可以看出:(1)在 halfcheetah-medium-v2 任务中,λ 越大则 ODACUE 的性能越好. 这是因为中等质量数据集分布比较狭窄,λ 越大对应的 OOD 动作值函数损失分配权重越小,从而可以利用较好的 OOD 样本学习更优的策略;(2)在 hopper-medium-replay-v2 任务中,λ 取值较小则 ODACUE 的性能越好. 这是由于中等回放数据集分布范围较大,λ 取值较小,可以较多地利用数据集内样本学习更优的策略;(3)对于 walker2d-medium-expert-v2 和 walker2d-expert-v2 任务来说,当 λ 取值区间为 0.4 ~ 0.65 时,ODACUE 的性能较好. 这是因为含有专家样本的数据集质量高,λ 越小对应的数据集内动作值函数损失分配权重越小,可以更多地利用数据集内的专家样本进行学习.

图 5 显示了集成 Q 网络个数对 ODACUE 性能的影响,可以看出:在 halfcheetah-medium-v2 任务中,当 $N=4$ 时,ODACUE 的性能稍差,当 $N=2$ 或 $N\geqslant 6$ 时,ODACUE 的性能均较好且相近;在 hopper-medium-replay-v2 和 walker2d-medium-expert-v2 任务中,当 $N\geqslant 4$ 时,ODACUE 的性能均较好且相近;在 walker2d-expert-v2 任务中,N 取值越大则 ODACUE 性能越好. 综合 4 个任务,当 $N\geqslant 8$ 时,ODACUE 可以获得更好的性能,说明了随着集成 Q 网络个数的增多,Q 值函数的学习泛化能力和不确定性估计也随之提高.

表 2 离线强化学习算法在最后 10 次评估上的标准平均分

任务	离线数据集	UWAC	BEAR	CQL	Fisher-BRC	TD3+BC	ODACUE
halfcheetah	random	2.3	2.3	15.6	**26.9**	10.4	25.0
	medium	42.5	42.5	46.9	48.1	48.3	**64.2**
	medium-replay	36.1	36.0	45.2	46.2	44.5	**57.5**
hopper	random	2.7	17.3	7.4	16.3	8.5	**31.7**
	medium	51.9	50.8	62.0	**94.1**	58.0	72.7
	medium-replay	27.4	30.2	90.6	93.5	59.1	**101.5**
walker2d	random	1.3	1.4	4.3	6.1	2.4	**11.1**
	medium	69.1	67.3	82.0	82.7	83.9	**90.4**
	medium-replay	27.9	23.5	79.4	73.6	79.6	**93.0**
	medium-expert	99.5	92.6	109.5	109.6	110.1	**112.1**
	expert	108.2	108.4	109.5	108.7	110.1	**114.1**
总得分		468.9	472.3	652.4	705.8	614.9	**773.3**
性能提升率		64.92%	63.73%	18.53%	9.56%	25.76%	

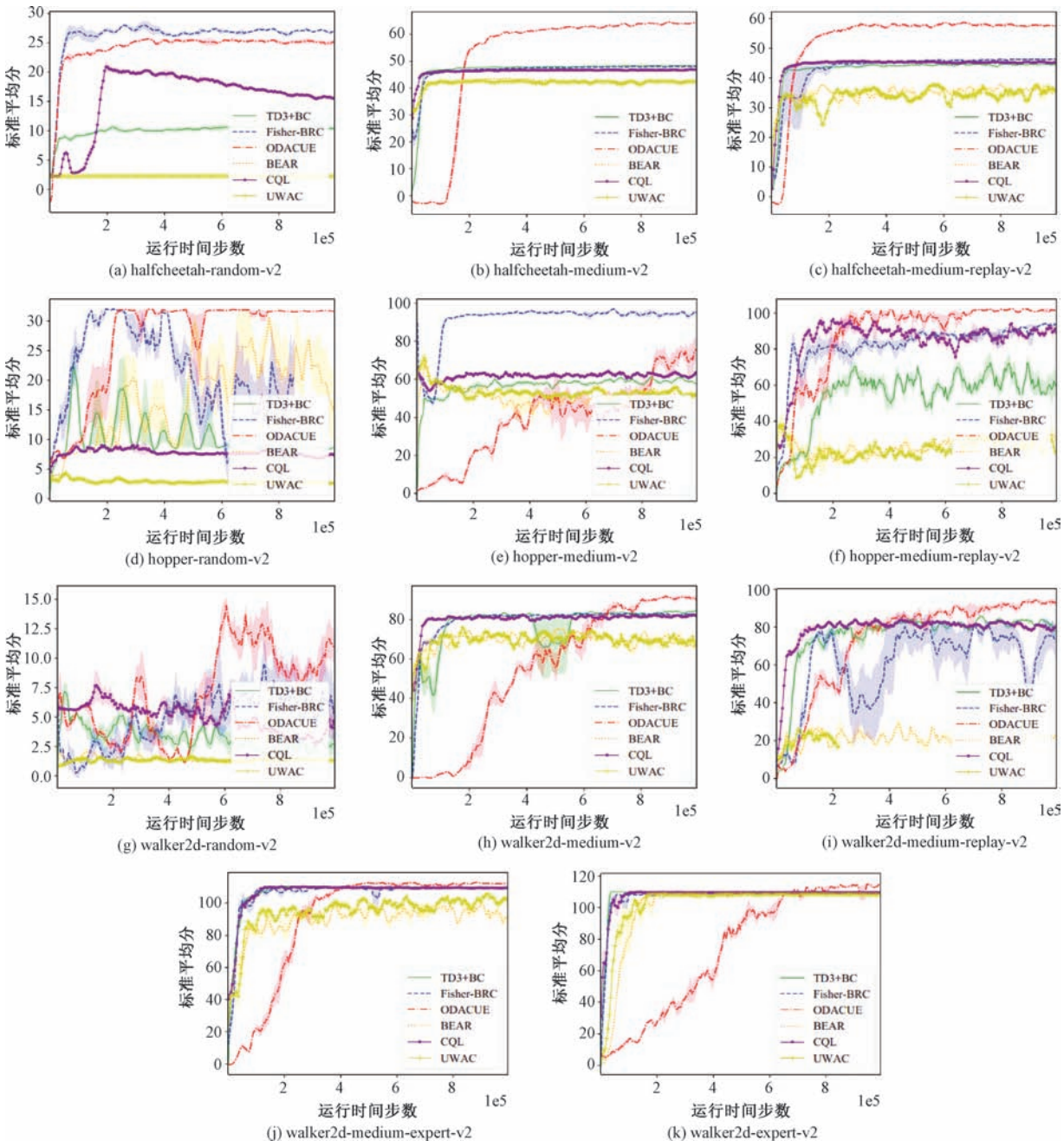


图 3　ODACUE 与离线强化学习基线方法的学习曲线对比

不确定性惩罚系数 α_{ood} 主要用于惩罚 OOD 动作值函数的不确定性估计大小. 当不确定性估计较大时,OOD 动作具有高不确定性,选用较大的 α_{ood} 惩罚 OOD 动作值函数,以减小 OOD 动作对值函数估计的影响. 当 OOD 动作的不确定性较小时,选用较小的 α_{ood} ,从而可以充分利用 OOD 动作提升 Q 值函数的泛化能力. 固定数据集内动作值函数学习的不确定性惩罚系数 $\alpha_{\text{in}}=0.01$,图 6 显示了 OOD 采样动作值函数的不确定性惩罚系数 α_{ood} 对 ODACUE 性能的影响,可以看出:(1)在 halfcheetah-medium-v2 和 hopper-medium-replay-v2 任务中,随着 α_{ood} 的增大,ODACUE 性能均逐渐下降,且 $\alpha_{\text{ood}}=0.1$ 时 ODACUE 的性能表现最好. 这是因为这两个数据集任务的策略是次优策略,通过设置较小的不确定性惩罚系数 α_{ood} 可以放松 OOD 约束,更好地学习 OOD 采样动作值函数,从而提高 OOD 动作不确定性估计可信度,利用可信的 OOD 动作值函数学习更优的策略;(2)在 walker2d-medium-expert-v2 和 walker2d-expert-v2 任务中,随着 α_{ood} 的增大,ODACUE 性能均有所提高,$\alpha_{\text{ood}}=0.5$ 时的性能表现最好,

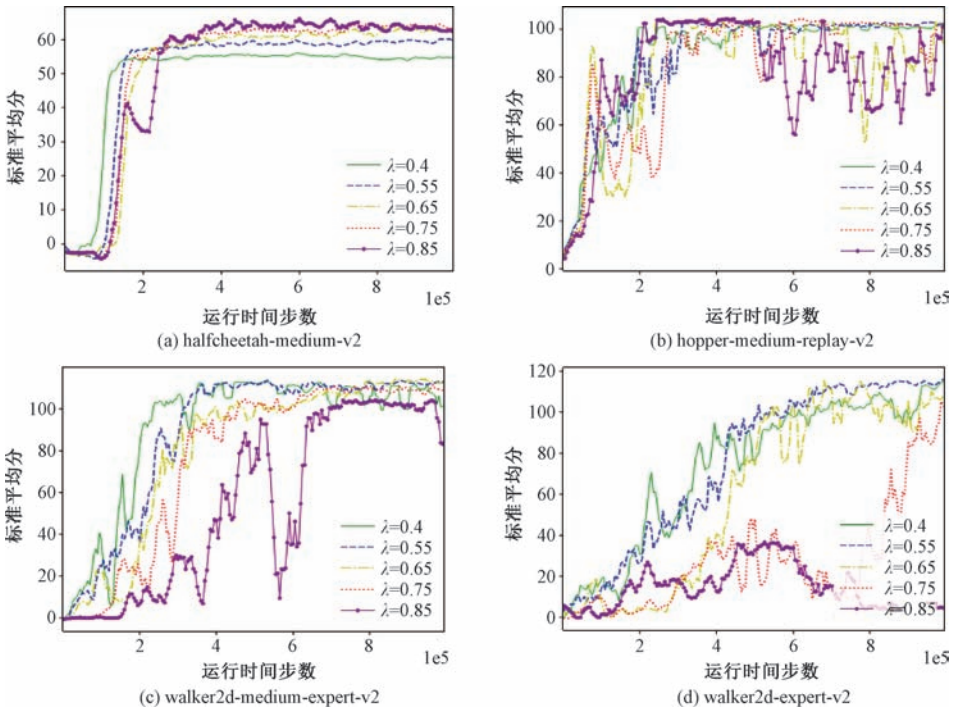


图 4 权重系数 λ 对 ODACUE 性能的影响

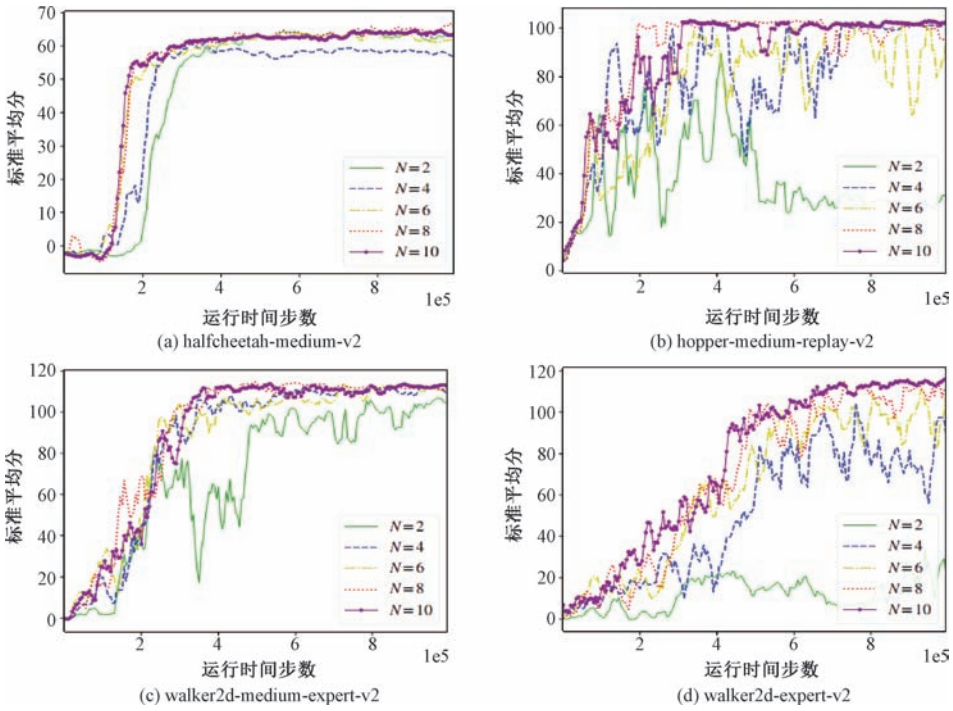


图 5 集成 Q 网络个数对 ODACUE 性能的影响

$\alpha_{ood}=1.0$ 时的性能在训练后期略有下降. 这是因为含有专家样本的数据集策略优于其他质量等级的数据集策略,在学习中需要更多地利用专家样本数据来学习 Q 值函数,因此需要设置较大的 α_{ood} 来惩罚 OOD 动作的影响. 但是,若 $\alpha_{ood}=1.0$,由于过度惩罚则会导致性能有所下降.

固定集成 Q 网络个数 $N=10$,图 7 显示了 λ 和 α_{ood} 对 ODACUE 性能的关联影响,图中数据表示的是最后 10 次评估的标准平均分. 可以看出:(1)在中等数据集任务中,当 λ 越大,α_{ood} 越小时,ODACUE 性能越好. 当 $0.65\leqslant\lambda\leqslant0.85$ 且 $\alpha_{ood}=0.1$ 时,ODACUE 可以获得稳定的性能,说明了 ODACUE 对该区间参数变化具有较强的鲁棒性;(2)在中等回放数据集任务中,当 λ 越小,α_{ood} 越小时,ODACUE 性能越

好.当 $0.75 \leqslant \lambda \leqslant 0.85$ 且 $\alpha_{ood} \geqslant 0.3$ 时,ODACUE 可以获得稳定的性能;(3)在专家数据集任务中,当 λ 越小,α_{ood} 越大时,ODACUE 性能越好.当 $0.55 \leqslant \lambda \leqslant 0.65$ 且 $\alpha_{ood} \geqslant 0.5$ 时,ODACUE 性能更稳定.

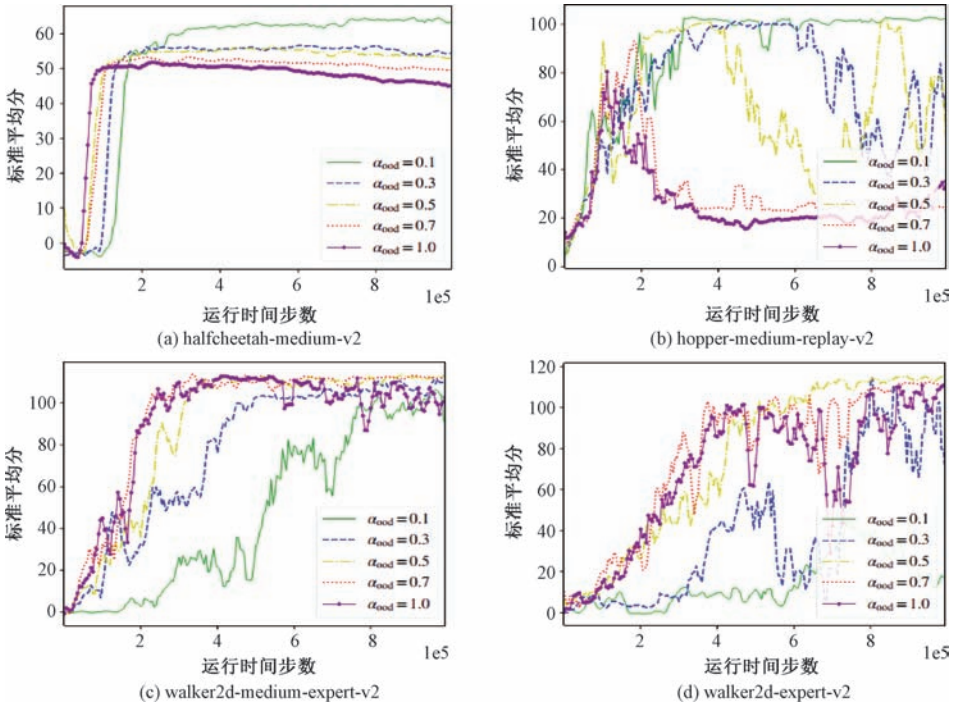


图 6　不确定性惩罚系数 α_{ood} 对 ODACUE 性能的影响

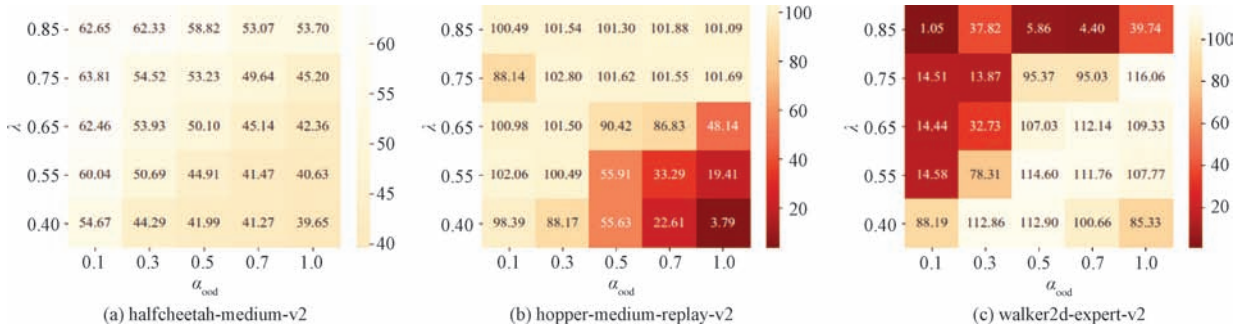


图 7　λ 和 α_{ood} 对 ODACUE 性能的关联影响

4.4　Q 值估计与消融实验

为测试 ODACUE 的各组成部件对其性能的影响,通过移除 ODACUE 不同的组件进行对比实验,实验中未移除的组件参数与 ODACUE 设置相同.不同组件对应的方法如下:

ODACUE-OODT0:将 ODACUE 中的 OOD 采样动作值函数的目标值设置为 0,即 $\hat{\Gamma}_{UE}(Q_{\varphi_n}(\boldsymbol{s}^{in}, \boldsymbol{a}^{ood}))=0$ 表示对 ODACUE 的 OOD 采样动作值函数进行过度惩罚,认为 OOD 采样动作都是不好的.

ODACUE-NOODS:移除 ODACUE 中的 OOD 采样组件,即无 OOD 动作值函数的学习,只根据数据集内样本学习 Q 值函数,Critic 学习损失函数为数据集内动作值函数损失:

$$L_{in}=E_{(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}',r)\sim D}[(Q_{\varphi_n}(\boldsymbol{s},\boldsymbol{a})-\hat{\Gamma}_{UE}(Q_{\varphi_n}(\boldsymbol{s},\boldsymbol{a})))^2]$$

ODACUE-NOOD-NUE:移除 ODACUE 中的 OOD 采样和不确定性估计组件,即无 OOD 动作值函数的学习和数据集内动作值函数的不确定性惩罚,相应的 Critic 学习损失函数为式(3),Actor 学习目标为式(4).

参考文献[33]的评估指标,图 8 给出了 ODACUE 各组件在标准平均分、离线数据 Q 值和学习策略 Q 值方面的性能,其中,离线数据 Q 值表示的是 $Q(\boldsymbol{s},\boldsymbol{a})$,$(\boldsymbol{s},\boldsymbol{a}) \in D$,学习策略 Q 值表示的是 $Q(\boldsymbol{s}, \pi(\boldsymbol{s}))$.从图 8(a)、8(d)、8(g)和 8(j)中可以看出:在 4 个任务中,ODACUE 性能最优;除了 ODACUE-NOODS 在 halfcheetah-medium-v2 和 hopper-medium-replay-v2 任务中的性能表现略好外,OD-

ACUE-OODT0、ODACUE-NOODS 和 ODACUE-NOOD-NUE 的性能均极度下降. 结合图 8 中离线数据 Q 值和学习策略 Q 值的性能来分析原因，可知：

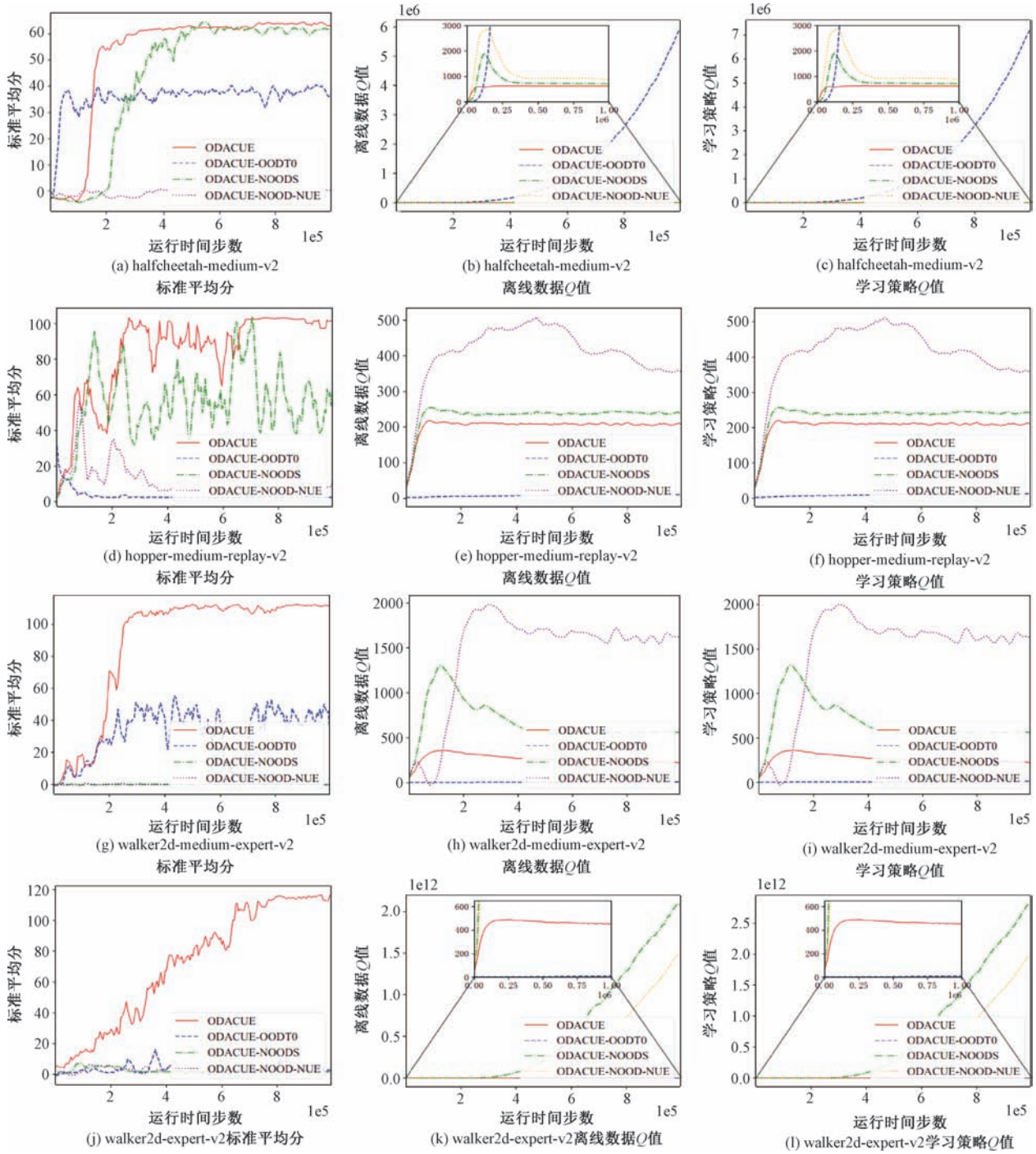


图 8 ODACUE 消融性能对比

(1) ODACUE-OODT0 过度惩罚 OOD 动作值函数会导致学到过度悲观的 Q 值函数. 离线数据 Q 值和学习策略 Q 值除了在 halfcheetah-medium-v2 任务中发散外，在其余三个任务中均趋近于 0. 由此可见，过度悲观的 Q 值函数在学习过程中引入了很大的外推误差，导致策略学习性能下降.

(2) ODACUE-NOOD-NUE 在值函数学习中对 OOD 动作不做任何处理，Q 值函数往往容易被高估. 离线数据 Q 值和学习策略 Q 值均远远高于 ODACUE 的 Q 值，且在 walker2d-expert-v2 任务中发散. 由此可见，Q 值高估带来的外推误差导致了性能下降.

(3) 相较于 ODACUE-NOOD-NUE，ODAC-

UE-NOODS 根据不确定性估计惩罚数据集内动作值函数的 OOD 动作,可以减小 OOD 对值函数估计准确性的影响. 离线数据 Q 值和学习策略 Q 值除了在 walker2d-expert-v2 任务中发散外,在其余三个任务中均小于 ODACUE-NOOD-NUE 的 Q 值. 此外,在 halfcheetah-medium-v2 和 hopper-medium-replay-v2 任务中,ODACUE-NOODS 的 Q 值与 ODACUE 的 Q 值相差最小,说明此处对 OOD 动作进行了较大的不确定性惩罚,可以学习到略好的 Q 值,从而不确定性估计的准确性略好. 在 walker2d-medium-expert-v2 和 walker2d-expert-v2 任务中,ODACUE-NOODS 对 OOD 动作进行了较小的不确定性惩罚,不足以抑制 OOD 动作的影响,导致学到偏高估或者发散的 Q 值,从而外推误差偏大导致性能下降. 由此可见,不确定性估计的准确性对 Q 值函数学习具有重要影响.

(4)在 ODACUE-NOODS 基础上,ODACUE 增加了 OOD 动作值函数学习过程,提高了值函数的学习泛化能力,相应地提高了不确定性估计准确性,从而也提高了值函数估计的准确性. ODACUE 学到离线数据 Q 值和学习策略 Q 值均比 ODACUE-NOODS 小,且全部收敛,可以学到更优的策略.

图 9 显示了 ODACUE 在 4 个任务上对 OOD 动作值函数 $Q(\boldsymbol{s}^{\text{in}},\boldsymbol{a}^{\text{ood}})$ 的 Q 值估计,可以看出:OOD 动作值函数的 Q 值曲线与图 8 中的离线数据 Q 值和学习策略 Q 值曲线变化相似,且都趋于稳定,说明了 ODACUE 能够学到好的 Q 值函数,验证了 OOD 采样方法能够提高 Q 值函数的学习泛化能力.

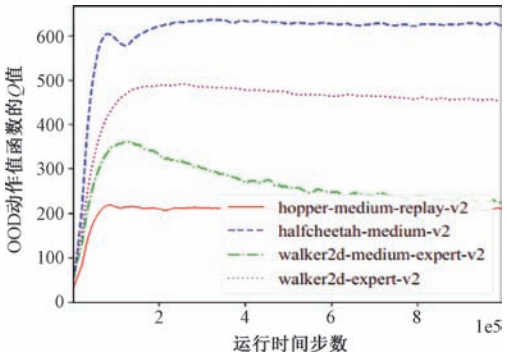


图 9　OOD 动作值函数 $Q(\boldsymbol{s}^{\text{in}},\boldsymbol{a}^{\text{ood}})$ 的 Q 值估计

图 10 显示了 Q 值函数的评估误差,其中,最小 Q 值评估误差是集成 Q 网络中的最小 Q 值与蒙特卡洛回报之间的差值,最大 Q 值评估误差是集成 Q 网络中的最大 Q 值与蒙特卡洛回报之间的差值,可以看出:最大 Q 值评估误差与最小 Q 值评估误差相差 5 左右,说明了 Q 值函数学习的稳定性,这也与图 8 和图 9 中 Q 值估计结果相一致. 此外,最大 Q 值评估误差均小于 0,说明了 ODACUE 学到的是一个悲观 Q 值,同时验证了 3.1 节不确定性估计算子的悲观性.

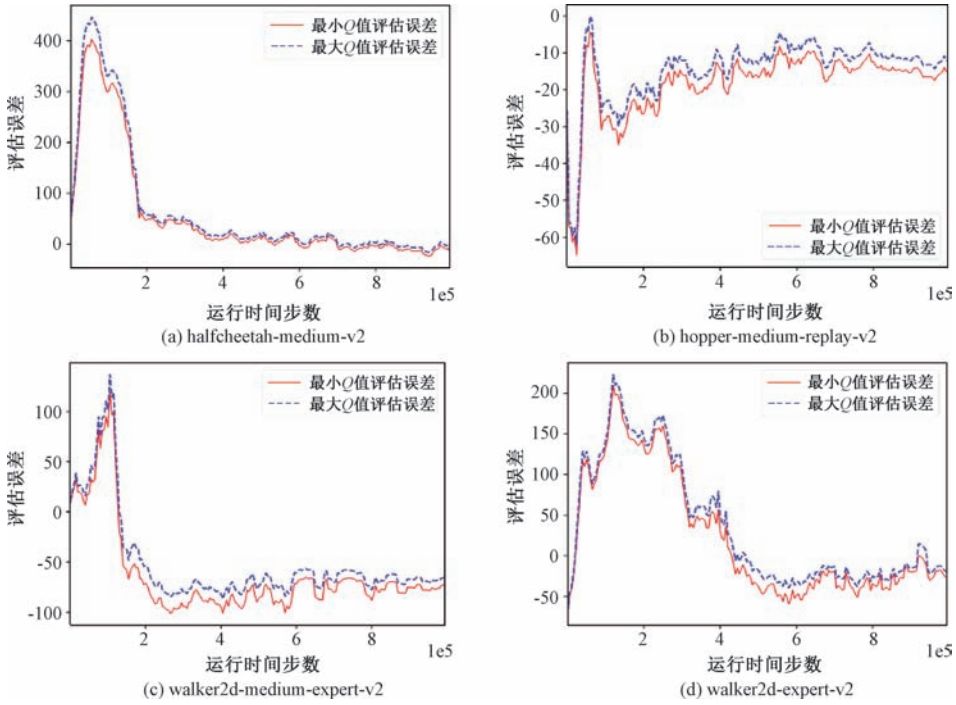


图 10　Q 值函数评估误差

5 结 论

为了解决离线强化学习方法因过于保守而很难学到优于行为策略的问题,本文利用自举集成不确定性估计方法惩罚 OOD 动作,结合 OOD 采样方法估计 OOD 动作值函数,定义了一种值函数不确定性估计算子.通过凸组合形式将不确定性估计算子应用于确定型策略梯度方法中,提出了一种基于不确定性估计的离线确定型 Actor-Critic 算法(ODACUE).在 MuJoCo 连续控制任务上进行了对比和消融实验,实验结果表明 ODACUE 可以获得优于或等同于先前基线算法的性能.此外,ODACUE 涉及多个参数,如何采用自适应调参方法提升算法性能是有待进一步解决的问题.

参 考 文 献

[1] Liu Yang, He Ze-Zhong, Wang Chun-Yu, Guo Mao-Zu. Terminal guidance law design based on DDPG algorithm. Chinese Journal of Computers, 2021, 44(9): 1854-1865 (in Chinese)
(刘扬, 何泽众, 王春宇, 郭茂祖. 基于 DDPG 算法的末制导律设计研究. 计算机学报, 2021, 44(9): 1854-1865)

[2] Xiang Chao-Can, Li Yao-Yu, Feng Liang, Chen Chao, et al. Near-optimal vehicular crowdsensing task allocation empowered by deep reinforcement learning. Chinese Journal of Computers, 2022, 45(5): 918-934 (in Chinese)
(向朝参, 李耀宇, 冯亮, 陈超,等. 基于深度强化学习的智联网汽车感知任务分配. 计算机学报, 2022, 45(5): 918-934)

[3] Jiang Yu-Bin, Liu Quan, Hu Zhi-Hui. Actor-Critic algorithm with maximum-entropy correction. Chinese Journal of Computers, 2020, 43(10): 1897-1908 (in Chinese)
(姜玉斌, 刘全, 胡智慧. 带最大熵修正的行动者评论家算法. 计算机学报, 2020, 43(10): 1897-1908)

[4] Levine S, Kumar A, Tucker G, et al. Offline reinforcement learning: tutorial, review, and perspectives on open problems. ArXiv preprint arXiv: 2005.01643, 2020

[5] Zhu Fei, Wu Wen, Fu Yu-Chen, Liu Quan. A dual deep network based secure deep reinforcement learning method. Chinese Journal of Computers, 2019, 42(8): 1812-1826 (in Chinese)
(朱斐, 吴文, 伏玉琛, 刘全. 基于双深度网络的安全深度强化学习方法. 计算机学报, 2019, 42(8): 1812-1826)

[6] Kiran B R, Sobh I, Talpaert V, et al. Deep reinforcement learning for autonomous driving: A survey. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(6): 4909-4926

[7] Yu C, Liu J, Nemati S. Reinforcement learning in healthcare: A survey. ArXiv preprint arXiv: 1908.08796, 2019

[8] Singh B, Kumar R, Singh V P. Reinforcement learning in robotic applications: A comprehensive survey. Artificial Intelligence Review, 2022, 55(2): 945-990

[9] Fujimoto S, Meger D, Precup D. Off-policy deep reinforcement learning without exploration//Proceedings of the 36th International Conference on Machine Learning. California, USA, 2019: 2052-2062

[10] Kumar A, Fu J, Soh M, et al. Stabilizing off-policy q-learning via bootstrapping error reduction//Proceedings of the 33th Conference on Neural Information Processing Systems. Vancouver, Canada, 2019: 11761-11771

[11] Peng X B, Kumar A, Zhang G, et al. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. ArXiv preprint arXiv: 1910.00177, 2019

[12] Wang Z, Novikov A, Zolna K, et al. Critic regularized regression//Proceedings of the 34th Conference on Neural Information Processing Systems. Virtual, Online, 2020: 7768-7778

[13] Ghasemipour S K S, Schuurmans D, Gu S S. EmaQ: expected-max Q-learning operator for simple yet effective offline and online RL//Proceedings of the 38th International Conference on Machine Learning. Virtual, Online, 2021: 3682-3691

[14] Cheng Yu-Hu, Huang Long-Yang, Hou Di-Yuan, Zhang Jia-Zhi, et al. Generalized offline actor-critic with behavior regularization. Chinese Journal of Computers, 2023, 46(4): 843-855 (in Chinese)
(程玉虎, 黄龙阳, 侯棣元, 张佳志,等. 广义行为正则化离线 Actor-Critic. 计算机学报, 2023, 46(4): 843-855)

[15] Kumar A, Zhou A, Tucker G, et al. Conservative Q-learning for offline reinforcement learning//Proceedings of the 34th Conference on Neural Information Processing Systems. Virtual, Online, 2020: 1179-1191

[16] Singh A, Yu A, Yang J, et al. Cog: connecting new skills to past experience with offline reinforcement learning. ArXiv preprint arXiv: 2010.14500, 2020

[17] Yu T, Kumar A, Rafailov R, et al. Combo: conservative offline model-based policy optimization//Proceedings of the 35th Conference on Neural Information Processing Systems. Online, 2021: 28954-28967

[18] Lockwood O, Si M. A review of uncertainty for deep reinforcement learning//Proceedings of the 18th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. California, USA, 2022: 155-162

[19] Wu Y, Zhai S, Srivastava N, et al. Uncertainty weighted actor-critic for offline reinforcement learning//Proceedings of the 38th International Conference on Machine Learning. Virtual, Online, 2021: 11319-11328

[20] Osband I, Blundell C, Pritzel A, et al. Deep exploration via bootstrapped dqn//Proceedings of the 30th Conference on Neural Information Processing Systems. Barcelona, Spain, 2016: 4026-4034

[21] Chen X, Wang C, Zhou Z, et al. Randomized ensembled double q-learning: Learning fast without a model. ArXiv preprint arXiv: 2101.05982, 2021

[22] Kidambi R, Rajeswaran A, Netrapalli P, et al. Morel: model-based offline reinforcement learning//Proceedings of the 34th Conference on Neural Information Processing Systems. Online, 2020: 21810-21823

[23] Yu T, Thomas G, Yu L, et al. Mopo: model-based offline policy optimization//Proceedings of the 34th Conference on Neural Information Processing Systems. Online, 2020: 14129-14142

[24] Agarwal R, Schuurmans D, Norouzi M. An optimistic perspective on offline reinforcement learning//Proceedings of the 34th International Conference on Machine Learning. Online, 2020: 104-114

[25] An G, Moon S, Kim J H, et al. Uncertainty-based offline reinforcement learning with diversified q-ensemble//Proceedings of the 35th Conference on Neural Information Processing Systems. Online, 2021: 7436-7447

[26] Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor//Proceedings of the 35th International Conference on Machine Learning. Stockholm, Sweden, 2018: 1856-1865

[27] Lee S, Seo Y, Lee K, et al. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble//Proceedings of the 5th Conference on Robot Learning. London, UK, 2022: 1702-1712

[28] Bai C, Wang L, Yang Z, et al. Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning//Proceedings of the 10th International Conference on Learning Representations. Online, 2022

[29] Ghasemipour S K S, Gu S S, Nachum O. Why so pessimistic? Estimating uncertainties for offline RL through ensembles, and why their independence matters. ArXiv preprint arXiv: 2205.13703, 2022

[30] Fujimoto S, Gu S S. A minimalist approach to offline reinforcement learning//Proceedings of the 35th Conference on Neural Information Processing Systems. Online, 2021: 20132-20145

[31] Liu Quan, Zhai Jian-Wei, Zhang Zong-Zhang, Zhong Shan, et al. A survey on deep reinforcement learning. Chinese Journal of Computers, 2018, 41(1): 1-27 (in Chinese)
(刘全, 翟建伟, 章宗长, 钟珊, 等. 深度强化学习综述. 计算机学报, 2018, 41(1): 1-27)

[32] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. Nature, 2015, 518(7540): 529-533

[33] Lyu J, Ma X, Li X, et al. Mildly conservative Q-learning for offline reinforcement learning. ArXiv preprint arXiv: 2206.04745, 2022

[34] Jin Y, Yang Z, Wang Z. Is pessimism provably efficient for offline RL? //Proceedings of the 38th International Conference on Machine Learning. Online, 2021: 5084-5096

[35] Fujimoto S, Hoof H, Meger D. Addressing function approximation error in actor-critic methods//Proceedings of the 35th International Conference on Machine Learning. Stockholm, Sweden, 2018: 1582-1591

[36] Fu J, Kumar A, Nachum O, et al. D4RL: datasets for deep data-driven reinforcement learning. ArXiv preprint arXiv: 2004.07219, 2020

[37] Kostrikov I, Fergus R, Tompson J, et al. Offline reinforcement learning with fisher divergence critic regularization//Proceedings of the 38th International Conference on Machine Learning. Online, 2021: 5774-5783

附录 A

由于 $\hat{\mathrm{B}}Q(\boldsymbol{s},\boldsymbol{a})=\hat{\Gamma}_{\mathrm{UE}}Q(\boldsymbol{s},\boldsymbol{a})+\beta U(Q(\boldsymbol{s},\boldsymbol{a}))$，当 $(\boldsymbol{s},\boldsymbol{a})\in D$ 时，有

$$
\begin{aligned}
&|\hat{\mathrm{B}}Q(\boldsymbol{s},\boldsymbol{a})-\mathrm{B}Q(\boldsymbol{s},\boldsymbol{a})|\\
&=|\hat{\Gamma}_{\mathrm{UE}}Q(\boldsymbol{s},\boldsymbol{a})+\beta U(Q(\boldsymbol{s},\boldsymbol{a}))-\mathrm{B}Q(\boldsymbol{s},\boldsymbol{a})|\\
&=|r(\boldsymbol{s},\boldsymbol{a})+\gamma E_{\boldsymbol{s}'\sim p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})}[y(\boldsymbol{s}',\pi(\boldsymbol{s}'))-\alpha_{\mathrm{in}}U(Q(\boldsymbol{s}',\pi(\boldsymbol{s}')))]\\
&\quad+\alpha_{\mathrm{in}}U(Q(\boldsymbol{s},\boldsymbol{a}))-[r(\boldsymbol{s},\boldsymbol{a})+\gamma E_{\boldsymbol{s}'\sim p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})}(Q(\boldsymbol{s}',\pi(\boldsymbol{s}')))]|\\
&=|\gamma E_{\boldsymbol{s}'\sim p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})}[y(\boldsymbol{s}',\pi(\boldsymbol{s}'))-\alpha_{\mathrm{in}}U(Q(\boldsymbol{s}',\pi(\boldsymbol{s}')))\\
&\quad-Q(\boldsymbol{s}',\pi(\boldsymbol{s}'))]+\alpha_{\mathrm{in}}U(Q(\boldsymbol{s},\boldsymbol{a}))| \qquad (19)
\end{aligned}
$$

当 $\alpha_{\mathrm{in}}\geqslant\dfrac{y(\boldsymbol{s}',\pi(\boldsymbol{s}'))-Q(\boldsymbol{s}',\pi(\boldsymbol{s}'))}{U(Q(\boldsymbol{s}',\pi(\boldsymbol{s}')))}$ 时，则式(19)满足以下不等式：

$$
|\hat{\mathrm{B}}Q(\boldsymbol{s},\boldsymbol{a})-\mathrm{B}Q(\boldsymbol{s},\boldsymbol{a})|\leqslant\alpha_{\mathrm{in}}U(Q(\boldsymbol{s},\boldsymbol{a})) \qquad (20)
$$

当 $\boldsymbol{s}\in D,\boldsymbol{a}=\pi(\boldsymbol{s})$ 时，有

$$
\begin{aligned}
&|\hat{\mathrm{B}}Q(\boldsymbol{s},\boldsymbol{a})-\mathrm{B}Q(\boldsymbol{s},\boldsymbol{a})|\\
&=|\hat{\Gamma}_{\mathrm{UE}}Q(\boldsymbol{s},\boldsymbol{a})+\beta U(Q(\boldsymbol{s},\boldsymbol{a}))-\mathrm{B}Q(\boldsymbol{s},\boldsymbol{a})|\\
&=|Q(\boldsymbol{s},\pi(\boldsymbol{s}))-\alpha_{\mathrm{ood}}U(Q(\boldsymbol{s},\pi(\boldsymbol{s})))+\\
&\quad\alpha_{\mathrm{ood}}U(Q(\boldsymbol{s},\pi(\boldsymbol{s})))-Q(\boldsymbol{s},\pi(\boldsymbol{s}))|\\
&=0 \qquad (21)
\end{aligned}
$$

综合式(20)和(21)，通过选取合适的不确定性惩罚参数 β，即 $\alpha_{\mathrm{in}}\geqslant\dfrac{y(\boldsymbol{s}',\pi(\boldsymbol{s}'))-Q(\boldsymbol{s}',\pi(\boldsymbol{s}'))}{U(Q(\boldsymbol{s}',\pi(\boldsymbol{s}')))}$，可以保证：

$$
P_D\{|\hat{\mathrm{B}}Q(\boldsymbol{s},\boldsymbol{a})-\mathrm{B}Q(\boldsymbol{s},\boldsymbol{a})|\leqslant\beta U(\boldsymbol{s},\boldsymbol{a})\}\geqslant 1-\xi
$$

因此，$\beta U(Q(\boldsymbol{s},\boldsymbol{a}))$ 是 ξ −不确定性估计. 证毕.

附录 B

当 $(\boldsymbol{s},\boldsymbol{a})\in D$ 时，有

$$
Q^{*}(\boldsymbol{s},\boldsymbol{a})-\hat{\Gamma}_{\mathrm{UE}}Q(\boldsymbol{s},\boldsymbol{a})
$$

$$
\begin{aligned}
&= BQ^*(s,a) - \hat{\Gamma}_{UE}Q(s,a)\\
&= [r(s,a) + \gamma E_{s'\sim p(s'|s,a)} \max_{a'\in A} Q^*(s',a')] -\\
&\quad [r(s,a) + \gamma E_{s'\sim p(s'|s,a)}(y(s',a') - \alpha_{in}U(Q(s',a')))]\\
&= \gamma E_{s'\sim p(s'|s,a)}[\max_{a'\in A} Q^*(s',a') - y(s',a') + \alpha_{in}U(Q(s',a'))]\\
&\geqslant \gamma E_{s'\sim p(s'|s,a)}[\max_{a'\in A} Q^*(s',a') - y(s',a')]\\
&\geqslant 0
\end{aligned} \tag{22}
$$

当 $s \in D, a = \pi(s)$ 时，有

$$
\begin{aligned}
&Q^*(s,a) - \hat{\Gamma}_{UE}Q(s,a)\\
&= \max_{a\in A} Q(s,a) - [Q(s,a) - \alpha_{ood}U(Q(s,a))]\\
&= \max_{a\in A} Q(s,a) - Q(s,a) + \alpha_{ood}U(Q(s,a))\\
&\geqslant \max_{a\in A} Q(s,a) - Q(s,a)\\
&\geqslant 0
\end{aligned} \tag{23}
$$

综合式(22)和(23)，可得：$\hat{\Gamma}_{UE}Q(s,a) < Q^*(s,a)$，即根据 $\hat{\Gamma}_{UE}Q(s,a)$ 学习得到的 Q 值函数是最优 Q^* 值函数的一种悲观估计. 证毕.

FENG Huan-Ting, Ph. D. candidate. Her main research interest is reinforcement learning.

CHENG Yu-Hu, Ph. D., professor. His research interests cover machine learning and intelligent system.

WANG Xue-Song, Ph. D., professor. Her research interests cover machine learning and pattern recognition.

Background

Offline reinforcement learning, as a data-driven reinforcement learning paradigm, has been widely studied and applied in autonomous driving, robotics, healthcare, and so on. The main challenge of offline reinforcement learning methods is the distribution shift between the learned and behavior policies. In order to solve the distribution shift problem, model-free offline reinforcement learning methods are mainly divided into policy constraint and value function regularization. Both of these methods consider the out-of-distribution (OOD) action to be bad and may learn a conservative policy by restricting the agent access to OOD actions, which hinders the generalization performance of Q value function. In order to balance the conservation and generalization of offline reinforcement learning methods, this paper proposed an offline deterministic actor-critic method based on uncertainty estimation (ODACUE). According to the different estimation methods of Q value function for the in-dataset and OOD actions, the definition of uncertainty estimation operator of Q value function is given for the deterministic policy. According to the uncertainty estimation operator, the convex combination of the in-dataset and OOD action value functions is carried out to balance the conservation and generalization of value function learning. Experimental results show that ODACUE performs better than comparative algorithms.

This paper is supported by the National Natural Science Foundation of China (62373364, 62176259) and the Key Research and Development Program of Jiangsu Province (BE2022095). These projects aim to enrich the reinforcement learning theory and develop effective and reliable methods to expand the applicability of reinforcement learning in real-world tasks.