工程科学学报
Chinese Journal of Engineering

基于时间差分误差的离线强化学习采样策略

张龙飞 冯旸赫 梁星星 刘世旋 程光权 黄金才

Sample strategy based on TD-error for offline reinforcement learning

ZHANG Longfei, FENG Yanghe, LIANG Xingxing, LIU Shixuan, CHENG Guangquan, HUANG Jincai

引用本文:
张龙飞, 冯旸赫, 梁星星, 刘世旋, 程光权, 黄金才. 基于时间差分误差的离线强化学习采样策略[J]. 工程科学学报, 2023, 45(12): 2118–2128. doi: 10.13374/j.issn2095–9389.2022.10.22.001
ZHANG Longfei, FENG Yanghe, LIANG Xingxing, LIU Shixuan, CHENG Guangquan, HUANG Jincai. Sample strategy based on TD–error for offline reinforcement learning[J]. *Chinese Journal of Engineering*, 2023, 45(12): 2118–2128. doi: 10.13374/j.issn2095–9389.2022.10.22.001

在线阅读 View online: https://doi.org/10.13374/j.issn2095–9389.2022.10.22.001

工程科学学报, 第 45 卷, 第 12 期: 2118−2128, 2023 年 12 月
Chinese Journal of Engineering, Vol. 45, No. 12: 2118−2128, December 2023
https://doi.org/10.13374/j.issn2095-9389.2022.10.22.001; http://cje.ustb.edu.cn

基于时间差分误差的离线强化学习采样策略

张龙飞✉, 冯旸赫✉, 梁星星, 刘世旋, 程光权, 黄金才

国防科技大学系统工程学院, 长沙 410073

✉通信作者, 张龙飞, E-mail: zhanglongfei@nudt.edu.cn; 冯旸赫, E-mail: fengyanghe@nudt.edu.cn

摘　要　离线强化学习利用预先收集的专家数据或其他经验数据, 在不与环境交互的情况下离线学习动作策略. 与在线强化学习相比, 离线强化学习具有样本效率高、交互成本低的优势. 强化学习中通常使用 Q 值估计函数或 Q 值估计网络表示状态−动作的价值. 因无法通过与环境交互及时修正 Q 值估计误差, 离线强化学习往往面临外推误差严重、样本利用率低的问题. 为此, 提出基于时间差分误差的离线强化学习采样方法, 使用时间差分误差作为样本优先采样的优先度度量, 通过使用优先采样和标准采样相结合的采样方式, 提升离线强化学习的采样效率并缓解分布外误差问题. 同时, 在使用双 Q 值估计网络的基础上, 根据目标网络的不同计算方法, 比较了 3 种时间差分误差度量所对应的算法的性能. 此外, 为消除因使用优先经验回放机制的偏好采样产生的训练偏差, 使用了重要性采样机制. 通过在强化学习公测数据集—深度数据驱动强化学习数据集上与已有研究成果相比, 基于时间差分误差的离线强化学习采样方法在最终性能、数据效率和训练稳定性上均有更好的表现. 消融实验表明, 优先采样和标准采样相结合的采样方式对算法性能的发挥至关重要, 同时, 使用最小化双目标 Q 值估计的时间差分误差优先度度量所对应的算法, 在多个任务上具有最优的性能. 基于时间差分误差的离线强化学习采样方法可与任何基于 Q 值估计的离线强化学习方法结合, 具有性能稳定、实现简单、可扩展性强的特点.

关键词　离线; 强化学习; 采样策略; 经验回放; 时间差分误差

分类号　TG142.71

Sample strategy based on TD-error for offline reinforcement learning

ZHANG Longfei✉, *FENG Yanghe*✉, *LIANG Xingxing*, *LIU Shixuan*, *CHENG Guangquan*, *HUANG Jincai*

College of Systems Engineering, National University of Defense Technology, Changsha 410073, China

✉Corresponding authors, ZHANG Longfei, E-mail: zhanglongfei@nudt.edu.cn; FENG Yanghe, E-mail: fengyanghe@nudt.edu.cn

ABSTRACT　Offline reinforcement learning uses pre-collected expert data or other empirical data to learn action strategies offline without interacting with the environment. Offline reinforcement learning is preferable to online reinforcement learning because it has lower interaction costs and trial-and-error risks. However, offline reinforcement learning often faces the issues of severe extrapolation errors and low sample utilization because the Q-value estimation errors cannot be corrected in time by interacting with the environment. To this end, this paper suggests an effective sampling strategy for offline reinforcement learning based on TD-error, using TD-error as the priority measure for priority sampling, and enhancing the sampling efficacy of offline reinforcement learning and addressing the issue of out-of-distribution error by using a combination of priority sampling and uniform sampling. Meanwhile, based on the use of the dual Q-value estimation network, this paper examines the performance of the algorithms corresponding to their time-difference error measures when determining the target network using three approaches, including the minimum, the maximum, and the convex combined of dual Q-value network, according to the various calculation techniques of the target network. Furthermore, to eliminate the training bias arising from preference sampling using priority sampling, this paper uses a significant sampling mechanism. By comparing with existing offline reinforcement learning research results combining sampling strategies on the D4RL baseline, the algorithm proposed

收稿日期: 2022−10−22

基金项目: 国家自然科学基金面上资助项目(62273352)

shows better performance in terms of the final performance, data efficiency, and training stability. To confirm the contribution of each research point in the algorithm, two experiments were performed in the ablation experiment section of this study. Experiment 1 shows that the algorithm using the sampling method with a combination of uniform sampling and priority sampling outperforms the algorithm using uniform sampling alone and the algorithm using priority sampling alone in terms of sample utilization and strategy stability, while experiment 2 compares the effect on the performance of the algorithm based on the double Q-value estimation network produced by the double network of a maximum, minimum, and maximum-minimum convex combination of values based on the dual Q-value estimation network with a total of three different time-difference calculation methods on the performance of the algorithm. Experimental evidence shows that the algorithm in the research that uses the least amount of dual networks performs better overall and in terms of data utilization than the other two algorithms, but its strategy variance is higher. The approach described in this paper can be used in conjunction with any offline reinforcement learning method based on Q-value estimation. This approach has the advantages of stable performance, straightforward implementation, and high scalability, and it supports the use of reinforcement learning techniques in real-world settings.

KEY WORDS offline; reinforcement learning; sample strategy; experience replay buffer; TD-error

近年来，强化学习技术凭借其自主学习的优势在游戏[1]、自动驾驶[2]、核能利用[3]、算法发现[4]和兵棋推演[5]等领域取得重要进展. 与传统基于搜索的决策算法不同，强化学习通过与环境交互试错，根据获取的反馈奖励学习状态−动作的价值函数，不断学习更新行动策略. 这种数据驱动的学习方式决定了强化学习的策略学习依赖于大规模的学习数据，存在训练效率低、学习收敛慢、过程不稳定等问题. 为此，研究者们提出离策略(Off-Policy)强化学习方法，比如异步优势演员−评论者(A3C)[6]、软性演员−评论者(SAC)[7]、双延迟深度确定性策略梯度(TD3)[8]等，基于模型的强化学习方法，比如深度规划网络(PlaNet)[9]、想象者(Dreamer)[10]、想象者-v2(Dreamer-v2)[11]等，可以有效提升样本利用率、加快学习效率. 尽管这些方法在多个复杂决策和控制任务上展现出优异性能，但仍需通过与环境交互获取训练数据，属于在线强化学习方法的范畴.

与在线强化学习方法不同，批量−约束的深度Q学习(BCQ)[12]、基于数据增强的离线强化学习方法(ORAD)[13]、基于模型的离线−在线强化学习方法(MOORE)[14]，双延迟深度确定性策略梯度−行为克隆(TD3-BC)[15]等离线强化学习方法，是指智能体无需与环境进行实时交互，而是利用预先收集的离线数据，在离线的情况下学习策略. 在交互成本高、在线样本采集难的场景下，离线强化学习方法具有一定节省交互成本、降低交互风险的优势. 然而，离线强化学习不与环境交互的特性，也给其训练带来了“外推误差”的问题. 当离线训练的智能体在实际环境中进行交互时，会对训练时未见过的新状态、新动作出现乐观估计，导致学习的价值函数估计误差累积，从而引发性能下降. 为此，大量离线强化学习方法通过在目标函数中增加惩罚项、正则项等方式，使离线学习的策略保持在预先收集的数据分布的附近，从而约束了离线学习的策略在实际环境中的过度探索. 尽管这些方法在一定程度上缓解了外推误差的问题，但因引入了额外的约束项和超参数，使得训练更为复杂，限制了其在实际场景中的推广应用.

本文提出一种基于时间差分误差的采样策略，通过将优先经验回放的优先采样方法与标准经验回放的标准采样方法相结合，在不同的训练阶段采用不同的训练机制，以提升训练效率，降低样本采样率. 然而，以往的研究经验表明，使用优先经验采样会因采样偏好导致出现训练偏差，为消除这种偏差，本文使用重要性采样机制. 时间差分误差作为强化学习中衡量策略优劣的指标，通常被用于优先采样的优先度. 本文结合离线强化学习算法——保守Q估计离线强化学习方法[16](Conservative Q-learning for offline reinforcement learning, CQL)，基于不同的目标函数估计方法，比较了以3种不同的时间差分误差值作为优先采样的优先度度量指标时的算法性能. 经过在离线数据集——深度数据驱动强化学习数据集(Datasets for deep data-driven reinforcement learning, D4RL)上的实验可以看出，本文的方法在样本利用率、最终性能和训练稳定性上都明显优于其他同类算法[17]. 此外，通过消融实验也可发现，优先采样和标准采样相结合的采样方式是本文算法的关键. 同时，使用最小化双Q值目标估计的时间差分误差优先度度量所对应的算法，在多个任务上具有最优的性能. 本文的方法主要贡献如下：(1)提出了一种新

的包含优先采样和标准采样的离线强化学习采样策略, 可有效提升样本效率; (2) 比较了 3 种基于时间差分误差的优先度度量指标所对应的算法的性能. 本文的方法聚焦于样本采样策略, 无需在算法的目标函数中额外引入偏置归纳项, 避免了对原始算法的修改, 易于与其他离线学习算法结合, 具有较好的实用性和可扩展性.

1 强化学习中的采样方法

1.1 基于经验回访池的采样方法

强化学习需利用智能体和环境交互产生的数据不断更新策略直至收敛, 这些数据往往质量不一、标签稀缺, 且无法重复使用. 为提升数据利用率, 强化学习引入了经验回放机制, 即将智能体与环境在一定时间段内交互生成的数据存储起来组成一个经验池, 训练时再从经验池中采集训练样本. 为保持样本的多样性, 一般都使用随机采样的方法. 优先经验回放 (Prioritized experience replay, PER) 则是考虑到数据的质量不同, 部分具有更高价值的训练数据应被更多次采样, 因此使用每条数据的时间差分误差值来定义优先度, 每次训练时按照数据的优先度进行采样, 可有效提升样本利用率[18]. 此外, 竞争经验回放 (Competitive experience replay, CER) 通过使用双经验回放池, 从 2 个经验池中采样训练数据, 鼓励 2 个智能体相互竞争实现对环境的探索, 引导智能体在目标导向的任务中探索出最优策略[19]. Fu 等[20] 在优先采样策略的基础上, 定义了时间差分误差、N 步奖励返回值、虚拟计数、似然估计、通用自模仿学习、虚拟计数 6 种不同的优先度指标, 通过大量实验表明, 以上指标均可有效提升离线强化学习的采样效率. Lee 等[21] 提出针对离线训练的智能体在线微调再学习时, 易因数据分布偏差产生严重的自举误差的问题, 使用一个平衡的经验回放机制, 优先采样遇到的在线样本, 同时也鼓励使用来自离线数据集的接近策略的样本, 该方法主要是解决数据分布偏差问题, 并未对离线数据本身进行优先度度量以提升学习效率.

1.2 时间差分误差

时间差分误差是智能体自举产生的目标值与价值网络评估值之间的差异, 反映的是在当前策略和价值函数下, 状态或状态–动作对的相对优劣, 是对状态或状态–动作对的评估指标[22]. 按照时间差分中使用的自举步数, 可将其分为 1 步时间差分和 N(N>1) 步时间差分. PER 使用 1 步时间差分误差作为数据元组优先度的度量指标, 并把数据存储在 SumTree 结构中, 每次按照优先采样方式采集出批量数据进行训练, 并在每次网络更新后重新计算时间差分误差, 以赋予数据元组新的优先度. PER 使得时间差分误差值大的数据获得更多采样次数, 有效提升了深度 Q 网络算法 (Deep Q-network, DQN) 的样本利用率和最终性能. Rainbow 则是采用 N 步返回值计算时间差分误差, 通过 N 步返回值计算目标值与价值网络估计值的差值, 以该差值作为数据元组的优先度[23]. 相比于 1 步时间差分误差, N 步时间差分误差具有更小的估计偏差, 但同时也增加了后续推演的计算量, 增大了算法的复杂度.

1.3 重要性采样机制

重要性采样是通过给样本添加重要性系数, 修正因偏好采样产生的训练偏差. 优先经验回放中的优先采样, 尽管是从完整数据集中采集样本, 但因偏好优先度高的数据, 易造成神经网络梯度更新偏向于优先度高的数据集, 带来训练偏差. 为此, 很多算法采用了重要性采样来修正该训练偏差. 鉴于本文赋予数据元组的初始优先度为 1, 且随着训练步数增长不断降低优先度, 数据元组相对应的重要性权重也应遵循随训练步数增大逐渐递减的变化规律. 重要性采样在强化学习算法中应用广泛, 置信域策略优化[24] (Trust region policy optimization, TRPO) 为随机策略更新添加了一个置信域, 通过 KL 散度 (Kullback-Leibler divergence) 惩罚来约束梯度更新, 保证了梯度的正向更新. 近端策略优化[25] (Proximal policy optimization, PPO) 则是计算新策略与旧策略的重要性比率, 以此衡量新策略与旧策略的差异, 通过设置截断比率的范围值, 约束新策略的梯度正向更新.

2 离线强化学习方法

2.1 强化学习

强化学习通常被表示为马尔科夫决策过程[22] (Markov decision process, MDP), $\boldsymbol{M}=(\boldsymbol{S},\boldsymbol{A},\boldsymbol{P},\boldsymbol{R},p_0,\gamma,H)$, 其中 $s_t\in\boldsymbol{S}$ 表示状态空间, $a_t\in\boldsymbol{A}$ 表示动作空间, $\boldsymbol{P}: s_{t+1}\sim P(\cdot|s_t,a_t)$ 表示状态转移概率函数, $\boldsymbol{R}: r_t=R(s_t,a_t)$ 表示奖励函数, p_0 表示初始状态分布, γ 表示折扣因子, H 表示任务的尺度, 一般是指片段环境中一个片段的固定步数, s_t, a_t, p_0, r_t 分别表示 t 时刻的状态、动作、0 时刻的状态转移概率和 t 时刻的奖励. 强化学习的目标是使用经验来学习一个行动策略 $\pi:\boldsymbol{S}\to\boldsymbol{A}$, 以最大化累积折扣期望

奖励: $J(\pi)=E_\pi[\sum_{t=0}^{\infty}\gamma^t r_t(s_t,a_t,s_{t+1})]$, 其中 $s_0\sim p_0$, E_π 表示当使用策略 π 时的累积折扣期望奖励, $J(\pi)$ 表示关于策略 π 的目标函数, 并且对于每个时间步 t, 都有 $a_t\sim\pi(\cdot|s_t)$, $s_{t+1}\sim P(\cdot|s_t,a_t)$, 且 $r_t=R(s_t,a_t)$. 每个策略 π 都有一个相对应的 Q 值函数 $Q^\pi(s_t,a_t)=E_\pi[R_t|s_t,a_t]$, 表示在状态 s_t 采取动作 a_t 后, 执行策略 π 所获得的奖励期望. Q 值函数可通过如下贝尔曼算子得到:

$$\mathcal{T}^\pi Q(s_t,a_t)=E_{s_{t+1}}[r_t+\gamma Q(s_{t+1},\pi(s_{t+1}))] \tag{1}$$

其中, $\mathcal{T}^\pi$ 表示贝尔曼算子. Q 学习(Q-learning)[26] 是通过迭代使用贝尔曼算子 $Q^{k+1}=\mathcal{T}^\pi Q^k$, 直至 Q^k 收敛到 Q^π, 之后通过贪婪动作选择获取最优值函数 $Q^*(s_t,a_t)=\max_\pi Q^\pi(s_t,a_t)$ 所对应的最优策略, 其中 $Q^*(s_t,a_t)$ 表示在状态 s_t 采取动作 a_t 的最优 Q 值. DQN 是在 Q 学习基础上, 使用深度网络表示连续状态空间或巨大动作空间所对应的值函数, 按照如公式(1)所示的计算方法, 以值迭代的方式更新网络参数, 获取最优价值函数[27]. DQN 通过最小化如下目标实现值函数的更新:

$$\begin{aligned}J_Q(\phi)=E_{(s_t,a_t,s_{t+1})\sim D}[(Q_\phi(s_t,a_t)-(r(s_t,a_t)+\\ \gamma E_{a_{t+1}\sim\pi(\cdot|s_{t+1})}[Q_{\phi'}(s_{t+1},a_{t+1})]))^2]\end{aligned} \tag{2}$$

其中, ϕ 表示价值网络参数, ϕ' 表示延迟更新的价值网络参数.

2.2 保守 Q 估计离线强化学习方法

保守 Q 估计离线强化学习方法(CQL)是通过在 SAC 算法的基础上, 为 Q 值估计网络添加一个保守约束项, 惩罚 Q 函数对分布外数据的乐观估计. SAC 算法的策略评估通过最小化软贝尔曼残差来更新价值函数网络的参数:

$$J(\pi)=E_{s_t,a_t,r_t,s_{t+1}\sim D}[(Q(s_t,a_t)-r_t-\gamma\bar{V}(s_t+1))^2] \tag{3}$$

其中, $\bar{V}(s_t)$ 是软价值函数, 其定义为在标准价值函数基础上添加策略函数分布熵的正则项:

$$\bar{V}(s_t)=E_{a_t\sim\pi}[Q(s_t,a_t)-\alpha\lg\pi(a_t|s_t)] \tag{4}$$

其中, α 是温度系数, 正则项 $\log\pi(a_t|s_t)$ 鼓励策略进行探索和增强对噪声的鲁棒性. SAC 算法的策略提升则是通过最小化策略 $\pi(\cdot|s)$ 和一个基于价值网络 $Q(s_t,\pi(s_t))$ 的波兹曼分布之间的 KL 散度来实现的, 策略网络的目标函数表示如下:

$$J(\pi)=E_{s_t\sim D}[D_{KL}(\pi(\cdot|s_t)\|Q(s_t,\cdot))] \tag{5}$$

其中, $Q(s_t,\cdot)$ 表示基于能量的策略, $Q(s_t,\cdot)\propto\exp\left\{\frac{1}{\alpha}Q(s_t,\cdot)\right\}$. D 表示数据集分布, D_{KL} 表示 KL 散度. CQL 的通用目标函数如下:

$$\begin{aligned}&\min_Q\max_\mu\alpha(E_{s_t\sim\mathcal{D},a\sim\mu(a_t|s_t)}[Q(s_t,a_t)]-\\ &E_{s_t\sim\mathcal{D},a_t\sim\hat{\pi}_\beta(a_t|s_t)}[Q(s_t,a_t)])+\\ &\frac{1}{2}E_{s_t,a_t,s_{t+1}\sim\mathcal{D}}[(Q(s_t,a_t)-\mathcal{B}^{\pi_k}\hat{Q}^k(s_t,a_t))^2]+\mathcal{R}(\mu)\end{aligned} \tag{6}$$

其中, $\mu(a_t|s_t)$ 是用来近似可以最大化当前 Q 值函数迭代的策略, $\mathcal{B}^{\pi_k}$ 表示策略 π_k 对应的贝尔曼算子, $\hat{Q}^k(s_t,a_t)$ 表示目标 Q 值函数, $\hat{\pi}_\beta(a_t|s_t)$ 表示某个近似 $\mu(a_t|s_t)$ 的策略, $\mathcal{R}(\mu)$ 表示约束 $\mu(a_t|s_t)$ 和上一次迭代的策略分布的正则项. 如果选择 $\mathcal{R}(\mu)$ 为 $\mu(a_t|s_t)$ 和上一次迭代的策略的 KL 散度, 且假设上一次的策略分布为标准分布, 则 CQL 的目标可以定义为:

$$\begin{aligned}&\min_Q\alpha E_{s\sim\mathcal{D}}\left[\lg\sum_a\exp(Q(s,a))-E_{a\sim\hat{\pi}_\beta(a|s)}[Q(s,a)]\right]+\\ &\frac{1}{2}E_{a,s,s'\sim\mathcal{D}}\left[(Q-\mathcal{B}^{\pi_k}\hat{Q}^k)^2\right]\end{aligned} \tag{7}$$

从 CQL 的目标函数可以看出, 其是对数据集中的样本进行价值函数的乐观估计, 而对数据集之外的数据进行保守估计, 通过惩罚项为真正的 Q 值函数估计设定一个下界, 防止其在分布外数据上的过高估计.

2.3 优先经验回放机制

经验回放是离策略强化学习中的重要机制. 智能体通过从存放有历史数据的经验回放池中标准采样, 既打破了数据间的时序依赖关系, 使得输入神经网络的数据符合独立同分布的特性, 又能利用历史数据更新当前网络参数, 提升数据利用率. 然而, 经验数据的价值不一, 对所有经验数据采取等价采样概率的标准采样, 忽略了数据本身的价值特性. 为此, PER 提出使用时间差分误差(TD-error)作为数据元组的优先度度量指标, 并依据优先度进行采样, 进而更新神经网络参数. 时间差分误差的定义如下:

$$\delta=Q_\phi(s_t,a_t)-[r(s_t,a_t)+\gamma Q_{\phi'}(s_{t+1},a_{t+1})] \tag{8}$$

其中, ϕ 和 ϕ' 表示 Q 网络参数. 在实际应用中, 通常使用 $|\delta|$ 作为优先度度量, 并通过如下公式计算最终的采样概率:

$$p(i)=\frac{p_i^\alpha}{\sum_{j=0}^{N}p_j^\alpha},\ p_i=|\delta(i)|+\epsilon \tag{9}$$

其中 N 表示数据量, i 表示数据的序号, $\epsilon>0$ 表示一个小数值的常量, 用以保证即使在 $|\delta(i)|$ 为 0 的情况下, 其对应的数据元组的采样概率 p_i 始终为正. 但优先经验回放会因采样偏好而引起样本偏差, 通常把重要性采样作为优先采样数据的梯度更新系

数来修正该偏差:

$$w_i = \left(\frac{1}{N}\cdot\frac{1}{P(i)}\right)^{\beta} \tag{10}$$

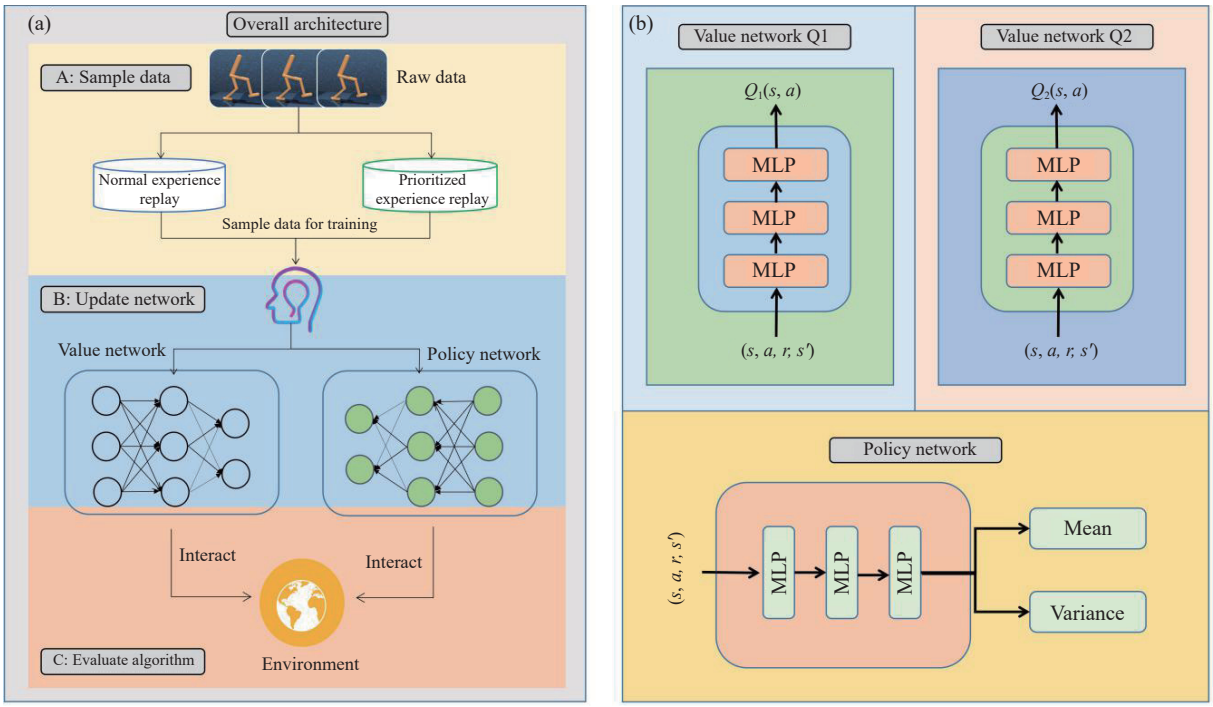
其中$\beta\in[0,1]$, 当$\beta=0$时, $w_i=1$, 表示重要度为 1 , 即不对采样的数据进行修正, 完全按照优先采样的数据更新网络梯度; 当$\beta=1$时, $w_i=\left(\frac{1}{N}\cdot\frac{1}{P(i)}\right)$, 即使用$w_i$对优先采样的数据进行修正. 实际应用中, 本文设置β的初始值为 1 , 使用线性退化机制, 随着训练进行逐渐降低β的值直至为 0 . 即在训练初始阶段, 加大对数据的修正值, 随着训练进行, 网络参数更新趋于稳定, 减小修正直至不再进行修正.

3　基于时间差分误差的离线强化学习采样方法

3.1　高效样本采样方法

本文提出一种新的离线强化学习智能体采样策略, 在训练初期采用优先采样策略对高价值数据进行优先偏好采样, 之后采用标准采样策略对数据进行无偏采样. 优先采样策略的执行步数区间, 按照不同任务特点, 以超参数的形式设定. 根据实验经验, 一般在整个训练过程的前1/5至1/3步数之前使用优先经验回放, 之后再使用标准采样策略. 为此, 本文为智能体分别建立 2 个经验回放池: 优先经验回放池和标准经验回放池. 为保证离线数据集完整性, 两个经验回放池的大小相同, 初始化数据一致. 优先经验回放池中的数据初始优先度均设置为 1, 以保证采样时对新数据的优先探索. 与以往方法类似, 本文采用时间差分误差作为数据元组的优先度度量, 同时采用重要性采样来纠正因优先采样引起的训练偏差. 除在训练不同阶段使用不同的采样策略外, 算法其余部分包括超参数设置均保持一致. 本文使用 CQL 作为基础离线强化学习算法. 本文的算法整体结构如图 1 所示.

如图 1 所示, 图 1(a)中包括了本文提出的方法的 3 个步骤: 数据采样、网络更新和算法评估, 其中数据采样部分包括 2 个经验回放池, 智能体从 2 个经验回放池中采集训练样本; 网络更新部分是指利用采样的数据进行算法的离线策略学习; 算法评估部分是指对学习的智能体在实际环境中进行性能评估. (b)本文的算法主要涉及策略网络和价值网络的训练. 本文采用双价值网络架构, 鉴于本文实验环境中的任务都是连续动作空间, 本文假设动作服从高斯分布, 策略网络的输出为动作分布的均值和方差, 最终的动作通过在策略网络的分布中采样得到.



Notes: MLP represents the multiple layer perception

图 1　本文方法的框架和网络的具体架构. (a) 本文方法的框架; (b) 网络的具体架构

Fig.1　Framework of the method and the specific architecture of the network in this paper: (a) framework of the method in this paper; (b) the specific architecture of the network

3.2 多种类时间差分误差度量

正如本文在2.2章节中的介绍, 强化学习算法通常使用时间差分误差来衡量当前策略和价值函数, 并据此对策略和价值函数进行更新. PER等使用时间差分误差作为样本优先度度量均取得较好性能, 结合计算复杂度考量, 本文的方法采用1步时间差分误差作为优先经验回放的优先度度量. 根据时间差分误差计算公式(8)可知, 时间差分误差取决于目标Q值函数$Q_{\phi'}(s_{t+1},a_{t+1})$. 本文的双Q价值网络采用3种计算目标Q值的方法:

最小化的双Q值网络: 使用双目标Q值网络的最小值作为Q值网络的目标值.

$$Q_{\text{target}_{\min}}(s_t,a_t)=r+\gamma\max_{a_{t+1}}\left[\min_{j=1,2}Q_{\varphi_j'}(s_{t+1},a_{t+1})\right] \tag{11}$$

最大化的双Q值网络: 使用双Q值目标网络的最大值作为Q值网络的目标值.

$$Q_{\text{target}_{\max}}(s_t,a_t)=r+\gamma\max_{a_{t+1}}\left[\min_{j=1,2}Q_{\varphi_j'}(s_{t+1},a_{t+1})\right] \tag{12}$$

凸组合的双Q值网络: 使用两个目标Q值网络的凸组合作为Q值网络的目标值.

$$Q_{\text{target}_{\text{comb}}}(s_t,a_t)=r+\gamma\max_{a_{t+1}}\left[\lambda\min_{j=1,2}Q_{\varphi_j'}(s_{t+1},a_{t+1})+(1-\lambda)\max_{j=1,2}Q_{\varphi_j'}(s_{t+1},a_{t+1})\right] \tag{13}$$

其中$\lambda\in(0,1)$是温度系数, 表示凸组合中最小化的双Q值的权重, j表示网络的序号. 可以看出, 最小化和最大化的双Q值是凸组合的双Q价值网络的特殊形式, 分别代表$\lambda=1$和$\lambda=0$时的凸组合的双Q值. 当目标Q值确定后, 本文方法中的时间差分误差可表示为:

$$\delta_{(s_t,a_t)}=Q(s_t,a_t)-Q_{\text{target}}(s_t,a_t) \tag{14}$$

其中, $\delta_{(s_t,a_t)}$表示数据(s_t,a_t)的时间差分误差, $Q_{\text{target}}(s_t,a_t)$表示$Q_{\text{target}_{\min}}(s_t,a_t)$, $Q_{\text{target}_{\max}}(s_t,a_t)$和$Q_{\text{target}_{\text{comb}}}(s_t,a_t)$共3种计算目标Q值方式之一. 与公式(9)类似, 基于时间差分误差的样本优先度如下:

$$p_{(s_t,a_t)}=\left|\delta_{(s_t,a_t)}\right|+\epsilon \tag{15}$$

其中, $p_{(s_t,a_t)}$表示数据(s_t,a_t)的样本优先度, $\epsilon>0$保证每个数据元组都有被采样的可能性. 本文的方法也使用了重要性采样, 但与PER不同的是, 本文只在计算样本优先度$p_{(s_t,a_t)}$时使用了重要性采样, 在梯度更新时, 则并未使用. 实验结果表明, 本文这种改动并未影响实验效果. 本文的方法中的策略网络与CQL中的策略网络更新方法一致:

$$\phi\leftarrow\phi+\eta E_{s_t\sim\mathcal{D},a\sim\pi_\theta(\cdot|s_t)}\left[Q_{\varphi(s_t,a_t)}-\lg\pi_\theta(a_t|s_t)\right] \tag{16}$$

其中, η为策略网络的更新步长, θ为策略π的参数. 本文的算法伪代码如下:

算法1: 基于时间差分误差的离线强化学习采样方法(CQL版本)

初始化: 双Q值网络Q_{φ_1}, Q_{φ_2}, 双Q值目标网络$Q_{\varphi_1'}$, $Q_{\varphi_2'}$, 策略网络π_θ, Q值网络更新步长τ, 策略网络π_θ的参数更新步长η, 片段长度为H, 批处理大小为N, 初始优先度设置为1, 最大训练步数T, 优先采样的最大步数T_p, 标准经验回放池B, 优先经验回放池B_p, Q网络参数φ的梯度Δ, Q网络参数φ的更新步长ζ, 数据序号i, 目标Q值网络参数软更新系数τ.

对于训练步数$t<T$:

如果$t<T_p$(即从优先经验池中采样):

1. 据公式(15)计算优先采样率并从优先经验池中采样N个批处理数据

2. 算重要性采样权重: $w_i=(N\cdot P(i))^{-\beta}/\max_i w_i$

3. 据公式(11)、(12)或(13)估计Q目标值$Q_{\text{target}}(s_t,a_t)$

4. 算Q值网络梯度变化

$$\Delta\leftarrow\Delta+\delta_i\cdot\nabla_\phi\left[\left(E_{(s_t,a_t)\sim\mathcal{D}}\left[Q_\varphi(s_t,a_t)\right]-Q_{\text{target}}(s_t,a_t)\right)^2\right]$$

5. 新Q值网络: $\varphi\leftarrow\varphi+\zeta\cdot\Delta$

6. 更新目标Q值网络: $\varphi'\leftarrow\tau\varphi+(1-\tau)\varphi'$

7. 据公式(16)更新策略网络

#否则(即从标准经验池中采样):

8. 根据公式(11)、(12)或(13)估计Q目标值$Q_{\text{target}}(s_t,a_t)$

9. 计算Q值网络梯度变化

$$\Delta\leftarrow\Delta+\nabla_\varphi\left[\left(E_{(s_t,a_t)\sim\mathcal{D}}\left[Q_\varphi(s_t,a_t)\right]-Q_{\text{target}}(s_t,a_t)\right)^2\right]$$

10. 更新Q值网络: $\varphi\leftarrow\varphi+\zeta\cdot\Delta$

11. 软更新目标Q值网络: $\varphi'\leftarrow\tau\varphi+(1-\tau)\varphi'$

12. 根据公式(16)更新策略网络

4 基于时间差分误差的离线强化学习采样方法的实验验证

4.1 实验设置

本文在离线强化学习公测数据集D4RL上进行实验. D4RL是DeepMind公司开发的一套仿真模拟数据集, 包括机器人、迷宫、自动驾驶等多种任务下的数据. 每种任务分别使用随机策略、中等策略、中等回放、中等—专家策略生成4种数据集, 不同策略生成的数据质量不同, 这些生成数据的策略被称为行动策略(Behavior policy). 每个数据集有约1000000或2000000个(s,a,r,s',done)数据元组, 其中done为一个片段(episode)的结束标志. 本文的实验是在D4RL中的Hopper(图2(a))、Half-Cheetah(图2(b))和Walker2d(图2(c))3种任务的

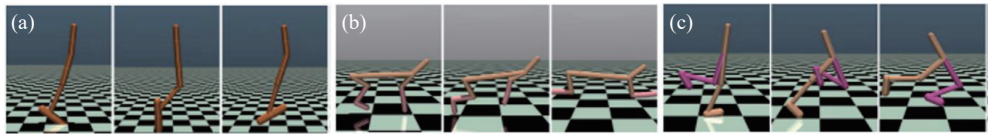


图 2　本文实验所使用的 DMControl 中的 3 个仿真环境. (a) Hopper; (b) HalfCheetah; (c) Walker2d

Fig.2　The three simulation environments in DMControl used for the experiments in this paper: (a) Hopper; (b) HalfCheetah; (c) Walker2d

数据集上进行训练, 在对应的 DMControl 实际环境中进行验证, 这 3 种任务分别对应不同维度的连续状态空间和动作空间. 本文在实验中使用 3 种任务相对应的随机策略、中等策略和中等—专家策略生成的 9 个数据集, 具体如表 1 所示.

表 1　实验所用的 D4RL 数据集

Table 1　D4RL dataset used in our experiment

Task	Datasets	Samples/10^4
Hopper	Hopper-random	1
	Hopper-medium	1
	Hopper-medium-expert	2
Halfcheetah	Halfcheetah-random	1
	Halfcheetah-medium	1
	Halfcheetah-medium-expert	2
Walker2d	Walker2d-random	1
	Walker2d-medium	1
	Walker2d-medium-expert	2

4.2　主要实验

为验证本文方法的性能, 本文利用上述提到的数据来训练智能体, 并和已有的结合采样机制的离线强化学习算法进行比较. 设置训练总步数为 1000000 步, 优先经验采样的步数为 200000, 之后的训练从标准经验回放中标准采样. 为公平比较, 本文在所有的实验中均使用相同的超参数. 算法评估是通过智能体与 DMControl 中相对应的环境交互实现的, 每 5000 步进行一次评估. 每个任务共评估 10 次, 每次评估执行 1000 步, 以 1000 步的累积奖励值作为当次评估的值, 以 10 次评估的平均值作为最终的评估值. 本文采用最小化双 Q 值作为本次实验的 3 种算法的目标 Q 值. 本次实验主要涉及 3 种算法: (a) CQL_PER 算法: 表示使用优先采样方式进行样本采样的 CQL 算法, 该方法为常用的结合采样策略的离线强化学习方法之一; (b) CQL_H 算法: 本文提出的方法, 表示使用标准采样和优先采样相结合的方式进行样本采集的 CQL 算法; (c) CQL_PER_N_return: 表示使用 N 步返回计算样本优先度并使用优先采样方式进行样本采集的 CQL 算法, 该方法为常用的结合采样策略的离线强化学习方法之一.

如图 3 所示, 主要实验部分是对 CQL_PER、CQL_H(本文的方法) 和 CQL_PER_N_return 这 3 种方法在 DMControl 仿真控制平台上的 Hopper(图 3(a))、HalfCheetah(图 3(b))和 Walker2d(图 3(c))这 3 种环境中进行了性能测试. 其中, 横轴表示不同质量的数据类型(Data types), 纵轴表示片段奖励值(Episode return), 柱状图中的红色误差棒表示算法性能的方差.

按照实验设定, 本文针对 3 种任务上的 random、medium 和 medium-expert 3 类数据, 分别使用 3 种离线强化学习方法进行了策略训练. 由图 3 可知, 在进行 1000000 步的网络梯度更新后, CQL_H 在 Hopper、HalfCheetah 中的 3 类数据上的策略的最终性能均优于 CQL_PER 和 CQL_PER_N_return. 在 Hopper 环境中, CQL_H 在 medium 数据集的最终性能超过 CQL_PER 约 40.6%, 超过 CQL_PER_N_return 约 4.9%; 在 medium-expert 数据集上的最终性能超过 CQL_PER 约 4.2%, 超过 CQL_PER_N_return 约 11.2%; 在 HalfCheetah 环境中, CQL_H 在 medium 数据集上的最终性能超过 CQL_PER 约 15.5%, 超过 CQL_PER_N_return 约 7.5%; 而在 Walker2d 中的 medium 数据集上, CQL_PER 的效果反而相对而言更优, 这可能是因为 Walker2d 的 medium 数据集更易于策略进行优先采样.

此外, 图 3 也展示了 3 种算法在不同环境不同类型数据集上的性能的方差, 可以看出, CQL_H 相比于其他 2 种方法, 在大部分环境中具有最小的误差, 表明使用 CQL_H 学习到的策略性能更为稳定.

4.3　消融实验

下面针对 CQL_H 开展消融实验, 主要分为 2 部分:

(1) CQL_H 使用标准采样和优先采样相组合的采样方式, 将使用该采样方式与单独使用标准采样方法和单独使用优先采样方法的性能进行比较. (2) 采用 3 种时间差分优先度时, CQL_H 对应 3 种不同的算法, 研究在相同的仿真环境下, 不同

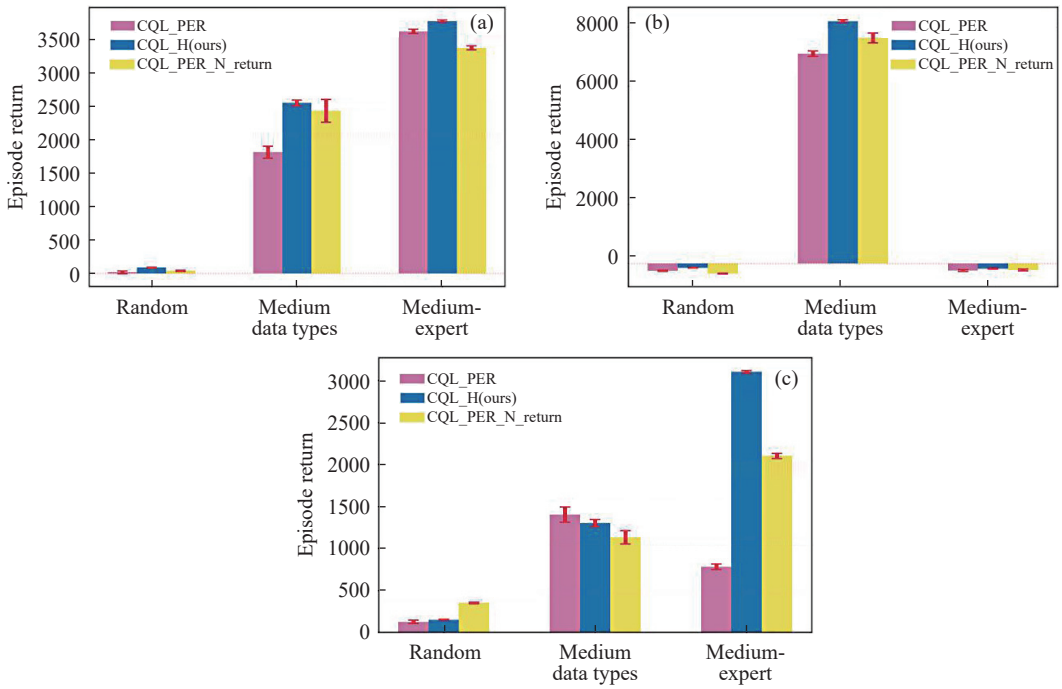


图 3 本文的方法 CQL_H 与 CQL_PER、CQL_PER_N_return 算法在 3 种环境的 3 类数据上的性能比较. (a) Hopper; (b) HalfCheetah; (c) Walker2d

Fig.3 The performance of the methods CQL_H and CQL_PER, CQL_PER_N_return algorithms in this paper is compared on three types of data for three environments: (a) Hopper; (b) HalfCheetah; (c) Walker2d

优先度的算法对算法性能的影响.

在消融实验(1)中, 本次实验使用 500000 步的训练次数, 即对网络参数更新 500000 次. 与主要实验的设置一致, 为缓解训练偏差, 本文每 1000 步进行一次评估. 每个任务共评估 10 次, 每次评估执行 1000 步, 以 1000 步的累积奖励值作为当次评估的值, 使用 10 次训练的均值和方差作为最终的均值和方差, 每次训练均使用随机种子. 本实验对 CQL_H 和 CQL_Uniform、CQL_Priority 进行性能对比, 这 3 种算法分别是: (a)CQL_H, 本文的方法, 表示使用标准采样策略和优先采样策略的 CQL_H 算法; (b)CQL_Uniform, 表示完全使用标准采样的策略的 CQL_H 算法; (c)CQL_Priority, 表示完全使用优先采样策略的 CQL_H 算法, 其中优先采样中的优先度度量使用最小化双 Q 值的时间差分误差.

如图 4 所示, 本文对 CQL_Uniform、CQL_H(本文的方法) 和 CQL_Priority 这 3 种方法在 Hopper (图 4(a)) 、HalfCheetah(图 4(b)) 和 Walker2d (图 4(c))这 3 种环境中进行了性能测试.

从图中可知, 在进行 500000 步的网络梯度更新后, CQL_H 在 Hopper、HalfCheetah 中的 3 类数据上的策略的最终性能都优于 CQL_Uniform 和 CQL_Priority. 在 Hopper 环境中, CQL_H 在 medium 数据集的最终性能超过 CQL_Uniform 约 10.4%, 超过 CQL_Priority 约 7.3%; 在 medium-expert 数据集上的最终性能超过 CQL_Uniform 约 11.8%, 超过 CQL_Priority 约 17.2%; 在 HalfCheetah 环境中, CQL_H 在 medium 数据集上的最终性能超过 CQL_Uniform 约 43.9%, 超过 CQL_Priority 约 14.7%. 而在 Walker2d 中的 medium 数据集上, CQL_Priority 的效果相对而言更优, 这与主要实验中的结果一致, 即 Walker2d 数据集更适合使用优先采样的采样方式训练策略.

从图 4 中的误差棒也可看出, CQL_H 相比于其他 2 种方法, 在大部分环境中具有最小的性能方差, 这也反映出使用标准采样和优先采样组合的采样方式的 CQL_H 算法相比于仅使用标准采样的 CQL_Uniform 算法和仅使用优先采样的 CQL_Priority 算法, 策略稳定性更强.

在消融实验(2)中, 本文采用 3 种基于双 Q 值网络目标值的时间差分误差, 分别利用最小化双 Q 网络值、最大化双 Q 网络值和凸组合的双 Q 网络值计算得到. 基于 3 种时间差分误差, 本文采用 3 种不同的优先采样策略, 从优先经验回放池中采集训练数据. 除了目标 Q 值计算方式不同, 3 种算法的网络结构和其余的超参数均设置一致. 鉴于计算资源有限, 本文只在 Hopper-medium, HalfCheetah-medium,Walker2d-medium 这 3 种任务上进行训练和评估. 本次训练和评估的设置和上文中的介

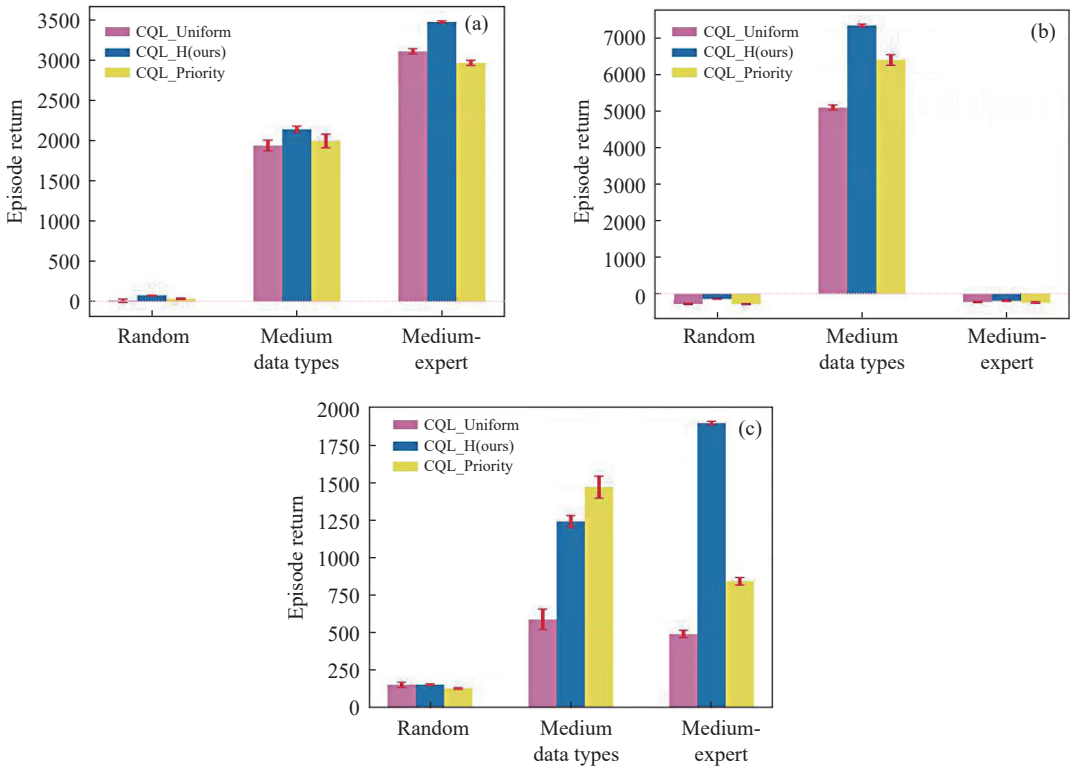


图 4　本文的方法 CQL_H 使用不同采样方式在 3 种环境的 3 类数据上的性能比较. (a) Hopper; (b) HalfCheetah; (c) Walker2d

Fig.4　The performance of the method CQL_H using different sampling methods in this paper is compared on 3 types of data for a total of 3 environments: (a) Hopper; (b) HalfCheetah; (c) Walker2d

绍相同. 和 BCQ 算法一致, 本文将凸组合的双 Q 网络值的计算公式 (13) 中的调节因子超参数 λ设置为 0.75. 3 种算法分别为: (a)CQL_H_min, 表示以双 Q 值估计的最小值计算时间差分误差, 以此作为 CQL_H 算法中优先经验采样策略部分的优先度度量, 是相对而言更为悲观的优先度度量方式, 也是本文中的 CQL_H 默认形式; (b)CQL_H_max, 表示以双 Q 值估计的最大值计算时间差分误差, 以此作为 CQL_H 算法中优先经验采样策略部分的优先度度量, 是相对而言更为乐观的优先度度量方式; (c)CQL_H_medium_0.75, 表示以双 Q 值估计的最大、最小值的凸优化组合计算时间差分误差, 以此作为 CQL_H 算法中优先经验采样策略部分的优先度度量, 是比较中庸的优先度度量方式. 其中 0.75 表示最小化双 Q 值的温度系数.

如图 5 所示, 本次实验是对使用不同方式计算的时间差分误差的 CQL_H 算法的性能进行验证, 以比较不同计算方式的时间差分对算法的性能有何不同影响. 图中的曲线表示评估值均值, 阴影部分表示评估值的方差. 可以看出, CQL_H_min 在 3 个环境中相比 CQL_H_max 和 CQL_H_medium_0.75 具有更好的数据利用率和最终性能. 在 Hopper-medium(图 5(a))环境中, CQL_H_min 在 250000 训练步即达到了 3000 左右, 但随即又下降, 直至最终和 CQL_H_max、CQL_H_medium_0.75 收敛至 2000. 在 HalfCheetah-medium(图 5(b))环境中, CQL_H_min 在 200000 训练步即达到了 7400, 在后续的训练中稳定在 5000 至 7500, 而 CQL_H_max、CQL_H_medium_0.75 则未能学习到有效策略, 网络一直未收敛. 在 Walker2d-medium(图 5(c))环境中, CQL_H_min 在 250000 训练步后逐步稳定, 最终收敛在 1500 附近. CQL_H_max 则收敛在 200 附近, 而 CQL_H_medium_0.75 则未能学习到有效策略, 策略网络未收敛. 从图中可以看出, 本文的 3 种方法在 Hopper-medium 上性能较好, 但 CQL_H_max 和 CQL_H_medium_0.75 在 HalfCheetah-medium, Walker2d-medium 上性能较差, CQL_H_min 在 3 种任务上均表现较好.

从图 5 中代表策略方差值的阴影部分可知, 尽管 CQL_H_min 在数据利用率和最终性能上具有较大优势, 但策略在部分训练环节的方差较大.

5　结论

本文围绕离线强化学习数据利用率低、易加剧外推误差的问题, 提出一种新的结合优先采样策略和标准采样策略的离线强化学习采样策略,

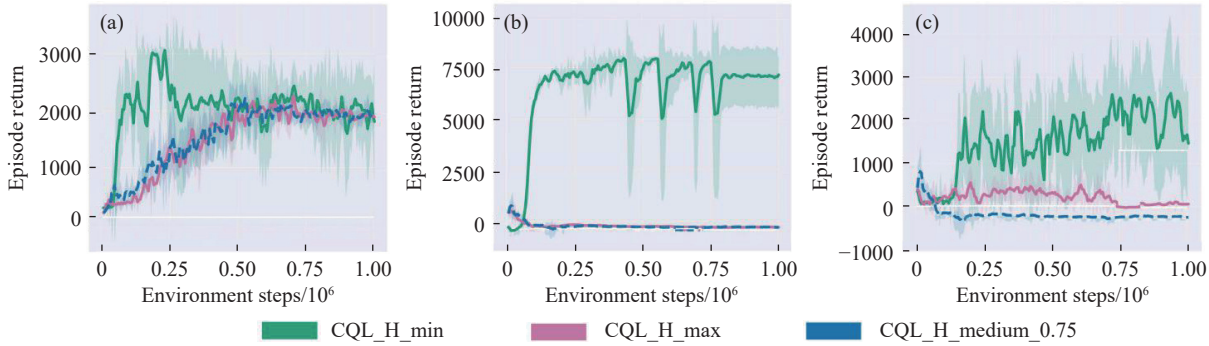


图 5 本文的方法 CQL_H 使用 3 种时间差分误差的优先采样策略的性能比较. (a) Hopper-medium; (b) HalfCheetah-medium; (c) Walker2d-medium

Fig.5 Comparison of CQL_H with three different TD-error: (a) Hopper-medium; (b) HalfCheetah-medium; (c) Walker2d-medium

通过设立 2 个经验回放池, 在训练初期从优先经验回放池中优先采样, 之后从标准经验回放池中标准采样, 并使用双 Q 价值网络计算的时间差分误差作为优先度度量, 可与任何基于 Q 值估计的离线强化学习算法相结合. 在 D4RL 上的实验表明, 本文的方法相比于已有的结合采样机制的基于 CQL 的离线强化学习算法, 具有更优的样本利用率和最终性能, 较好地缓解了外推误差问题, 且训练过程更稳定. 消融实验表明, 优先采样和标准采样相结合的采样方式对算法至关重要, 此外, 与最大化双 Q 值和凸组合的双 Q 值相比较, 使用最小化双 Q 值目标估计的时间差分误差更适合作为优先采样度量. 本文的方法是从采样策略的角度研究离线强化学习算法, 与之前的研究相比, 无需调整算法的基本架构和网络的基本结构, 最大限度避免增加网络参数, 实现简单, 可扩展性强. 本文的研究旨在提升数据利用率, 克服离线强化学习外推误差问题, 为解决数据采集难、样本数据有限、质量不一的控制决策任务, 提供了新的研究思路和方法支撑, 有助于相关研究者探索高效、安全、鲁棒的离线强化学习算法, 进一步推动强化学习在复杂实际场景中的落地应用.

参 考 文 献

[1] Vinyals O, Babuschkin I, Czarnecki W M, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 2019, 575(7782): 350

[2] Kiran B R, Sobh I, Talpaert V, et al. Deep reinforcement learning for autonomous driving: A survey. *IEEE Trans Intell Transp Syst*, 2022, 23(6): 4909

[3] Degrave J, Felici F, Buchli J. et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 2022, 602(7897): 414

[4] Fawzi A, Balog M, Huang A, et al. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 2022, 610(7930): 47

[5] Liang X X, Feng Y H, Huang J C, et al. Novel deep reinforcement learning algorithm based on attention-based value function and autoregressive environment model. *J Softw*, 2020, 31(4): 948 (梁星星, 冯旸赫, 黄金才, 等. 基于自回归预测模型的深度注意力强化学习方法. 软件学报, 2020, 31(4): 948)

[6] Mnih V, Badia A P, Mirza M, et al. Asynchronous methods for deep reinforcement learning//*International Conference on Machine Learning*. New York, 2016: 1928

[7] Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor//*International Conference on Machine Learning*. Stockholm, 2018: 1861

[8] Fujimoto S, Hoof H, Meger D. Addressing function approximation error in actor-critic methods // *International Conference on Machine Learning*. Stockholm, 2018: 1587

[9] Hafner D, Lillicrap T, Fischer I, et al. Learning latent dynamics for planning from pixels // *International Conference on Machine Learning*. California, 2019: 2555

[10] Hafner D, Lillicrap T, Ba J, et al. Dream to control: Learning behaviors by latent imagination[J/OL]. *arXiv preprint* (2020-05-17) [2022-10-22].https://arxiv.org/abs/1912.01603

[11] Hafner D, Lillicrap T, Norouzi M, et al. Mastering atari with discrete world models[J/OL]. *arXiv preprint* (2022-02-12) [2022-10-22].https://arxiv.org/abs/2010.02193

[12] Fujimoto S, Meger D, Precup D. Off-policy deep reinforcement learning without exploration // *International Conference on Machine Learning*. California, 2019: 2052

[13] Zhang L F, Zhang Y L, Liu S X, et al. ORAD: A new framework of offline Reinforcement Learning with Q-value regularization. *Evol Intel*, 2022: 1

[14] Mao Y H, Wang C, Wang B, et al. MOORe: Model-based offline-to-online reinforcement learning[J/OL]. *arXiv preprint* (2022-01-25) [2022-10-22]. https://arvix.org/abs/2201.10070

[15] Fujimoto S, Gu S S. A minimalist approach to offline reinforcement learning. *Adv Neural Inf Process Syst*, 2021, 34:

20132

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

[17] Fu J, Kumar A, Nachum O, et al. D4rl: Datasets for deep data-driven reinforcement learning[J/OL]. *arXiv preprint* (2021-02-06) [2022-10-22]. https://arxiv.org/abs/2004.07219

[18] Schaul T, Quan J, Antonoglou I, et al. Prioritized experience replay[J/OL]. *arXiv preprint* (2016-02-25) [2022-10-22]. https://arxiv.org/abs/1511.05952

[19] Liu H, Trott A, Socher R, et al. Competitive experience replay[J/OL]. *arXiv preprint* (2019-02-17) [2022-10-22]. https://arxiv.org/abs/1902.00528

[20] Fu Y W, Wu D, Boulet B. Benchmarking sample selection strategies for batch reinforcement learning[J/OL]. *OpenReview. net* (2022-01-29) [2022-10-22]. https://openreview.net/forum?id=WxBFVNbDUT6

[21] Lee S, Seo Y, Lee K, et al. Offline-to-online reinforcement learning *via* balanced replay and pessimistic q-ensemble // *Conference on Robot Learning*. London, 2022: 1702

[22] Bellman R. A Markovian decision process. *J Math Mech*, 1957: 679

[23] Hessel M, Modayil J, Van Hasselt H, et al. Rainbow: Combining improvements in deep reinforcement learning // *The Thirty-Second AAAI Conference on Artificial Intelligence*. New Orleans, 2018: 3215

[24] Schulman J, Levine S, Abbeel P, et al. Trust region policy optimization // *International Conference on Machine Learning*. Lille, 2015: 1889

[25] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[J/OL]. *arXiv preprint* (2017-08-28) [2022-10-22]. https://arxiv.org/abs/1707.06347

[26] Sutton R S, Barto A G. *Reinforcement Learning: An Introduction*. Cambridge: MIT press. 2018

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