小 型 微 型 计 算 机 系 统
Journal of Chinese Computer Systems
DOI:10.20009/j.cnki.21-1106/TP.2025-0354
2026 年 5 月 第 5 期
Vol.47 No.5 2026

离线强化学习研究综述

李晓峰,蒋佳慧,王雪娆

(河海大学 人工智能与自动化学院,南京 210000)
(东南大学 网络空间安全学院,南京 210000)
(安徽大学 人工智能学院,合肥 230601)
E-mail:xfli90@hhu.edu.cn

摘 要: 深度强化学习结合了深度学习的特征学习和强化学习的序贯决策能力,在诸多挑战性任务中表现出超越人类的水平.但是,在线强化学习以"试错"方式与环境交互,存在采样成本高、探索风险大和样本效率低的问题,阻碍了其在实际系统中的落地.离线强化学习是一种完全从静态数据集中学习目标策略的框架,将数据收集与策略学习过程分离,有效避免了交互过程中的潜在危险.本文将首先介绍强化学习基础知识,并分析在线学习方式存在的瓶颈.在此基础上,构建离线强化学习问题的形式化描述并指出其关键问题.进一步,对相关代表性算法和最新成果进行全面系统梳理,并介绍主要应用领域和常用基准测试平台.最后,总结分析面临的挑战,探讨未来发展方向.

关 键 词: 强化学习;深度强化学习;离线强化学习;策略提升;分布偏移
中图分类号: TP18 **文献标识码:** A **文 章 编 号:** 1000-1220(2026)05-1056-14

Offline Reinforcement Learning: a Survey

LI Xiaofeng, JIANG Jiahui, WANG Xuerao
(College of Artificial Intelligence and Automation, Hohai University, Nanjing 210000, China)
(School of Cyber Science and Engineering, Southeast University, Nanjing 210000, China)
(School of Artificial Intelligence, Anhui University, Hefei 230601, China)

Abstract: Deep reinforcement learning algorithms achieve impressive performance in multiple challenging tasks by combing the powerful representation learning capability of deep learning together with the sequential decision ability of reinforcement learning. However, as for some risk-aware real-world systems, collecting the data based on trial-and-error method is inaccessible because it is dangerous, expensive and sample inefficient. The active learning framework is an important reason that hinders the widespread applications of online reinforcement learning algorithms. Offline reinforcement learning is a data-driven paradigm that can learn exclusively from the static dataset without interaction with the environment during the training process. Due to the ability of learning from the previously collected data, offline reinforcement learning is appealing to deal with real-world applications. In this paper, the fundamentals of reinforcement learning is first introduced. Then, we analyze the challenges of this active learning framework to deal with practical systems. Second, the problem formulation of offline reinforcement learning is provided. A comprehensive review of important algorithms, common benchmarks and main practical applications in this field is given. Finally, we summarize the primary challenges and discuss research directions.

Keywords: reinforcement learning; deep reinforcement learning; offline reinforcement learning; policy improvement; distribution shift

0 引 言

近年来,深度强化学习结合了深度学习的特征学习和函数拟合以及强化学习的序贯决策能力,可以在复杂非线性环境中实现端到端(end-to-end)控制[1-3].强化学习是一种特殊的机器学习方法,其目的是寻找能够获得最大长期回报奖励的最优策略[4].在标准强化学习框架中,智能体基于"试错"思想,通过与环境交互收集轨迹数据和奖励反馈,并利用所得经验对策略进行优化[5,6].深度强化学习在诸多挑战性场景中取得了令人惊叹的表现,如:视频游戏[7-9]、棋牌游戏[10-12]、机器人控制[13-15]、智能电网[16]等.

目前为止,深度强化学习已实现的成功应用主要集中在虚拟仿真环境或者具有高逼真度模拟器的领域.主要原因在于现有强化学习方法大多采用在线学习的形式.智能体根据当前策略与环境交互收集数据,根据交互数据对策略进行更新,再利用更新后的策略采样新的轨迹.如此交替反复,直至策略收敛或达到期望的性能表现.在真实场景中,这种在线收集样本的方式存在采样成本高、探索风险大和样本效率

收稿日期:2025-09-02 收修改稿日期:2025-10-20 基金项目:国家自然科学基金项目(62203005)资助;中央高校基础研究基金项目(B250201086)资助. 作者简介:**李晓峰**(通信作者),男,1990 年生,博士,讲师,CCF 会员,研究方向为强化学习、无人系统自主决策、非线性系统最优决策;**蒋佳慧**,女,1993 年生,博士研究生,研究方向为多智能体强化学习、智能交通;**王雪娆**,女,1996 年生,博士,讲师,研究方向为强化学习、多智能体协同控制.

(sample efficiency)低的问题[17]. 一种常用的解决思路是先构建环境模拟器,在模拟器中训练策略,然后将学习到的策略从模拟环境迁移到真实世界(sim-to-real)中[18,19]. 即使借助于环境模拟器,样本效率也是困扰强化学习的关键问题之一[20].

离线强化学习(offline reinforcement learning)是一种从静态数据集学习策略的数据驱动框架,避免了在线交互的风险[21]. 另一方面,基于静态数据集的策略学习可以实现历史数据的重复利用,提高数据利用效率,提高训练过程的稳定性,有助于推动强化学习在自动驾驶[22-25]、医疗健康[26-28]、无人机导航与控制[29,30]等复杂危险场景中的实际应用.

但是,缺乏即时交互轨迹数据使得智能体当前策略对应的轨迹分布与静态数据集中的轨迹数据分布存在明显的分布偏移(distribution shift)[31]. 并且,随着训练过程的不断深入,两者之间的分布偏移不断加大. 更严重的是,利用深度网络对值函数或策略进行拟合会进一步放大分布偏移对策略学习的负面影响[32]. 因此,如何设计合理的策略约束方法,在分布外动作空间的探索和抑制外推误差之间取得平衡是离线强化学习需要解决的关键核心问题,也是当前的研究热点问题. 根据是否预训练环境动态模型可以将已有离线强化学习划分为无模型(model-free)方法和基于模型(model-based)的方法. 无模型方法直接从静态数据中学习可能的最优策略,通过训练过程中策略或值函数进行约束,抑制目标策略与行为策略(behavior policy)之间的分布偏移[33-39]. 另一方面,基于模型的算法先从离线数据学习环境动态模型,基于模型的不确定性、值函数保守或对抗训练等方法,对目标策略的动作选择进行约束[40-45]. 2020 年,Levine 等对离线强化学习方向的研究背景和意义、关键问题、重要算法、常见应用场景等做了较为详细的介绍,并指出了未来可能的发展方向和亟待解决的问题[21]. 在此基础上,文献[46]着重对近期相关研究成果进行了补充介绍,并对已有算法进行了分类汇总. Fu 等针对策略训练和评估问题设计了离线强化学习基准数据集 D4RL,包含了 7 类环境下总计 42 个任务以及多种离线强化学习基准算法,方便了后续算法的设计和以及不同算法间性能的比较[47].

近 5 年来,离线强化学习的相关研究得到了快速发展,取得了相对丰富的研究成果. 本文旨在对离线强化学习的基本概念、关键问题、研究进展和研究平台进行全面整理,帮助对该领域有兴趣的研究人员全面了解该领域的全貌. 重点围绕分布偏移如何抑制问题,分别从无模型和基于模型的角度,对代表性离线强化学习进行分类整理,内容偏重自 2020 年以来发表在各重要会议的研究成果. 此外,还介绍了离线强化学习常用的 3 种基准测试平台:D4RL[47]、RL Unplugged[48]和 NeoRL[49]以及离线强化学习在机器人控制、大语言模型、推荐系统等实际场景中的应用.

本文结构如下:第 1 节对强化学习的基础知识和在线强化学习算法进行简要介绍;第 2 节构建了离线强化学习问题的形式化描述并对分布偏移问题进行了分析;第 3 节对离线强化学习进行了分类并代表性工作进行了综述;第 4 节介绍了离线强化学习基准策略平台;第 5 节总结了离线强化学习实际应用场景;最后 1 节进行总结并探讨未来研究方向. 本文常用符号表示及说明如表 1 所示.

表 1　常用符号表示及说明

Table 1　Description of notations

符号	表示	符号	表示
π_b	行为策略	π	目标策略
M	真实 MDP 模型	$\hat{M}$	近似 MDP 模型
$T(s'\|s,a)$	真实状态转移函数	$\hat{T}(s'\|s,a)$	近似状态转移函数
$V^{\pi}(s)$	状态值函数	$Q^{\pi}(s,a)$	状态-动作值函数
D	静态数据集	N	正态分布

1　强化学习概述

1.1　强化学习基础

强化学习框架主要包括了环境和智能体两个部分,智能体通过“试错”方式探索环境获取轨迹数据和奖励反馈,进而根据数据对策略进行优化. 一般假设环境具有马尔可夫性质,可以用马尔可夫决策过程(Markov Decision Process,MDP)描述,表示为 $\mathbf{M}=(\mathbf{S},\mathbf{A},\mathrm{T},\mathrm{d}_0,\mathrm{r},\gamma)$[50]. $\mathbf{S}$ 表示所有状态的集合,称为状态空间. $\mathbf{A}$ 表示所有可执行动作的集合,称为动作空间. $T(s_{t+1}|s_t,a_t)$ 表示状态转移概率分布函数,用来描述在状态 s_t 下执行动作 a_t 时,$t+1$ 时刻的状态为 s_{t+1} 的概率分布. d_0 表示初始状态的概率分布. $r:\mathbf{S}\times\mathbf{A}\to\mathbb{R}$ 表示奖励函数,是环境对状态 s_t 下选择动作 a_t 的评价反馈. 最后,$\gamma\in(0,1]$ 表示折扣因子,用于平衡不同时刻的奖励反馈对全局回报的影响. 马尔可夫性质决定了 $t+1$ 时刻环境的状态完全由时刻的状态和动作决定,不会受到先前轨迹的影响. 值得注意的是,在很多真实环境下,其全局状态 s_t 无法直接观测,仅能获取观测信号 O_t,此时环境可以构建为部分可观马尔科夫模型(Partial Observable Markov Decision Process,POMDP). 针对 POMDP 环境,研究人员提出了多种在线强化学习算法来克服部分状态不客观的限制,对策略进行高效学习[51-53]. 在离线强化学习框架下,相关研究工作主要围绕状态完全可观的环境,针对 POMDP 环境的研究相对较少,是未来有待于进一步深度研究的方向之一[54].

智能体策略是指动作关于状态的概率分布函数,一般用 $\pi(a_t|s_t)$ 表示. 强化学习以最大化期望累积奖励回报为目标对策略进行优化. 目标函数定义为:

$$J(\pi)=\mathbf{E}_{\tau\sim p_{\pi}(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right] \tag{1}$$

从概率分布的角度出发,目标函数还可以表示为:

$$J(\pi)=\mathbf{E}_{s\sim d_{\pi}(s)}\mathbf{E}_{a\sim\pi(a|s)}[r(s,a)] \tag{2}$$

其中 $\tau=\{(s_0,a_0,r_0),(s_1,a_1,r_1),\cdots\}$ 表示由策略 π 生成的轨迹,$p_{\pi}(\tau)$ 表示轨迹 τ 的概率分布,$d_{\pi}(s)=\sum_{t=0}^{\infty}\gamma^t p(s_t=s|\pi)$ 表示由策略 π 生成的轨迹数据中的状态分布,$p(s_t=s|\pi)$ 表示执行策略 π 第 t 步后状态为 s 的概率.

根据强化学习的目标,最优策略 π^* 可以表示为:

$$\pi^*=\arg\max_{\pi}J(\pi) \tag{3}$$

通常情况下,定义关于策略 π 的状态值函数为:

$$V^{\pi}(s)=\mathbf{E}_{\pi}\left[\sum_{k=t}^{\infty}\gamma^{(k-t)}r(s_k,a_k)\,|\,s_t=s\right] \tag{4}$$

定义关于策略 π 的状态-动作值函数(也称为 Q 函数)为:

$$Q^{\pi}(s,a)=\mathbf{E}_{\pi}\left[\sum_{k=t}^{\infty}\gamma^{(k-t)}r(s_k,a_k)\,|\,s_t=s,a_t=a\right] \tag{5}$$

Q 函数与状态值函数之差可以直观描述当前状态下选择任意动作 a 与策略生成的动作导致的累积奖励回报的差异，一般用优势函数 $A^{\pi}(s,a)$ 表示：

$$A^{\pi}(s,a)=Q^{\pi}(s,a)-V^{\pi}(s) \tag{6}$$

当模型 $T(s'|s,a)$ 和奖励函数 $r(s,a)$ 已知时，利用动态规划(dynamic programming)思想将问题分解为单步优化子任务，通过值迭代或策略迭代方法可以逐步逼近最优策略[55]. 当模型未知时，基于采样估计思想，利用蒙特卡洛(monte-carlo)方法或时间差分(temporal difference)方法，可以从交互数据中学习策略[56-58]. 两者的区别在于，蒙特卡洛方法必须等到整个回合结束后方能计算轨迹的累积回报来更新值函数. 因此，蒙特卡洛方法估计的值函数方差较大，且当样本数量有限时估计误差难以避免. 时间差分方法则根据当前时刻的转移数据 (s_t,a_t,r_t,s_{t+1})，利用后继状态值函数估计更新当前状态的值函数：

$$V(s_t)\leftarrow V(s_t)+\alpha[r_t+\gamma V(s_{t+1})-V(s_t)] \tag{7}$$

时间差分方法可以显著提升样本利用效率，抑制了值函数估计方差，是强化学习最重要的核心内容.

1.2 在线强化学习算法

1.2.1 同策略强化学习算法

SARSA 是一种经典的基于时间差分学习的同策略在线强化学习算法，利用 Q 表格记录所有状态-动作对的 Q 值[59]. Q 函数更新公式如下：

$$\begin{aligned}Q(s_t,a_t)&=Q(s_t,a_t)+\alpha\delta_t\\ \delta_t&=r_t+\gamma Q(s_{t+1},a_{t+1})-Q(s_t,a_t)\end{aligned} \tag{8}$$

动作 a_t 和 a_{t+1} 均根据 Q 表格和 ϵ-贪心策略选择. 根据式(8)更新 Q 函数直至收敛，最优动作可以根据 $\text{argmax}_{a\in\mathbf{A}}Q^{*}(s,a)$ 确定. 但是，基于表格形式存储 Q 值只适用于有限离散状态和动作空间. 当动作空间连续时，$\text{argmax}_{a\in A}Q(s,a)$ 显然是无法实现的. 解决的思路是利用多项式或神经网络等逼近器参数化策略，然后利用策略梯度方法直接对目标策略进行优化. Sutton 等提出并从理论上证明了策略梯度定理，为策略梯度式的强化学习方法奠定了重要的理论基础. 假设使用神经网络近似目标策略，权值用 π_{θ} 表示. 目标函数关于 θ 的梯度为：

$$\nabla_{\theta}J(\pi_{\theta})=\mathbf{E}_{\pi_{\theta}}[Q^{\pi_{\theta}}(s,a)\nabla_{\theta}\log\pi_{\theta}(a|s)] \tag{9}$$

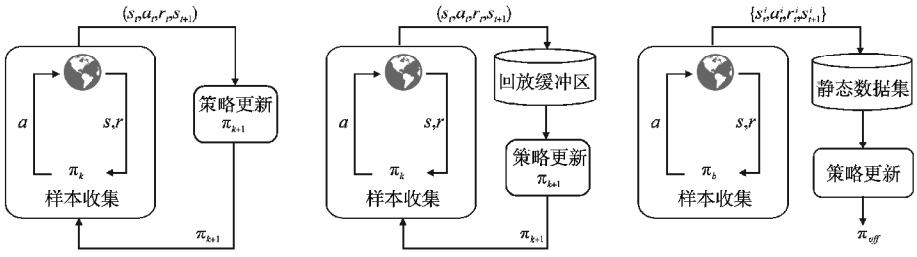
在策略梯度定理的基础上，Sutton 等提出一种同策略强化学习算法：REINFORCE 算法[60]. 智能体利用当前策略采样轨迹数据，利用蒙特卡洛方法统计奖励回报并根据式(9)计算策略梯度，最后根据梯度上升方法更新策略. 通过理论分析可以证明 REINFORCE 算法的局部最优性. 但是，蒙特卡罗方法估计 Q 函数方差较大的问题，且必须等回合结束后方能更新 Q 函数. 为了缓解过高方差导致的训练过程的不稳定，通常将执行器-评价器(Actor-Critic，AC)结构与策略梯度方法相结合来设计强化学习算法[61]. 评价网络的目标值可以由时间差分学习方法计算，可以有效缓解高方差对训练过程的影响. 另一方面，虽然策略梯度公式指明了策略更新的方向，但当更新幅度步长过大时可能出现策略性能下降的问题. 信任区域策略优化(Trust Region Policy Optimization，TRPO)算法通过在策略提升过程中限制新旧策略差异在合理距离范围内，构建了策略空间内的信任区域，保证了训练过程的稳定性和单调性[62]. Schulman 等在 TRPO 算法基础上对梯度计算过程进行简化，通过截断目标函数来限制提升前后的策略差异，被称为近端策略优化(Proximal Policy Optimization，PPO)算法[63]. 相比于 TRPO 算法，PPO 方法显著降低了算法的实现难度，并且在大多数实验环境中均取得了更好的表现. TRPO 与 PPO 均属于同策略在线强化学习算法，常被用作基准算法来评估新算法的有效性.

值得一提的是，以 PPO 为代表的同策略强化学习算法通常可以在模拟器环境中取得亮眼的表现. 但是，同策略学习阻碍了历史数据的重复利用. 每次策略提升后，需要丢弃先前采样的数据并重新收集新的轨迹数据，严重影响了强化学习样本效率.

1.2.2 异策略强化学习算法

如图 1(b)所示，异策略在线强化学习可以将历史采样数据保存到回放缓冲区，然后从中随机抽取样本进行策略评估和策略提升，明显提高了样本效率. Q 学习(Q-learning)是一种典型的异策略强化学习算法，利用 Q 表格存储 Q 值，并根据轨迹数据 (s_t,a_t,r_t,s_{t+1}) 对 Q 函数进行更新[64,65]：

$$\begin{aligned}Q(s_t,a_t)&=Q(s_t,a_t)+\alpha\delta_t\\ \delta_t&=r_t+\gamma\max_{a}Q(s_{t+1},a)-Q(s_t,a_t)\end{aligned} \tag{10}$$



(a)同策略在线强化学习 (b)异策略在线强化学习 (c)离线强化学习

图 1 在线和离线强化学习框架示意图

Fig. 1 Illustration of online and offline reinforcement learning architecture

与 SARASA 不同，式(10)中 $t+1$ 时刻的动作由 $\text{argmax}_{a}Q(s,a)$ 确定，而 t 时刻的动作可以是训练过程中的不同策略决定. 通过比较不难发现同策略和异策略强化学习的区别. 同样受到 Q 表格的限制，经典 Q 学习算法只适合用于

有限离散状态和动作空间. 当环境的状态空间连续时,利用神经网络作为函数估计器可以增加 Q 函数的泛化能力. 深度 Q 网络(Deep Q Network,DQN)将深度神经网络与 Q 学习算法相结合,利用经验回放和目标网络技术提高训练过程的稳定性,在多种雅达利游戏中取得了超越人类水平的表现[1]. 针对 DQN 算法经验回放过程中的样本选择问题,Schaul 等根据时间差分误差大小对轨迹数据进行优先级排序,增加高信息容量数据被采样的可能性,加速策略学习过程并提高性能表现[66]. 文献[67]针对 DQN 算法中 Q 函数高估问题,巧妙利用目标网络构建双重深度 Q 网络(double DQN,DDQN)架构,分别利用训练网络选择动作和目标网络计算 Q 函数目标值,有效抑制了估计误差随迭代次数增加而不断累积. 竞争深度 Q 网络(dueling DQN)算法对 DQN 算法中的神经网络结构进行改进,将 Q 函数分解为状态值函数和优势值函数,实现了算法性能提升[68].

当环境的动作空间连续时,Sliver 等提出深度确定性策略梯度(Deep Deterministic Policy Gradient,DDPG)算法,在执行器-评价器框架下利用梯度上升方法实现目标函数的最大化[69]. 确定性策略梯度公式为[70]:

$$\nabla_\theta \boldsymbol{J}(\pi_\theta)=\mathbf{E}[\nabla_\theta \mu_\theta(s)\nabla_a Q_\omega^\mu(s,a)|_{a=\mu_\theta(s)}] \tag{11}$$

与 DQN 算法相同,DDPG 算法中的 Q 函数也存在被过高估计的问题. Fujimoto 等针对 AC 框架下 Q 函数过高估计问题,提出双延迟深度确定性策略梯度(twin delayed DDPG,TD3)算法,结合双 Q 学习截断、策略网络延迟软更新、目标策略平滑正则化等方法对 DDPG 算法进行改进[71]. 文献[72]从最大熵强化学习的角度出发,提出 SAC(Soft Actor-Critic)算法,通过在目标函数中加入策略熵的方式,提高策略的随机性和对环境的探索能力.

异策略强化学习方法允许重复使用历史数据,很大程度提高了算法的样本效率,但是需要持续利用更新后的策略采样数据,避免经验回放池中的数据分布与当前策略对应轨迹的数据分布出现较大偏移. 总的来说,同策略学习和异策略学习均采用在线方式,需要在训练过程中不断利用更新后的策略与环境交互获取轨迹数据. 尤其在训练初始阶段,一般通过随机选择动作的方式对环境进行探索. 这种“在线试错”的学习方式更适用于模拟器环境,试错成本低且样本获取相对容易. 相反,在很多真实世界应用中,操作不当可能导致机器损坏甚至人身危险,因此不得不考虑数据收集过程中的安全性和犯错成本.

2　离线强化学习问题描述

离线强化学习旨在与环境不发生交互的情况下,完全从静态历史数据集中,采用类似监督学习的方式训练策略模型. 一般情况下假设环境为 MDP. 静态数据集(也称为离线数据集)包含多条历史轨迹数据,表示为 $\mathbf{D}=\{(s_t^i,a_t^i,r_t^i,s_{t+1}^i)\}$. 其中,$i$ 表示轨迹数据的索引,t 表示时间步长. $\mathbf{D}$ 可以由单一行为策略与环境交互获得,或者多种不同策略的采样数据混合构成. 在离线强化学习中,一般用行为策略 π_b 抽象表示生成数据集的策略. 离线强化学习的目标函数定义为:

$$J(\pi)=\mathbf{E}_{(s,a)\sim \mathrm{D}}[r(s,a)] \tag{12}$$

其中$(s,a)\sim$D 表示从数据集中采样训练数据且数据集中的状态分布用 $d_{\pi_b}(s)$ 表示.

离线强化学习期望从 D 中学习到的目标策略可以超越行为策略的表现,实现比模仿学习更好的泛化能力. 但是,在训练过程中无法与环境交互获取即时轨迹数据对策略的优化提出了新的挑战. 一方面,静态数据集包含的轨迹数据是有限的.

特别对于具有高维连续状态动作空间的环境而言,离线数据集只能覆盖的非常有限的区域,导致很多高奖励回报的状态-动作对没有包括在内. 期望目标策略超越行为策略表现就要求智能体尽可能探索数据集分布外的动作,以发现可能的更优动作选择. 另一方面,对于给定静态数据集,其与目标策略的偏差随着目标策略的提升不断增加,导致数据分布偏移问题,也是离线强化学习的关键核心问题.

分布偏移问题对离线强化学习的训练和测试过程均有影响. 在训练过程中,对于静态数据集内的任意轨迹数(s_t,a_t,r_t,s_{t+1}),根据目标策略选择的动作 a_{t+1} 可能落于数据集分布之外. 此时,关于(s_{t+1},a_{t+1})的 Q 值估计的准确性无法保证. 考虑到目标策略以最大化 Q 函数为目标,会倾向于选择使得 Q 函数估值最大的分布外动作. 更严重的是,估计误差会随着 Q 函数迭代次数的增加不断累积,最终导致 Q 值发散. 另一方面,目标策略对应的轨迹与静态数据集之间存在的状态分布偏移会影响目标策略的测试过程. 当智能体访问数据集分布外状态时,目标策略泛化能力的不足导致其可能选择预期外的动作,进而偏离期望轨迹. 在实际部署策略前利用静态数据集对策略进行离线评估是可以一定程度避免状态分布偏移的影响. 文献[73]对离线策略评估的主要方法、理论知识和模型机理进行了详细而全面的整理和总结. 这部分内容就不再重复介绍,本文将重点关注训练过程中动作分布偏移问题以及相关的研究成果.

3　离线强化学习分类

近年来,离线强化学习受到了广泛关注,取得了一系列研究成果. 这些方法的核心思想可以归纳为于鼓励目标策略在训练过程中选择分布内动作,同时减少分布外动作被选择的概率. 实现这一目的存在多种途径,包括策略约束、值函数保守估计、模型不确定估计等. 本文根据是否拟合环境模型将现有方法分为无模型方法和基于模型的方法. 其中,无模型方法大多通过施加策略约束或保守估计 Q 函数以抑制外推误差的产生和累积. 相反,基于模型的算法利用监督学习的方式训练环境动态,然后利用模型生成模拟估计来学习策略. 表 2 对离线强化学习的重要算法的思路和特点进行了汇总.

3.1　无模型离线强化学习

Fujimoto 等指出在无法与环境交互的情况下,异策略在线强化学习算法会受到外推误差(extrapolation error)的影响,训练后的策略表现甚至不如行为策略. 外推误差是指由于静态数据集的状态和动作分布与目标策略存在明显偏差,数据集分布外的状态-动作的价值函数可能被严重高估. 在无法获得即时交互数据的情况下,外推误差无法得到校正,且随着迭代次数的增加不断累积. 除此之外,外推误差还可能由于静态数据集数据量不足或者数据集对应的模型与实际模型存在偏

表 2 离线强化学习代表性算法对比

Table 2 Comparasion of offline reinforcement learning algorithms

算法类型	算法原理	算法名称	特点分析
无模型方法	策略约束	BCQ[33]	从数据集中拟合行为策略,通过限制与行为策略间的距离直接显式约束目标策略. 但约束可能过于严格,且学习到的目标策略的性能依赖于静态数据集质量和对状态动作空间的覆盖度
		BEAR[34]	基于行为策略支撑集对目标策略进行约束,避免分布外状态-动作对的估计误差的影响. 相比于 BCQ 算法适当放松了策略约束条件,因此取得了更好的性能表现. 但算法表现依赖于行为策略的准确估计
		BRAC[35]	基于执行器-评价器结构的通用算法框架,可以通过值函数惩罚或策略正则化方法对策略进行约束,兼容多种散度的度量方法. 但依赖最大对数似然估计方法拟合行为策略,且对超参数较为敏感
		AWR[36]	利用监督学习方法训练值函数和策略函数,利用 KL 散度对策略提升过程进行约束,算法实现过程简单且可兼容在线微调模式. 但样本效率不足,计算成本高,缺乏关于算法收敛性的理论分析
	值函数保守估计	CQL[37]	保守估计分布外状态-动作的 Q 函数,构建真实 Q 函数的下界函数并对其进行优化,算法实现简单,在诸多任务中表现优秀. 但存在 Q 函数估计过于保守,且缺乏在深度神经网络下的理论分析
		MCQ[74]	相比 CQL 算法适度放宽了 Q 函数估计的保守程度,提高了对分布外动作的泛化能力,取得了比 CQL 更好的性能表现,且理论分析过程完整. 但需要从静态数据集中拟合行为策略,需要手动调节权重系数
	模仿学习	TD3 + BC[75]	添加行为克隆项正则化目标策略,算法实现简单,训练速度快,需要调节的超参数少. 但算法对于静态数据集质量要求较高,训练后策略对未知状态和动作泛化能力较弱,在复杂环境中性能较差
		BAIL[76]	基于监督学习方法训练上包线函数,利用上包线函数选择最优状态-动作对并进行模仿学习,精准筛选高质量数据,无需复杂约束机制. 但仅适用于连续动作空间和有限步长环境,对超参数选择敏感
基于模型的方法	模型不确定估计	MOPO[42]	根据模型不确定估计对奖励函数添加惩罚项,间接对分布外状态-动作对值函数进行悲观估计,理论分析过程充分. 但算法依赖于对不确定性的准确估计,泛化能力不强,需要手动调整超参数,实现较复杂
		MOReL[41]	从静态数据集中训练学习悲观 MDP 模型,根据模型不确定性与设定阈值的比较修正奖励函数,间接对策略进行约束,理论上可以达到极小极大最优. 但对数据集覆盖要求高,对超参数敏感,对环境适配性较差
		BREMEN[78]	从静态数据集中学习包含 K 个确定性模型组成的集合,基于行为克隆拟合行为策略,对目标策略进行初始化,根据信任域迭代优化策略,但需要与环境进行若干次交互,且依赖行为策略拟合
	值函数保守估计	COMBO[43]	惩罚分布外数据的 Q 值,避免显示估计深度网络不确定的,可以结合真实数据和合成数据,间接抑制分布外偏移的影响,在多种任务中表现优异. 但依赖动力学模型精度,训练时间长,且理论分析不够完善
		RAMBO[44]	通过建立双人零和博弈模型,对对抗训练环境 MDP 模型引入保守性,间接限制目标策略选择分布外状态-动作对,具备扎实的理论基础. 但需要额外训练对抗模型,计算成本高,且对数据集质量要求较高
	对抗学习	PMDB[77]	构建交替马尔可夫博弈模型,基于系统动态的信用分布进行悲观采样,避免因信息丢失导致的策略过度保守问题,算法性能表现好,理论分析过程完整. 但算法实现复杂,计算成本高,且依赖初始动态信念
		ARMOR[45]	在参考策略的基础上,通过对抗训练 MDP 模型实现策略的鲁棒提升,对于数据集质量和超参数设定有较大的宽容度,性能表现优异. 但算法实现复杂,需要手动调节超参数,学习速度较慢
	轨迹优化	DT[79]	利用 Transformer 结构学习关于状态、动作和累积奖励回报的模型,根据期望回报和初始状态生成目标轨迹,训练稳定性高,在稀疏奖励、长时域任务中表现优异. 但模型训练时间长,计算成本高,依赖数据集质量
		TT[80]	基于 Transformer 联合建模轨迹序列,然后利用集束搜索和累积回报估计学习策略,具有结构简单、累计误差小、环境泛化能力强的优点. 但计算效率低,训练时间长,决策精度受限

差导致. BCQ(Batch Constrained Q-learning)算法通过限制目标策略所选动作与数据集之间的距离来抑制外推误差[33]. 在算法实现过程中,利用变分自编码器(Variational Auto-Encoder,VAE)近似数据集的行为策略 $\hat{\pi}_b(\cdot|s)$,并引入扰动模型 $\xi(s,a,\phi)$,在数据集分布临近空间范围内保持一定的探索能力. 由此,其目标策略修正为:

$$\pi_\theta(s,a)=\underset{a_i+\xi_\phi(s,a_i)}{\operatorname{argmax}} Q_\omega^\pi(s,a_i+\xi_\phi(s,a_i))$$
$$a_i\sim\hat{\pi}_b(\cdot|s),i=1,\cdots,N \tag{13}$$

其中,扰动模型 $\xi(s,a,\phi)$ 以最大化 Q 函数为目标:

$$\phi\leftarrow\underset{\phi}{\operatorname{argmax}}\sum_{(s,a)\in D}Q_\omega(s,a_i+\xi_\phi(s,a_i)) \tag{14}$$

除此之外,BCQ 算法基于双 Q 函数截断方法,将两个 Q 函数凸结合的方式构建 Q 函数的目标函数:

$$y=r+\gamma\max_{a_i}[\lambda\min_{j=1,2}Q_{\omega'_j}(s',a_i)+(1-\lambda)\max_{j=1,2}Q_{\omega'_j}(s',a_i)] \tag{15}$$

BCQ 算法可以保证目标策略生成的动作以相当大概率落于数据集内,避免了价值函数对分布外动作高估问题. 这种显式策略约束方法过于严格,限制了智能体对于分布外动作空间的探索能力. 目标策略的性能上界很大程度受限于行为策略的水平. 当静态数据集由随机策略或其它类型的低质量行为策略采样收集时,即使在数据充足的情况下 BCQ 算法仍难以学习到高水平的目标策略. Kumar 等在 Q 函数更新过程中对目标策略加入分布限制,来抑制分布外动作的高估误差随着迭代次数的增加而不断累积. BEAR(Bootstrapping Error Accumulation Reduction)算法采用执行器-评价器结构,并且执行器网络在策略提升的过程中在行为策略支撑集约束范围内寻找能最大化 Q 函数的权值[34]. 然后,根据执行器网络采样的动作来计算 Q 函数的目标值并更新评价网络的权值. 考虑到行为策略分布未知,BEAR 选用最大平均差异(Maximum Mean Discrepancy,MMD)指标,通过采样估计的方法近似计算目标策略与行为策略支撑集之间的距离,并利用对偶梯度下降方法来解决这个受约束优化问题. 与 BCQ 方法类似,BEAR 算法训练多个 Q 函数估计器,并基于最小的 Q 函数提升策略. 策略优化的目标为:

$$\pi_\theta(s)=\max_\pi\mathbf{E}_{s\sim D}\mathbf{E}_{a\sim\pi(\cdot|s)}[\min_{j=1,\cdots,K}\hat{Q}_j(s,a)]$$
$$s.t.\quad \mathbf{E}_{s\sim D}[\mathrm{MMD}(\mathrm{D}(s),\pi(\cdot|s))]\leqslant\varepsilon \tag{16}$$

本质上,BEAR 与 BCQ 算法都通过策略约束来抑制分布外动作偏移带来的误差. 为了深入探究这两种算法中各模块的功能及必要性,Wu 等设计了一种通用框架 BRAC(Behavior Regularized Actor Critic),通过选择不同的参数可以转化为 BEAR 和 BCQ 等算法[35].

在 BRAC 框架的基础上,Wu 等在 4 种具有连续动作空间的 Mujoco 环境中对正则化系数、Q 函数目标值计算方法、值函数或策略惩罚选择、散度评估方法等设计思路进行了大量实验. 实验结果表明,相比算法层面的模块设计选择,挑选合适的超参数对算法性能表现的影响更为显著. BCQ、BEAR 或 BRAC 算法都依赖于从静态数据集中估计行为策略. 当数据由多种类型的策略或方式采样获取时,行为策略难以准确描述,也可能影响算法效果. AWR(Advantage-Weighted Regression)算法将问题分解为两步监督学习,通过交替回归训练值函数和策略,实现异策略离线优化[36]. 一方面,基于蒙特卡洛方法计算状态累积奖励回报,并利用监督回归学习值函数,即:

$$V\leftarrow\underset{V}{\operatorname{argmin}}\mathbf{E}_{s,a\sim D}[\ \|\mathbf{R}^{\mathrm{D}}_{s,a}-V(s)\|^2] \tag{17}$$

$\mathbf{R}^{\mathrm{D}}_{s,a}$ 表示从数据集 D 中根据蒙特卡洛方法计算得到的累积回报奖励. 另一方面,基于优势加权回归方法拟合样本来提升策略提升:

$$\pi_{k+1}\leftarrow\arg\max_\pi\mathbf{E}_{s,a\sim\mathrm{D}}\left[\log\pi(a|s)\exp\left(\frac{1}{\beta}\hat{A}^\pi\right)\right] \tag{18}$$

其中 $\hat{A}^\pi=\mathbf{R}^{\mathrm{D}}_{s,a}-V^{\mathrm{D}}_k(\mathbf{s})$ 表示优势函数,β 是需要调节的超参数. AWR 算法通过监督学习对策略进行评估和提升,是训练过程得到简化且变得更加稳定.

Q 函数高估误差随着迭代不断累积的特性给策略学习过程造成严重且不可控的破坏. 对 Q 函数进行保守估计可以有效缓解动作分布偏移影响. Kumar 等提出了提出了一种值函数惩罚算法 CQL(Conservative Q-learning),在策略评估过程中引入对分布外动作 Q 值的惩罚项,使得迭代收敛后的 Q 函数在任意的值都小于真实 Q 函数的值[37]. 一般认为,Q 函数在分布内 (s,a) 上的估计比较准确. 将 $\mathbf{E}_{s\sim\mathrm{D},a\sim\hat{\pi}_b(a|s)}Q(s,a)$ 最大化加入优化目标,可以进一步提高 Q 函数估计的下界. Q 函数的迭代更新过程为:

$$\hat{Q}^{k+1}\leftarrow\ \operatorname{argmin}_Q\max_\zeta\alpha(\mathbf{E}_{s\sim D,a\sim\zeta(a|s)}[Q(s,a)]-\mathbf{E}_{s\sim D,a\sim\hat{\pi}_b(a|s)}[Q(s,a)])+\mathbf{R}(\zeta)+\frac{1}{2}\mathbf{E}_{s,a,s'\sim D}[(Q(s,a)-\hat{B}^{\pi_k}\hat{Q}^k(s,a))^2] \tag{19}$$

可以证明,当 $\zeta=\pi$ 时,式(19)迭代收敛后的 Q 函数满足 $\mathbf{E}_{\pi(a|s)}[\hat{Q}^\pi(s,a)]\leqslant V^\pi(s)$. 通过学习保守的 Q 函数可以避免在分布外动作上的 Q 函数高估误差的产生以及随迭代过程的累积. 但是,利用式(19)优化策略需要在每次策略更新后对目标策略进行完整的策略评估,计算开销很大. 考虑到迭代过程中策略是以最大化 Q 函数为目标的,可以将策略优化过程描述为:

$$\min_Q\max_\zeta\alpha(\mathbf{E}_{s\sim D,a\sim\zeta(a|s)}[Q(s,a)]-\mathbf{E}_{s\sim D,a\sim\hat{\pi}_b(a|s)}[Q(s,a)])+\frac{1}{2}\mathbf{E}_{s,a,s'\sim D}[(Q(s,a)-\hat{B}^{\pi_k}\hat{Q}^k(s,a))^2]+R(\zeta) \tag{20}$$

$R(\zeta)$ 是为了防止神经网络过拟合而增加的正则项. CQL 算法通过修改 Q 网络的目标函数,隐式约束了目标策略与行为策略分布的偏差,避免对行为策略的学习过程. 但是,CQL 算法对分布外动作的 Q 值估计往往过于保守,一定程度上降低了值函数的泛化能力. Lyu 等提出了 MCQ(Mildly Conservative Bellman)算法,通过对分布外动作的 Q 函数设置合理的替代值,在抑制外推误差的同时提高了 Q 函数的泛化能力[74]. 首先,MCQ 算子定义如下:

$$\mathbf{T}_{MCB}Q(s,a)=(\mathbf{T}_1\mathbf{T}_2)Q(s,a) \tag{21}$$

其中:

$$\mathrm{T}_1Q(s,a)=\begin{cases}Q(s,a), & \pi_b(a|s)>0\\ \max_{a'\sim\mathrm{S}(\pi(\cdot|s))}Q(s,a')-\delta, & otherwise\end{cases}$$

$$\mathrm{T}_2Q(s,a)=\begin{cases}r+\gamma\mathbf{E}_{s'}[\max_{a'\in A}Q(s',a')], & \pi_b(a|s)>0\\ Q(s,a), & otherwise\end{cases}$$

$\mathbf{S}(\pi(\cdot|s))$ 表示策略 π 的支撑集. 不难发现,当目标策略输出的动作在行为策略的支撑集内时,MCQ 算子根据标准

Bellman 方程更新 Q 函数. 当动作位于行为策略支撑集之外时,替代 Q 函数目标值为 $\max_{a'\sim S(\pi(\cdot|s))}Q(s,a')-\delta,\delta$ 表示任意小的常数. 根据 MCQ 算子对 Q 函数进行更新,可以保证分布外的动作无法被策略 $\operatorname{argmax}_{a\in A}$ 选中,阻断了外推误差的累积. 在算法实现过程中,为了判断所选动作是否在行为策略的支撑集内,可以利用条件变分自编码器来拟合行为策略. 然后,在 SAC 算法的基础上,根据 MCQ 算子对贝尔曼误差进行修改,实现对分布外动作的合理约束. 将离线强化学习与模仿学习结合也是抑制外推误差的一种可行思路. Fujimoto 等在 TD3 算法的基础上,在策略更新过程中加入行为克隆(behavior cloning,BC)项来帮助离线策略学习[75]. TD3 + BC 算法策略更新过程为:

$$\pi_{k+1}\leftarrow\operatorname*{argmax}_{\pi}\mathbf{E}_{(s,a)\sim D}[\lambda Q(s,\pi(s))-(\pi(s)-a)^2] \quad (22)$$

行为克隆项可以对策略进行正则化,超参数 λ 用来调节对目标策略与行为策略的偏差约束强度. 归一化处理数据集中的状态为均值 0 和标准差为 1 的标准正态分布 可以提高策略的稳定性. TD3 + BC 算法采用了一种极简的算法设计思路,分析了各模块、各超参数以及算法层面的改进对算法性能的影响. BAIL(Best Action Imitation Learning)算法从静态数据集中挑选出表现良好的状态-动作对,然后利用模仿学习方法训练策略,通过不选择分布外动作的方式避免了外推误差的产生[76]. 首先,根据蒙特卡洛方法计算数据 $D=\{(s_i,a_i,r_i,s'_i),i=1,\cdots,m\}$ 中各 s_i 对应的长期累积汇报 G_i. 然后,在数据集 $\{(s_i,G_i),i=1,\cdots,m\}$ 上利用监督学习方法学习 D 的上包线(upper envelope)函数,用 $V_\phi(s)$ 表示. 神经网路的权值和偏差用 $\phi=(\omega,b)$ 表示. 上包线函数是如下受约束优化问题的最优解:

$$\min_\phi\sum_{i=1}^{m}[V_\phi(s_i)-G_i]^2+\lambda\|\omega\|_2,s.t.\quad V_\phi(s_i)\geqslant G_i \quad (23)$$

其中 λ 是正则项系数,防止神经网络过拟合. 然后,利用上包线 $V(s)$ 从 D 中选择前 $p\%$ 的最优动作,一般设置 $p=25$ 置. 最后,在挑选出的状态-动作对上利用模仿学习方法训练策略神经网络. 与 BAIL 类似,IQL(Implicit Q-Learning)算法基于 expectile 回归模型,利用数据集分布内动作估计各状态下最优动作相关的 Q 函数,避免了外推误差的产生. 在收敛后 Q 函数的基础上,利用 AWR 方法提取目标策略,在分布受限下最大化 Q 函数[38].

3.2 基于模型的离线强化学习

强化学习中的模型一般包括环境的状态转移概率函数和奖励函数. 基于模型的的强化学习方法根据预先知道的模型或从交互数据中学习的模型模拟轨迹数据,然后综合利用真实轨迹和模拟数据进行策略搜索. 借助模型的泛化能力可以显著提升算法样本效率. 在离线学习框架下,静态数据集对状态和动作空间的覆盖范围有限,无法保证所学模型在全局范围的准确性. 特别地,当状态或动作没有出现在静态数据集中时,模型可能存在较大的泛化误差. 因此,基于模型的离线策略优化过程需要考虑对模型的信任程度. 基于模型的在线强化学习方法可以利用即时轨迹数据对模型偏差进行校正,难以直接应用于离线训练. 基于模型的离线强化学习结构如图 2 所示.

MOReL 算法基于静态数据集学习近似环境动态模型 $\hat{P}(\cdot|s,a)$,并利用静态数据集和 $\hat{P}(\cdot|s,a)$ 生成的模拟数据对策略进行优化[41]. 为了缓解分布偏移和模型误差的影响,算法定义未知状态-动作检测器(Unknown State-Action Detector,USAD)将状态空间划分为已知和未知两个区域,并对访问未知状态的状态-动作对给予负面奖励,实现对目标策略动作选择的间接约束. 在算法实现过程中,利用高斯动态模型近似环境模型,即:$\hat{P}(\cdot|s,a)\equiv\mathbf{N}(f_\phi(s,a),\Sigma)$,其中 $f_\phi(s,a)$ 表示平均值,Σ 表示方差. 通过设置不同初始权重并采样不同训练样本可以训练得到若干个模型组成的集合 $\{f_{\phi_1},f_{\phi_2},\cdots\}$. 该集合中各模型间的差异为:

$$\text{disc}=\max_{i,j}\|f_{\phi_i(s,a)-\phi_j(s,a)}\|_2 \quad (24)$$

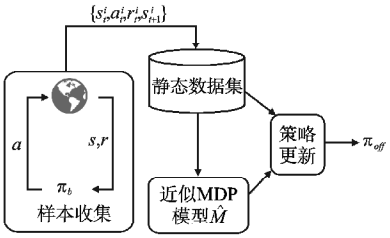


图 2 基于模型的离线强化学习结构示意图

Fig. 2 Architecture of model-based offline reinforcement learning

然后,根据模型的不确定性设置 USAD:

$$U(s,a)=\begin{cases}(s,a)\ is\ known, & if\ \text{disc}(s,a)\leqslant\Xi\\(s,a)\ is\ unknown, & otherwise\end{cases}$$

根据上述 USAD 修正奖励函数为:

$$r(s,a)=\begin{cases}-K, & if(s,a)\ is\ known\\r(s,a), & otherwise\end{cases} \quad (25)$$

MOReL 算法基于模型不确定性量化估计对状态空间进行划分,并对访问未知区域轨迹进行惩罚. 但是,式(25)设定未知区域内所有状态-动作对的奖励为 -K,这种硬约束条件无法区分状态-动作对偏离程度,一定程度限制了目标策略对临近未知空间的探索. Yu 等提出 MOPO(Model-Based Offline Policy Optimization)算法,同样基于模型不确定性量化估计思想对奖励函数进行修正,抑制分布偏移过大[42]. 首先,MOPO 算法基于静态数据集 **D** 训练包含 N 个模型的集合:

$$\{\hat{T}^i(s',r|s,a)=\mathbf{N}(\mu^i(s,a),\Sigma^i(s,a))\}_{i=1}^{N} \quad (26)$$

其中 μ^i 和 Σ^i 分别表示多变量高斯过程的均值和协方差矩阵,上标 i 表示集合中模型的索引. 接下来,利用高斯模型的标准差估计模型误差,惩罚不确定性较高的状态-动作对:

$$\tilde{r}(s,a)=\hat{r}(s,a)-\lambda\max_{i=1,\cdots,N}\|\sum_{\phi}^{i}(s,a)\|_F \quad (27)$$

其中,r 是 $\hat{T}$ 输出的预测奖励的平均值. 经过理论分析可以发现 MOPO 算法可以最大化真实 MDP 环境下回报函数的下界函数. 相比 MOReL 算法,可以实现对静态数据集分布外状态和动作空间的更多探索,平衡探索过程中可能得到的回报和需要承担的风险.

上述方法均依赖模型不确定性估计,利用量化后的不确定性构建并优化策略性能表现的下界函数. 但是,对于复杂模型(如深度神经网络),其不确定性量化过程难以实现且准确性不高,使得算法表现不如预期. 为了避免对模型不确定性进

行估计,Yu 等将 CQL 算法的值函数保守估计思想引入到基于模型的离线策略优化框架中,提出 COMBO(Conservative Offline Model-Based Policy Optimization)算法[43]. 首先,从静态数据集 D 中根据最大似然法学习环境动态模型:

$$\hat{T}_\theta(\mathbf{s}_{t+1},r|\mathbf{s},\mathbf{a})=\mathbf{N}(\mu_\theta(\mathbf{s}_t,\mathbf{a}_t),\Sigma_\theta(\mathbf{s}_t,\mathbf{a}_t))$$

已知 $\hat{T}$ 和 $\hat{r}$,可以构建 MDP:$\hat{M}=(S,A,\hat{T},\hat{r},\mu_0,\gamma)$. 在此基础上,根据模型 $\hat{T}_\theta$ 生成模拟轨迹数据,并利用静态数据集和模拟数据集对策略进行评估:

$$\hat{Q}^{k+1}\leftarrow\arg\min_Q\beta(\mathbf{E}_{s,a\sim\rho(s,a)}[Q(s,a)]-\mathbf{E}_{s,a\sim D}[Q(s,a)])+\frac{1}{2}\mathbf{E}_{s,a,s'\sim d_f}[(Q(s,a)-\hat{\mathbf{B}}^\pi\hat{Q}^k(s,a))^2] \tag{28}$$

其中,$\rho(s,a)$ 表示模拟数据集的状态-动作对分布,满足:

$$\rho(s,a)=d^\pi_{\hat{M}}(s)\pi(a|s) \tag{29}$$

其中,$d^\pi_{\hat{M}}(s)$ 表示在 $\hat{M}$ 中执行策略 π 所生成轨迹的状态分布. d^μ_f 表示从静态数据集和模拟数据集中采样数据的概率分布,满足:

$$d^\mu_f(s,a)=fd(s,a)+(1-f)d_{\hat{M}}(s,a) \tag{30}$$

其中,$f\in[0,1]$ 表示从 D 中采样数据的概率. 相应的,$1-f$ 表示从模拟数据集中采样数据的概率. 然后,根据 Q 函数的保守对策略进行更新:

$$\pi'\leftarrow\arg\max_\pi\mathbf{E}_{s\sim\rho,a\sim\pi(\cdot|s)}[\hat{Q}^\pi(s,a)] \tag{31}$$

通过上述策略评估方法,可以实现对模拟轨迹上状态-动作对 Q 函数值的保守估计,并同时提高了对静态数据集中状态-动作对 Q 函数值的估计. Yu 等从理论上证明了 COMBO 算法可以学习到真实 Q 函数的下界,从而可以有效抑制分布外数据的过高估计问题,间接限制目标策略与行为策略偏差过大. RAMBO(Robust Adversarial Model-Based Offline)算法引入了鲁棒对抗学习思想,将基于模型的离线强化学习重新构建为目标策略与对抗环境的零和博弈问题[44]. RAMBO 算法交替轮流更新目标策略和对抗环境模型,通过对抗模型的优化引入策略估计的保守性,抑制分布偏移的影响,避免了对于模型不确定性的量化估计. 实验结果表明,RAMBO 算法相比 COMBO 算法可以取得更高的奖励回报. 分析发现 COMBO 对值函数的悲观估计使得训练过程中 Q 函数更容易陷入局部最优. RAMBO 算法中对于值函数的估计的保守性随着训练的推进逐步增加,可以避免策略陷入局部最优,并取得更好的全局表现.

Bhardwaj 等基于对抗训练思想设计了 ARMOR(Adversarial Model for Offline Reinforcement Learning)算法,通过对抗训练 MDP 模型,可以在任意参考策略的基础上进一步优化目标策略,避免了数据集质量的影响[45]. 理论证明,在相当大的超参数范围内 ARMOR 可以提升参考策略的表现. 这种鲁棒策略提升特性使得 ARMOR 算法可以在实际应用中发挥重要作用. Guo 等在鲁棒 MDP 的基础上构建交替马尔可夫博弈(Alternating Markov Game,AMG)模型,在系统动态的信用分布(belief distribution)上进行悲观倾向的采样对策略进行评估或优化[77]. 与标准鲁棒 MDP 模型相比,AMG 模型每一步的模型候选集(candidate set)不固定且元素具有随机性,根据信任分布进行随机采样可以有效避免因信息丢失导致的策略过度保守问题. 实验结果也验证了算法的有效性.

Matsushima 等提出算法部署效率(deployment efficiency)的概念,其大小等于训练过程中样本采集策略的更改次数. 不难发现,在线强化学习算法需要至少成千上万次的部署来完成策略的学习,而上述离线强化学习只允许一次部署. 在训练过程中无法获得新的在线交互数据使得离线强化学习往往局限于在次优策略生成的数据集中取得良好的表现,但在随机生成的数据集上的表现难以令人满意. 在非完全批强化学习(semi-batch)框架下,Matsushima 等提出 BREMEN(Behavior-Regularized Model-ENsemble)算法,通过有限次数的样本收集和策略提升的迭代过程,实现策略的成功学习[78]. 首先,BREMEN 算法从采样数据集中学习 K 个动态模型组成的集合:$\hat{f}_\phi=\{\hat{f}_{\phi_1},\hat{f}_{\phi_2},\cdots,\hat{f}_{\phi K}\}$. 模型训练的目标函数为:

$$\mathrm{L}(\phi_i)=\frac{1}{D}\sum_{(s_t,a_t,s_{t+1})\in D}\frac{1}{2}\|s_{t+1}-\hat{f}_{\phi_i}(s_t,a_t)\|_2^2 \tag{32}$$

在策略优化的初始时刻,利用行为克隆方法近似行为策略 $\hat{\pi}_b$:

$$\hat{\pi}_b\leftarrow\arg\min_\pi\frac{1}{D}\sum_{(s_t,a_t)\in D}\frac{1}{2}\|a_t-\pi(s_t)\|_2^2 \tag{33}$$

然后,用标准差为 1 的高斯分布表示目标函数 π_θ,并利用 $\hat{\pi}_b$ 初始化目标策略的均值. 为了避免目标策略与真实行为策略偏差过大,基于 KL 散度的信赖区间优化方法对目标策略进行更新:

$$\theta_{k+1}\leftarrow\arg\max_\theta\mathbf{E}_{s,a\sim\pi_{\theta_k},\hat{f}_{\phi_i}}[\frac{\pi_\theta(s,a)}{\pi_{\theta_k}(s,a)}\mathbf{A}^{\pi_{\theta_k}}(s,a)]$$
$$\mathbf{E}_{s\sim\pi_{\theta_k},\hat{f}_{\phi_i}}[D_{KL}(\pi_\theta(\cdot|s)\|\pi_{\theta_k}(\cdot|s))]\leqslant\delta,\pi_{\theta_0}=\mathbf{N}(\hat{\pi}_b,1) \tag{34}$$

其中,$A^{\pi_{\theta_k}}(s,a)$ 表示基于学习模型计算的优势函数. 基于 BC 方法的策略初始化与有限次迭代的信赖区间优化方法相结合可以限制分布偏移的负面影响. 实验结果表明这种隐式正则方法相比显式 KL 散度惩罚更有效.

上述算法根据最优性原理或单步预测模型对复杂的长期决策问题进行分解,然后利用无模型或基于模型的强化学习方法进行求解. DT(Decision Transformer)算法将强化学习视为长时域序列决策建模问题,基于 Transformer 结构强大的特征学习和时序建模能力,构建了状态、动作和长期奖励回报三者之间的关系[79]. 然后根据期望的目标奖励回报和初始状态选择执行动作生成轨迹. Janner 等提出了 TT(Trajectory Transformer)算法,利用 Transformer 模型学习轨迹分布,然后结合集束搜索(beam search)技术和奖励回报估计对候选轨迹进行搜索和优化[80]. 相比于基于单步预测模型方法,轨迹优化方法需要消耗更多的计算资源和训练时间. 但是,模型可以考虑相当长时域的累积奖励,因此在稀疏奖励环境中可以取得良好的性能表现.

3.3　算法对比分析

无模型方法试图直接从离线数据中对策略和值函数进行优化. 基于模型的方法先从离线数据集中训练环境模型,再依托模型生成的模拟数据更新策略. 两者使用离线数据逻辑的不同导致算法在收敛速度、样本效率与最终性能上呈现鲜明对比. 基于模型的离线强化学习样本利用效率更高,可以显示地从离线数据中学习环境模型,并借助环境模型快速生成大量合成数据,往往具有更快的收敛速度[41-43]. 相比之下,无模型方法直接从离线数据中学习策略,且为规避分布偏移需引

入保守性约束(如 CQL 的对数求和项),导致策略更新需反复迭代数据批次以稳定值函数,收敛周期可能是基于模型方法的 2 ~ 3 倍[34-37]. 但是无模型方法在算法最终性能表现和理论上限方面更具优势. 基于模型的方法易受"模型估计"影响,合成数据与真实数据存在分布偏移,学习到的策略可能出现"失准"的情况. 无模型方法的训练数据源自真实轨迹记录,不存在模型偏差问题. 当离线数据质量较高且覆盖范围较广时,就能通过抑制分布外状态-动作 Q 值估计,学习到高性能的策略. 尤其在数据充足的场景中,无模型方法可以通过对海量真实交互数据的学习,突破模型偏差的限制,达到更接近人类水平的最终性能.

综上所述,基于模型的方法适合真实数据稀缺、需快速迭代的场景,无模型方法则在数据充足、追求最终策略高性能表现的任务中更应被选择.

3.4 算法理论成果

上述离线强化学习文献中也从理论层面对算法进行了分析,主要围绕解决分布偏移与价值高估核心挑战展开,分别从值函数保守估计、期望回报下界估计、数据集特性等角度构建了较为完整的理论分析基础. CQL 算法基于保守性理论,引入正则项对 Q 函数进行约束,使策略期望 Q 值不超数据分布期望,从数学上保证学习得到的 Q 值为真实值的下界,为抑制分布外偏移影响提供了理论支撑[37]. Robust-IQL 算法对上述框架进行了优化,引入 Huber 损失处理重尾噪声数据,在受污染的场景中仍保持稳定泛化性能,验证了保守性原则对于干扰数据的泛化能力[81]. MOPO 算法考虑了在训练环境动态模型的情况下,建立真实 MDP 模型回报误差与估计 MDP 模型动态误差的关联,并且证明了在存在可接受误差估计器的假设条件下,优化估计 MDP 模型的回报可以间接保证真实回报的保守估计,避免模型在分布外区域的误差滥用[42]. 此外,一些学者通过分析数据集的底层特性,揭示了泛化能力与数据多样性与覆盖度存在直接关联性[82]. Mediratta 等人以上下文马尔可夫决策过程(CMDP)为理论基础,分析论证了增加数据多样性有助于提高策略的泛化能力[83].

上述理论成果为离线强化学习泛化能力分析的提供了重要支撑. 保守性理论为算法设计正则化方法提供了理论基础,环境模型的引入开辟了模型驱动的泛化新路径,数据特性分析揭示样本效率的内在规律.

4 基准测试平台

高质量大规模的基准开源数据集为推动机器学习在诸多领域的发展发挥了重要作用. 例如:视觉目标识别领域常用的 ImageNet 数据集[84]和 COCO 数据集[85]. 强化学习也有一些常用的基准环境模拟器对算法进行测试和验证[3]. 但是,以往在线强化学习算法采用边采样边训练的方式,无法重复利用历史数据. 离线强化学习框架为序列决策问题中使用大规模历史轨迹数据提供了一种新的可能. 构建完备易用的静态数据集,可以为离线强化学习设计提供公平的基准测试平台,对推动相关算法的研究具有重要意义:1)可以观察不同数据收集办法和数据量大小对算法效果的影响,量化分析算法对数据集质量的依赖性;2)可以测试算法在不同环境和任务下的表现,评估算法的鲁棒性;3)可以为不同离线强化学习提供公平的基准测试平台. 本节将重点介绍 3 个目前主流的基准数据集,包含了多种不同特性的环境和任务,具有样性的数据来源方式,并集成了多个常用的离线强化学习方法. 表 3 对这 3 种基准平台的主要特点进行了整理.

表 3 离线强化学习基准测试平台对比

Table 3 Comparison of benchmarking platforms for offline reinforcement learning

基准测试平台	任务类型	数据集构建	基准算法	策略评估
D4RL	机器人导航、机器人控制、交通管理、自动驾驶	基于规则的策略、人类演示数据、专家策略等多种策行为策略混合生成	BCQ[33], BEAR[34], BRAC[35], AWR[36], SAC[72], BC[97], cREM[98], AlgaeDICE[99]	基于模拟器在线评估
RL Unplugged	机器人导航、机器人控制、视频游戏、真实世界挑战	在线训练过程数据	DQN[1], BCQ[33], BRAC[35], BC[97], D4PG[100], IQN[101], RABM[102], REM[103]	在线和离线策略评估
NeoRL	工业控制、金融交易、城市管理、机器人控制	多种行为策略混合生成	BCQ[33], CQL[37], MOPO[42], BREMEN[78], BC[97], PLAS[104], CRR[105]	离线策略评估

4.1 D4RL

D4RL 是当前离线强化学习领域应用最广泛的大规模基准测试平台[47]. 该数据集包含了多种类型的任务:机器人导航、Gym-MuJoCo 控制[86]、机械臂操作[87]、机器人多目标操作[88]、交通管理[89]和自动驾驶[90]. 数据集收集的方式包括人类演示、基于规则的控制器和在线强化学习过程中的策略. 为了有效评估离线强化学习在实际应用中的作用,D4RL 数据集在构建过程中考虑了以下特性:1)数据集分布较窄且存在偏差;2)多目标数据混合;3)奖励反馈稀疏;4)轨迹数据不是最优路径;5)行为策略难以准确表征或不满足马尔可夫性质;6)数据来自真实系统轨迹,存在传感器噪声. 文献[35]分析大量实验结果发现超参数调节对离线强化学习的性能表现有明显的影响. 完全基于离线数据的策略评估结果可能过于乐观,但是很多风险敏感的实际系统难以接受通过大量在线策略评估实验来选择合适的超参数. 针对超参数调节问题,D4RL 将一部分任务划分为训练环境,在这些环境中利用环境模拟器调整超参数. 将剩余任务设置为评估环境以实现对算法的评估. D4RL 还包含了多种代表性离线强化学习方法作为基准,详细情况可以参考表 3.

4.2 RL Unplugged

RL Unplugged 也是训练和评估离线强化学习常用的基准测试平台之一[48]. 该数据集中包含的环境有:机器人导航

与控制[91]、视频游戏[92]和真实世界挑战[93]. 数据集由在线强化学习训练过程中的数据构成,相比 D4RL 基准平台数据来源相对单一. 针对超参数调节问题,RL Unplugged 提供了在线和离线两种策略评估模式,并指明了不同任务适用的评估方法. 在线策略评估允许智能体与环境交互并利用在线采样数据测试策略性能. 离线策略评估一般采用异策略评估方法,基于重要性采样原理,从离线数据中计算策略奖励回报. 表 3 对 RL Unplugged 平台集成的多种基准算法进行了汇总.

4.3 NeoRL

NeoRL 是南栖仙策团队于 2022 年提出的离线强化学习基准数据集[49]. 数据集包含了 4 种环境:工业控制[94]、金融交易[95];城市管理[96]和机器人控制[86]. NeoRL 从实际系统数据收集过程出发,考虑轨迹生成时系统的稳定性和性能表现,对行为策略进行保守限制. 因此,数据集中的轨迹数据存在分布较窄,对状态和动作空间探索不够充分的局限,放大了数据分布偏移的影响. 在所有环境中,首先利用 SAC 算法训练策略直至神经网络收敛. 在训练过程分别记录下 4 种不同质量的策略,并在此基础上构建不同水平的训练数据集以及相对应的测试数据集. 作者指出离线强化学习的完整部署流程应该包括离线策略学习、离线策略验证以及在线测试,并将离线学习和离线评估模块集成到基准测试平台中. NeoRL 包含了多种无模型和基于模型的离线强化学习方法作为基准,为后续研究提供参考,见表 3.

4.4 对比分析

D4RL、RL Unplugged 和 NeoRL 作为离线强化学习领域的代表性基准数据集,为不同算法之间的对比提供了有效依据,为该领域的发展提供了重要支撑. 但是,它们各自有不同的侧重点,本节将从数据集采样方法、样本多样性和任务场景 3 个维度对三者进行对比分析.

首先,从数据集采样策略的角度来看,D4RL 采用了分层策略采样,通过人为控制专家、中期和随机策略生成数据的混合比例,构建了可量化质量梯度的数据集. 这种设计方式可以直观体现数据集质量对算法学习效果的影响,但划分方式依赖人工经验,可能与真实场景存在偏差. RL Unplugged 数据集保存了在线 RL 训练过程中的经验回放池,记录了完整的策略学习过程中的智能体演化轨迹. 这种方式可以比较充分的保留数据的原始分布特性,但对于数据集质量的划分不够清晰. NeoRL 最晚提出,在前两个工作的基础上,创新性地采用次优策略采样,选择回报在专家水平 50% ~80% 的 PID 控制器或保守强化学习策略生成数据,刻意模拟工业场景中“足够好而非最优”的操作记录,使得数据集可以尽可能模仿真实环境.

其次,在样本多样性方面,D4RL 构建了 5 类不同质量的数据集,但数据子集策略具有较高的一致性. 这种设计方式适合评估算法对数据质量的敏感程度,但数据分布相对静态. 相对来说,RL Unplugged 对策略学习的动态过程的覆盖程度更高,包含了策略训练的全过程数据,甚至包括了失败轨迹. 数据集的动态多样性对算法的泛化能力提出更高要求,但数据质量波动较大,难以进行量化. NeoRL 更加关注现实约束多样性,通过控制数据规模和时间延迟等实际工况表现,模拟真实场景中的数据稀缺性和外部干扰,其数据集的分布狭窄特性更接近现实世界的数据约束.

上述 3 个数据集在任务场景方面形成了从模拟到真实的完整覆盖. D4RL 专注于经典控制任务,覆盖了多种导航控制等标准化环境,适合基础算法性能的验证分析. RL Unplugged 拓展到复杂动态场景,包含了难度更高的 Atari 游戏、机器人运动控制、星际争霸等任务,可以进一步测试离线强化学习的能力上限. NeoRL 则更专注于专业领域场景,7 个核心任务覆盖工业管道控制、血糖调节、火箭回收等现实应用,考虑了外部干扰和安全约束等实际条件限制,将算法评估与实际应用需求紧密结合.

综上所述,D4RL 更适合用于对算法基础能力的标准化测试,也因此更受学术界研究人员的青睐. RL Unplugged 更适用于研究数据自然特性对策略学习的影响,NeoRL 则更关注算法在真实场景下的表现,为面向实际问题的算法研发提供了精准模拟.

5 离线强化学习应用

基于离线强化学习框架可以完全从已收集的历史轨迹数据中学习策略,实现对数据的重复利用,同时也规避了在线采样过程中的风险. 当前,离线强化学习已经被成功应用到多个实际系统,包括:机器人控制[106,107]、自动驾驶[108-110]、健康医疗[111]、推荐系统[112-114]、语言对话系统[115,116]等. Kumar 等针对机器人学习任务,将监督学习中“过拟合”和“欠拟合”概念引入到强化学习中,并设计了相应的评估方法为选择神经网络结构、正则项以及是否提前停止训练提供决策依据[117]. 实验结果验证了所设计的工作流程在模拟和真实机器人操作环境中的有效性. 文献[118]考虑了自动驾驶应用中策略训练过程的安全性、策略的可解释性以及可迁移性,提出了一种基于模型的离线强化学习框架,通过构建一个部分可观的随机动态模型,并利用规划方法统筹解决自动驾驶中的预测、规划和控制问题. 离线强化学习也被应用于提高大规模语言模型处理用户指定任务的能力. Snell 等提出了 ILQL(Implicit Language Q-Learning)算法,结合值保守估计和数据支撑集约束学习价值函数,可以从已收集的历史数据中学习高性能策略且提高策略优化过程的稳定性[119]. 在推荐系统领域,Xiao 等提出了一种基于执行器-评价器结构的离线强化学习框架,完全从离线反馈数据训练推荐策略,避免探索行为可能导致的用户流失[120]. 在此过程中交互式推荐系统被形式化为概率推断问题. 考虑到离线数据的分布偏移,在离线策略优化的过程中综合利用了支撑集限制、正则项、策略约束、对偶约束和奖励探索技术来抑制外推误差对算法性能的影响. 文献[121]也对离线强化学习的实际应用做了较为全面的汇总.

Gürtler 等针对真实机器人灵巧操作任务,构建了以 TriFinger 平台为核心的离线强化学习基准体系,旨在推动其在真实物理场景的应用[122]. 主要过程可以分为 3 步:第 1 步,在 Isaac Gym GPU 加速模拟器中,用 PPO 算法结合领域随机化技术训练专家策略,并选择真实系统表现最优的种子用于数据采集;第 2 步,在 PyBullet 模拟器和 TriFinger 机器人集群上,采集 Push 与 Lift 任务轨迹数据,并根据采样方式分为 4 个类别;第 3 步,基于 d3rlpy 算法库测试 5 中代表性离线强化

学习算法,然后通过远程机器人集群和模拟器验证策略性能.该场景下离线强化学习面临三大核心挑战:1)相比于模拟器生成的轨迹,真实数据存在噪声、传感器延迟、动态非平稳性等诸多干扰,两者之间的差异使得策略从模拟环境迁移到真实环境时会出现性能下降;2)绝大多数算法在质量较差的数据集上成功率骤降,算法在不同质量数据集上的鲁棒性不足;3)算法在需要多步操作的任务中的表现远差于单步任务,复杂任务适应性差.机器人控制作为离线强化学习的典型场景,上述处理过程和挑战具有相当的代表性.

6 未来研究方向

离线强化学习表现出可以推动强化学习在真实世界落地的潜在能力,受到了研究人员的广泛关注,相关的研究成果不断涌现.但是,完全基于静态数据集的学习方式也对策略的学习和评估提出了新的挑战.Levine 等列举了该领域一些亟待解决的关键问题[21].虽然其中的一些问题取得了一定进展,但是仍然有很多问题值得进一步探索和研究.

1)静态数据集构建方法研究.离线强化学习完全从静态数据集中学习和评估,数据集水平直接影响策略学习和评估结果.与监督学习不同,离线强化学习的数据集可以由不同的行为策略生成,甚至由多种策略生成轨迹数据混合组成.分析样本复杂性、轨迹质量、状态-动作覆盖率等数据集特征对算法的影响,设计合理的数据集构建方法,可以为比较不同算法表现提供公平的基准,也有助于离线强化学习在实际任务中的快速部署.

2)离线策略评估方法研究.在策略实际部署前离线评估其性能表现,可以规避潜在风险,是应用强化学习解决机器人、自动驾驶等实际问题不可或缺的步骤.离线策略评估完全依赖静态数据集估计目标策略的状态价值函数,估计误差主要来源于数据集状态分布偏移.离线策略评估相关研究已经取得了一定的进展,但目前仍缺乏通用的框架可以为不同算法提供一致性的策略评估.考虑非平稳环境和轨迹长度等情况,探索新的误差度量方式,设计一种新的离线策略评估框架为调节超参数、决定训练结束时间以及选择合适的策略提供帮助具有重要的研究意义.

3)将策略离线训练与在线微调相结合,构建混合学习框架.离线强化学习致力于完全从静态历史数据集中学习目标策略.在策略实际部署过程中,策略对于数据集分布外状态选择的动作质量并不能保证.为了减小状态分布偏移的影响,在离线训练完成后将策略与环境交互,并利用在线交互数据进一步对策略进行微调,可以提高策略的性能表现.因此,研究在线微调和离线训练的结合方式,避免在线学习过程中策略衰退问题是重要的研究方向之一.

7 结 论

离线强化学习希望借鉴监督学习的训练方式从历史数据集中训练决策模型.将样本收集与策略训练过程分开,可以避免在线交互的成本与风险,有利于强化学习方法在实际系统中的应用.本文对离线强化学习的问题描述、训练算法和基准数据库等进行了较为全面的梳理和总结,旨在为对该领域感兴趣的研究人员提供研究现状和思路.首先,对离线强化学习问题进行了形式化描述,分析了分布偏移对策略学习和评估的影响.然后,分类整理了离线强化学习代表性算法以及最新研究成果进行了.接下来,介绍了离线强化学习常用的基准测试平台以及可能应用的方向.最后,对该领域的未来研究方向进行了展望.

总的来说,离线强化学习为大规模智能决策模型提供了一种预训练的方法,以更经济安全的方式推动强化学习在实际系统中的部署.随着分布偏移、离线策略评估、离线数据集构建等问题的深入研究,有理由相信离线强化学习可以为推动通用人工智能的实现发挥重要作用.

References:

[1] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning [J]. Nature, 2020, 518(7540):529-533.

[2] Arulkumaran K, Deisenroth P M, Brundage M, et al. Deep reinforcement learning: a brief survey [J]. IEEE Signal Processing Magazine, 2017, 34(6):26-38.

[3] Henderson P, Islam R, Bachman P, et al. Deep reinforcement learning that matters[C]//Proceedings of the AAAI Conference on Artificial Intelligence, 2018, doi: https://doi.org/10.1609/aaai.v32i1.11694.

[4] Sutton S R, Barto G A. Reinforcement learning: an introduction [M]. MA, USA: MIT Press, 2018.

[5] Wang X S, Wang R R, Cheng Y H. Safe reinforcement learning: a survey[J]. Acta Automatica Sinica, 2023, 49(9):1-23.

[6] SUN Y W, LIU W Z, SUN C Y. Causality in reinforcement learning control: the state of the art and prospects[J]. Acta Automatica Sinica, 2023, 49(3):661-677.

[7] Vinyals O, Babuschkin I, Czarnecki W M, et al. Grandmaster level in starcraft II using multiagent reinforcement learning[J]. Nature, 2019, 575(7782):350-354.

[8] Liu I J, Jain U, Yeh R A, et al. Cooperative exploration for multi-agent deep reinforcement learning[C]//Proceedings of 38th International Conference on Machine Learning, 2021:6826-6836.

[9] Rashid T, Samvelyan M, De Witt C S, et al. Monotonic value function factorisation for deep multi-agent reinforcement learning[J]. Journal of Machine Learning Research, 2020, 21(1):7234-7284.

[10] Silver D, Huang A, Maddison C J, et al. Mastering the game of go with deep neural networks and tree search[J]. Nature, 2016, 529(7587):484-489.

[11] Silver D, Hubert T, Schrittwieser J, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play[J]. Science, 2018, 362(6419):1140-1144.

[12] Brown N, Sandholm T. Superhuman AI for multiplayer poker[J]. Science, 2019, 365(6456):885-890.

[13] Polydoros A S, Nalpantidis L. Survey of model-based reinforcement learning: applications on robotics[J]. Journal of Intelligent & Robotic Systems, 2017, 86(2):153-173.

[14] Zhu K, Zhang T. Deep reinforcement learning based mobile robot navigation: a review[J]. Tsinghua Science and Technology, 2021, 26(5):674-691.

[15] Wang C, Wang J, Shen Y, et al. Autonomous navigation of UAVs in large-scale complex environments: a deep reinforcement learning approach[J]. IEEE Transactions on Vehicular Technology, 2019, 68(3):2124-2136.

[16] Zhang D X, Han X Q, Deng C Y. Review on the research and practice of deep learning and reinforcement learning in smart grids[J]. CSEE Journal of Power and Energy Systems, 2018, 4(3): 362-370.

[17] Aradi S. Survey of deep reinforcement learning for motion planning of autonomous vehicles [J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 23(2): 740-759.

[18] Zhao W, Queralta J P, Westerlund T. Sim-to-real transfer in deep reinforcement learning for robotics: a survey[C]//Proceedings of IEEE Symposium Series on Computational Intelligence, 2020: 737-744.

[19] Scheikl P M, Tagliabue E, Gyenes B, et al. Sim-to-real transfer for visual reinforcement learning of deformable object manipulation for robot assisted surgery[J]. IEEE Robotics and Automation Letters, 2022, 8(2): 560-567.

[20] Xie T Y, Jiang N, Wang H, et al. Policy finetuning: bridging sample-efficient offline and online reinforcement learning[C]//Proceedings of Advances in Neural Information Processing Systems, 2021: 27395-27407.

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

[22] Zhang L, Zhang R, Wu T, et al. Safe reinforcement learning with stability guarantee for motion planning of autonomous vehicles [J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 32(12): 5435-5444.

[23] Kiran B R, Sobh I, Talpaert V, et al. Deep reinforcement learning for autonomous driving: a survey[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 23(6): 4909-4926.

[24] Chen J, Yuan B, Tomizuka M. Model-free deep reinforcement learning for urban autonomous driving[C]//Proceedings of IEEE Intelligent Transportation Systems Conference, 2019: 2765-2771.

[25] Chen J, Li S E, Tomizuka M. Interpretable end-to-end urban autonomous driving with latent deep reinforcement learning[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 23(6): 5068-5078.

[26] Liu S, See K C, Ngiam K Y, et al. Reinforcement learning for clinical decision support in critical care: comprehensive review[J]. Journal of Medical Internet Research, 2020, 22(7): e18477, doi: 10.2196/18477.

[27] Yu C, Liu J, Nemati S, et al. Reinforcement learning in healthcare: a survey[J]. ACM Computing Surveys, 2021, 55(1): 1-36.

[28] Coronato A, Naeem M, De Pietro G, et al. Reinforcement learning for intelligent healthcare applications: a survey[J]. Artificial Intelligence in Medicine, 2020, 109: 101964, doi: 10.1016/j.artmed.2020.101964.

[29] Wang C, Wang J, Wang J, et al. Deep reinforcement learning based autonomous UAV navigation with sparse rewards[J]. IEEE Internet of Things Journal, 2020, 7(7): 6180-6190.

[30] Koch W, Mancuso R, West R, et al. Reinforcement learning for UAV attitude control[J]. ACM Transactions on Cyber Physical Systems, 2019, 3(2): 1-21.

[31] Chen J, Jiang N. Information-theoretic considerations in batch reinforcement learning[C]//Proceedings of International Conference on Machine Learning, 2019: 1042-1051.

[32] Zhao D B, Shao K, Zhu Y H, et al. Review of deep reinforcement learning and discussions on the development of computer go[J]. Control Theory & Applications, 2016, 33(6): 701-717.

[33] Fujimoto S, Meger D, Precup D. O-policy deep reinforcement learning without exploration [C]//Proceedings of International Conference on Machine Learning, 2019: 2052-2062.

[34] Kumar A, Fu J, Tucker G, et al. Stabilizing o-policy q-learning via bootstrapping error reduction[C]//Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2019: 11784-11794.

[35] Wu Y, Tucker G, Nachum O. Behavior regularized offline reinforcement learning[J]. arXiv preprint arXiv: 1911.11361, 2019.

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

[37] Kumar A, Zhou A, Tucker G et al. Conservative q learning for offline reinforcement learning[C]//Proceedings of Advances in Neural Information Processing Systems, 2020: 1179-1191.

[38] Kostrikov I, Nair A, Levine S. Offline reinforcement learning with implicit q learning[J]. arXiv preprint arXiv: 2110.06169, 2021.

[39] Nair A, Gupta A, Dalal M, et al. Awac: accelerating online reinforcement learning with online datasets[J]. arXiv preprint arXiv: 2006.09359, 2020.

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

[41] Kidambi R, Rajeswaran A, Netrapalli P, et al. MOReL: model-based offline reinforcement learning[C]//Proceedings of Advances in Neural Information Processing Systems, 2020: 21810-21823.

[42] Yu T, Thomas G, Yu L, et al. MOPO: model-based offline policy optimization[C]//Proceedings of Advances in Neural Information Processing Systems, 2020: 14129-14142.

[43] Yu T, Kumar A, Rafailov R, et al. COMBO: conservative offline model-based policy optimization[C]//Proceedings of Advances in Neural Information Processing Systems, 2021: 28954-28967.

[44] Marc R, Lacerda B, Hawes N. Rambo-rl: robust adversarial model-based offline reinforcement learning[C]//Proceedings of Advances in Neural Information Processing Systems, 2022: 16082-16097.

[45] Bhardwaj M, Xie T, Boots B, et al. Adversarial model for offline reinforcement learning[J]. arXiv preprint arXiv: 2302.11048, 2023.

[46] Prudencio R F, Maximo M R O A, Colombini E L. A survey on offline reinforcement learning: taxonomy, review, and open problems [J]. IEEE Transactions on Neural Networks and Learning Systems, 2023, 35(8): 10237-10257.

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

[48] Gulcehre C, Wang Z, Novikov A, et al. Rl unplugged: a suite of benchmarks for offiine reinforcement learning[C]//Proceedings of Advances in Neural Information Processing Systems, 2020: 7248-7259.

[49] Qin R J, Zhang X, Gao S, et al. NeoRL: a near real-world benchmark for offline reinforcement learning[C]//Proceedings of Advances in Neural Information Processing Systems, 2022: 24753-24765.

[50] Otterlo M V, Wiering M. Reinforcement learning and markov decision processes[J]. Springer Berlin Heidelberg, 2012, doi: 10.1007/978-3-642-27645-3_1.

[51] Lauri M, Hsu D, Pajarinen J. Partially observable markov decision processes in robotics: a survey[J]. IEEE Transactions on Robotics, 2022, 39(1): 21-40.

[52] Hausknecht M, Stone P. Deep recurrent q-learning for partially observable MDPs[J]. arXiv preprint arXiv:1507.06527, 2015.
[53] Zhang X, Zheng K, Wang C, et al. A novel deep reinforcement learning for POMDP-based autonomous ship collision decision-making[J]. Neural Computing and Applications, 2025, 37(21): 15963-15977.
[54] Guo H, Cai Q, Zhang Y, et al. Provably efficient offline reinforcement learning for partially observable markov decision processes [C]//Proceedings of International Conference on Machine Learning, 2022: 8016-8038.
[55] Bertsekas D. Dynamic programming and optimal control [M]. Nashua, USA: Athena Scientific, 1995.
[56] Balhara S, Gupta N, Alkhayyat A, et al. A survey on deep reinforcement learning architectures, applications and emerging trends [J]. IET Communications, 2025, 19(1): 1-16.
[57] LI R Y, PENG H M, LI R G, et al. Overview on algorithms and applications for reinforcement learning[J]. Computer Systems & Applications, 2020, 29(12): 13-25.
[58] WEN G H, YANG T, ZHOU J L, et al. Reinforcement learning and adaptive/approximate dynamic programming: a survey from theory to applications in multi-agent systems[J]. Control and Decision, 2023, 38(5): 1200-1230.
[59] Bellman R. Dynamic programming and Lagrange multipliers[J]. Proceedings of the National Academy of Sciences, 1956, 42(10): 767-769.
[60] Sutton R S, Mcallester D A, Singh S P, et al. Policy gradient methods for reinforcement learning with function approximation[C]// Proceedings of Advances in Neural Information Processing Systems, 1999: 1057-1063.
[61] Mnih V, Badia A P, Mirza M, et al. Asynchronous methods for deep reinforcement learning[C]//Proceedings of the 33rd International Conference on Machine Learning, 2016: 1928-1937.
[62] Schulman J, Levine S, Abbeel P, et al. Trust region policy optimization[C]//Proceedings of the 32nd International Conference on Machine Learning, 2015: 1889-1897.
[63] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[J]. arXiv preprint arXiv: 1707.06347, 2017.
[64] Watkins C J, Dayan P. Q learning[J]. Machine Learning, 1992, 8: 279-292, doi: 10.1007/BF00992698.
[65] Rummery G A, Niranjan M. On-line q-learning using connectionist systems [D]. Cambridge: University of Cambridge, 1994.
[66] Schaul T, Quan J, Antonoglou I, et al. Prioritized experience replay [J]. arXiv preprint arXiv: 1511.05952, 2015.
[67] Hasselt H V, Guez A, Silver D. Deep reinforcement learning with double q learning[C]//Proceedings of the AAAI Conference on Artificial Intelligence, 2016: 2094-2100.
[68] Wang Z, Schaul T, Hessel M, et al. Dueling network architectures for deep reinforcement learning[C]//Proceedings of the 33rd International Conference on Machine Learning, 2016: 1995-2003.
[69] Lillicrap P T, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning [J]. arXiv preprint arXiv: 1509.02971, 2016.
[70] Silver D, Lever G, Heess N, et al. Deterministic policy gradient algorithms[C]//Proceedings of the 31st International Conference on Machine Learning, 2014: 387-395.
[71] Fujimoto S, Van Hoof H, Meger D. Addressing function approximation error in actor-critic methods[C]//Proceedings of the International Conference on Machine Learning, 2018: 1582-1591.
[72] Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: o-policy maximum entropy deep reinforcement learning with a stochastic actor[C]//Proceedings of the International Conference on Machine Learning, 2018: 1861-1870.
[73] Wang S R, Niu W J, Tong E D, et al. Research on o-policy evaluation in reinforcement learning: a survey [J]. Chinese Journal of Computers, 2022, 45(9): 1926-1948.
[74] Lyu J, Ma X, Li X, et al. Mildly conservative Q learning for offline reinforcement learning [C]//Proceedings of Advances in Neural Information Processing Systems, New Orleans, 2022: 1711-1724.
[75] Fujimoto S, Gu S S. A minimalist approach to offline reinforcement learning[J]. arXiv preprint arXiv: 2106.06860, 2021.
[76] Chen X, Zhou Z, Wang Z, et al. BAIL: best-action imitation learning for batch deep reinforcement learning [C]//Proceedings of Advances in Neural Information Processing Systems, 2020: 18353-18363.
[77] Guo K Y, Shao Y F, Geng Y H. Model-based offline reinforcement learning with pessimism-modulated dynamics belief[C]//Proceedings of Advances in Neural Information Processing Systems, 2022: 449-461.
[78] Matsushima T, Furuta H, Matsuo Y, et al. Deployment efficient reinforcement learning via model based offline optimization[J]. arXiv preprint arXiv: 2006.03647, 2020.
[79] Chen L, Lu K, Rajeswaran A, et al. Decision transformer: reinforcement learning via sequence modeling [C]//Proceedings of Advances in Neural Information Processing Systems, 2021: 15084-15097.
[80] Michael J, Li Q Y, Levine S. Offline reinforcement learning as one big sequence modeling problem[C]//Proceedings of Advances in Neural Information Processing Systems, 2021: 1273-1286.
[81] Yang R, Zhong H, Xu J, et al. Towards robust offline reinforcement learning under diverse data corruption [C]//Proceedings of 12th International Conference on Learning Representations, 2023: 1-32.
[82] Nguyen Tang T, Arora R. On sample-efficient offline reinforcement learning: data diversity, posterior sampling and beyond[C]//Proceedings of Advances in Neural Information Processing Systems, 2023: 61115-61157.
[83] Mediratta I, You Q, Jiang M, et al. The generalization gap in offline reinforcement learning[J]. arXiv preprint arXiv: 2312.05742, 2023.
[84] Deng J, Dong W, Socher R, et al. Imagenet: a large-scale hierarchical image database [C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2009: 248-255.
[85] Veit A, Matera T, Neumann L, et al. Cocotext: dataset and benchmark for text detection and recognition in natural images[J]. arXiv preprint arXiv: 1601.07140, 2016.
[86] Todorov E, Erez T, Tassa Y. MuJoCo: a physics engine for model-based control [C]//Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012: 5026-5033.
[87] Rajeswaran A, Kumar V, Gupta A, et al. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations[J]. arXiv preprint arXiv: 1709.10087, 2018.
[88] Gupta A, Kumar V, Lynch C, et al. Relay policy learning: solving long-horizon tasks via imitation and reinforcement learning[J]. arXiv preprint arXiv: 1910.11956, 2019.
[89] Vinitsky E, Kreidieh A, Flem L L, et al. Benchmarks for reinforcement learning in mixed-autonomy traffic[C]//Proceedings of the Conference on Robot Learning, 2018: 399-409.
[90] Dosovitskiy A, Ros G, Codevilla F, et al. CARLA: an open urban driving simulator[J]. arXiv e-prints, arXiv-1711, 2017.

[91] Tassa Y, Doron Y, Muldal A, et al. DeepMind control suite[J]. arXiv preprint arXiv:1801.00690,2018.
[92] Bellemare M G, Naddaf Y, Veness J, et al. The arcade learning environment: an evaluation platform for general agents[J]. Journal of Artificial Intelligence Research,201,3(47):253-279.
[93] Dulac G, Mankowitz D, Hester T. Challenges of real-world reinforcement learning[J]. arXiv preprint arXiv:1904.12901,2019.
[94] Hein D, Depeweg S, Tokic M, et al. A benchmark environment motivated by industrial control problems[C]//Proceedings of IEEE Symposium Series on Computational Intelligence,2017:1-8.
[95] Liu X Y, Yang H, Chen Q, et al. FinRL: a deep reinforcement learning library for automated stock trading in quantitative finance [J]. arXiv preprint arXiv:2011.09607,2020.
[96] VSzquez J R, Kampf J, Henze G, et al. Citylearn v1.0: an openai gym environment for demand response with deep reinforcement learning[C]//Proceedings of the 6th ACM International Conference on Systems for Energy Efficient Buildings, Cities, and Transportation,2019:356-357.
[97] Pomerleau D A. Alvinn: an autonomous land vehicle in a neural network[C]//Proceedings of Advances in Neural Information Processing Systems,1989:305-313.
[98] Wang C, Wu Y, Vuong Q, et al. Striving for simplicity and performance in off policy drl: output normalization and nonuniform sampling[J]. arXiv preprint arXiv:1910.02208,2019.
[99] Nachum O, Dai B, Kostrikov I, et al. Algaedice: policy gradient from arbitrary experience [J]. arXiv preprint arXiv: 1912.02074,2019.
[100] Barth M G, Hoffman M W, Budden D, et al. Distributed distributional deterministic policy gradients [J]. arXiv preprint arXiv: 1804.08617,2018.
[101] Dabney W, Ostrovski G, Silver D, et al. Implicit quantile networks for distributional reinforcement learning[J]. arXiv preprint arXiv: 1806.06923,2018.
[102] Siegel N Y, Springenberg J T, Berkenkamp F, et al. Keep doing what worked: behavioral modelling priors for offline reinforcement learning[J]. arXiv preprint arXiv:2002.08396,2020.
[103] Agarwal R, Schuurmans D, Norouzi M. An optimistic perspective on offline reinforcement learning[C]//Proceedings of the International Conference on Machine Learning,2020:104-114.
[104] Zhou W, Bajracharya S, Held D. Plas: latent action space for offline reinforcement learning[C]//Proccceedings of the Conference on Robot Learning,2020:1719-1735.
[105] Wang Z, Novikov A, Zolna K, et al. Critic regularized regression [C]//Proceedings of the Advances in Neural Information Processing Systems,2020:7768-7778.
[106] Singh B, Kumar R, Singh V. Reinforcement learning in robotic applications: a comprehensive survey[J]. Artifical Intelligence Review,2022,55(2):945-990.
[107] Zhou G, Dean V, Srirama M K, et al. Train offline, test online: a real robot kearning benchmark[J]. arXiv preprint arXiv: 2306.00942,2023.
[108] Fang X, Zhang Q, Gao Y, et al. Offline reinforcement learning for autonomous driving with real world driving data[C]//Proceedings of IEEE 25th International Conference on Intelligent Transportation Systems,2022:3417-3422.
[109] Hu B, Li J. A deployment efficient energy management strategy for connected hybrid electric vehicle based on offline reinforcement learning[J]. IEEE Transactions on Industrial Electronics,2021,69(9):9644-9654.
[110] He H, Niu Z, Wang Y, et al. Energy management optimization for connected hybrid electric vehicle using offline reinforcement learning[J]. Journal of Energy Storage, 2023, 72: 108517, doi: 10.1016/j.est.2023.108517.
[111] Wang L, Zhang W, He X, et al. Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018: 2447-2456.
[112] Swaminathan A, Krishnamurthy A, Agarwal A, et al. Off policy evaluation for slate recommendation[C]//Proceedings of Advances in Neural Information Processing Systems,2017:3632-3642.
[113] Covington P, Adams J, Sargin E. Deep neural networks for youtube recommendations[C]//Proceedings of the 10th ACM Conference on Recommender Systems,2016:191-198.
[114] Chen M, Beutel A, Covington P, et al. Top-k o ffpolicy correction for a reinforce recommender system[C]//Proceedings of the 12th ACM International Conference on Web Search and Data Mining, 2018:456-464.
[115] Verma S, Fu J, Yang M, et al. Chai: a chatbot ai for task-oriented dialogue with offline reinforcement learning [J]. arXiv preprint arXiv:2204.08426,2022.
[116] Jaques N, Ghandeharioun A, Shen J H, et al. Way offpolicy batch deep reinforcement learning of implicit human preferences in dialog[J]. arXiv preprint arXiv:1907.00456,2019.
[117] Kumar A, Singh A, Tian S, et al. A workflow for offline model free robotic reinforcement learning [J]. arXiv preprint arXiv: 2109.10813,2021.
[118] Diehl C, Sievernich T S, Krjger M, et al. Uncertainty aware model based offline reinforcement learning for automated driving [J]. IEEE Robotics and Automation Letters,2023,8(2):1167-1174.
[119] Snell C, Kostrikov I, Su Y, et al. Offline RL for natural language generation with implicit language q learning[J]. arXiv preprint arXiv:2206.11871,2022.
[120] Xiao T, Wang D. A general offline reinforcement learning framework for interactive recommendation [C]//Proceedings of the AAAI Conference on Artificial Intelligence,2021:4512-4520.
[121] Fu W Y, Di B B. Batch reinforcement learning in the real world: a survey[C]//Proceedings of Offline RL Workshop,2020:1-13.
[122] Gürtler N, Blaes S, Kolev P, et al. Benchmarking offline reinforcement learning on real-robot hardware[J]. arXiv preprint arXiv: 2307.15690,2023.

附中文参考文献:

[6] 孙悦雯,柳文章,孙长银. 基于因果建模的强化学习控制:现状及展望[J]. 自动化学报,2023,49(3):661-677.
[57] 李茹杨,彭慧民,李仁刚,等. 强化学习算法与应用综述[J]. 计算机系统应用,2020,29(12):13-25.
[58] 温广辉,杨　涛,周佳玲,等. 强化学习与自适应动态规划:从基础理论到多智能体系统中的应用进展综述[J]. 控制与决策,2023,38(5):1200-1230.