不动点视角下的强化学习算法综述

陈兴国[1),2)] 孙丁源昊[1)] 杨 光[2),3)] 杨尚东[1),2)] 高 阳[2),3)]

[1)](南京邮电大学大数据安全与智能处理重点实验室 南京 210023)

[2)](南京大学计算机软件新技术国家重点实验室 南京 210046)

[3)](南京大学深圳研究院 广东 深圳 518057)

摘 要 近年来,强化学习已成为求解序贯决策任务的范式.然而,在实际应用中,强化学习算法仍存在三个问题:(1)什么解最优?(2)如何保证算法的稳定性?(3)如何加速算法的收敛?本文从不动点视角总结了强化学习算法的设计原理.首先,分析了值函数估计最优解与可行解的构造问题;其次,根据Banach不动点定理和Lyapunov第二判定定理,总结了已有基于值函数强化学习算法的稳定性问题,包括基于表格、线性估计、非线性估计、非参估计等值函数的算法在同策略和异策略情况下的收敛性;然后,从不动点的偏差与方差控制角度,解读了多种提高算法准确性或收敛速度的改进思想;最后总结和展望了强化学习算法的改进方向.

关键词 强化学习;值函数估计;稳定性;同策略;异策略;偏差与方差控制

中图法分类号 TP18 **DOI号** 10.11897/SP.J.1016.2023.01246

A Survey of Reinforcement Learning Algorithms from a Fixed Point Perspective

CHEN Xing-Guo[1),2)] SUN Dingyuanhao[1)] YANG Guang[2),3)] YANG Shang-Dong[1),2)] GAO Yang[2),3)]

[1)](*Jiangsu Key Laboratory of Big Data Security & Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing* 210023)

[2)](*National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing* 210046)

[3)](*Shenzhen Research Institute of Nanjing University, Shenzhen, Guangdong* 518057)

Abstract Reinforcement Learning has been developed for nearly 40 years since it was proposed. In recent years, with the breakthrough of deep learning, reinforcement learning has achieved many achievements, such as AlphaGo, AlphaZero, DouZero, and so on. Reinforcement learning has become one of the most promising paths to strong artificial intelligence. More and more researchers are trying to apply reinforcement learning to solve sequential decision-making tasks in their specific fields. However, practice studies show that applying classical reinforcement learning algorithm does not directly meet the practical needs. There is still a great challenge for researchers and engineers to design efficient reinforcement learning algorithms for real world decision problems. There are still three problems for reinforcement learning applications: (1) What is the optimal solution? (2) How to ensure the stability of the algorithm? (3) How to speed up the convergence of the algorithm? In recent years, reinforcement learning has grown rapidly with the rise of deep learning, and a dizzying array of algorithms, techniques and tools has emerged. There is an urgent need for researchers to view the latest reinforcement learning techniques from a unified perspective. From the unique perspective of the fixed point, reinforcement learning algorithm design includes value function-based reinforcement learning and policy gradient-based reinforcement learning. Since

收稿日期:2022-02-09;在线发布日期:2023-01-11.本课题得到国家自然科学基金(62276142,62206133,62202240,62192783)、科技创新2030-"新一代人工智能"重大项目(2018AAA0100905)、江苏省产业前瞻与关键核心技术竞争项目(BE2021028)、深圳市中央引导地方科技发展资金(2021Szvup056)资助.**陈兴国**,博士,讲师,硕士生导师,主要研究领域为强化学习、智能博弈、机器学习.E-mail:chenxg@njupt.edu.cn.**孙丁源昊**,硕士,主要研究方向为强化学习、智能博弈.**杨 光**,硕士,主要研究方向为强化学习、智能博弈.**杨尚东**,博士,主要研究方向为强化学习、智能博弈、机器学习.**高 阳**(通信作者),博士,教授,主要研究领域为机器学习、强化学习.E-mail:gaoy@nju.edu.cn.

there is relatively little research on fixed points for policy gradient-based reinforcement learning, this paper focuses mainly on the fundamentals of value function-based reinforcement learning algorithm design: (1) the optimal solution problem and feasible solution construction for value function estimation. (2) The stability problem of the algorithm, i. e., whether convergence is guaranteed. (3) How quickly the algorithm converges. To this end, this paper summarizes the design principles of reinforcement learning algorithms from a fixed point perspective. First of all, this paper introduces the reinforcement learning model, analyzes the optimal solution of the value function reinforcement learning algorithm and the feasible solution of the value estimation reinforcement learning algorithm, and summarizes and compares the key conditions for the convergence of reinforcement learning algorithms under different value function types. Then, the stability of the reinforcement learning algorithm based on the table value function and various improved algorithms are summarized. After that, the stability principles of reinforcement learning algorithms with linear value function estimation is proposed, and the differences between same-strategy reinforcement learning and different-strategy reinforcement learning in terms of design ideas and convergence key conditions under linear value estimation are summarized. Later, the stability principles of nonlinear value estimation and various improved algorithms are summarized. Furthermore, the research progress of nonparametric value estimation algorithms is presented. Moreover, various improvement ideas to improve the accuracy or speedup convergence are interpreted from the perspective of the bias and variance control of the fixed point. Finally, future improvement directions on reinforcement learning are prospected.

Keywords　reinforcement learning; value function approximation; stability; on-policy; off-policy; bias and variance control

1 引　言

强化学习(Reinforcement Learning)自提出以来至今已有近 40 年的发展历史. 近年来,随着深度学习的突破,强化学习取得了诸多成果,如围棋 AI AlphaGo、AlphaZero、斗地主 AI DouZero 等. 强化学习成为通向强人工智能最有希望的途径之一. 越来越多的研究者开始使用强化学习,试图求解各自领域的序贯决策任务. 然而实践表明,套用经典的强化学习算法并不能直接满足实际需求. 如何针对实际决策问题设计高效的强化学习算法仍是研究人员、工程师们面临的巨大挑战.

国内外众多学者对强化学习做了各种类型的综述,如早期的强化学习综述[1-4];从不同视角总结强化学习的算法:如面向迁移学习的强化学习综述[5-6]、面向安全的强化学习算法综述[7]、采用分层结构的强化学习综述[8-9]、采用深度神经网络的强化学习综述[10-16]、针对稀疏奖励的强化学习综述[17]、逆强化学习综述[18-19]、多智能体强化学习综述[9,20-22]、可解释强化学习综述[23-24]、深度强化学习的攻防与安全性分析综述[25];面向应用的强化学习综述,如综合能源系统管理[26]、组合优化[27-28]、自动驾驶[29]、自然语言处理[30]、机器翻译[31]、智慧医疗[32]、智能运输[33-34]、网络攻防[35]等. 这些工作主要从特定场景或应用角度对强化学习进行归类或总结.

最近几年,随着深度学习的兴起,强化学习得到了飞速发展,各种算法层出不穷,技巧和手段日新月异,令人眼花缭乱. 如何从一个统一的视角看待最新的强化学习技术成为研究者的迫切需求. 从不动点这个独特的视角,强化学习算法设计包括基于值函数的强化学习和基于策略梯度的强化学习. 由于策略梯度强化学习的不动点研究相对较少(见总结与展望),本文主要关注基于值函数强化学习算法设计的基本原理:

(1)值函数估计的最优解问题与可行解构造;

(2)算法的稳定性问题,即能否保证收敛;

(3)算法如何快速收敛.

如图 1 所示,本文第 2 节介绍强化学习模型,分析值函数强化学习算法的最优解和值估计强化学习

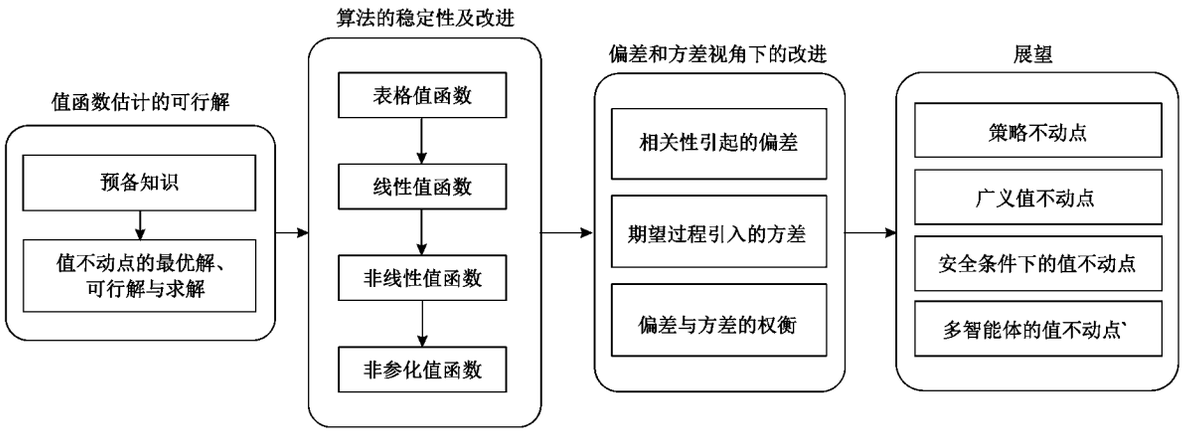

图 1 本文的整体架构

算法的可行解，并针对不同值函数类型下强化学习算法收敛的关键条件做总结和对比；第 3 节分析基于表格值函数强化学习算法的稳定性，并总结各种改进算法；第 4 节分析线性值函数估计的强化学习算法的稳定性原理，总结线性值估计下同策略强化学习和异策略强化学习在设计思想和收敛性关键条件上的差异；第 5 节分析和总结非线性值估计的稳定性原理和各种改进算法；第 6 节总结非参值估计算法的研究进展；第 7 节从算法收敛到不动点的偏差和方差角度，对强化学习算法的一些典型改进做分析和总结；最后对未来工作进行展望.

2 值函数估计的可行解

2.1 马氏决策过程与 Bellman 等式

强化学习智能体(Agent)感知环境(Environment)的状态 s 并采取动作 a，环境转移到后继状态 s' 并反馈给智能体奖赏 r. 智能体与环境的这种交互过程持续不断或达到终止状态结束. 这类序贯决策任务通常满足马氏属性，由马氏决策过程(Markov Decision Process，MDP)建模[36-37]. MDP 由四元组 $\langle \boldsymbol{S},\boldsymbol{A},\boldsymbol{R},\boldsymbol{T}\rangle$ 构成，其中 $\boldsymbol{S}$ 表示状态空间，$\boldsymbol{A}$ 表示可执行的动作空间，$\boldsymbol{R}:\boldsymbol{S}\times\boldsymbol{A}\times\boldsymbol{S}\to\mathbb{R}$ 为有界奖赏函数，$\boldsymbol{T}:\boldsymbol{S}\times\boldsymbol{A}\times\boldsymbol{S}\to[0,1]$ 为状态转移概率函数，满足 $\forall s,a,\sum_{s'\in\boldsymbol{S}}\boldsymbol{T}(s,a,s')=1$. 与基于局部策略的规划算法不同，强化学习设计了全局策略 $\boldsymbol{\pi}:\boldsymbol{S}\times\boldsymbol{A}\to[0,1]$，其目标是最大化长期的带折扣奖赏 $Return=\mathbb{E}_{\boldsymbol{\pi}}[\sum_{t=0}^{\infty}\gamma^t r_t]$，其中 $\gamma\in(0,1]$ 为折扣率，$r_t=\boldsymbol{R}(s_t,a_t,s_{t+1})$ 为 t 时刻的奖赏，$a_t\sim\boldsymbol{\pi}(s_t,a_t)$.

最大化长期奖赏等价于最大化值函数，如状态值函数：

$$\boldsymbol{V}^{\boldsymbol{\pi}}(s)\doteq\mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{t=0}^{\infty}\gamma^t r_t\,|\,s_0=s\right] \tag{1}$$

和状态动作值函数：

$$\boldsymbol{Q}^{\boldsymbol{\pi}}(s,a)\doteq\left[\sum_{t=0}^{\infty}\gamma^t r_t\,|\,s_0=s,a_0=a\right] \tag{2}$$

以状态值函数为例，根据定义，可得 Bellman 等式：

$$\begin{aligned}\boldsymbol{V}^{\boldsymbol{\pi}}(s)&=\mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{t=0}^{\infty}\gamma^t r_t\,|\,s_0=s\right]\\&=\mathbb{E}_{\boldsymbol{\pi}}\left[\boldsymbol{R}(s,a,s')+\gamma\sum_{t=0}^{\infty}\gamma^t r_t\,|\,s_0=s'\right]\\&=\mathbb{E}_{\boldsymbol{\pi}}[\boldsymbol{R}(s,a,s')+\gamma\boldsymbol{V}^{\boldsymbol{\pi}}(s')]\end{aligned} \tag{3}$$

拓展至所有状态，令 $\boldsymbol{R}^{\boldsymbol{\pi}}(s)$ 为奖赏列向量，其元素为

$$\begin{aligned}\boldsymbol{R}^{\boldsymbol{\pi}}(s)&=\mathbb{E}_{\boldsymbol{\pi}}[\boldsymbol{R}(s,a,s')]\\&=\sum_a\boldsymbol{\pi}(s,a)\sum_{s'}\boldsymbol{T}(s,a,s')\boldsymbol{R}(s,a,s')\end{aligned} \tag{4}$$

$\boldsymbol{P}^{\boldsymbol{\pi}}$ 为状态转移概率矩阵，其元素为

$$\boldsymbol{P}^{\boldsymbol{\pi}}(s,s')=\sum_a\boldsymbol{\pi}(s,a)\boldsymbol{T}(s,a,s') \tag{5}$$

所有状态值函数构成列向量 $\boldsymbol{V}^{\boldsymbol{\pi}}$，其元素为

$$\boldsymbol{V}^{\boldsymbol{\pi}}(s)=\boldsymbol{R}^{\boldsymbol{\pi}}(s)+\gamma\sum_{s'}\boldsymbol{P}^{\boldsymbol{\pi}}(s,s')\boldsymbol{V}(s') \tag{6}$$

至此，所有状态的 Bellman 等式由向量和矩阵构成：

$$\begin{aligned}\boldsymbol{V}^{\boldsymbol{\pi}}&=\boldsymbol{R}^{\boldsymbol{\pi}}+\gamma\boldsymbol{P}^{\boldsymbol{\pi}}\boldsymbol{V}^{\boldsymbol{\pi}}\\&\doteq\mathcal{T}^{\boldsymbol{\pi}}\boldsymbol{V}^{\boldsymbol{\pi}}\end{aligned} \tag{7}$$

其中，$\mathcal{T}^{\boldsymbol{\pi}}:\mathbb{R}^{|\boldsymbol{S}|}\to\mathbb{R}^{|\boldsymbol{S}|}$ 为 Bellman 评估算子. 由式(7)可知

$$\boldsymbol{V}^{\boldsymbol{\pi}}=(\boldsymbol{I}-\gamma\boldsymbol{P}^{\boldsymbol{\pi}})^{-1}\boldsymbol{R}^{\boldsymbol{\pi}} \tag{8}$$

最优策略 $\boldsymbol{\pi}^*$ 满足 $\forall s\in\boldsymbol{S},\boldsymbol{V}^*(s)=\max_{\boldsymbol{\pi}}\boldsymbol{V}^{\boldsymbol{\pi}}(s)$，其中 $\boldsymbol{V}^*$ 为最优值函数，满足 Bellman 最优等式：

$$\begin{aligned}\boldsymbol{V}^*(s)&=\max_a\mathbb{E}\left[\boldsymbol{R}(s,a,s')+\gamma\boldsymbol{V}^*(s')\right]\\&\doteq\mathcal{T}\boldsymbol{V}^*(s)\end{aligned} \tag{9}$$

其中，$\mathcal{T}:\mathbb{R}^{|S|}\to\mathbb{R}^{|S|}$ 为 Bellman 最优算子.

2.2　线性值函数的最优解

当状态空间 $|\boldsymbol{S}|=n$ 较小时，强化学习采用表格值函数，为每个状态分配一个值. 式(7)是一个 n 元一次方程组，共有 n 个方程. 那么，该方程组有唯一解. 当状态空间大到计算机无法存储时，只能采用远小于状态空间($m\ll n$)的参数对值函数进行估计，如线性函数、非线性函数和非参函数. 由于线性函数有利于理论分析，以线性值函数 $\boldsymbol{V}_{\boldsymbol{\theta}}(s)$ 为例，

$$\boldsymbol{V}_{\boldsymbol{\theta}}(s)\doteq\sum_{i=0}^{m}\boldsymbol{\phi}_i(s)\boldsymbol{\theta}_i=\boldsymbol{\phi}(s)^{\top}\boldsymbol{\theta}\tag{10}$$

其中，$\boldsymbol{\theta}=(\boldsymbol{\theta}_0,\boldsymbol{\theta}_1,\cdots,\boldsymbol{\theta}_m)$ 为权重参数，$\boldsymbol{\phi}(s)=(\boldsymbol{\phi}_0(s),\boldsymbol{\phi}_1(s),\cdots,\boldsymbol{\phi}_m(s))^{\top}$ 为状态 s 的特征. 所有状态值函数向量为

$$\boldsymbol{V}_{\boldsymbol{\theta}}=\boldsymbol{\Phi}\boldsymbol{\theta}\tag{11}$$

$\boldsymbol{\Phi}$ 是所有状态的特征矩阵，其每一行为对应状态的特征向量转置 $\boldsymbol{\Phi}(s)=\boldsymbol{\phi}(s)^{\top}$. 代入式(7)，

$$\begin{aligned}\boldsymbol{V}_{\boldsymbol{\theta}}&=\boldsymbol{R}^{\pi}+\gamma\boldsymbol{P}^{\pi}\boldsymbol{V}_{\boldsymbol{\theta}}\\&=(\boldsymbol{I}-\gamma\boldsymbol{P}^{\pi})^{-1}\boldsymbol{R}^{\pi}\\&=\boldsymbol{V}^{\pi}\end{aligned}\tag{12}$$

式(12)是一个 m 元一次方程组，共有 n 个方程. 由于 $m\ll n$，该方程组是病态方程，无解，只能逼近. 采用最小二乘法进行逼近，

$$\begin{aligned}\boldsymbol{\theta}&=\arg\min_{\boldsymbol{\theta}}\|\boldsymbol{V}_{\boldsymbol{\theta}}-\boldsymbol{V}^{\pi}\|_{\boldsymbol{D}}\\&=\arg\min_{\boldsymbol{\theta}}(\boldsymbol{V}_{\boldsymbol{\theta}}-\boldsymbol{V}^{\pi})^{\top}\boldsymbol{D}(\boldsymbol{V}_{\boldsymbol{\theta}}-\boldsymbol{V}^{\pi})\end{aligned}\tag{13}$$

其中，状态分布矩阵 $\boldsymbol{D}$ 为对角矩阵，其对角线元素 $\boldsymbol{D}_{ss}$ 为状态 s 的概率 d_s. 对式(13)求导可得

$$\boldsymbol{\theta}=(\boldsymbol{\Phi}^{\top}\boldsymbol{D}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\top}\boldsymbol{D}\boldsymbol{V}^{\pi}\tag{14}$$

式(14)是线性值函数的最优解. 然而由于 $\boldsymbol{V}^{\pi}$ 未知，式(14)不可直接求解. 因此，当状态空间很大时，对值函数估计的可行解进行构造是关键.

2.3　值函数估计的可行解构造

事实上，采用值函数估计 $\boldsymbol{V}_{\boldsymbol{\theta}}$ 的问题是 Bellman 等式不再成立，即 $\boldsymbol{V}_{\boldsymbol{\theta}}\neq\mathcal{T}^{\pi}\boldsymbol{V}_{\boldsymbol{\theta}}=\boldsymbol{R}^{\pi}+\gamma\boldsymbol{P}^{\pi}\boldsymbol{V}_{\boldsymbol{\theta}}$. 其原因在于 $\boldsymbol{R}^{\pi}$ 是由奖赏函数和策略 $\boldsymbol{\pi}$ 共同决定的，不受特征函数 $\boldsymbol{\Phi}$ 约束[38]. 这就导致了值函数估计 $\boldsymbol{V}_{\boldsymbol{\theta}}$ 与 $\mathcal{T}^{\pi}\boldsymbol{V}_{\boldsymbol{\theta}}$ 的偏差.

可行解构造需要充分利用 $\boldsymbol{V}_{\boldsymbol{\theta}}$ 与 $\mathcal{T}^{\pi}\boldsymbol{V}_{\boldsymbol{\theta}}$. 带参投影算子 $\boldsymbol{\Pi}_{\boldsymbol{X}}=\boldsymbol{\Phi}(\boldsymbol{X}^{\top}\boldsymbol{\Phi})^{-1}\boldsymbol{X}^{\top}$ 将 $\mathcal{T}^{\pi}\boldsymbol{V}_{\boldsymbol{\theta}}$ 映射到特征空间，使得

$$\boldsymbol{V}_{\boldsymbol{\theta}}=\boldsymbol{\Pi}_{\boldsymbol{X}}\mathcal{T}^{\pi}\boldsymbol{V}_{\boldsymbol{\theta}}\tag{15}$$

其中，$\boldsymbol{X}$ 为投影方向. 因此，问题转化为寻找一个合适的投影方向 $\boldsymbol{X}$.

为了更直观地比较各种可行解，采用期望表达式替换不动点表达式. 令投影方向 $\boldsymbol{X}=\boldsymbol{D}\boldsymbol{\Phi}$，以正投影算子 $\boldsymbol{\Pi}=\boldsymbol{\Pi}_{\boldsymbol{D}\boldsymbol{\Phi}}=\boldsymbol{\Phi}(\boldsymbol{\Phi}^{\top}\boldsymbol{D}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\top}\boldsymbol{D}$ 为例.

$$\boldsymbol{V}_{\boldsymbol{\theta}}=\boldsymbol{\Pi}\mathcal{T}^{\pi}\boldsymbol{V}_{\boldsymbol{\theta}}\tag{16}$$

分别代入 $\boldsymbol{V}_{\boldsymbol{\theta}}$ 和 $\boldsymbol{\Pi}$，可得 $\boldsymbol{\theta}=(\boldsymbol{\Phi}^{\top}\boldsymbol{D}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\top}\boldsymbol{D}\mathcal{T}^{\pi}\boldsymbol{V}_{\boldsymbol{\theta}}$，化简后，$\boldsymbol{\Phi}^{\top}\boldsymbol{D}(\mathcal{T}^{\pi}\boldsymbol{V}_{\boldsymbol{\theta}}-\boldsymbol{V}_{\boldsymbol{\theta}})=0$. 采用期望形式，可得

$$\begin{aligned}0&=\boldsymbol{\Phi}^{\top}\boldsymbol{D}(\mathcal{T}^{\pi}\boldsymbol{V}_{\boldsymbol{\theta}}-\boldsymbol{V}_{\boldsymbol{\theta}})\\&=\sum_s d_s(\mathbb{E}_{\pi}[r+\gamma\boldsymbol{V}_{\boldsymbol{\theta}}(s')]-\boldsymbol{V}_{\boldsymbol{\theta}}(s))\boldsymbol{\phi}(s)\\&=\mathbb{E}_{\pi}[\delta\boldsymbol{\phi}]\end{aligned}\tag{17}$$

其中，$\delta=r+\gamma\boldsymbol{V}_{\boldsymbol{\theta}}(s')-\boldsymbol{V}_{\boldsymbol{\theta}}(s)$ 为时序误差(TD error). 因此，两种表达方式是等价的.

$$\boldsymbol{V}_{\boldsymbol{\theta}}=\boldsymbol{\Pi}\mathcal{T}^{\pi}\boldsymbol{V}_{\boldsymbol{\theta}}\Leftrightarrow\mathbb{E}_{\pi}[\delta\boldsymbol{\phi}]=0\tag{18}$$

表 1 总结了已有值函数可行解，分别从值函数类型、参数个数、投影方向、解的性质、可行解的不动点表达式和期望表达式方面，对比了各种可行解. 值函数不动点的求解用不动点式作迭代，可以分解为两个算子：优化 $\mathcal{T}$ 或评估 $\mathcal{T}^{\pi}$ 算子，以及投影算子. 其中，投影算子将解投影到可行区域，不同值函数需要不同的投影算子，具体如下：

表格值函数中参数 $\boldsymbol{\theta}$ 个数 m 与状态空间 n 相等，是 One-Hot 编码为特征 $\boldsymbol{\Phi}=\boldsymbol{I}$ 的线性求和函数. 投影方向为 $\boldsymbol{X}=\boldsymbol{D}\boldsymbol{\Phi}=\boldsymbol{D}\boldsymbol{I}=\boldsymbol{D}$，投影算子 $\boldsymbol{\Pi}_{\boldsymbol{D}}=\boldsymbol{\Phi}(\boldsymbol{X}^{\top}\boldsymbol{\Phi})^{-1}\boldsymbol{X}^{\top}=\boldsymbol{I}(\boldsymbol{D}\boldsymbol{I})^{-1}\boldsymbol{D}=\boldsymbol{I}$ 为单位向量，即不动点为方程 $\boldsymbol{V}_{\boldsymbol{\theta}}=\mathcal{T}^{\pi}\boldsymbol{V}_{\boldsymbol{\theta}}$ 的解.

线性值函数估计无法直接求最优解，只能构造可行解. 若采用正投影 $\boldsymbol{\Pi}$，则如上述所示，不动点是方程 $\boldsymbol{V}_{\boldsymbol{\theta}}=\mathcal{T}^{\pi}\boldsymbol{V}_{\boldsymbol{\theta}}$ 的解，其期望形式是 $\mathbb{E}_{\pi}[\boldsymbol{\delta}\boldsymbol{\phi}]=0$；若采用斜投影 $\boldsymbol{\Pi}_{\boldsymbol{D}(\boldsymbol{I}-\gamma\boldsymbol{P})\boldsymbol{\Phi}}$，则不动点是方程 $\boldsymbol{V}_{\boldsymbol{\theta}}=\boldsymbol{\Pi}_{\boldsymbol{D}(\boldsymbol{I}-\gamma\boldsymbol{P})\boldsymbol{\Phi}}\mathcal{T}^{\pi}\boldsymbol{V}_{\boldsymbol{\theta}}$ 的解，其期望形式是 $\mathbb{E}[\delta(\boldsymbol{\phi}-\gamma\boldsymbol{\phi}')]=0$.

表 1　值函数可行解的对比

投影方向 X	类型	参数个数	解的性质	不动点表达式	期望表达式
$\boldsymbol{D}$	表格值	$m=n$	最优	$\boldsymbol{V}_{\boldsymbol{\theta}}=\mathcal{T}^{\pi}\boldsymbol{V}_{\boldsymbol{\theta}}$	$\mathbb{E}[\delta^2]=0$
$\boldsymbol{D}\boldsymbol{\Phi}$	线性	$m\ll n$	非最优	$\boldsymbol{V}_{\boldsymbol{\theta}}=\boldsymbol{\Pi}\mathcal{T}^{\pi}\boldsymbol{V}_{\boldsymbol{\theta}}$	$\mathbb{E}[\delta\boldsymbol{\phi}]=0$
$\boldsymbol{D}(\boldsymbol{I}-\gamma\boldsymbol{P})\boldsymbol{\Phi}$	线性	$m\ll n$	非最优	$\boldsymbol{V}_{\boldsymbol{\theta}}=\boldsymbol{\Pi}_{\boldsymbol{D}(\boldsymbol{I}-\gamma\boldsymbol{P})\boldsymbol{\Phi}}\mathcal{T}^{\pi}\boldsymbol{V}_{\boldsymbol{\theta}}$	$\mathbb{E}[\boldsymbol{\delta}(\boldsymbol{\phi}-\gamma\boldsymbol{\phi}')]=0$
$\boldsymbol{D}\boldsymbol{\Phi}_{\boldsymbol{\theta}}$	非线性	$m\ll n$	非最优	$\boldsymbol{V}_{\boldsymbol{\theta}}=\boldsymbol{\Pi}_{\boldsymbol{D}\boldsymbol{\Phi}_{\boldsymbol{\theta}}}\mathcal{T}^{\pi}\boldsymbol{V}_{\boldsymbol{\theta}}$	$\mathbb{E}[\delta(\boldsymbol{\theta})\nabla\boldsymbol{V}_{\boldsymbol{\theta}}(s)]=0$

同理,非线性值函数估计更难表达最优解.此外,特征随参数变化而变化,投影算子也是带参的,$\boldsymbol{\Pi}_{D\boldsymbol{\Phi}_\theta}=\boldsymbol{\Phi}_\theta(\boldsymbol{\Phi}_\theta^{\mathrm{T}}\boldsymbol{D}\boldsymbol{\Phi}_\theta)^{-1}\boldsymbol{\Phi}_\theta^{\mathrm{T}}\boldsymbol{D}$,可以看成是局部特征空间的投影,不动点是方程 $\boldsymbol{V}_\theta=\boldsymbol{\Pi}_{D\boldsymbol{\Phi}_\theta}\mathcal{T}^\pi\boldsymbol{V}_\theta$ 的解,期望形式是 $\mathbb{E}[\delta(\boldsymbol{\theta})\nabla\boldsymbol{V}_\theta(s)]=0$.

综上所述,除了表格值函数,值函数估计的可行解构造并不唯一,哪种构造方式更逼近最优解是当前的开放问题.

2.4 值函数估计可行解的求解

假设给定可行解的形式,那么求解的高效算法需要满足两个要求:既快又稳.基于值函数强化学习的可行解是不动点(15).一般而言,函数 $\boldsymbol{f}(x)$ 的不动点 x^* 指的是方程 $x=\boldsymbol{f}(x)$ 的解.计算机算法通常采用迭代法来求解不动点,如给定初始值 x_0,

$$x_{n+1}=\boldsymbol{f}(x_n) \tag{19}$$

其中,$n\in\mathbb{N}$ 且 $n\geq 0$,$\boldsymbol{f}$ 是关于 x 的迭代函数.如果 $\lim\limits_{n\to\infty}x_n\to x^*$,则说明该迭代函数 $\boldsymbol{f}(x)$ 是稳定的.

表 2 分别从函数类型、迭代算法收敛性分析时的期望表达式、收敛性判定的理论依据和关键条件角度对比总结了值函数强化学习算法的稳定性.接下来,我们分别从表格值函数、线性值函数、非线性值函数以及非参化函数的角度,总结已有强化学习算法的稳定性,并从快速收敛的角度总结已有强化学习改进算法的思想.

表 2 不同值函数类型算法的稳定性

函数类型	迭代函数	关键条件	相关判定定理
表格值	$x_{n+1}=\mathcal{T}^\pi(x_n)$	压缩映射	巴拿赫不动点定理
线性	$\dot{\boldsymbol{\theta}}(t)=-\boldsymbol{A}\boldsymbol{\theta}(t)+\boldsymbol{b}$	正定矩阵	李雅普诺夫第二判定法推论 1
非线性	$\dot{\boldsymbol{\theta}}(t)=-\boldsymbol{g}'(\boldsymbol{\theta}(t))$	凸函数且二阶导数不等于 0	李雅普诺夫第二判定法推论 2

3 表格值强化学习

表格值强化学习算法为每个状态或者状态动作对分配一个值,适用于小规模强化学习问题.

3.1 表格值强化学习稳定性原理

巴拿赫不动点定理不仅给出了不动点的存在性判定,还构造了不动点的求解方法[39].

定理 1. 巴拿赫(Banach)不动点定理.设 $\boldsymbol{X}$ 是完备的度量空间,$\boldsymbol{f}:\boldsymbol{X}\to\boldsymbol{X}$ 是压缩映射,则存在唯一的不动点 $x=\boldsymbol{f}(x)$.该不动点可以通过迭代法不断逼近:

$$x_{n+1}=\boldsymbol{f}(x_n) \tag{20}$$

其中,压缩映射 $\boldsymbol{f}$ 满足以下条件:存在常数 $a\in[0,1)$,对于任何的 $x,y\in\boldsymbol{X}$,有 $d(\boldsymbol{f}(x),\boldsymbol{f}(y))\leq ad(x,y)$.其中,最小的常数 a 称为利普希茨(Lipschitz)常数.

根据理论分析,$\mathcal{T}^\pi$ 是一个压缩映射[40].根据巴拿赫不动点定理,存在唯一不动点 $\boldsymbol{V}^*=\mathcal{T}^\pi\boldsymbol{V}^*$,可以通过迭代法 $\boldsymbol{V}_{n+1}=\mathcal{T}^\pi\boldsymbol{V}_n$ 不断逼近不动点.该迭代方法就是动态规划中的策略评估算法的关键步骤.

$\mathcal{T}$ 也是压缩映射[40].存在唯一不动点 $\boldsymbol{V}^*=\mathcal{T}\boldsymbol{V}^*$,并可以通过迭代法 $\boldsymbol{V}_{n+1}=\mathcal{T}\boldsymbol{V}_n$ 不断逼近.该迭代方法就是值迭代算法的关键步骤.

综合多步 Bellman 评估算子 $\mathcal{T}^\pi$,可得满足压缩映射的算子 $\mathcal{T}^{\lambda,\pi}$:

$$\mathcal{T}^{\lambda,\pi}\boldsymbol{V}\doteq(1-\lambda)\sum_{i=0}^{\infty}\lambda^i(\mathcal{T}^\pi)^{i+1}\boldsymbol{V} \tag{21}$$

并由此得到 λ-策略迭代算法.

另外,算子 $\mathcal{T}$ 和 $\mathcal{T}^\pi$ 还满足单调性.假设 $\forall s\in\boldsymbol{S}$,$\boldsymbol{V}(s)\geq\boldsymbol{U}(s)$,则 $\mathcal{T}\boldsymbol{V}(s)\geq\mathcal{T}\boldsymbol{U}(s)$,$\mathcal{T}^\pi\boldsymbol{V}(s)\geq\mathcal{T}^\pi\boldsymbol{U}(s)$.因此,策略迭代算法中策略改进可以收敛到最优值函数和最优策略,值迭代算法也可以收敛到最优.

3.2 表格值强化学习

表 3 总结了基于表格值不动点的强化学习算法,主要包含两类:基于模型的强化学习和免模型强化学习.由于表格值函数都可以收敛到不动点及最优策略,下文主要从高效计算、快速收敛的角度对经典算法进行总结.

基于模型的强化学习算法分为两类:一类给定模型;另一类学习未知模型.基于给定模型的强化学习算法,策略迭代(Policy Iteration)及其改进算法不断重复两个过程:策略评估(Policy Evaluation)和策略改进(Policy Improvement),直至收敛到最优策略和最优值函数[40].原始的策略迭代算法在每次策略评估时要求值函数收敛到不动点再做策略改进.注意到,我们所要求的是最优策略和最优值函数,当前非最优策略的评估没有必要收敛到不动点.因此改进的策略迭代算法(Modified Policy Iteration)在策略评估阶段,只迭代 m_k 次,减小了运算量.进一步,如果所有的 $m_k=1$,则变成值迭代(Value Iteration)算法.多级搜索策略迭代(Multistage Lookahead Policy Iteration)在策略改进阶段,通过搜索以进一步逼近最优值函数从而得到更优的改进策略.当状态空间较大时,对所有状态做一次扫描的耗

表 3　基于表格值不动点的强化学习算法

分类	算法名称	不动点	迭代式的关键步骤
基于模型	策略迭代[40]	$\boldsymbol{V}=\mathcal{T}^{\pi}\boldsymbol{V}$	评估：$\boldsymbol{V}^{\pi_n}=\lim\limits_{k\to\infty}(\mathcal{T}^{\pi_n})^k\boldsymbol{V}_0$，改进：$\mathcal{T}^{\pi_{n+1}}\boldsymbol{V}^{\pi_n}=\mathcal{T}\boldsymbol{V}^{\pi_n}$
	多级搜索策略迭代[40]	$\boldsymbol{V}=\mathcal{T}^{\pi}\boldsymbol{V}$	评估：$\boldsymbol{V}^{\pi_n}=\lim\limits_{k\to\infty}(\mathcal{T}^{\pi_n})^k\boldsymbol{V}_0$，改进：$\mathcal{T}^{\pi_{n+1}}\mathcal{T}^{k-1}\boldsymbol{V}^{\pi_n}=\mathcal{T}\boldsymbol{V}^{\pi_n}$，$1<k<\infty$
	改进的策略迭代[40]	$\boldsymbol{V}=\mathcal{T}^{\pi}\boldsymbol{V}$	评估：$\boldsymbol{V}_{n+1}=(\mathcal{T}^{\pi_n})^{m_k}\boldsymbol{V}_n$，$1\leq m_k\leq\infty$，改进：$\mathcal{T}^{\pi_{n+1}}\boldsymbol{V}_{n+1}=\mathcal{T}\boldsymbol{V}_{n+1}$
	异步策略迭代[40]	$\boldsymbol{V}=\mathcal{T}^{\pi}\boldsymbol{V}$	$\forall s\in\boldsymbol{S}_n\subset\boldsymbol{S}$，评估：$\boldsymbol{V}_{n+1}(s)=(\mathcal{T}^{\pi_n}\boldsymbol{V}_n)(s)$，改进：$(\mathcal{T}^{\pi_{n+1}}\boldsymbol{V}_{n+1})(s)=(\mathcal{T}\boldsymbol{V}_{n+1})(s)$
	λ-策略迭代[40]	$\boldsymbol{V}=\mathcal{T}^{\lambda,\pi}\boldsymbol{V}$	评估：$\boldsymbol{V}_{n+1}=\mathcal{T}^{\lambda,\pi_n}\boldsymbol{V}_n$，改进：$\mathcal{T}^{\pi_{n+1}}\boldsymbol{V}_{n+1}=\mathcal{T}\boldsymbol{V}_{n+1}$
	值迭代[40]	$\boldsymbol{V}=\mathcal{T}\boldsymbol{V}$	$\boldsymbol{V}_{n+1}=\mathcal{T}\boldsymbol{V}_n$
	异步值迭代[40]	$\boldsymbol{V}=\mathcal{T}\boldsymbol{V}$	$\boldsymbol{V}_{n+1}(s_n)=(\mathcal{T}\boldsymbol{V}_n)(s_n)$，$s_n\in\boldsymbol{S}$
免模型	蒙特卡洛[36]	无	$\boldsymbol{V}(s)=\text{average}(Return(s))$
	TD[43]	$\boldsymbol{V}=\mathcal{T}^{\pi}\boldsymbol{V}$	$\Delta\boldsymbol{V}(s)=\alpha(r+\gamma\boldsymbol{V}(s')-\boldsymbol{V}(s))$
	TD(λ)[44]	$\boldsymbol{V}=\mathcal{T}^{\lambda,\pi}\boldsymbol{V}$	$\forall x\in\boldsymbol{S}$，$\boldsymbol{e}_n(x)=\gamma\lambda\boldsymbol{e}_{n-1}(x)+\mathbb{I}(x,s)$，$\Delta\boldsymbol{V}(s)=\alpha(r+\gamma\boldsymbol{V}(s')-\boldsymbol{V}(s))\boldsymbol{e}_n(s)$
	Sarsa[45]	$\boldsymbol{Q}=\mathcal{T}^{\pi}\boldsymbol{Q}$	$\Delta\boldsymbol{Q}(s,a)=\alpha(r+\gamma\boldsymbol{Q}(s',a')-\boldsymbol{Q}(s,a))$
	Q-learning[36]	$\boldsymbol{Q}=\mathcal{T}\boldsymbol{Q}$	$\Delta\boldsymbol{Q}(s,a)=\alpha(r+\gamma\max\limits_b\boldsymbol{Q}(s',b)-\boldsymbol{Q}(s,a))$
	Per-Decision[46]	$\boldsymbol{Q}=\mathcal{T}^{\lambda}\boldsymbol{Q}$	$\forall s\in\boldsymbol{S},a\in\boldsymbol{A},\boldsymbol{e}_t(s,a)=\gamma\lambda\rho_t\boldsymbol{e}_{t-1}(s,a)$，$\boldsymbol{e}_t(s_t,a_t)=1$，$\Delta\boldsymbol{Q}(s,a)=\alpha(r+\gamma\rho_t\boldsymbol{Q}(s',a')-\boldsymbol{Q}(s,a))\boldsymbol{e}_t(s,a)$
	Tree Backup[46]	$\boldsymbol{Q}=\mathcal{T}^{\lambda,\pi}\boldsymbol{Q}$	$\forall s\in\boldsymbol{S},a\in\boldsymbol{A},\boldsymbol{e}_t(s,a)=\gamma\lambda\rho_t\boldsymbol{e}_{t-1}(s,a)$，$\boldsymbol{e}_t(s_t,a_t)=1$，$\Delta\boldsymbol{Q}(s,a)=\alpha\left(r+\gamma\sum\limits_{a\in\boldsymbol{A}}(s',a)\boldsymbol{Q}(s',a)-\boldsymbol{Q}(s,a)\right)\boldsymbol{e}_t(s,a)$
	Double Q-learning[47]	$\boldsymbol{Q}=\mathcal{T}\boldsymbol{Q}$	随机选择 $i\in\{0,1\}$，$\Delta\boldsymbol{Q}^i(s,a)=\alpha(r+\gamma\boldsymbol{Q}^{1-i}(s',\arg\max\limits_b\boldsymbol{Q}^i(s',b))-\boldsymbol{Q}^i(s,a))$
	Speedy Q-learning[48]	$\boldsymbol{Q}=\mathcal{T}\boldsymbol{Q}$	$\Delta\boldsymbol{Q}(s,a)=\alpha(\mathcal{T}_k\boldsymbol{Q}_{k-1}(s,a)-\boldsymbol{Q}_k(s,a))+(1-\alpha)(\mathcal{T}_k\boldsymbol{Q}_k(s,a)-\mathcal{T}_k\boldsymbol{Q}_{k-1}(s,a))$
	GSQ-learning[49]	$\boldsymbol{Q}=\mathcal{T}\boldsymbol{Q}$	替换 Speedy Q 中 $\mathcal{T}_k$ 为 $(\boldsymbol{H}^{\omega}\boldsymbol{Q})(s,a)=\omega\mathcal{T}_k\boldsymbol{Q}(s,a)+(1-\omega)\max\limits_{b\in\boldsymbol{A}}\boldsymbol{Q}(s,b)$
	DSQ-learning[50]	$\boldsymbol{Q}=\mathcal{T}\boldsymbol{Q}$	采用 Double estimator 分别替换 $\boldsymbol{Q}_{k-1}$ 和 $\boldsymbol{Q}_k$
	SCQ-learning[51]	$\boldsymbol{Q}=\mathcal{T}\boldsymbol{Q}$	替换 Double Q 中 target：$\boldsymbol{Q}_n^{\beta}(s',a)=\boldsymbol{Q}_n(s',a)-\beta[\boldsymbol{Q}_n(s',a)-\boldsymbol{Q}_{n-1}(s',a)]$

时较长，因此异步策略迭代(Asynchronous Policy Iteration)每轮只对状态空间的真子集 $\boldsymbol{S}_n\subset\boldsymbol{S}$ 进行迭代，同样减少了运算量. 进一步，异步值迭代(Asynchronous Value Iteration)的每次迭代只针对状态空间的一个样本. λ-策略迭代在策略评估阶段，通过综合所有的多步 Bellman 评估算子，只做一次迭代，可以更快速地逼近不动点. 广义策略迭代(Generalized Policy Iteration，GPI)框架从粒度(迭代次数、评估状态集合的大小等)角度抽象了上述算法，进一步推广了策略评估与策略改进的交互过程.

基于学习模型的强化学习算法：当模型未知时，可以通过学习模型逼近真实模型，并利用前向规划算法如 Dyna-Q 学习[36]、后向规划算法如 Prioritized Sweeping[36] 以及经验回放(Experience Replay)[41]等对强化学习算法的不动点求解进行加速. 此时，强化学习算法设计的挑战来源于模型的不同特性，如随机性(Stochasticity)、不确定性(Uncertainty)、部分可观察性(Partial observability)、非稳态(Non-stationarity)、多步预测(Multi-step Prediction)、状态抽象(State abstraction)、时间抽象(Temporal Abstraction)等，更多详情请关注基于模型的强化学习综述[42].

免模型强化学习算法. 不需要模型的动态信息，直接与环境进行交互，以此评估和改进策略. 经典的免模型强化学习算法包括：(1) 蒙特卡洛(Monte Carlo)算法，每次在游戏结束时统计奖赏总和，其优点是无偏估计，但代价是方差大，同时要等游戏结束才能更新值函数，速度较慢；(2) TD(Temporal Difference)学习算法是学习预测方法，每次只更新一个样本，可以做到高效的策略评估，代价是估计有偏差[43]；(3) TD(λ)算法[44]通过采样 $\mathcal{T}^{\lambda,\pi}$ 的一条轨迹，可以快速逼近不动点(21)；(4) Sarsa 算法[45]，顾名思义通过学习元组 $\langle s,a,r,s',a'\rangle$ 进行控制，利用无偏估计 $\hat{\boldsymbol{Q}}(s,a)$ 逼近状态动作值函数 $\boldsymbol{Q}(s,a)$，即 $\mathbb{E}[\hat{\boldsymbol{Q}}(s,a)]=\boldsymbol{Q}(s,a)$，其特点是行为策略和目标策略是相同的，属于同策略(On-Policy)学习；(5) Q-learning 算法根据最优状态动作值的采样 $\max\limits_a\hat{\boldsymbol{Q}}(s,a)$ 来更新值函数，其特点是行为策略和目标策略是不同的，属于异策略(Off-Policy)学习. 2000 年，Precup 等人提出了基于重要性采样率 $\rho_t=\dfrac{\boldsymbol{\pi}(s_t,a_t)}{\boldsymbol{b}(s_t,a_t)}$ 的异策略 Per-Decision 算法，$\Delta\boldsymbol{Q}(s,a)=\alpha\boldsymbol{e}_t(s,a)\delta_t$，其中，误差项为 $\delta_t=R_t+\gamma\sum\limits_{a\in\boldsymbol{A}}\rho_t\boldsymbol{Q}(s_{t+1},a_{t+1})-\boldsymbol{Q}(s_t,a_t)$，$\forall s,a,\boldsymbol{e}_t(s,a)=\gamma\lambda\rho_t\boldsymbol{e}_{t-1}$，针对当前执行的

s_t,a_t有$\boldsymbol{e}_t(s_t,a_t)=1$;同时他还提出了 Tree Backup 算法,与 Per-Decision 算法不同的是误差项 $\delta_t=R_t+\gamma\sum_{a\in\boldsymbol{A}}\boldsymbol{\pi}(s_{t+1},a)\boldsymbol{Q}(s_{t+1},a_{t+1})-\boldsymbol{Q}(s_t,a_t)$;此外,Precup 等人还指出,$\lambda$ 可以是依赖状态的数值,而不是常数[46]. 根据 Jensen 不等式可知,$\mathbb{E}[\max_a\boldsymbol{Q}(s,a)]\geq\max_a\mathbb{E}[\hat{\boldsymbol{Q}}(s,a)]=\max_a\boldsymbol{Q}(s,q)$,这就是过估计(Over-Estimation)问题. 2010 年,van Hasselt 等人提出了Double Q-learning,采用两个无偏估计$\hat{\boldsymbol{Q}}_0(s,a)$和$\hat{\boldsymbol{Q}}_1(s,a)$分别逼近$\boldsymbol{Q}(s,a)$,以欠估计(Underestimation)的方式,即用$\hat{\boldsymbol{Q}}_{1-i}(s,\arg\max_a\boldsymbol{Q}_i(s,a))$替换$\max_a\hat{\boldsymbol{Q}}_i(s,q)$,其中$i\in\{0,1\}$,缓解过估计的问题[47]. 2011 年,Azar 等人用$\alpha_k=\frac{1}{k+1}$学习时序 TD 误差,并用更大的学习率$(1-\alpha_k)$学习 Q 函数的时序误差,获得高速 Q 学习算法(Speedy Q-learning)[48]. 2016 年,Munos 等人提出基于截断重要性采样 $c_t=\min\{1,\rho_t\}$的 Retrace(λ)算法,并证明了该算法在异策略下的收敛性[52]. 2019 年,Lv 等人提出 SDQ(Stochastic Double Q-learning),随机选择过估计和欠估计以尽量逼近真实估计[53]. 2020 年,John 等人将 Speedy Q-learning 的思想推广到广义 Bellman 算子

$$(\boldsymbol{H}^{\omega}\boldsymbol{Q})(s,a)=\omega[\boldsymbol{R}(s,a)+\gamma\sum_{s'\in\boldsymbol{S}}\boldsymbol{P}(s,a,s')\max_{b\in\boldsymbol{A}}\boldsymbol{Q}(s',b)]+(1-\omega)\max_{c\in\boldsymbol{A}}\boldsymbol{Q}(s,c)\tag{22}$$

得到Generalized Speedy Q-learning[49]. 2020年,Zheng 等人通过直接将 Speedy Q-learning 和 Double Q-learning 进行结合,得到能减轻过估计并加速收敛的 Double Speedy Q-learning 算法[50]. 2021 年,Zhu 等人借鉴 Double Q-learning 以及时序上延迟的 Target network 思想[54],通过对两个值函数进行插值,提出了自矫正 Q-learning 算法(Self-Correcting Q-learning)[51].

从不动点视角看,表格值强化学习算法由于满足压缩映射性质,都有收敛性保证. 近期的工作主要集中在样本复杂度分析[55-57]、后悔界[58]等方面. 在实际应用中,大家更关心的是,当 MDP 的一些假设被打破时如何应对,如半马氏属性[59]、部分可观察性[60]、非稳态[61]等. 当序贯决策问题的规模较大时,表格值强化学习算法不再适用. 接下来,我们介绍不动点视角下基于参数化估计或无参估计的强化学习.

4 线性值强化学习

基于表格值的迭代函数 $\boldsymbol{f}(x_n)$通常可以满足利普希茨常数小于 1 的压缩映射假设. 然而采用函数估计的迭代函数 $\boldsymbol{f}(x_n)$后,无论是线性函数估计,还是非线性函数估计,利普希茨常数不能保证小于 1[62]. 因此,不能直接使用巴拿赫不动点定理证明线性值估计强化学习算法的稳定性.

4.1 线性值强化学习的稳定性原理

引入一个步长参数 $\alpha(0<\alpha<1)$,通过梯度上升或下降方式进行迭代,从而保证采用估计的迭代函数 $\boldsymbol{f}(x_n)$利普希茨常数小于 1. 设 $\boldsymbol{f}(x_n)$是形如 $x_n+\alpha g(x_n)$的梯度上升或下降的函数,即迭代式为 $x_{n+1}=x_n+\alpha g(x_n)$. 此时,迭代式可以转化为动力(常微分)方程[63]:

$$\dot{x}(t)=g(x(t))\tag{23}$$

其中 t 表示连续时间.

定理 2. 李雅普诺夫(Lyapunov)第二判定法. 系统 $\dot{x}(t)=g(x)$在 $x=x_0$处有一个平衡点. 若存在 Lyapunov 函数 $\boldsymbol{L}:\mathbb{R}^n\to\mathbb{R}$,满足(1)$\boldsymbol{L}(x)\geq0$,仅在 $x=x_0$处等号成立;(2)对于$x\neq x_0$,一阶导数 $\dot{\boldsymbol{L}}(x)<0$. 则系统渐进收敛到平衡点.

利用李雅普诺夫第二判定法判定一个系统渐进收敛性的技巧在于找到合适的 Lyapunov 函数.

推论 1. 线性系统的稳定性. 假设矩阵 $\boldsymbol{A}$ 是正定矩阵,则系统 $\dot{\boldsymbol{\theta}}(t)=-\boldsymbol{A}\boldsymbol{\theta}(t)+\boldsymbol{b}$ 渐进稳定.

证明. 设Lyapunov函数$\boldsymbol{L}(\boldsymbol{\theta}(t))=(-\boldsymbol{A}\boldsymbol{\theta}(t)+\boldsymbol{b})^{\top}(-\boldsymbol{A}\boldsymbol{\theta}(t)+\boldsymbol{b})/2$,容易验证:

(1)$\boldsymbol{L}(\boldsymbol{\theta}(t))\geq0$,仅在 $\boldsymbol{\theta}(t)=\boldsymbol{A}^{-1}\boldsymbol{b}$ 处等号成立;

(2)一阶导数

$$\dot{\boldsymbol{L}}(\boldsymbol{\theta}(t))=(-\boldsymbol{A}\boldsymbol{\theta}(t)+\boldsymbol{b})^{\top}(-\boldsymbol{A}\dot{\boldsymbol{\theta}}(t))=-(\boldsymbol{A}\boldsymbol{\theta}(t)+\boldsymbol{b})^{\top}\boldsymbol{A}(-\boldsymbol{A}\boldsymbol{\theta}(t)+\boldsymbol{b}),$$

由于 $\boldsymbol{\theta}(t)\neq\boldsymbol{A}^{-1}\boldsymbol{b}$,$(-\boldsymbol{A}\boldsymbol{\theta}(t)+\boldsymbol{b})\neq0$,并且根据正定矩阵的定义,$\dot{\boldsymbol{L}}(\boldsymbol{\theta}(t))<0$. 根据 Lyapunov 第二判定法,该系统渐进稳定,并收敛至 $\boldsymbol{A}^{-1}\boldsymbol{b}$. 证毕.

线性系统对应的梯度迭代式是

$$x_{n+1}=(\boldsymbol{I}-\alpha\boldsymbol{A})x_n+\alpha\boldsymbol{b}\tag{24}$$

此时,利普希茨常数为$\|\boldsymbol{I}-\alpha\boldsymbol{A}\|$,其中$\|\cdot\|$为任意的矩阵范数. 根据巴拿赫不动点定理,迭代式(24)的收敛性要求$\|\boldsymbol{I}-\alpha\boldsymbol{A}\|<1$. 又根据假设,$\boldsymbol{A}$ 是正定矩阵,设 $\rho(\boldsymbol{A})$为矩阵 $\boldsymbol{A}$ 的谱半径. 因此,步长满足 $0<\alpha<\frac{2}{\rho(\boldsymbol{A})}$即可.

线性系统对应的随机梯度迭代式是

$$x_{n+1}=x_n+\alpha_n(b_t\boldsymbol{a}_t-\boldsymbol{a}_t\boldsymbol{a}_t^{\top}x_n+M_{n+1})\tag{25}$$

其中,随机性体现在 $\boldsymbol{a}_t$是从矩阵 $\boldsymbol{A}$ 中随机选择的第 t 列,b_t是向量 $\boldsymbol{b}$ 中第 t 个值;对应于梯度迭代式,$\boldsymbol{a}_t\boldsymbol{a}_t^{\top}$

的期望等于正定矩阵$\boldsymbol{A}$，即$\mathbb{E}[\boldsymbol{a}_t\boldsymbol{a}_t^\top]=\boldsymbol{A}$；$\{M_n\}$是均值为0，方差有界的噪音序列；此时，为了满足渐进收敛，步长序列$\{\alpha_n\}$满足$0<\alpha_n<1$，非递增，且$\sum_n^\infty\alpha_n=\infty$，初始值$\alpha_0$不要太大即可. 此外，注意到迭代式(25)中有噪音项M_n，根据Kolmogorov定理，α_nM_{n+1}收敛的充分必要条件是$\sum_n^\infty\alpha_n^2$收敛[64]. 因此，步长序列$\{\alpha_n\}$需要满足的另一个条件是$\sum_n^\infty\alpha_n^2<\infty$. 随机梯度迭代中$\boldsymbol{a}_t\boldsymbol{a}_t^\top$的期望等于正定矩阵$\boldsymbol{A}$，然而每个$\boldsymbol{a}_t\boldsymbol{a}_t^\top$是不同的，这就给学习过程带来了方差，控制住这个方差将有利于提高收敛速度[65].

接下来，主要从算法的稳定性角度分别对同策略强化学习算法、异策略强化学习算法进行详细的梳理和总结.

4.2　线性值同策略强化学习

表4分别从优化方法、求解的不动点表达式、收敛性保证的关键条件正定矩阵以及误差界等角度，分析和总结了基于线性值估计的同策略强化学习算法.

表4　基于线性不动点的同策略强化学习算法

分类	年份	算法	不动点	正定矩阵	误差界
随机梯度下降	1988	TD(0)[44]	$\mathbb{E}[\delta\boldsymbol{\phi}]=0$	$\boldsymbol{A}_{\text{on}}$	$\|\boldsymbol{\Pi}\boldsymbol{V}^*-\boldsymbol{V}^*\|/(1-\gamma)$[62]
	1988	TD(λ)[44]	$\mathbb{E}[\delta\boldsymbol{e}]=0$	$\boldsymbol{A}_\lambda$	$\|\boldsymbol{\Pi}\boldsymbol{V}^*-\boldsymbol{V}^*\|/(1-\gamma(1-\lambda)/(1-\gamma\lambda))$[62]
	1994	NTD(0)[66]	$\mathbb{E}[\delta\boldsymbol{\phi}]=0$	$\boldsymbol{A}_{\text{on}}$	$\|\boldsymbol{\Pi}\boldsymbol{V}^*-\boldsymbol{V}^*\|/(1-\gamma)$[62]
	1994	NTD(λ)[66]	$\mathbb{E}[\delta\boldsymbol{e}]=0$	$\boldsymbol{A}_\lambda$	$\|\boldsymbol{\Pi}\boldsymbol{V}^*-\boldsymbol{V}^*\|/(1-\gamma(1-\lambda)/(1-\gamma\lambda))$[62]
	2011	Accelerated-TD[70]	$\mathbb{E}[\delta\boldsymbol{\phi}]=0$	$\boldsymbol{I}$	$\|\boldsymbol{\Pi}\boldsymbol{V}^*-\boldsymbol{V}^*\|/(1-\gamma)$[62]
	2014	True TD(λ)[71]	$\mathbb{E}[\delta\boldsymbol{e}]=0$	$\boldsymbol{A}_\lambda$	$\|\boldsymbol{\Pi}\boldsymbol{V}^*-\boldsymbol{V}^*\|/(1-\gamma(1-\lambda)/(1-\gamma\lambda))$[62]
	2021	GCTD[72]	$\mathbb{E}[\delta\boldsymbol{e}]=0$	$\boldsymbol{A}_\lambda$	$\|\boldsymbol{\Pi}\boldsymbol{V}^*-\boldsymbol{V}^*\|/(1-\gamma(1-\lambda)/(1-\gamma\lambda))$[62]
最小二乘	1996	LSTD(λ)[67]	$\mathbb{E}[\delta\boldsymbol{\phi}]=0$	$\boldsymbol{A}_{\text{on}}$	$\|\boldsymbol{\Pi}\boldsymbol{V}^*-\boldsymbol{V}^*\|/(1-\gamma)$[62]
	1996	RLSTD[67]	$\mathbb{E}[\delta\boldsymbol{\phi}]=0$	$\boldsymbol{A}_{\text{on}}$	$\|\boldsymbol{\Pi}\boldsymbol{V}^*-\boldsymbol{V}^*\|/(1-\gamma)$[62]
	1996	LSPE(λ)[40]	$\mathbb{E}[\delta\boldsymbol{e}]=0$	$\boldsymbol{C}^{-1}\boldsymbol{A}_\lambda$	$\|\boldsymbol{\Pi}\boldsymbol{V}^*-\boldsymbol{V}^*\|/(1-\gamma(1-\lambda)/(1-\gamma\lambda))$[39]
	2002	LSTD(λ)[73]	$\mathbb{E}[\delta\boldsymbol{e}]=0$	$\boldsymbol{A}_\lambda$	$\|\boldsymbol{\Pi}\boldsymbol{V}^*-\boldsymbol{V}^*\|/(1-\gamma(1-\lambda)/(1-\gamma\lambda))$[62]
	2002	RLSTD(λ)[74]	$\mathbb{E}[\delta\boldsymbol{e}]=0$	$\boldsymbol{A}_\lambda$	$\|\boldsymbol{\Pi}\boldsymbol{V}^*-\boldsymbol{V}^*\|/(1-\gamma(1-\lambda)/(1-\gamma\lambda))$[62]
	2003	LSPI[75]	$\mathbb{E}[\delta\boldsymbol{\phi}(s,a)]=0$	$\boldsymbol{A}_{\text{on}}$	$\|\boldsymbol{Q}^{\pi_\infty}-\boldsymbol{Q}^*\|\leq\frac{2\gamma\epsilon}{(1-\gamma)^2}$[73]
	2006	iLSTD[76]	$\mathbb{E}[\delta\boldsymbol{\phi}]=0$	$\boldsymbol{A}_{\text{on}}$	$\|\boldsymbol{\Pi}\boldsymbol{V}^*-\boldsymbol{V}^*\|/(1-\gamma)$[62]
	2007	iLSTD(λ)[77]	$\mathbb{E}[\delta\boldsymbol{e}]=0$	$\boldsymbol{A}_\lambda$	$\|\boldsymbol{\Pi}\boldsymbol{V}^*-\boldsymbol{V}^*\|/(1-\gamma(1-\lambda)/(1-\gamma\lambda))$[62]
	2009	HLSTD[78]	$\mathbb{E}[\delta(\boldsymbol{\phi}-\omega\gamma\boldsymbol{\phi}')]=0$	$\boldsymbol{A}_H$	无
	2015	Forgetful LSTD(λ)[79]	$\mathbb{E}[\delta\boldsymbol{e}]=0$	$\boldsymbol{A}_\lambda$	$\|\boldsymbol{\Pi}\boldsymbol{V}^*-\boldsymbol{V}^*\|/(1-\gamma(1-\lambda)/(1-\gamma\lambda))$[62]
	2020	Uncorrected LSTD(λ)[80]	$\mathbb{E}[\delta\boldsymbol{e}]=0$	$\boldsymbol{A}_\lambda$	$\|\boldsymbol{\Pi}\boldsymbol{V}^*-\boldsymbol{V}^*\|/(1-\gamma(1-\lambda)/(1-\gamma\lambda))$[62]

1988年，Sutton提出TD(λ)算法，并给出了TD(0)算法的收敛性证明[44]. 其中，TD(0)收敛到不动点$\boldsymbol{V}_{\boldsymbol{\theta}}=\boldsymbol{\Pi}\boldsymbol{T}^\pi\boldsymbol{V}_{\boldsymbol{\theta}}$，其期望形式是方程$\mathbb{E}_\pi[\delta\boldsymbol{\phi}]=0$的解，收敛性证明的关键性在于同策略(on-policy)[62]，即行为策略与目标策略相同，使得状态分布与状态转移概率满足不动点$\boldsymbol{d}_\pi=\boldsymbol{P}^{\pi\top}\boldsymbol{d}_\pi$，由此可得正定矩阵

$$\boldsymbol{A}_{\text{on}}=\mathbb{E}_\pi[(\boldsymbol{\phi}_k-\gamma\boldsymbol{\phi}_k')\boldsymbol{\phi}_k^\top]=\boldsymbol{\Phi}^\top\boldsymbol{D}_\pi(\boldsymbol{I}-\gamma\boldsymbol{P}^\pi)\boldsymbol{\Phi}\tag{26}$$

1997年，Tsitsiklis等人证明了TD(λ)算法以概率1收敛到不动点$\boldsymbol{V}_{\boldsymbol{\theta}}=\boldsymbol{\Pi}\boldsymbol{T}^{\lambda,\pi}\boldsymbol{V}_{\boldsymbol{\theta}}$，其期望形式为方程$\mathbb{E}[\delta\boldsymbol{e}]=0$的解，收敛的关键因素在于正定矩阵

$$\boldsymbol{A}_\lambda=\mathbb{E}_\pi[(\boldsymbol{\phi}_k-\gamma\boldsymbol{\phi}_k')\boldsymbol{e}_k^\top]\tag{27}$$

同时证明了综合多步回报算子$\boldsymbol{T}^\lambda$满足压缩映射，给出了TD(λ)算法的误差界$\frac{\|\boldsymbol{\Pi}\boldsymbol{V}^*-\boldsymbol{V}^*\|}{1-\gamma(1-\lambda)/(1-\gamma\lambda)}$[62].

1994年，针对特征函数过大(>1)导致算法容易发散的问题，Bradtke在博士毕业论文中提出了特征正则化的NTD(λ)算法(Normalized TD)，并基于TD(0)收敛性证明，通过调整学习率证明了NTD(λ)算法$\boldsymbol{\phi}$的收敛性[66]. 其中，特征由$\boldsymbol{\phi}^k$正则化为$\frac{\boldsymbol{\phi}_k}{\epsilon+\boldsymbol{\phi}_k^\top\boldsymbol{\phi}_k}$，$\epsilon$为一个很小的正数. 为了更高效地利用已有数据，1996年，Bradtke采用最小二乘法和递归最小二乘法对方程$\boldsymbol{A}\boldsymbol{\theta}=\boldsymbol{b}$直接进行求解：

$$\boldsymbol{\theta}_t=\boldsymbol{A}_t^{-1}\boldsymbol{b}_t\tag{28}$$

分别得到LSTD(Least Squared TD)算法和RLSTD(Recursive LSTD)算法，只需可逆性判定，即可证明LSTD算法的收敛性. 它们的另一个优势是没有学习率，不需要手工调参[67]. 1996年，Bertsekas等人综合多步回报提出了LSPE(λ)算法(λ-Policy Iteration)[40,68]，其更新公式与LSTD不同，本质上属于Jacobi方法[69]：

$$\boldsymbol{\theta}_{t+1}=\boldsymbol{\theta}_t+\alpha\boldsymbol{C}_t^{-1}(-\boldsymbol{A}_t\boldsymbol{\theta}_t+\boldsymbol{b}_t)\tag{29}$$

更进一步，2002年，Boyan综合多步回报，提出了LSTD(λ)算法[73]. 2002年，Xu等人综合多步回报，提出了RLSTD(λ)算法[74]. 与TD类算法相比，LSTD类算法也有缺点，它们的每步计算复杂度和存储复

杂度都是 $O(n^2)$，而 TD 类算法每步计算复杂度和存储复杂度都是 $O(n)$. 2006 年，Geramifard 等人针对大规模特征空间，采用贪心选择方式更新误差最显著的特征项对应的参数，使得每步的计算复杂度降为 $O(n)$，得到 iLSTD(incremental LSTD)算法[76]. 2007 年，Geramifard 等人综合多步回报，以及利用特征选择机制，提出了 iLSTD 的改进 iLSTD(λ)算法，并证明了随机特征选择机制下，iLSTD(λ)算法的收敛性[77]. 2003 年，Lagoudakis 等人将针对状态值评估的 LSTD 推广到状态动作值评估 LSTDQ 算法，并采用策略迭代方式更新策略，得到保证收敛的 LSPI(Least Squared Policy Iteration)算法[75]. 2009 年，Johns 等人通过对 TD 不动点与 BR 不动点进行插值，并用最小二乘法进行求解，得到 HLSTD(Hybrid LSTD)[78]算法，其中关键矩阵为

$$\begin{aligned}\boldsymbol{A}_H&=\mathbb{E}_{\pi}[(\boldsymbol{\phi}-\gamma\boldsymbol{\phi}')(\boldsymbol{\phi}-w\gamma\boldsymbol{\phi}')^{\top}]\\&=\boldsymbol{\Phi}^{\top}(\boldsymbol{I}-w\gamma\boldsymbol{P}^{\pi})^{\top}D_{\pi}(I-\gamma P^{\pi})\boldsymbol{\Phi}\end{aligned}\tag{30}$$

2011 年，Ueno 通过定义广义的估计函数，对 TD、TD(λ)、RG 算法做了总结，并针对正定矩阵做了修改. 以 TD 为例，

$$\boldsymbol{\theta}_{t+1}=\boldsymbol{\theta}_t+\alpha\boldsymbol{A}_t^{-1}\delta_t\boldsymbol{\phi}_t\tag{31}$$

由此得到了 Accelerated-TD 算法[70]. 2014 年，van Seijen 等人采用截断多步回报，证明了前后向视角等价的真正(True)的 TD(λ)算法[71,81]. 2015 年，van Seijen 等人从经验回放角度，设置了遗忘率参数，对最小二乘方法做了改进，得到 Forgetful LSTD(λ)[79]. 2020 年，Osogami 根据 True TD(λ)的思想，对 LSTD(λ)进行改进，提出了 Uncorrected LSTD(λ)[80]. 2021 年，Wang 等人从梯度补偿角度而非资格迹角度提出了梯度补偿时序差分 GCTD 学习算法(Gradient Compensation TD)[72].

4.3 线性值异策略强化学习

异策略(Off-Policy)指的是负责探索的行为策略 $\boldsymbol{\mu}$ 和目标策略 $\boldsymbol{\pi}$ 不一样. 通过在期望式 $\mathbb{E}_{\mu}[\cdot]$ 加入重要性采样率 $\rho_t=\dfrac{\boldsymbol{\pi}(s_t,a_t)}{\boldsymbol{\mu}(s_t,a_t)}$，使得从行为策略的行为转化为目标策略，即设依赖于 s_t,a_t,s_{t+1} 的变量 $\boldsymbol{z}_{t+1}$，可得

$$\begin{aligned}\mathbb{E}_{\mu}[\rho_t\boldsymbol{z}_{t+1}\mid s_t=s]&=\sum_a\boldsymbol{\mu}(s,a)\frac{\boldsymbol{\pi}(s,a)}{\boldsymbol{\mu}(s,a)}\boldsymbol{z}_{t+1}\\&=\sum_a\boldsymbol{\pi}(s,a)\boldsymbol{z}_{t+1}\\&=\mathbb{E}_{\pi}[\boldsymbol{z}_{t+1}\mid s_t=s]\end{aligned}\tag{32}$$

此时，核心要素矩阵为

$$\begin{aligned}\boldsymbol{A}_{\rho,\mathrm{off}}&=\mathbb{E}_{\mu}[\rho_t\boldsymbol{\phi}_t(\boldsymbol{\phi}_t-\gamma\boldsymbol{\phi}_{t+1})^{\top}]\\&=\sum_s\boldsymbol{d}_{\mu}(s)\mathbb{E}_{\mu}[\rho_t\boldsymbol{\phi}_t(\boldsymbol{\phi}_t-\gamma\boldsymbol{\phi}_{t+1})^{\top}]\\&=\sum_s\boldsymbol{d}_{\mu}(s)\mathbb{E}_{\pi}[\boldsymbol{\phi}_t(\boldsymbol{\phi}_t-\gamma\boldsymbol{\phi}_{t+1})^{\top}]\\&=\boldsymbol{\Phi}^{\top}\boldsymbol{D}_{\mu}(\boldsymbol{I}-\gamma\boldsymbol{P}^{\pi})\boldsymbol{\Phi}\end{aligned}\tag{33}$$

由于状态的分布不满足不变性，即 $\boldsymbol{d}_{\mu}\neq\boldsymbol{P}^{\pi\top}\boldsymbol{d}_{\mu}$，矩阵 $\boldsymbol{A}_{\rho,\mathrm{off}}$ 无法保证正定性，因此异策略 TD 算法不能保证收敛[82].

为了确保算法的收敛，本文从四个角度分别阐述了异策略下强化学习的改进方案：(1)投影法. 将 $\mathcal{T}\boldsymbol{V}$ 利用某种投影函数投影到 $\boldsymbol{D\Phi}$ 平面，使得投影后的 Bellman 方程能成立；(2)分布矫正思想. 针对矩阵 $\boldsymbol{A}_{\rho,\mathrm{off}}$ 无法保证正定性问题，采用分布矫正的方式使得矫正后的矩阵确保正定；(3)压缩映射. 如果算法满足压缩映射，那么根据前述理论，很容易证明收敛性；(4)工程解决方案. 不关心理论上保持收敛的做法，直接看效果，做最轻量的改变. 表 5 主要针对稳定性问题，分别从求解的不动点、正定矩阵、优化方法和误差界角度，分析和总结了基于值估计不动点的异策略强化学习算法.

投影法. 1995 年，Baird 等人提出了一个 TD 算法不能收敛的“7-态”反例，从优化角度提出了残差梯度算法(Residual Gradient，RG)[83]. Bellman 残差法可保证收敛到残差不动点 $\boldsymbol{V}_{\boldsymbol{\theta}}=\boldsymbol{\Pi}_{\boldsymbol{X}}\mathcal{T}\boldsymbol{V}_{\boldsymbol{\theta}}$，写成期望形式是方程 $\mathbb{E}_{\mu}[\delta(\boldsymbol{\phi}-\gamma\boldsymbol{\phi}')]=0$ 的解. 其中，斜投影 $\boldsymbol{\Pi}_{\boldsymbol{X}}=\boldsymbol{\Phi}(\boldsymbol{\Phi}^{\top}(\boldsymbol{I}-\gamma\boldsymbol{P})^{\top}\boldsymbol{D}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\top}(\boldsymbol{I}-\gamma\boldsymbol{P})^{\top}\boldsymbol{D}$[84]. 其中，Bellman 残差法收敛的关键因素在于正定矩阵

$$\boldsymbol{A}_{\mathrm{rg}}=\mathbb{E}_{\mu}[(\boldsymbol{\phi}_k-\gamma\boldsymbol{\phi}_k')(\boldsymbol{\phi}_k-\gamma\boldsymbol{\phi}_k')^{\top}]\tag{34}$$

此外，Baird 等人结合 Bellman 残差法的收敛性和 TD 算法的快速学习特点，采用批量梯度下降优化方式，通过插值思想设置 w，在 TD 与 RG 同向时选择 TD，在 TD 与 RG 不同向时，找一个与 TD 方向最近并且与 RG 角度略小于 90°的方向进行更新，得到了残差算法(Residual)[83]. Residual 算法如果收敛的话，将收敛至 $\mathbb{E}_{\mu}[\delta(\boldsymbol{\phi}-w\gamma\boldsymbol{\phi}')]=0$，这里“如果”指的是矩阵很难形式化，更无法证明它的正定性. 2008 年，Sutton 等人直接假设异策略中状态 s 和后继状态 s' 服从两个独立的分布. 那么更一般的矩阵为

$$\boldsymbol{A}_{\mathrm{off}}=\mathbb{E}[(\boldsymbol{\phi}_k-\gamma\boldsymbol{\phi}_k')\boldsymbol{\phi}_k^{\top}]\tag{35}$$

异策略不动点是方程 $\mathbb{E}_{\mu}[\delta\boldsymbol{\phi}]=0$ 的解，矩阵形式为 $\boldsymbol{A}_{\mathrm{off}}\boldsymbol{\theta}=\boldsymbol{b}$，通过最小化 $\mathrm{NEU}(\boldsymbol{\theta})=\mathbb{E}_{\mu}[\delta\boldsymbol{\phi}]^{\top}\mathbb{E}_{\mu}[\delta\boldsymbol{\phi}]$ 得到能保证收敛的 GTD(Gradient TD)算法，其核心要素为正定矩阵[85].

表 5　基于线性不动点的异策略强化学习算法

分类	年份	算法	不动点	正定矩阵	优化方法	误差界
投影法	1995	RG[83]	$\mathbb{E}_{\mu}[\delta(\boldsymbol{\phi}-\gamma\boldsymbol{\phi}')]=0$	$\boldsymbol{A}_{\mathrm{rg}}$	随机梯度下降	无
	1995	Residual[83]	$\mathbb{E}_{\mu}[\delta(\boldsymbol{\phi}-w\gamma\boldsymbol{\phi}')]=0$	未知	批量梯度下降	
	2008	GTD[85]	$\mathbb{E}_{\mu}[\delta\boldsymbol{\phi}]=0$	$\boldsymbol{A}_{\mathrm{GTD}}$	随机梯度下降	
	2009	GTD2[38]	$\mathbb{E}_{\mu}[\delta\boldsymbol{\phi}]=0$	$\boldsymbol{A}_{\mathrm{GTD2}}$	随机梯度下降	
	2009	TD[38]	$\mathbb{E}_{\mu}[\delta\boldsymbol{\phi}]=0$	$\boldsymbol{A}_{\mathrm{off}}^{\top}\boldsymbol{C}^{-1}\boldsymbol{A}_{\mathrm{off}}$	随机梯度下降	
	2010	GQ(λ)[86]	$\mathbb{E}_{\mu}[\delta\boldsymbol{e}]=0$	$\boldsymbol{A}_{\lambda}^{\top}\boldsymbol{C}^{-1}\boldsymbol{A}_{\lambda}$	随机梯度下降	
	2011	AB[91]	$\mathbb{E}_{\mu}[\delta\boldsymbol{\phi}]=0$	$\boldsymbol{A}_{\mathrm{off}}^{\top}\boldsymbol{A}_{\mathrm{on}}^{-1}\boldsymbol{A}_{\mathrm{off}}$	随机梯度下降	
	2011	HybridGQ[91]	$\mathbb{E}_{\mu}[\delta\boldsymbol{\phi}]=0$	$\boldsymbol{A}_{\mathrm{off}}^{\top}\boldsymbol{A}_{\mathrm{on}}^{-1}\boldsymbol{A}_{\mathrm{off}}$	随机梯度下降	
	2013	TDGQ[90]	$\mathbb{E}_{\mu}[\delta\boldsymbol{\phi}]=0$	未知	随机梯度下降	
	2016	GTD-MP[92]	$\mathbb{E}_{\mu}[\delta\boldsymbol{\phi}]=0$	$\boldsymbol{A}_{\mathrm{GTD}}$	近端优化	
	2016	GTD2-MP[92]	$\mathbb{E}_{\mu}[\delta\boldsymbol{\phi}]=0$	$\boldsymbol{A}_{\mathrm{GTD2}}$	近端优化	
	2017	ATD(λ)[93]	$\mathbb{E}_{\mu}[\delta\boldsymbol{e}]=0$	$(\boldsymbol{A}_{\lambda}+\epsilon\boldsymbol{I})^{-1}\boldsymbol{A}_{\lambda}$	随机梯度下降	
	2017	RTD[94]	$\mathbb{E}_{\mu}[\delta(\boldsymbol{\phi}-w\gamma\boldsymbol{\phi}')]=0$	$\boldsymbol{A}_{\mathrm{RTD}}$	随机梯度下降	
	2019	RLSTDC[95]	$\mathbb{E}_{\mu}[\delta\boldsymbol{\phi}]=0$	$\boldsymbol{A}_{\mathrm{off}}+\epsilon\boldsymbol{I}$	最小二乘	
	2019	RC(λ)[95]	$\mathbb{E}_{\mu}[\delta\boldsymbol{e}]=0$	$\boldsymbol{A}_{\lambda}+\epsilon\boldsymbol{I}$	最小二乘	
	2020	TDRC[96]	$\mathbb{E}_{\mu}[\delta\boldsymbol{\phi}]=0$	$\boldsymbol{A}_{\mathrm{off}}^{\top}\boldsymbol{C}^{-1}\boldsymbol{A}_{\mathrm{off}}$	随机梯度下降	
分布矫正	2001	IS-TD(λ)[97]	$\mathbb{E}_{\mu}[\delta\boldsymbol{e}]=0$	$\boldsymbol{A}_{\lambda}$	随机梯度下降	$\|\boldsymbol{\Pi}\boldsymbol{V}^{*}-\boldsymbol{V}^{*}\|/(1-\gamma(1-\lambda)/(1-\gamma\lambda))$[92]
	2014	WIS-LSTD(λ)[98]	$\mathbb{E}_{\mu}[\delta\boldsymbol{e}]=0$	$\boldsymbol{A}_{\lambda}$	最小二乘	$\|\boldsymbol{\Pi}\boldsymbol{V}^{*}-\boldsymbol{V}^{*}\|/(1-\gamma(1-\lambda)/(1-\gamma\lambda))$[92]
	2015	WIS-TD 族[99]	$\mathbb{E}_{\mu}[\delta\boldsymbol{e}]=0$	$\boldsymbol{A}_{\lambda}$	随机梯度下降	$\|\boldsymbol{\Pi}\boldsymbol{V}^{*}-\boldsymbol{V}^{*}\|/(1-\gamma(1-\lambda)/(1-\gamma\lambda))$[92]
	2016	ETD(λ)[82]	$\mathbb{E}_{\mu}[\delta\boldsymbol{e}]=0$	$\boldsymbol{\Phi}^{\top}\boldsymbol{M}(\boldsymbol{I}-\boldsymbol{P}^{\lambda,\pi})\boldsymbol{\Phi}$	随机梯度下降	$\|\boldsymbol{\Pi}\boldsymbol{V}^{*}-\boldsymbol{V}^{*}\|/(1-\sqrt{\gamma(1-\lambda)/(1-\lambda\gamma)})$[96]
压缩映射	2009	LSTD(λ)[88]	$\mathbb{E}_{\mu}[\delta\boldsymbol{e}]=0$	$\boldsymbol{A}_{\lambda}$	最小二乘	无
	2016	ETD(λ)[82]	$\mathbb{E}_{\mu}[\delta\boldsymbol{e}]=0$	$\boldsymbol{\Phi}^{\top}\boldsymbol{M}(\boldsymbol{I}-\boldsymbol{P}^{\lambda,\pi})\boldsymbol{\Phi}$	随机梯度下降	$\|\boldsymbol{\Pi}\boldsymbol{V}^{*}-\boldsymbol{V}^{*}\|/(1-\sqrt{\gamma(1-\lambda)/(1-\lambda\gamma)})$[96]

$$\boldsymbol{A}_{\mathrm{GTD}}=\begin{pmatrix}\sqrt{\eta}\boldsymbol{I} & \boldsymbol{A}_{\mathrm{off}}\\ -\boldsymbol{A}_{\mathrm{off}}^{\top} & 0\end{pmatrix} \tag{36}$$

2009 年，Sutton 等人在此基础上，通过最小化 $\mathrm{MSPBE}(\boldsymbol{\theta})=\|\boldsymbol{V}_{\boldsymbol{\theta}}-\boldsymbol{\Pi}\mathcal{T}\boldsymbol{V}_{\boldsymbol{\theta}}\|_{\boldsymbol{D}}^{2}=\mathbb{E}_{\mu}[\delta\boldsymbol{\phi}]^{\top}\mathbb{E}[\boldsymbol{\phi}_k\boldsymbol{\phi}_k^{\top}]^{-1}\mathbb{E}_{\mu}[\delta\boldsymbol{\phi}]$得到能保证收敛的 GTD2 和 TDC(TD with Gradient Correction)算法[38]. 其中，GTD2 算法收敛的核心要素为正定矩阵：

$$\boldsymbol{A}_{\mathrm{GTD2}}=\begin{pmatrix}\sqrt{\eta}\boldsymbol{C} & \boldsymbol{A}_{\mathrm{off}}\\ -\boldsymbol{A}_{\mathrm{off}}^{\top} & 0\end{pmatrix} \tag{37}$$

TDC 算法收敛的核心要素为正定矩阵 $\boldsymbol{A}_{\mathrm{off}}^{\top}\boldsymbol{C}^{-1}\boldsymbol{A}_{\mathrm{off}}$，$\boldsymbol{C}=\mathbb{E}[\boldsymbol{\phi}_k\boldsymbol{\phi}_k^{\top}]$. Bellman 残差与 MSPBE 的几何关系如图 2 所示，MSPBE 是 $\boldsymbol{V}_{\boldsymbol{\theta}}$ 与 $\mathcal{T}\boldsymbol{V}_{\boldsymbol{\theta}}$ 在 $\boldsymbol{\Phi},\boldsymbol{D}$ 平面投影的距离，因此可以优化到 0. 2010 年，Maei 等人从状态动作、Option 角度对 TDC 算法进行了扩展，提出了异策略 GQ(λ)算法[86]，该算法收敛到投影 λ-Bellman 不动点解 $\boldsymbol{Q}_{\boldsymbol{\theta}}=\boldsymbol{\Pi}\mathcal{T}_{\pi}^{\lambda\beta}\boldsymbol{Q}_{\boldsymbol{\theta}}$，也可以写成 $\mathbb{E}[\delta\boldsymbol{e}]=0$，其中 $\boldsymbol{e}_t=(1-\beta_t)\lambda_t\rho_t\boldsymbol{e}_{t-1}+\boldsymbol{\phi}_t$，重要性采样比例 $\rho_t=\dfrac{\boldsymbol{\pi}(s_t,a_t)}{\boldsymbol{b}(s_t,a_t)}$，$\beta_t:\boldsymbol{S}\to[0,1]$表示目标策略短暂终止执行的概率，$\lambda_t=\lambda(s_t)$，$\delta_t$是基于修正 Q 值函数的 TD 误差. 2010 年，Yu 通过 e-chains[87]证明了不动点 $\boldsymbol{\theta}=\boldsymbol{A}^{-1}\boldsymbol{b}$ 中 $\boldsymbol{A}$ 和 $\boldsymbol{b}$ 的收敛性，从而给出了异策略 LSTD(λ)算法在更一般条件下(对比[88])的收敛性证明[89]. 通过观察 TDC 算法收敛的核心要素，2013 年 Hackman 在硕士毕业论文中指出采用任意正定矩阵替换 $\boldsymbol{C}$，不影响正定性，也不影响收敛的不动点，但可以加速收敛速度[90]. 他提出了三种改进算法，其中前两种是 AB 算法和 Hybrid-GQ 算法[91]，把 $\boldsymbol{C}$ 替换为行为策略的正定矩阵 $\boldsymbol{A}_{\mathrm{on}}$，即 $\boldsymbol{A}_{\mathrm{off}}^{\top}\boldsymbol{A}_{\mathrm{on}}^{-1}(-\boldsymbol{A}_{\mathrm{off}}\boldsymbol{\theta}+\boldsymbol{b})$，第三种算法是 TDGQ 算法，其思想是当行为策略与目标策略的选择一致时采用 On-Policy 的 TD 进行学习，不同时则采用 Gradient TD 算法来更新，以此加速收敛[90]. 为了进一步提高收敛速度，2016 年 Liu 等人通过把目标函数 NEU 和 MSPBE 建模成鞍点问题：

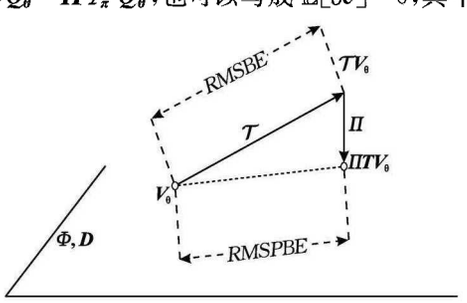

图 2　Bellman 残差[83]与 MSPBE[38]的几何关系

$$\min_{\boldsymbol{\theta}}\max_{y}\left(\boldsymbol{L}(\boldsymbol{\theta},y)=\langle\boldsymbol{b}-\boldsymbol{A}\boldsymbol{\theta},y\rangle-\frac{1}{2}\|y\|_{\boldsymbol{M}}^{2}\right) \tag{38}$$

通过近端梯度(Proximal Gradient)优化方法分别得到 GTD-MP(Mirror-Prox)和 GTD2-MP 算法[92,100-101]. 其中，NEU 中 $\boldsymbol{M}=\boldsymbol{I}$，MSPBE 中 $\boldsymbol{M}=\boldsymbol{C}=\mathbb{E}[\boldsymbol{\phi}_k\boldsymbol{\phi}_k^{\top}]$. 2017 年，Pan 等人考虑 MSPBE 的导数

$$-\frac{1}{2}\nabla\mathrm{MSPBE}(\boldsymbol{\theta})=\boldsymbol{A}^{\top}\boldsymbol{C}^{-1}\mathbb{E}_{\mu}[\delta\boldsymbol{e}] \tag{39}$$

其 Hessian 矩阵是 $\boldsymbol{H}=\boldsymbol{A}^{\top}\boldsymbol{C}^{-1}\boldsymbol{A}$，因此通过更新公式中加入 $\boldsymbol{H}^{-1}$ 获得了加速的 ATD(λ)算法[93]. 2017 年，吴等人提出了一般化斜投影的异策略 RTD[94]算法(Residual TD)，它们都收敛到一般化斜投影 Bellman 不动点解，即 $\boldsymbol{V}_{\theta}=\boldsymbol{\Pi}_{\boldsymbol{X}_w}\mathcal{T}\boldsymbol{V}_{\theta}$，写成期望形式为 $\mathbb{E}[\delta(\boldsymbol{\phi}-w\gamma\boldsymbol{\phi}')]=0$. 其中，斜投影为 $\boldsymbol{\Pi}_{\boldsymbol{X}_w}=\boldsymbol{\Pi}(\boldsymbol{\Phi}^{\top}(\boldsymbol{I}-w\gamma\boldsymbol{P})^{\top}\boldsymbol{D}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\top}(\boldsymbol{I}-w\gamma\boldsymbol{P})^{\top}\boldsymbol{D}$，算法收敛的核心要素是正定矩阵

$$\boldsymbol{A}_{\mathrm{RTD}}=\begin{pmatrix}\sqrt{\eta}(1-w)\boldsymbol{I} & 0 & (1-w)\boldsymbol{A}_{\mathrm{off}}\\ 0 & \sqrt{\eta}w\boldsymbol{I} & w\boldsymbol{A}_{\mathrm{rg}}\\ -(1-w)\boldsymbol{A}_{\mathrm{off}}^{\top} & -w\boldsymbol{A}_{\mathrm{rg}}^{\top} & 0\end{pmatrix} \tag{40}$$

2019 年，Song 等人采用基于递归最小二乘的岭回归求解目标函数 MSPBE，得到 RLSTDC 算法(RLSTD with Gradient Correction)，并综合多步回报得到 RC(λ)算法[95]. 其中，关键要素是

$$\begin{aligned}\mathbb{E}_{\mu}[\boldsymbol{\phi}(\boldsymbol{\phi}-\gamma\boldsymbol{\phi}')+\boldsymbol{I}]&=\mathbb{E}_{\mu}[\boldsymbol{\phi}(\boldsymbol{\phi}-\gamma\boldsymbol{\phi}')]+\epsilon\boldsymbol{I}\\&=\boldsymbol{A}_{\mathrm{off}}+\epsilon\boldsymbol{I}\end{aligned} \tag{41}$$

2020 年，Ghiassian 等人通过将正定矩阵 $\boldsymbol{C}$ 替换成 $\boldsymbol{C}+\beta\boldsymbol{I}$ 来更新第二个参数，得到 TDRC(TD with Regularized Corrections)算法，用参数 β 控制第二个参数的方差，使得它比 TDC 更快收敛[96].

分布矫正. 2001 年，Precup 等人在引入重要性采样率的基础上，从时间步 0 开始把所有重要性采样率进行连乘，强制把状态分布由 $\boldsymbol{d}_{\mu}$ 转成 $\boldsymbol{d}_{\pi}$，最终将 $\boldsymbol{A}_{\rho,\mathrm{off}}=\boldsymbol{\Phi}^{\top}\boldsymbol{D}_{\mu}(\boldsymbol{I}-\gamma\boldsymbol{P}^{\pi})\boldsymbol{\Phi}$ 转化成正定矩阵 $\boldsymbol{A}_{\mathrm{on}}=\boldsymbol{\Phi}^{\top}\boldsymbol{D}_{\pi}(\boldsymbol{I}-\gamma\boldsymbol{P}^{\pi})\boldsymbol{\Phi}$，从而在理论上保证了所提 IS-TD(0)算法的收敛性[97]. 经过状态分布的矫正以及综合多步回报，得到异策略 IS-TD(λ)算法，其收敛性结果与 TD(λ)相同. 由于连乘引起的超大方差，IS-TD(λ)算法的收敛速度受到了限制，在 Episode 内任意时刻对重要性采样率的连乘做重置，可以一定程度上降低算法的方差[97]. 此外，通过带权重的重要性采样率 WIS-TD(λ)，可以有效地降低方差，提高收敛速度[97]. 与 2001 年 Precup 等人将行为策略的状态分布 $\boldsymbol{d}_{\mu}$ 转化成目标策略的状态分布 $\boldsymbol{d}_{\pi}$ 不同，2014 年，Mahmood 等人从监督学习角度的目标函数出发，对比了重要性采样率和带权重的重要性采样率的异同之处，并据此提出了带权重重要性采样率的离线 WIS-LSTD(λ)算法，其收敛速度相比 2009 年和 2010 年 off-policy LSTD(λ)算法有显著提高[98]. 同样基于带权重的重要性采样率，2015 年，Mahmood 等人提出了 WIS-TD(λ)、WIS-GTD(λ)、WISTO-TD(λ)、U-TD(λ)、U-TO-TD(λ)等算法[99]. 2016 年，Sutton 通过引入权值 $\boldsymbol{f}^{\top}=\boldsymbol{d}_{\mu}^{\top}(\boldsymbol{I}-\gamma\boldsymbol{P}^{\pi})^{-1}$ 来抵消无法保证正定性的矩阵 $\boldsymbol{A}_{\rho,\mathrm{off}}=\boldsymbol{\Phi}^{\top}\boldsymbol{D}_{\mu}(\boldsymbol{I}-\gamma\boldsymbol{P}^{\pi})\boldsymbol{\Phi}$ 中的项 $(\boldsymbol{I}-\gamma\boldsymbol{P}^{\pi})$，由此构成了能满足收敛性的正定矩阵

$$\begin{aligned}\boldsymbol{A}&=\mathbb{E}_{\mu}[\boldsymbol{F}\rho\boldsymbol{\phi}(\boldsymbol{\phi}-\gamma\boldsymbol{\phi}')^{\top}]\\&=\boldsymbol{\Phi}^{\top}\boldsymbol{F}(\boldsymbol{I}-\gamma\boldsymbol{P}^{\pi})\boldsymbol{\Phi}\end{aligned} \tag{42}$$

所得到的 ETD(0)算法收敛到的不动点是方程 $\mathbb{E}_{\mu}[\boldsymbol{F}\rho\delta\boldsymbol{\phi}]=0$ 的解[82]. 其中，矩阵 $\boldsymbol{F}$ 是由 $\boldsymbol{f}$ 向量为对角线构成的对角矩阵，$\boldsymbol{f}(s)\doteq\boldsymbol{d}_{\mu}(s)\lim\limits_{t\to\infty}\mathbb{E}_{\mu}[\boldsymbol{F}_t\mid S_t=s]$，$\boldsymbol{F}_t\doteq\gamma\rho_{t-1}\boldsymbol{F}_{t-1}+1$，$\boldsymbol{F}_0=1$. 经理论分析 ETD 的算子是 $\sqrt{\gamma(1-\boldsymbol{\kappa})}$-压缩映射，其中 $\boldsymbol{\kappa}=\min\limits_{s}\frac{\boldsymbol{d}_{\mu}(s)}{\boldsymbol{f}(s)}$. 根据压缩映射原理，可以得到 ETD 算法的误差界是 $\frac{\|\boldsymbol{\Pi}\boldsymbol{V}^{*}-\boldsymbol{V}^{*}\|}{1-\sqrt{\gamma(1-\boldsymbol{\kappa})}}$[102]. 同时，综合多步回报，得到能保证收敛的 ETD(λ)算法[82]. ETD(λ)算法收敛的关键要素是正定矩阵 $\boldsymbol{\Phi}^{\top}\boldsymbol{M}(\boldsymbol{I}-\boldsymbol{P}^{\lambda,\pi})\boldsymbol{\Phi}$，收敛到的不动点是方程 $\mathbb{E}_{\mu}[\delta\boldsymbol{e}]=0$ 的解，其中，$\boldsymbol{e}_t=\rho_t(\gamma_t\lambda_t\boldsymbol{e}_{t-1}+\boldsymbol{M}_t\boldsymbol{\phi}_t)$，$\boldsymbol{M}_t=\lambda_t i(s_t)+(1-\lambda_t)\boldsymbol{F}_t$，$\boldsymbol{F}_t=\rho_{t-1}\gamma_t\boldsymbol{F}_{t-1}+i(s_t)$. 经理论分析 ETD(λ)算法的误差界是 $\frac{\|\boldsymbol{\Pi}\boldsymbol{V}^{*}-\boldsymbol{V}^{*}\|}{1-\sqrt{\gamma(1-\lambda)/(1-\lambda\gamma)}}$[102].

压缩映射. 2009 年，Bertsekas 等人指出异策略 LSTD(λ)算法在 $\lambda\gamma\max\limits_{s,s'}\frac{\boldsymbol{p}(s,s')}{\boldsymbol{q}(s,s')}<1$ 的条件下，即 γ 较小时 $\boldsymbol{A}_{\lambda}$ 能保证正定，使得 $\boldsymbol{\Pi}\mathcal{T}^{\lambda}$ 成为一个压缩映射，从而得到算法的收敛性保证[88]. 2015 年，Hallak 等人证明 ETD(λ)算法满足压缩映射，并给出了误差界[102].

工程上的解决方案. 异策略强化学习的收敛性问题是由高度抽象的反例广为人知的，例如两状态问题、Baird 的七星反例、Tsitsiklis 反例[37]. 然而，上述可以保证收敛的算法通常有更多的设置，例如需要调节两个学习率参数等，在解决实际需求时，往往不被工程师选用. 常见的 Q-learning 依然是大家最乐于使用的算法. 针对异策略的发散或收敛速度慢等问题，下面枚举一些工程上的解决方案：(1) 让行为策略和目标策略尽量相似，即减轻异策略的程度，如基于优先权的经验回放[41]等；(2) 在异步架构中频繁地做同步操作[103].

其它. 此外，还有一些有参考价值的工作，尽管他们没有显式地从收敛性角度对异策略算法进行深入探讨. 2014 年，van Hasselt 等人将 True on-policy TD(λ)算法推广到 Off-Policy True TD(λ)算法[104].

2014 年，Sutton 等人进一步推广到控制问题，提出了 PQ(λ)算法[105]. 2017 年，Devraj 等人从方差约减的角度出发，根据 Newton-Raphson 方法设计了最优渐进方差的 Zap Q(λ)算法，其算法收敛性的要素是通过构造假设的正定矩阵[106]. 2017 年，Mahmood 等人提出了多步TD反馈、不需要重要性采样率的ABQ算法[107]，$\boldsymbol{e}_t=\lambda_t\gamma_t\rho_t\boldsymbol{e}_{t-1}+\boldsymbol{\phi}_t$，其中 $\lambda_t=\beta\dfrac{\min\{1,\rho_{t-1}\}}{\rho_{t-1}}$. 2018 年，Yu 等人提出了基于广义 Bellman 等式 LSTD(λ)[108]，其收敛值为 $\mathbb{E}[\delta\boldsymbol{e}]=0$，其中，$\delta_t=\rho_t(\boldsymbol{R}_t+\gamma_{t+1}\boldsymbol{V}_{\boldsymbol{\theta}}(s_{t+1})-\boldsymbol{V}_{\boldsymbol{\theta}}(s_t))$，$\boldsymbol{e}_t=\lambda_t\boldsymbol{e}_{t-1}+\boldsymbol{\phi}_t$，$\lambda_t=\min\left\{\gamma_t\rho_{t-1},\dfrac{\boldsymbol{C}_{s_{t-1}s_t}}{\|\boldsymbol{e}_{t-1}\|_2}\right\}$. 2018 年，Dalal 等人对 GTD、GTD2、TDC 等双时间尺度随机逼近算法(Two Timescale Stochastic Approximation)做了有限样本分析[109]. 2019 年，Gupta 等人对双时间尺度随机逼近算法做了固定学习率的有限时间分析，并提出了一个自适应学习率机制，可以有效地提高收敛速度[110]. 2020 年，Dalal 等人对双时间尺度随机逼近算法的收敛率做了进一步分析[111]. 2021 年，Xu 等人对双时间尺度逼近算法的样本复杂度进行了分析[112].

线性值函数估计采用状态或状态动作对的特征值与权重参数的线性和，更新公式的期望可以将权重参数提出来，从而有利于收敛性分析. 因此，大家可以看到线性值函数估计始终是强化学习中最活跃的理论研究领域. 当专家给出比较好的特征函数时，如 2048 游戏的 N-Tuple 特征[113]，Tetris 游戏的 DT 特征[114]，线性值函数估计的效果是非常高效的. 然而，实际任务如围棋、斗地主等特别复杂，专家很难抽象出好的特征函数. 幸运的是，随着深度学习的兴起，非线性值函数估计得到了越来越多的关注.

5　非线性值强化学习

在线性值函数估计情况下，近年来研究者已提出了若干保证稳定性的算法. 然而采用以神经网络为代表的非线性值函数时，稳定性仍是一个巨大的挑战.

2018 年，Sutton 等人指出死亡三元组[37]：函数估计(Function Approximation)、自举(Bootstrapping)、异策略学习(Off-Policy Learning)，它们的组合容易导致算法的发散. 2018 年，van Hasselt 等人围绕 DQN 及其变种对死亡三元组做了实证检验，验证了六个假设[115]：(1) 深度卷积网络与 Q-learning 结合时不容易无限发散；(2) 目标网络有助于缓解发散；(3) 过估计的矫正有助于缓解发散；(4) 多步回报有助于缓解发散；(5) 大且灵活的网络有助于缓解发散；(6) 差异(Off-Policy)越大越容易发散. 然而，van Hasselt 同时指出这些尝试并不能真正解决死亡三元组.

5.1　非线性值强化学习的稳定性原理

推论 2.　非线性系统的稳定性. 假设 $g(\boldsymbol{\theta}(t))$ 是关于 $\boldsymbol{\theta}(t)$ 的凸函数且二阶导数 $g''(\boldsymbol{\theta}(t))\neq0$，则系统 $\dot{\boldsymbol{\theta}}(t)=-g'(\boldsymbol{\theta}(t))$ 渐进稳定.

证明.　设 Lyapunov 函数

$$\boldsymbol{L}(\boldsymbol{\theta}(t))=(-g'(\boldsymbol{\theta}(t)))^{\top}(-g'(\boldsymbol{\theta}(t)))/2,$$

容易验证：

(1) $\boldsymbol{L}(\boldsymbol{\theta}(t))\geqslant0$，仅在 $g'(\boldsymbol{\theta}(t))=0$ 处等号成立；

(2) 一阶导数

$$\begin{aligned}\dot{\boldsymbol{L}}(\boldsymbol{\theta}(t))&=(-g'(\boldsymbol{\theta}(t)))^{\top}(-g''(\boldsymbol{\theta}(t)))\dot{\boldsymbol{\theta}}(t)\\&=(-g'(\boldsymbol{\theta}(t)))^{\top}(-g''(\boldsymbol{\theta}(t)))(-g'(\boldsymbol{\theta}(t)))\\&=-(g'(\boldsymbol{\theta}(t)))^{\top}g''(\boldsymbol{\theta}(t))(g'(\boldsymbol{\theta}(t)))\end{aligned}\tag{43}$$

由于 $g(\boldsymbol{\theta}(t))$ 是关于 $\boldsymbol{\theta}(t)$ 的凸函数且二阶导数 $g''(\boldsymbol{\theta}(t))\neq0$，则 $-g''(\boldsymbol{\theta}(t))<0$，即 $\boldsymbol{L}(\boldsymbol{\theta}(t))<0$. 根据 Lyapunov 第二判定法，该系统渐进稳定，并收敛至 $g'(\boldsymbol{\theta}(t))=0$.　　证毕.

注意，凸函数是收敛性的强保证. 然而，非线性系统的凸性是很难保证的，并且通常是非凸的.

5.2　基于非线性值函数的强化学习

2016 年，Yang 等人提出了两阶段方法，第一阶段采用针对任务的监督学习预训练方法学习奖赏函数，第二阶段将学到的神经网络修改并扩展成深度神经网络进行强化学习. 实验结果表明该方法有利于基于神经网络的强化学习算法的收敛性，但该方法缺乏理论保障，并且可操作性和拓展性都值得商榷[116].

投影法. 为了克服神经网络的非凸性，研究者设计了紧凸集，将值投影至紧凸集则可满足收敛性. 2009 年，Maei 等人针对非线性模型，采用最小化带参投影均分误差则得到非线性 GTD2 和非线性 TDC 算法，其目标函数是非线性投影 Bellman 误差，

$$\begin{aligned}&\text{Nonlinear MSPBE}(\boldsymbol{\theta})=\\&\mathbb{E}[\delta(\boldsymbol{\theta})\nabla\boldsymbol{V}_{\boldsymbol{\theta}}(s)]^{\top}\mathbb{E}[\nabla\boldsymbol{V}_{\boldsymbol{\theta}}(s)\nabla\boldsymbol{V}_{\boldsymbol{\theta}}(s)^{\top}]^{-1}\mathbb{E}[\delta(\boldsymbol{\theta})\nabla\boldsymbol{V}_{\boldsymbol{\theta}}(s)]\end{aligned}\tag{44}$$

利用一个带参 $\boldsymbol{\theta}$ 的投影 $\boldsymbol{\Pi}_{\boldsymbol{\theta}}=\boldsymbol{\Phi}_{\boldsymbol{\theta}}(\boldsymbol{\Phi}_{\boldsymbol{\theta}}^{\top}\boldsymbol{D}\boldsymbol{\Phi}_{\boldsymbol{\theta}})^{-1}\boldsymbol{\Phi}_{\boldsymbol{\theta}}^{\top}\boldsymbol{D}$，可以证明非线性 GTD2 和非线性 TDC 收敛到局部最优解[117]. 2014 年，Lee 等人提出了保证收敛的两层神经网络的 Nonlinear Greedy-GQ 算法[118]. 2019

年，Qu 等人提出了分布式非线性 GTD 类算法[119]. 2019 年，Cai 等人指出非线性情况下 MSPBE 是非凸的，所以会陷入局部最优或者鞍点，提出了基于投影的 NTD 算法(Neural TD)，该算法可以保证全局收敛性[120]. 2019 年，Wang 等人通过将策略评估问题转化为非凸的原始对偶有限和优化(Finite-Sum Optimization)问题，其原始子问题是非凸的，对偶子问题是强凹的，采用方差减小的单时间尺度原始对偶梯度算法能获得更快的收敛速度[121]. 2020 年，Wai 等人将 MSBE 的最小化问题转化成极小极大的优化问题，提出了异策略 NeuralGTD 算法[122]. 2020 年，Zhang 等人详细对比了 TD 和 BR 方法的优势和劣势，尤其是 Residual 方法在收敛性方面的优势，并针对 DDPG 方法提出了改进的双向残差 DDPG(Bi-Res DDPG)算法[123]. 2021 年，Yang 等人采用随机混合优化的方式对 DQN 进行改进，实验表明去除了目标网络后算法仍可以稳定地学习[124]. 2021 年，Cao 等人将分布矫正思想引入投影算法，提出异策略评估 ETD-LVC 算法(Emphatic TD with Lower-Variance Gradient Correction)，该算法在线性和非线性函数估计上均能保证收敛[125]. 2021 年，Wang 等人对非线性两个时间尺度 TDC 算法做了非渐进性分析[126].

浅层更新法. 另一个思路是将神经网络看成一种表征学习，把最后一层的线性函数权重用强化学习来更新，由此形成了一系列浅层更新方法. 2017 年，Levine 等人提出了针对深度强化学习的浅层更新方法：采用最小二乘更新最后一层权重的最小二乘 DQN 算法(LSDQN)[127]. 2020 年，Ghosh 等人针对利用辅助任务进行表征学习，从辅助任务的目标函数角度进一步分析了异策略强化学习算法的稳定性，指出采用 Schur 基函数在一般情况下都能保证收敛性，在固定的奖赏函数前提下 Krylov 子空间的正交基更稳定[128].

几何视角. 2020 年，Brandfonbrener 等人基于非线性函数估计同策略 TD 学习算法，从发散的几何螺旋结构角度出发，指出了一种保证收敛的方案和一种增强收敛性的方案，即采用平滑且齐次的非线性值函数可以保证收敛性，根据由环境和策略生成的状态转移函数，定义该函数的对称程度，并推导出多步 TD 算法更趋向于稳定[129].

核损失函数法. 2019 年，Feng 等人提出了核损失函数[130]：

$$\boldsymbol{L}_K(\boldsymbol{V})=\mathbb{E}_{s_i,s_j\sim\mu}[\boldsymbol{K}(s_i,s_j)\mathcal{R}\boldsymbol{V}(s_i)\mathcal{R}\boldsymbol{V}(s_j)] \tag{45}$$

其中，$\mathcal{R}\boldsymbol{V}=\mathcal{T}\boldsymbol{V}-\boldsymbol{V}$，核函数 $\boldsymbol{K}(\cdot,\cdot)$ 是整体严格正定的，如高斯核. 因此，目标函数是明确定义的，问题转化为传统的监督学习. 对于批量的经验数据 $\{(s_i,a_i,r_i,s_i')\}_{1\le i\le n}$，损失的估计是

$$\hat{\boldsymbol{L}}_K(\boldsymbol{V}_{\boldsymbol{\theta}})=\frac{1}{n^2}\sum_{1\le i,j\le n}\boldsymbol{K}(s_i,s_j)\hat{\mathcal{R}}\boldsymbol{V}_{\boldsymbol{\theta}}(s_i)\hat{\mathcal{R}}\boldsymbol{V}_{\boldsymbol{\theta}}(s_j) \tag{46}$$

其中，$\hat{\mathcal{R}}\boldsymbol{V}_{\boldsymbol{\theta}}(s_i)=r_i+\gamma\boldsymbol{V}_{\boldsymbol{\theta}}(s_i')-\boldsymbol{V}_{\boldsymbol{\theta}}(s_i)$. 其梯度是

$$\nabla\hat{\boldsymbol{L}}_K(\boldsymbol{V}_{\boldsymbol{\theta}}):=\frac{2}{n^2}\sum_{1\le i,j\le n}\boldsymbol{K}(s_i,s_j)\hat{\mathcal{R}}\boldsymbol{V}_{\boldsymbol{\theta}}(s_i)\nabla\hat{\mathcal{R}}\boldsymbol{V}_{\boldsymbol{\theta}}(s_j) \tag{47}$$

有趣的是，用线性函数估计时，式(47)等价于 NEU($\boldsymbol{\theta}$). 2020 年，Feng 等人提出了核损失函数在变分框架下的紧置信区间的应用[131].

近年来，深度强化学习算法得到了广泛关注，取得了一定的理论和应用成果. 然而，相比基于表格值函数和线性值函数的强化学习算法，非线性值函数强化学习算法的渐进收敛性、非渐进收敛性、有限样本分析等理论仍有待建立和进一步完善.

6 非参化值强化学习

非参优化方法无需在训练前确定参数的个数，具备灵活的表达能力，有望获得更好的预测性能. 因此，非参值函数估计是强化学习的重要分支之一. 非参化值强化学习算法在值函数的结构和参数个数确定后，算法的稳定性与参数化值强化学习的稳定性相同，在此不做赘述. 本章从核方法、正则化和决策树的角度分别介绍非参值函数估计的研究进展.

6.1 核方法

2002 年，Ormoneit 等人首次将该方法引入强化学习[132]. 2003 年，Engel 等人将值函数建模成高斯过程，并结合在线核稀疏化提出了 GPTD 算法(Gaussian Process Temporal Differences)[133]. 2007 年，Xu 等人将核方法用到 LSPI 算法得到 KLSPI 算法(Kernelized LSPI)，其中核字典的构建选择采用计算复杂度为 $O(n^2)$ 的近似线性依赖法 ALD(Approximate Linear Dependence)[134]. 2010 年，Geist 等人基于值函数的高斯过程建模，采用了无导数的统计线性化方案和递归最小二乘法，提出了 KTD(Kalman Temporal Difference)算法[135]. 2011 年，Kroemer 等人提出了非参动态规划 NPDP 方法(Non-Parametric Dynamic Programming)[136]. 2013 年，Chen 等人采用计算复杂度为 $O(n)$ 的新颖标准 NC(Novelty Criterion)方法在线构建核字典，并结

合 TD 算法得到 OSKTD 算法(Online Selective Kernel-based TD)[137]. 2016 年,Song 等人提出了基于核方法的最小二乘梯度矫正 TD 算法(Kernel-based Least Square Temporal Difference with Gradient Correction,KLS-TDC),其中字典构造采用了 ALD 算法. 2016 年,Barreto 等人根据随机分解技巧对 KBRL 做了进一步改良,使得改进后的抽象 MDP 在保持原 MDP 特性的前提下更加小,从而大大减小了计算代价[138]. 2020 年,Koppel 等人提出了高效内存的核梯度时序差分学习 PKGTD 算法(Parsimonious Kernel Gradient Temporal Difference),其中字典的构造采用了核正交匹配跟踪(Kernel Orthogonal Matching Pursuit)方法[139]. 2022 年,Yang 等人提出了基于注意力核方法的 OAKTD 算法(Online Attentive Kernel-based Temporal Difference Learning)[140].

6.2　正则化

机器学习中正则化方法有利于处理过拟合的现象. 由于一些改进方法从特征选择等入手,不能直接列出不动点和正定矩阵. 2009 年,Kolter 等人通过 l_1 正则化,并对 MSPBE 目标函数做最小化,从而得到了特征选择的 LARS-TD 算法(Least Angle Regression)[141]. 其中,LARS-TD 算法的目标函数是

$$\min_{\boldsymbol{\omega}} \|\boldsymbol{A}_{\text{on}}\boldsymbol{\omega}-\boldsymbol{b}\|^2+\lambda\|\boldsymbol{\omega}\|_1 \tag{48}$$

为了方便对比,该目标函数还可以写成

$$\begin{cases}\boldsymbol{\omega}_{\boldsymbol{\theta}}=\arg\min\limits_{\boldsymbol{\omega}\in\mathbb{R}^m}\|\boldsymbol{R}+\gamma\boldsymbol{\Phi}'\boldsymbol{\theta}-\boldsymbol{\Phi}\boldsymbol{\omega}\|^2+\lambda\|\boldsymbol{\omega}\|_1\\ \boldsymbol{\theta}^*=\arg\min\limits_{\boldsymbol{\theta}\in\mathbb{R}^m}\|\boldsymbol{\Phi}'\boldsymbol{\omega}_{\boldsymbol{\theta}}-\boldsymbol{\Phi}\boldsymbol{\theta}\|^2+\lambda\|\boldsymbol{\theta}\|_1\end{cases} \tag{49}$$

2011 年,Geist 等人针对 LARS-TD 做了修改,将正则项用于残差,得到 l_1-PBR 算法(l_1-Penalty Projection of Bellman Residual)[142]. 其目标函数如下:

$$\begin{cases}\boldsymbol{\omega}_{\boldsymbol{\theta}}=\arg\min\limits_{\boldsymbol{\omega}\in\mathbb{R}^m}\|\boldsymbol{R}+\gamma\boldsymbol{\Phi}'\boldsymbol{\theta}-\boldsymbol{\Phi}\boldsymbol{\omega}\|^2\\ \boldsymbol{\theta}^*=\arg\min\limits_{\boldsymbol{\theta}\in\mathbb{R}^m}\|\boldsymbol{\Phi}'\boldsymbol{\omega}_{\boldsymbol{\theta}}-\boldsymbol{\Phi}\boldsymbol{\theta}\|^2+\lambda\|\boldsymbol{\theta}\|_1\end{cases} \tag{50}$$

2011 年,Liu 等人提出了基于随机投影的一般化斜投影方法,该方法可以有效地将高维特征压缩到低维[143]. 2013 年,Nguyen 等人提出了基于模型的在线特征选择强化学习 loreRL 算法(RL with Regularized Logistic Regression)[144]. 2016 年,Gehring 等人针对低秩假设,对矩阵做奇异值分解,得到更快的增量式低秩 LSTD(λ)算法(increment Allow-Rank LSTD(λ))[145]. 2016 年,Li 等人提出了基于 l_1 正则化特征选择的 LS-TDC 算法:LARS-TDC 算法(Least Angle Regression Squares TDC)[146]. 2016 年,Farahmand 等人采用正则化方法对最小二乘和贝尔曼残差最小化进行了扩展,得到基于正则化的近似策略迭代算法 REG-LSPI 算法(Regularized Least Squares Policy Improvement)和 REG-BRM 算法(Regularized Bellman Residual Minimization),对 REG-LSPI 做了统计分析,并提供了策略估计的误差界和与最优策略的性能差,指出策略评估的误差界与样本量是一种极小极大最优关系[147]. 2020 年,Song 等人提出了一种基于 l_1 正则化和近端交替方向乘子(Alternating Direction Method of Multipliers with Proximal Operator)最小化 MSPBE 的递归最小二乘算法,l_1-RC 算法(Recursive Correction)[148]. 2020 年,Amit 等人展示了减小折扣因子与在损失项中添加正则化的等价性[149]. 2020 年,Li 等人通过正交匹配跟踪(Orthogonal Matching Pursuit)对 LSTDC 进行特征选择的改进得到 OMP-TDC 算法[150]. 2021 年,Hao 等人采用稀疏特征选择的 Lasso FQI(Fitted Q Iteration)算法使得批量强化学习算法具有更高的样本效率[151]. 2021 年,Song 等人通过交替进行随机梯度下降和对偶平均方法得到在线稀疏 TD 学习算法[152].

6.3　决策树

采用一阶逻辑表示状态、动作,可以描述复杂关系,具有更丰富的表达能力,是缓解维度灾难的可行途径之一.

2001 年 Džeroski 等人首次提出基于归纳逻辑编程(Inductive Logic Programming)的关系强化学习 RRL 算法(Relational Reinforcement Learning),其中 Q-tree 是通过逻辑回归树(Logical Regression Tree)表示的 Q 值函数[153]. 2003 年,Driessens 等人为增强 RRL 算法的鲁棒性提出了基于增量关系实例的回归(Incremental Relational Instance based Regression)算法[154]. 2004 年,Driessens 等人针对关系 MDP 中的稀疏奖励问题采用向导对关系强化学习进行加速学习[155]. 2006 年,Driessens 等人提出了基于图核协方差函数的高斯过程对 Q 值进行逼近[156]. 2009 年,Sanner 等人提出了一阶 MDP 的实用求解技术,他们尝试在关系级别降低计算复杂度,以生成与域无关的策略[157].

相较于深度强化学习的研究热度,非参化值强化学习的研究显得有些停滞. 但这并不意味着非参化值强化学习的灭绝. 深度强化学习领域最受人诟病的是黑盒模型缺乏可解释性. 非参化值函数研究领域中,核方法直接对状态进行抽象、正则化方法采

用低秩假设、决策树具有丰富的表达能力，这都为强化学习的可解释性提供了基础. 2021 年，Liu 等人通过对象表示的决策树对深度强化学习进行解释，提出了基于表示和模拟的 RAMi 框架(Represent And Mimic)[158]. 我们相信，可解释的强化学习将成为下一个研究热点，非参化值函数仍会有它的一席之地.

7 不动点偏差和方差控制视角下强化学习算法改进

在特征已给定并构造了不动点的情况下，我们关心的是：

(1)算法能否求解到这个不动点？

(2)算法能否快速地收敛？

这两点对应于不动点的偏差问题和求解算法的方差问题. 偏差导致所求结果不准确，方差导致学习效率低下、收敛速度慢、甚至不收敛. 接下来，我们从偏差和方差控制的角度详细阐述各类方法的思想. 根据解的分析，不失一般性，我们以求解 $\mathbb{E}[\delta e]$ 为例，仔细分析已有偏差和方差的来源.

7.1 由于相关性引起的偏差

样本在时序上的相关性. 机器学习通常假设样本是独立同分布的，而强化学习的过程是 MDP 的一个决策轨迹，样本之间在时间序列上是相关的，这就导致了偏差. 经验回放(Experience Replay)采用各种方式批量更新经验池中的样本，更高效地利用了样本，同时打破了相关性，因此可以有效地减小偏差. 这在深度神经网络的训练中尤为奏效，如经典的经验回放[54]、基于优先级的经验回放[41]、基于分层的经验回放[159]、事后经验回放[160-161]、基于分布匹配的选择性经验回放[162]、基于 agent 策略与 replay 策略交互优化的经验回放[163]、竞争型经验回放[164]、基于记忆和遗忘的 ReF-ER 经验回放[165]、基于 λ 回报调和的经验回放[166]、基于示例和交互的动态经验回放[167]、基于注意力的经验回放[168]以及经验回放的要素分析[169]. 经验回放的优点在于它将强化学习容易陷入局部最优中解脱出来[170]，缺点就是它是一种离线学习方式，需要批量更新，增加了计算复杂度，同时由于更新的策略与当前行为策略不一致，加剧策略的差异(Off-Policy)程度，可能导致算法收敛速度慢甚至不收敛.

样本及后继样本在特征空间上的相关性. 泛化性能指的是函数估计器在一个样本上学到的知识能传播到其它样本. 因此，即使样本相互之间是独立的，样本在特征空间上可能是相关的，例如不同样本之间如果共享相同的参数将引入偏差，导致灾难性遗忘和干扰. 解决的办法是对状态进行稀疏表示，使得不同的状态更新的参数尽量不同，如显式的稀疏表达 Tile 编码[171]、N-tuple 网络[172]、ReLU 激活函数[173]、随机出局 Dropout[174]、k 稀疏表示[175]、胜者全拿[176]、选择性核函数[137]、分布正则化稀疏表示[177]、从低维映射到高维就可以减轻神经网络学习的干扰[178]以及时序差分学习中泛化与干扰的关系分析[179]等.

期望乘积的相关性. 2008 年和 2009 年，Sutton 等人提出了 Gradient TD 算法族如 GTD、GTD2、TDC 以及后期的 GTD2(λ)，GQ(λ)等算法. 其基本思想是最小化目标函数

$$\boldsymbol{J}(\boldsymbol{\theta})=\|\mathbb{E}[\delta\boldsymbol{\phi}]\|^2_{\boldsymbol{M}^{-1}}=\|\boldsymbol{\Phi}^{\top}\boldsymbol{D}(\boldsymbol{T}\boldsymbol{V}_{\boldsymbol{\theta}}-\boldsymbol{V}_{\boldsymbol{\theta}})\|^2_{\boldsymbol{M}^{-1}} \tag{51}$$

其中，NEU 中 $\boldsymbol{M}=\boldsymbol{I}$，MSPBE 中 $\boldsymbol{M}=\boldsymbol{C}$. 以 NEU 为例，$-\frac{1}{2}\nabla\mathrm{NEU}(\boldsymbol{\theta})=\mathbb{E}[\boldsymbol{\phi}(\boldsymbol{\phi}-\gamma\boldsymbol{\phi}')^{\top}]^{\top}\mathbb{E}[\delta\boldsymbol{\phi}]$. 这里涉及到两个期望的乘积，如果只做一次采样，则带来估计的偏差. 而采用另一个变量去估计 $\mathbb{E}[\delta\boldsymbol{\phi}]$，则不是真正的随机梯度下降算法. 因此，2015 年、2016 年、2018 年 Liu 等人提出了将目标函数转化成极大极小问题，从而转成了可以用随机梯度下降方法求解的问题，并完成了有限样本复杂度分析[92,100-101].

乘积中后继样本之间的相关性. 在前面求解不动点时，我们提到了 BR 方法，它的目标是 $\mathbb{E}[\delta(\boldsymbol{\phi}-\gamma\boldsymbol{\phi}')]$，这里 $\delta=r+\gamma\boldsymbol{V}(s')-\boldsymbol{V}(s)$，$\boldsymbol{\phi}=\boldsymbol{\phi}(s)$，$\boldsymbol{\phi}'=\boldsymbol{\phi}(s')$可以看到在求解期望时，同时出现了两个后继状态 s'. 在非确定性环境中，如果我们采用相同的 s' 即只做一次采样，那么就引起了期望中乘积的相关性，出现了求解的偏差. 常见的解决方案是做两次采样，消除后继状态的相关性[37,83]. 但同时，两次采样增加了复杂度，并且依赖于模型或规划，减少了适用场景. 改进的方法是从将来借用动态性[180]以获得无偏估计.

期望算子与最大化算子之间的相关性. 观察 Bellman 最优等式 $\boldsymbol{V}^*(s)=\max\limits_{a\in\boldsymbol{A}}\mathbb{E}[r+\gamma\boldsymbol{V}^*(s')]$. 随机梯度下降算法每次只更新一个样本，即 $\max\limits_{a\in\boldsymbol{A}}[r+\gamma\boldsymbol{V}^*(s')]$. 根据已有的理论分析方法，对随机梯度下降算法的分析采用求期望的方式. 那么就是先求 max 再求 $\mathbb{E}$，这与最优 Bellman 等式先求 $\mathbb{E}$ 再求 max 正好相反，引起了偏差. 根据 Jensen 不等式，这个偏差是过大的方向，因此是过估计. 改进有 Double Q[47]、Double-DQN[181]、Averaged-DQN[182]、Target-

TD算法[183]、基于MelloMax算子[184]的MellowMax-DQN[185]、基于SoftMax算法的SoftMaxDQN[186]以及基于Rankmax的软贪心方法[187].

7.2 期望过程 $\mathbb{E}[\cdot]$ 引入的方差

随机梯度下降算法. 每次只更新一个样本，可以更加迅速地进行调整，它的计算复杂度很低，是在线学习的不二选择. 尽管在期望条件下，梯度是往目标方向移动的，但在不同样本上更新的方向实际上是各不相同的. 因此，随机梯度下降算法对比批量更新方法有更大的方差[65]. 改进的方法有：基于底线的策略梯度算法[188]、蒙特卡洛树搜索中的方差约减[189]、基于方差约减的随机梯度下降[190]、非线性平滑函数估计策略评估的方差约减[121]、中心化CTD[191,192]、直接控制方差的VTD[193].

期望集合的大小. 直观上，参与求期望的集合大小越大，则方差更容易偏大. 通过对Tetris动作空间进行多个属性的约简，使得Tetris游戏动作的平均数从17下降到4个左右，有效地提升了策略迭代的学习效果[194].

训练后期奖赏坠崖引起的学习震荡. 深度强化学习训练中经常会遇到奖赏突然坠崖式地剧烈震荡，使得学习曲线往下掉，采用奖赏塑形可以有效地减轻这种震荡，如面向方差的奖赏平滑方法(Variance Aware Rewards，VAR)[195].

针对最大化算子的探索与利用权衡. 由于状态动作空间大，采用随机梯度下降的过程中，无法保证当前的动作是真的最优动作. 需要不断探索，找到真正的最优动作. 过多的探索会引入较大的方差. 然而，这对策略的提升必不可少，否则即使很快收敛了，也只会陷入局部最优策略. 因此，需要权衡探索和利用，做高效的探索势在必行. 常用的探索方法有ε-greedy、SoftMax[36]、基于计数的方法[196-197]、乐观初始化[198-200]、基于上置信区间的探索(如UCRL[201]、UCRL2[202]、UCRL-γ[203])、基于信息增益的探索[204]、基于汤普森采样概率匹配探索[205]等.

Off-Policy中的策略差异. 强化学习的目的是求解最优策略，不得不进行探索. 因此，我们往往采用了Off-Policy方法，即采用两个策略，一个策略作为目标策略，一个策略作为行为策略. 由于这两个策略的不同导致了很大的方差. 改进的方法有基于重要性采样的GQ(λ)算法[86]和LSTD(λ)算法[89]、带权重的重要性采样WIS-LSTD(λ)[98]和WIS-TD(λ)、WIS-GTD(λ)、WIS-TO-TD(λ)、U-TD(λ)、U-TO-TD(λ)等算法[99]、ETD(λ)[82]、截断带权重的重要性采样的Retrace(λ)算法[52]以及基于广义Bellman等式LSTD(λ)[108].

7.3 偏差与方差的权衡

偏差-方差窘境. 在函数估计中，对比TD算法与Monte Carlo(MC)算法，我们知道MC是无偏估计，但方差很大，而TD算法是有偏估计、方差较小[37]. 因此，TD(λ)、LSTD(λ)、多步LSTD(λ)[206]等算法通过λ权衡偏差与方差. 针对单步DQN的改进有基于回报的R-DQN[207]等.

TD误差 δ. 在随机梯度算法中，e 指的是梯度更新的方向，δ 是更新的长度. 更新量的大小很大程度上影响着方差，如自动调节学习步长的TCL(Temporal Coherence Learning)[113,208-210]. 在策略梯度算法中，基于状态的Baseline $b(s)$ 或者常数 b 能使得策略梯度算法更快收敛，在Actor-Critic学习架构中，Critic的值函数与Baseline同样起到了减小方差的作用[37,188]，预先定义的常值Baseline可以在值迭代算法中显著提高学习速率，如在Tetris游戏中的成功应用[211]以及多臂老虎机问题[212]. 这些方法在控制方差的同时产生了偏差. 幸运的是这些改变并不影响最优策略[213].

给定不动点解，给定特征后，学习算法的目标就是保证求到该解，并且快速地求到. 这涉及到迭代算法的收敛性、收敛速度问题. 其本质在于最小化学习算法的方差. 学习算法的方差有多种来源. 目前已有的研究并未对方差的来源全面考虑，此外针对特定问题，如何抓住并控制该算法的主要方差来源是关键.

8 总结与展望

本文从不动点视角，总结了基于表格值和值函数估计强化学习算法的解和稳定性，并从不动点解的偏差和方差控制视角分析了现有改进算法的思想. 尽管基于值函数的强化学习已取得了丰硕的成果，然而在不动点视角下强化学习还有许多有待解决的问题，总结如下：

策略不动点. 基于值函数的强化学习由两个算子构成：改进T算子和投影$\boldsymbol{\Pi}$算子，值函数是公式$\boldsymbol{V}_\theta=\boldsymbol{\Pi}T\boldsymbol{V}_\theta$的不动点，可以由值迭代或策略迭代算法求解. 传统的观点认为，基于策略梯度的强化学习是对奖赏总和期望的梯度上升方法. 令人惊喜的是，2020年，Ghosh等人从不动点视角总结了基于策略梯度的强化学习算法如REINFORCE[214]、PPO[215]、

MPO[216]等，指出策略梯度强化学习也可以看成由两个算子构成：改进 $\mathcal{I}$ 算子和投影 $\mathcal{P}$ 算子，策略是公式 $\boldsymbol{\pi}_\theta=\mathcal{PI}\boldsymbol{\pi}_\theta$ 的不动点[217]. 策略不动点为策略梯度强化学习研究开辟了新方向. 为了保证算法的收敛性并加速收敛，如何优化策略梯度强化学习的改进算子和投影算子是未来的研究热点.

广义值不动点. 在面对高维表示的大规模状态动作空间的强化学习问题时，由于高维引起的维度灾难，我们不得不采用函数估计. 与传统的监督学习不同，强化学习没有监督信号，只能通过即时奖赏来逼近真实值函数，即延时反馈. 这样的逼近需要权衡偏差与方差. 另外，由于即时奖赏不受函数估计的特征空间约束，目前已有的四类不动点解都不是最优解. 因此，需要研究如何构造广义不动点解，并针对不同的强化学习问题对广义不动点解进行搜索、集成和优化. 图 3 从稳定性和偏差两个角度总结了现有不动点的表达式. 其中，稳定性从 TD 和 BR 角度进行对比，偏差从一步 Bellman 方程和综合多步 Bellman 方程角度进行对比. 从该视角出发，广义不动点 $\mathbb{E}[\delta\boldsymbol{e}]$ 是显而易见能填补研究空白的，其中 $\boldsymbol{e}_t=\gamma_t\lambda_t\boldsymbol{e}_{t-1}+(\boldsymbol{\phi}-\gamma\boldsymbol{\phi}')$.

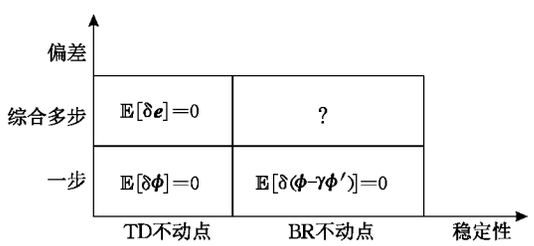

图 3 基于值函数估计的强化学习不动点

安全性条件下的值不动点. 不确定性环境中，存在一些危险或者风险. 智能体在最大化奖赏总和时，还需要尽量避免危险或风险带来的损伤，以保证自身的安全. 把不确定因素考虑进来时有多种理念：(1) 最坏情况. 采用极大极小法来替换原先的值函数 $\max\limits_{\pi\in\Pi}\min\limits_{\omega\in\Omega}\mathbb{E}_{\pi,\omega}\left(\sum\limits_{t=0}^{\infty}\gamma^t r_t\right)$[218]，其中 $\boldsymbol{\Omega}$ 表示所有可能的模型. $\hat{Q}$-learning 算法采用项 $\min(\hat{\boldsymbol{Q}}(s_t,a_t),r_{t+1}+\gamma\max\limits_{a'\in\boldsymbol{A}}\hat{\boldsymbol{Q}}(s_{t+1},a'))$ 进行参数更新[218]. 2003 年，Gaskett 提出了 β 悲观 Q 学习算法，采用项 $r_{t+1}+\gamma((1-\beta)\max\limits_{a'\in\boldsymbol{A}}\boldsymbol{Q}_\beta(s_{t+1},a')+\beta\min\limits_{a'\in\boldsymbol{A}}\boldsymbol{Q}_\beta(s_{t+1},a'))$ 进行参数更新[219]. 2021 年，Jin 等人探讨了悲观主义在离线强化学习中的有效性[220]；(2) 效用函数. 采用指数效用函数的方式进行修正，如 $\max\limits_{\pi\in\Pi}\beta^{-1}\log\mathbb{E}_\pi[e^{\beta\sum_{t=0}^{\infty}\gamma^t r_t}]$[221]，考虑方差的效用函数，如 $\max\limits_{\pi\in\Pi}\mathbb{E}_\pi[\boldsymbol{R}]-\beta\mathrm{Var}(\boldsymbol{R})$[222]、$\max\limits_{\pi\in\Pi}\mathbb{E}_\pi[\boldsymbol{R}]\sqrt{\mathrm{Var}(\boldsymbol{R})}$[223] 等；(3) 约束函数. 采用带约束的函数限制一些特殊动作，如 $\max\limits_{\pi\in\Pi}\mathbb{E}_\pi[\boldsymbol{R}],\mathrm{Var}(\boldsymbol{R})\leq\alpha$[223] 等. 然而这些解法是不是不动点，能否保证收敛，仍需进一步研究；(4) 不确定性贝尔曼等式. 2018 年，O'Donoghue 等人提出了不确定性贝尔曼等式，并证明了该等式的不动点是任何策略 Q 值后验分布下的方差上界，基于该不动点的汤普森采样优于基于计数和上置信的探索方法[224].

多智能体值不动点. 对于二人零和博弈，1953 年，Shapley 提出了基于表格值的值迭代算法，并根据压缩映射原理，可以收敛到纳什均衡[225]. 1994 年，Littman 提出了 MinimaxQ learning 算法[226]. 1999 年，Szepesvári 等人证明 MinimaxQ learning 算法收敛到纳什均衡[227]. 2002 年，Lagoudakis 等人证明了线性值函数估计下 MinimaxQ learning 算法的收敛性[228]. 与单智能体不同，多智能体的扩展有诸多形式：(1) 收益的不同，如零和博弈、广义和博弈；(2) 智能体之间的竞争与合作关系的不同，如纯合作型博弈、竞争型博弈、竞争合作型博弈；(3) 建模的不同，如 Markov 博弈、扩展式博弈等. 目前多智能体的理论分析相对局限在单一场景如二人零和博弈，其它相关算法的理论分析还任重道远. 更多关于多智能体的综述与挑战请阅读文献[229].

致 谢 该工作凝练成文得益于国防科技大学冯旸赫老师的鼓励. 稿件的改进和完善离不开审稿人的意见和建议以及编辑同志的善意提醒和耐心等待！

参 考 文 献

[1] Kaelbling L P, Littman M L, Moore A W. Reinforcement learning: A survey. Journal of Artificial Intelligence Research, 1996, 4: 237-285

[2] Gao Yang, Chen Shi-Fu, Lu Xin. Research on reinforcement learning technology: A review. Acta Automatica Sinica, 2004, 30(1): 86-100(in Chinese)
(高阳，陈世福，陆鑫. 强化学习研究综述. 自动化学报，2004，30(1)：86-100)

[3] Geist M, Pietquin O. Algorithmic survey of parametric value function approximation. IEEE Transactions on Neural Networks and Learning Systems, 2013, 24(6): 845-867

[4] Chen Xing-Guo, Yu Yang. Reinforcement learning and its application to the game of Go. Acta Automatica Sinica, 2016, 42(5): 685-695(in Chinese)

(陈兴国，俞扬. 强化学习及其在电脑围棋中的应用. 自动化学报，2016，42(5)：685-695)

[5] Wang Hao，Gao Yang，Chen Xing-Guo. Transfer of reinforcement learning：The state of the art. Acta Electronica Sinica，2008，36(12A)：39-43(in Chinese)
(王皓，高阳，陈兴国. 强化学习中的迁移：方法和进展. 电子学报，2008，36(12A)：39-43)

[6] Taylor M E，Stone P. Transfer learning for reinforcement learning domains：A survey. Journal of Machine Learning Research，2009，10：1633-1685

[7] Garcia J，Fernández F. A comprehensive survey on safe reinforcement learning. Journal of Machine Learning Research，2015，16(1)：1437-1480

[8] Zhou Wen-Ji，Yu Yang. Summarize of hierarchical reinforcement learning. CAAI Transactions on Intelligent Systems，2017，12(5)：590-594(in Chinese)
(周文吉，俞扬. 分层强化学习综述. 智能系统学报，2017，12(5)：590-594)

[9] Yin Chang-Sheng，Yang Ruo-Peng，Zhu Wei，et al. A survey on multi-agent hierarchical reinforcement learning. CAAI Transactions on Intelligent Systems，2020，15(4)：646-655(in Chinese)
(殷昌盛，杨若鹏，朱巍等. 多智能体分层强化学习综述. 智能系统学报，2020，15(4)：646-655)

[10] Zhao Dong-Bin，Shao Kun，Zhu Yuan-Heng，et al. Review of deep reinforcement learning and discussions on the development of computer Go. Control Theory & Applications，2016，33(6)：701-717(in Chinese)
(赵冬斌，邵坤，朱圆恒等. 深度强化学习综述：兼论计算机围棋的发展. 控制理论与应用，2016，33(6)：701-717)

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

[12] Li Y. Deep reinforcement learning：An overview. arXiv preprint arXiv：1701.07274，2017

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

[14] Zhao Xing-Yu，Ding Shi-Fei. Research on deep reinforcement learning. Computer Science，2018，45(7)：1-6(in Chinese)
(赵星宇，丁世飞. 深度强化学习研究综述. 计算机科学，2018，45(7)：1-6)

[15] Liu Jian-Wei，Gao Feng，Luo Xiong-Lin. Survey of deep reinforcement learning based on value function and policy gradient. Chinese Journal of Computers，2019，42(6)：1406-1438(in Chinese)
(刘建伟，高峰，罗雄麟. 基于值函数和策略梯度的深度强化学习综述. 计算机学报，2019，42(6)：1406-1438)

[16] Wan Li-Peng，Lan Xu-Guang，Zhang Han-Bo，et al. A review of deep reinforcement learning theory and application. Pattern Recognition and Artificial Intelligence，2019，32(1)：67-81(in Chinese)
(万里鹏，兰旭光，张翰博等. 深度强化学习理论及其应用综述. 模式识别与人工智能，2019，32(1)：67-81)

[17] Yang Wei-Yi，Bai Chen-Jia，Cai Chao，et al. Survey on sparse reward in deep reinforcement learning. Computer Science，2020，47(3)：182-191(in Chinese)
(杨惟轶，白辰甲，蔡超等. 深度强化学习中稀疏奖励问题研究综述. 计算机科学，2020，47(3)：182-191)

[18] Zhang Kai-Feng，Yu Yang. Methodologies for imitation learning via inverse reinforcement learning：A review. Journal of Computer Research and Development，2019，56(2)：254-261(in Chinese)
(张凯峰，俞扬. 基于逆强化学习的示教学习方法综述. 计算机研究与发展，2019，56(2)：254-261)

[19] Arora S，Doshi P. A survey of inverse reinforcement learning：Challenges，methods and progress. Artificial Intelligence，2021，297：103500

[20] Busoniu L，Babuska R，De Schutter B. A comprehensive survey of multiagent reinforcement learning. IEEE Transactions on Systems，Man，and Cybernetics，Part C (Applications and Reviews)，2008，38(2)：156-172

[21] Du Wei，Ding Shi-Fei. Overview on multi-agent reinforcement learning. Computer Science，2019，46(8)：1-8(in Chinese)
(杜威，丁世飞. 多智能体强化学习综述. 计算机科学，2019，46(8)：1-8)

[22] Sun Chang-Yin，Mu Chao-Xu. Important scientific problems of multi-agent deep reinforcement learning. Acta Automatica Sinica，2020，46(7)：1301-1312(in Chinese)
(孙长银，穆朝絮. 多智能体深度强化学习的若干关键科学问题. 自动化学报，2020，46(7)：1301-1312)

[23] Puiutta E，Veith E M. Explainable reinforcement learning：A survey//Proceedings of the International Cross-Domain Conference for Machine Learning and Knowledge Extraction. Dublin，Ireland，2020：77-95

[24] Glanois C，Weng P，Zimmer M，et al. A survey on interpretable reinforcement learning. arXiv preprint arXiv：2112.13112，2021

[25] Chen Jin-Yin，Zhang Yan，Wang Xue-Ke，et al. A survey of attack，defense and related security analysis for deep reinforcement learning. Acta Automatica Sinica，2022，48(1)：21-39(in Chinese)
(陈晋音，章燕，王雪柯等. 深度强化学习的攻防与安全性分析综述. 自动化学报，2022，48(1)：21-39)

[26] Xiong Luo-Lin，Mao Shuai，Tang Yang，et al. Reinforcement learning based integrated energy system management：A survey. Acta Automatica Sinica，2021，47(10)：2321-2340(in Chinese)
(熊珞琳，毛帅，唐漾等. 基于强化学习的综合能源系统管理综述. 自动化学报，2021，47(10)：2321-2340)

[27] Li Kai-Wen，Zhang Tao，Wang Rui，et al. Research reviews of combinatorial optimization methods based on deep reinforcement learning. Acta Automatica Sinica，2021，47(11)：2521-2537(in Chinese)

(李凯文，张涛，王锐等. 基于深度强化学习的组合优化研究进展. 自动化学报，2021，47(11)：2521-2537)

[28] Mazyavkina N, Sviridov S, Ivanov S, et al. Reinforcement learning for combinatorial optimization: A survey. Computers & Operations Research, 2021, 134: 105400

[29] 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

[30] Luketina J, Nardelli N, Farquhar G, et al. A survey of reinforcement learning informed by natural language//Proceedings of the 28th International Joint Conference on Artificial Intelligence. Macao, China, 2019: 6309-6317

[31] Alharin A, Doan T N, Sartipi M. Reinforcement learning interpretation methods: A survey. IEEE Access, 2020, 8: 171058-171077

[32] Coronato A, Naeem M, De Pietro G, et al. Reinforcement learning for intelligent healthcare applications: A survey. Artificial Intelligence in Medicine, 2020, 109: 101964

[33] Yau K L A, Qadir J, Khoo H L, et al. A survey on reinforcement learning models and algorithms for traffic signal control. ACM Computing Surveys, 2017, 50(3): 1-38

[34] Haydari A, Yilmaz Y. Deep reinforcement learning for intelligent transportation systems: A survey. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(1): 11-32

[35] Wang W, Sun D, Jiang F, et al. Research and challenges of reinforcement learning in cyber defense decision-making for intranet security. Algorithms, 2022, 15(4): 134

[36] Sutton R, Barto A. Reinforcement Learning: An Introduction, 1st Edition. Cambridge, USA: MIT Press, 1998

[37] Sutton R S, Barto A G. Reinforcement learning: An introduction, 2nd Edition. Cambridge, USA: MIT Press, 2018

[38] Sutton R, Maei H, Precup D, et al. Fast gradient-descent methods for temporal-difference learning with linear function approximation//Proceedings of the 26th Annual International Conference on Machine Learning. New York, USA, 2009: 993-1000

[39] Tang Yan. Iterative Approximation of Fixed Point and Zero Point and Its Application. Chongqing: Chongqing University Press, 2019(in Chinese)
(唐艳. 不动点与零点的迭代逼近及应用. 重庆：重庆大学出版社，2019)

[40] Bertsekas D, Tsitsiklis J N. Neuro-Dynamic Programming. New Hampshire, USA: Athena Scientific Press, 1996

[41] Schaul T, Quan J, Antonoglou I, et al. Prioritized experience replay. arXiv preprint arXiv:1511.05952, 2015

[42] Moerland T M, Broekens J, Jonker C M. Model-based reinforcement learning: A survey. arXiv preprint arXiv: 2006.16712, 2020

[43] Sutton R S. Temporal Credit Assignment in Reinforcement Learning[Ph. D. dissertation]. University of Massachusetts Amherst, Massachusetts, USA, 1984

[44] Sutton R S. Learning to predict by the methods of temporal differences. Machine Learning, 1988, 3(1): 9-44

[45] Rummery G A, Niranjan M. On-line Q-learning using connectionist systems. Cambridge University Engineering Department, Cambridge, England: Technical Report: CUED/F-INFENG/TR 166, 1994

[46] Precup D, Sutton R S, Singh S. Eligibility traces for off-policy policy evaluation//Proceedings of the 17th International Conference on Machine Learning. Stanford, USA, 2000: 759-766

[47] van Hasselt H. Double Q-learning//Advances in Neural Information Processing Systems. Vancouver, Canada, 2010: 2613-2621

[48] Azar M G, Munos R, Ghavamzadeh M, et al. Speedy Q-learning//Advances in Neural Information Processing Systems. Granadam, Spain, 2011: 2411-2419

[49] John I, Kamanchi C, Bhatnagar S. Generalized speedy Q-learning. IEEE Control Systems Letters, 2020, 4(3): 524-529

[50] Zheng Shuai, Luo Fei, Gu Chun-Hua, et al. Improved Speedy Q-learning algorithm based on double estimator. Computer Science, 2020, 47(7): 179-185(in Chinese)
(郑帅，罗飞，顾春华等. 基于双估计器的改进 Speedy Q-learning 算法. 计算机科学，2020，47(7)：179-185)

[51] Zhu R, Rigotti M. Self-correcting Q-learning//Proceedings of the AAAI Conference on Artificial Intelligence. California, USA, 2021: 11185-11192

[52] Munos R, Stepleton T, Harutyunyan A, et al. Safe and efficient off-policy reinforcement learning//Advances in Neural Information Processing Systems. Barcelona, Spain, 2016: 1054-1062

[53] Lv P, Wang X, Cheng Y, et al. Stochastic double deep Q-network. IEEE Access, 2019, 7: 79446-79454

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

[55] Kakade S M. On the Sample Complexity of Reinforcement Learning[Ph. D. dissertation]. University of London, London, England, 2003

[56] Lattimore T, Hutter M, Sunehag P. The sample-complexity of general reinforcement learning//Proceedings of the 30th International Conference on Machine Learning. Atlanta, Georgia, 2013: 28-36

[57] Dann C, Brunskill E. Sample complexity of episodic fixed horizon reinforcement learning//Advances in Neural Information Processing Systems. Montreal, Canada, 2015: 2818-2826

[58] Yang L, Wang M. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound//Proceedings of the 37th International Conference on Machine Learning. Virtual, 2020: 10746-10756

[59] Schmoll S, Schubert M. Semi-Markov reinforcement learning for stochastic resource collection//Proceedings of the 29th International Conference on International Joint Conferences on Artificial Intelligence. Yokohama, Japan, 2021: 3349-3355

[60] Xiang X, Foo S. Recent advances in deep reinforcement learning applications for solving partially observable Markov decision processes(POMDP) problems: Part 1—Fundamentals and applications in games, robotics and natural language processing. Machine Learning and Knowledge Extraction, 2021, 3(3): 554-581

[61] Padakandla S. A survey of reinforcement learning algorithms for dynamically varying environments. ACM Computing Surveys, 2021, 54(6): 1-25

[62] Tsitsiklis J N, Van Roy B. Analysis of temporal-difference learning with function approximation//Advances in Neural Information Processing Systems. Denver, USA, 1997: 1075-1081

[63] Borkar V S, Meyn S P. The ode method for convergence of stochastic approximation and reinforcement learning. SIAM Journal on Control and Optimization, 2000, 38(2): 447-469

[64] Borkar V S. Stochastic Approximation: A Dynamical Systems Viewpoint. New Delhi, India: Springer, 2009

[65] Johnson R, Zhang T. Accelerating stochastic gradient descent using predictive variance reduction//Advances in Neural Information Processing Systems. Nevada, USA, 2013: 315-323

[66] Bradtke S J. Incremental Dynamic Programming for On-Line Adaptive Optimal Control[Ph. D. dissertation]. University of Massachusetts, Massachusetts, USA, 1994

[67] Bradtke S J, Barto A G. Linear least-squares algorithms for temporal difference learning. Machine Learning, 1996, 22(1-3): 33-57

[68] Nedić A, Bertsekas D P. Least squares policy evaluation algorithms with linear function approximation. Discrete Event Dynamic Systems, 2003, 13(1): 79-110

[69] Yu H, Bertsekas D P. Convergence results for some temporal difference methods based on least squares. IEEE Transactions on Automatic Control, 2009, 54(7): 1515-1531

[70] Ueno T, Maeda S I, Kawanabe M, et al. Generalized TD learning. The Journal of Machine Learning Research, 2011, 12(6): 1977-2020

[71] van Seijen H, Sutton R. True online TD(λ)//Proceedings of the 31st International Conference on Machine Learning. Beijing, China, 2014: 692-700

[72] Wang Bi, Li Xuelian, Gao Zhiqiang, et al. Gradient compensation traces based temporal difference learning. Neurocomputing, 2021, 442: 221-235

[73] Boyan J A. Technical update: Least-squares temporal difference learning. Machine Larning, 2002, 49(2-3): 233-246

[74] Xu X, He H G, Hu D. Efficient reinforcement learning using recursive least-squares methods. Journal of Artificial Intelligence Research, 2002, 16: 259-292

[75] Lagoudakis M G, Parr R. Least-squares policy iteration. Journal of Machine Learning Research, 2003, 4: 1107-1149

[76] Geramifard A, Bowling M, Sutton R S. Incremental least-squares temporal difference learning//Proceedings of the National Conference on Artificial Intelligence. Boston, USA, 2006: 356-361

[77] Geramifard A, Bowling M, Zinkevich M, et al. iLSTD: Eligibility traces and convergence analysis//Advances in Neural Information Processing Systems. Vancouver, Canada, 2006: 441-448

[78] Johns J, Petrik M, Mahadevan S. Hybrid least-squares algorithms for approximate policy evaluation. Machine Learning, 2009, 76(2-3): 243-256

[79] van Seijen H, Sutton R. A deeper look at planning as learning from replay//Proceedings of the 32nd International Conference on Machine Learning. Lille, France, 2015: 2314-2322

[80] Osogami T. Uncorrected least-squares temporal difference with λ-return//Proceedings of the 34th AAAI Conference on Artificial Intelligence. New York, USA, 2020: 5323-5330

[81] van Seijen H, Mahmood A R, Pilarski P M, et al. True online temporal-difference learning. The Journal of Machine Learning Research, 2016, 17(1): 5057-5096

[82] Sutton R S, Mahmood A R, White M. An emphatic approach to the problem of off-policy temporal-difference learning. The Journal of Machine Learning Research, 2016, 17(1): 2603-2631

[83] Baird L, et al. Residual algorithms: Reinforcement learning with function approximation//Proceedings of the 12th International Conference on Machine Learning. California, USA, 1995: 30-37

[84] Scherrer B. Should one compute the temporal difference fix point or minimize the Bellman residual? The unified oblique projection view//Proceedings of the 27th International Conference on International Conference on Machine Learning. Haifa, Israel, 2010: 959-966

[85] Sutton R S, Maei H R, Szepesvari C. A convergent $O(\lambda)$ temporal-difference algorithm for off-policy learning with linear function approximation//Advances in Neural Information Processing Systems. Vancouver, Canada, 2008, 21: 1609-1616

[86] Maei H R, Sutton R S. GQ(λ): A general gradient algorithm for temporal-difference prediction learning with eligibility traces//Proceedings of the 3rd Conference on Artificial General Intelligence. Lugano, Switzerland, 2010: 91-96

[87] Meyn S, Tweedie R L. Markov Chains and Stochastic Stability. Cambridge, UK: Cambridge University Press, 2009

[88] Bertsekas D P, Yu H. Projected equation methods for approximate solution of large linear systems. Journal of Computational and Applied Mathematics, 2009, 227(1): 27-50

[89] Yu H. Convergence of least squares temporal difference methods under general conditions//Proceedings of the 27th International Conference on Machine Learning. Haifa, Israel, 2010: 1207-1214

[90] Hackman L M. Faster Gradient-TD Algorithms [M. S. dissertation]. University of Alberta, Canada, 2013

[91] Maei H R. Gradient Temporal-Difference Learning Algorithms [Ph. D. dissertation]. University of Alberta, Canada, 2011

[92] Liu B, Liu J, Ghavamzadeh M, et al. Proximal gradient temporal difference learning algorithms//Proceedings of the 25th International Joint Conference on Artificial Intelligence. New York, USA, 2016: 4195- 4199

[93] Pan Y, White A, White M. Accelerated gradient temporal difference learning//Proceedings of the 31st AAAI Conference on Artificial Intelligence. California, USA, 2017: 2464-2470

[94] Wu Yu-Shuang, Chen Xiao-Yu, Ma Jing-Wen, et al. Off-policy linear temporal difference learning algorithms with a generalized oblique projection. Journal of Nanjing University (Natural Science), 2017, 53(6): 1052-1062(in Chinese)
(吴毓双, 陈筱语, 马静雯等. 基于一般化斜投影的异策略时序差分学习算法. 南京大学学报（自然科学版）, 2017, 53(6): 1052-1062)

[95] Song T, Li D, Yang W, et al. Recursive least-squares temporal difference with gradient correction. IEEE Transactions on Cybernetics, 2019, 51: 4251-4264

[96] Ghiassian S, Patterson A, Garg S, et al. Gradient temporal difference learning with regularized corrections//Proceedings of the 37th International Conference on Machine Learning. Virtual, 2020: 3524-3534

[97] Precup D, Sutton R S, Dasgupta S. Off-policy temporal difference learning with function approximation//Proceedings of the 18th International Conference on Machine Learning. Massachusetts, USA, 2001: 417-424

[98] Mahmood A R, van Hasselt H, Sutton R S. Weighted importance sampling for off-policy learning with linear function approximation//Advances in Neural Information Processing Systems. Montreal, Canada, 2014: 3014-3022

[99] Mahmood A R, Sutton R S. Off-policy learning based on weighted importance sampling with linear computational complexity//Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence. Virginia, USA, 2015: 552-561

[100] Liu B, Liu J, Ghavamzadeh M, et al. Finite-sample analysis of proximal gradient TD algorithms//Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence. Amsterdam, Netherlands, 2015: 504-513

[101] Liu B, Gemp I, Ghavamzadeh M, et al. Proximal gradient temporal difference learning: Stable reinforcement learning with polynomial sample complexity. Journal of Artificial Intelligence Research, 2018, 63: 461-494

[102] Hallak A, Tamar A, Mannor S. Emphatic TD Bellman operator is a contraction. arXiv preprint arXiv:1508. 03411, 2015

[103] Mnih V, Badia A P, Mirza M, et al. Asynchronous methods for deep reinforcement learning//Proceedings of the 33rd International Conference on Machine Learning. New York, USA, 2016: 1928-1937

[104] van Hasselt H, Mahmood A R, Sutton R S. Off-policy TD (λ) with a true online equivalence//Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence. Quebec City, Canada, 2014: 330-339

[105] Sutton R, Mahmood A R, Precup D, et al. A new $Q(\lambda)$ with interim forward view and Monte Carlo equivalence//Proceedings of the 31st International Conference on Machine Learning. Beijing, China, 2014: 568- 576

[106] Devraj A M, Meyn S P. Zap Q-learning//Advances in Neural Information Processing Systems. California, USA, 2017: 2232-2241

[107] Mahmood A R, Yu H, Sutton R S. Multi-step off-policy learning without importance sampling ratios. arXiv preprint arXiv:1702. 03006, 2017

[108] Yu H, Mahmood A R, Sutton R S. On generalized Bellman equations and temporal-difference learning. The Journal of Machine Learning Research, 2018, 19(1): 1864-1912

[109] Dalal G, Thoppe G, Szörényi B, et al. Finite sample analysis of two-timescale stochastic approximation with applications to reinforcement learning//Proceedings of the 31st Conference on Learning Theory. Stockholm, Sweden, 2018: 1199-1233

[110] Gupta H, Srikant R, Ying L. Finite-time performance bounds and adaptive learning rate selection for two time-scale reinforcement learning//Advances in Neural Information Processing Systems. Vancouver, Canada, 2019: 4704-4713

[111] Dalal G, Szorenyi B, Thoppe G. A tale of two-timescale reinforcement learning with the tightest finite-time bound//Proceedings of the 34th AAAI Conference on Artificial Intelligence. New York, USA, 2020: 3701-3708

[112] Xu T, Liang Y. Sample complexity bounds for two timescale value based reinforcement learning algorithms//Proceedings of the 24th International Conference on Artificial Intelligence and Statistics. Virtual, 2021: 811-819

[113] Jaśkowski W. Mastering 2048 with delayed temporal coherence learning, multistage weight promotion, redundant encoding, and carousel shaping. IEEE Transactions on Games, 2017, 10(1): 3-14

[114] Scherrer B, Ghavamzadeh M, Gabillon V, et al. Approximate modified policy iteration and its application to the game of Tetris. Journal of Machine Learning Research, 2015, 16: 1629-1676

[115] van Hasselt H, Doron Y, Strub F, et al. Deep reinforcement learning and the deadly triad. arXiv preprint arXiv: 1812. 02648, 2018

[116] Yang Y, Li X, Zhang L. Task-specific pre-learning to improve the convergence of reinforcement learning based on a deep neural network//Proceedings of the 2016 12th World Congress on Intelligent Control and Automation. Guilin, China, 2016: 2209-2214

[117] Maei H R, Szepesvari C, Bhatnagar S, et al. Convergent temporal-difference learning with arbitrary smooth function approximation//Advances in Neural Information Processing Systems. Vancouver, Canada, 2009: 1204-1212

[118] Lee M, Anderson C W. Convergent reinforcement learning control with neural networks and continuous action search//Proceedings of the 2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning. Orlando, USA, 2014: 1-8

[119] Qu C, Mannor S, Xu H. Nonlinear distributional gradient temporal-difference learning//Proceedings of the 36th International Conference on Machine Learning. California, USA, 2019: 5251-5260

[120] Cai Q, Yang Z, Lee J D, et al. Neural temporal-difference learning converges to global optima//Advances in Neural Information Processing Systems. Vancouver, Canada, 2019: 11312-11322

[121] Wai H T, Hong M, Yang Z, et al. Variance reduced policy evaluation with smooth function approximation//Advances in Neural Information Processing Systems. Vancouver, Canada, 2019: 5784-5795

[122] Wai H T, Yang Z, Wang Z, et al. Provably efficient neural GTD for off-policy learning//Advances in Neural Information Processing Systems. Virtual, 2020: 10431-10442

[123] Zhang S, Boehmer W, Whiteson S. Deep residual reinforcement learning//Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems. Virtual, 2020: 1611- 1619

[124] Yang G, Li Y, Huang T, et al. DHQN: A stable approach to remove target network from deep Q-learning network//Proceedings of the 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence. Virtual, 2021: 1474-1479

[125] Cao J, Liu Q, Zhu F, et al. Gradient temporal-difference learning for off-policy evaluation using emphatic weightings. Information Sciences, 2021, 580: 311-330

[126] Wang Y, Zou S, Zhou Y. Non-asymptotic analysis for two timescale TDC with general smooth function approximation //Advances in Neural Information Processing Systems. Virtual, 2021: 9747-9758

[127] Levine N, Zahavy T, Mankowitz D J, et al. Shallow updates for deep reinforcement learning//Proceedings of the 31st International Conference on Neural Information Processing Systems. California, USA, 2017: 3138-3148

[128] Ghosh D, Bellemare M G. Representations for stable off-policy reinforcement learning//Proceedings of the 37th International Conference on Machine Learning. Virtual, 2020: 3556-3565

[129] Brandfonbrener D, Bruna J. Geometric insights into the convergence of non-linear TD learning//Proceedings of the 8th International Conference on Learning Representations. Virtual, 2020

[130] Feng Y, Li L, Liu Q. A kernel loss for solving the Bellman equation//Advances in Neural Information Processing Systems. Vancouver, Canada, 2019: 15456-15467

[131] Feng Y, Ren T, Tang Z, et al. Accountable off-policy evaluation with kernel Bellman statistics//Proceedings of the 37th International Conference on Machine Learning. Virtual, 2020: 3102-3111

[132] Ormoneit D, Sen S. Kernel-based reinforcement learning. Machine Learning, 2002, 49(2-3): 161-178

[133] Engel Y, Mannor S, Meir R. Bayes meets Bellman: The Gaussian process approach to temporal difference learning//Proceedings of the 20th International Conference on Machine Learning. Washington, USA, 2003: 154-161

[134] Xu X, Hu D, Lu X. Kernel-based least squares policy iteration for reinforcement learning. IEEE Transactions on Neural Networks, 2007, 18(4): 973-992

[135] Geist M, Pietquin O. Kalman temporal differences. Journal of Artificial Intelligence Research, 2010, 39: 483-532

[136] Kroemer O B, Peters J R. A non-parametric approach to dynamic programming//Advances in Neural Information Processing Systems. California, USA, 2011: 1719-1727

[137] Chen X, Gao Y, Wang R. Online selective kernel-based temporal difference learning. IEEE Transactions on Neural Networks and Learning Systems, 2013, 24(12): 1944-1956

[138] Barreto A M, Precup D, Pineau J. Practical kernel-based reinforcement learning. The Journal of Machine Learning Research, 2016, 17(1): 2372-2441

[139] Koppel A, Warnell G, Stump E, et al. Policy evaluation in continuous MDPs with efficient kernelized gradient temporal difference. IEEE Transactions on Automatic Control, 2020, 66(4): 1856-1863

[140] Yang G, Chen X, Yang S, et al. Online attentive kernel-based temporal difference learning. arXiv preprint arXiv: 2201.09065, 2022

[141] Kolter J Z, Ng A Y. Regularization and feature selection in least squares temporal difference learning//Proceedings of the 26th Annual International Conference on Machine Learning. Montreal, Canada, 2009: 521-528

[142] Geist M, Scherrer B. l_1-penalized projected Bellman residual //Proceedings of the European Workshop on Reinforcement Learning. Athens, Greece, 2011: 89-101

[143] Liu B, Mahadevan S. Compressive reinforcement learning with oblique random projections. Department of Computer Science, University of Massachusetts, Massachusetts, USA: Technical Report: UM-CS2011-024, 2011

[144] Nguyen T, Li Z, Silander T, et al. Online feature selection for model-based reinforcement learning//Proceedings of the 30th International Conference on Machine Learning. Atlanta, USA, 2013: 498-506

[145] Gehring C, Pan Y, White M. Incremental truncated LSTD //Proceedings of the 25th International Joint Conference on Artificial Intelligence. New York, USA, 2016: 1505-1511

[146] Li D, Li L, Song T, et al. Regularization and feature selection in least squares temporal difference with gradient correction//Proceedings of the 2016 12th World Congress on Intelligent Control and Automation. Guilin, China, 2016: 2289-2293

[147] Farahmand A M, Ghavamzadeh M, Szepesvári C, et al. Regularized policy iteration with nonparametric function spaces. The Journal of Machine Learning Research, 2016, 17(1): 4809-4874

[148] Song T, Li D, Jin Q, et al. Sparse proximal reinforcement learning via nested optimization. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2018, 50(11): 4020-4032

[149] Amit R, Meir R, Ciosek K. Discount factor as a regularizer in reinforcement learning//Proceedings of the 37th International Conference on Machine Learning. Virtual, 2020: 269-278

[150] Li D, Ma C, Zhang J, et al. Orthogonal matching pursuit for least squares temporal difference with gradient correction //Proceedings of the 2020 Chinese Automation Congress. Shanghai, China, 2020: 4108-4112

[151] Hao B, Duan Y, Lattimore T, et al. Sparse feature selection makes batch reinforcement learning more sample efficient// Proceedings of the 38th International Conference on Machine Learning. Virtual, 2021: 4063-4073

[152] Song T, Li D, Xu X. Online sparse temporal difference learning based on nested optimization and regularized dual averaging. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2021, 50(4): 2042-2052

[153] Džeroski S, De Raedt L, Driessens K. Relational reinforcement learning. Machine Learning, 2001, 43(1): 7-52

[154] Driessens K, Ramon J. Relational instance based regression for relational reinforcement learning//Proceedings of the 20th International Conference on Machine Learning. Washington, USA, 2003: 123-130

[155] Driessens K, Džeroski S. Integrating guidance into relational reinforcement learning. Machine Learning, 2004, 57(3): 271-304

[156] Driessens K, Ramon J, Gärtner T. Graph kernels and Gaussian processes for relational reinforcement learning. Machine Learning, 2006, 64(1): 91-119

[157] Sanner S, Boutilier C. Practical solution techniques for first order MDPs. Artificial Intelligence, 2009, 173(5-6): 748-788

[158] Liu G, Sun X, Schulte O, et al. Learning tree interpretation from object representation for deep reinforcement learning// Advances in Neural Information Processing Systems. Virtual, 2021: 19622-19636

[159] Yin H, Pan S J. Knowledge transfer for deep reinforcement learning with hierarchical experience replay//Proceedings of the 31st AAAI Conference on Artificial Intelligence. San Francisco, USA, 2017: 1640- 1646

[160] Andrychowicz M, Wolski F, Ray A, et al. Hindsight experience replay//Advances in Neural Information Processing Systems. California, USA, 2017: 5055-5065

[161] Manela B, Biess A. Bias-reduced hindsight experience replay with virtual goal prioritization. Neurocomputing, 2021, 451: 305-315

[162] Isele D, Cosgun A. Selective experience replay for lifelong learning//Proceedings of the AAAI Conference on Artificial Intelligence. New Orleans, USA, 2018: 3302-3309

[163] Zha D, Lai K H, Zhou K, et al. Experience replay optimization //Proceedings of the 28th International Joint Conference on Artificial Intelligence. Honolulu, Hawaii, 2019: 4243-4249

[164] Liu H, Trott A, Socher R, et al. Competitive experience replay//Proceedings of the 6th International Conference on Learning Representations. Vancouver, Canada, 2018

[165] Novati G, Koumoutsakos P. Remember and forget for experience replay//Proceedings of the 36th International Conference on Machine Learning. Long Beach, USA, 2019: 4851-4860

[166] Daley B, Amato C. Reconciling λ-returns with experience replay//Advances in Neural Information Processing Systems. Vancouver, Canada, 2019: 1133-1142

[167] Luo J, Li H. Dynamic experience replay//Proceedings of the Conference on Robot Learning. Virtual, 2020: 1191-1200

[168] Sun P, Zhou W, Li H. Attentive experience replay// Proceedings of the AAAI Conference on Artificial Intelligence. New York, USA, 2020: 5900-5907

[169] Fedus W, Ramachandran P, Agarwal R, et al. Revisiting fundamentals of experience replay//Proceedings of the 37th International Conference on Machine Learning. Virtual, 2020: 3061-3071

[170] Liu Y, Mattar M, Behrens T, et al. Experience replay is associated with efficient nonlocal learning. Science, 2021, 372(6544): eabf1357

[171] Sutton R S. Generalization in reinforcement learning: Successful examples using sparse coarse coding//Advances in Neural Information Processing Systems. Denver, USA, 1995: 1038-1044

[172] Krawiec K, Szubert M G. Learning N-tuple networks for othello by coevolutionary gradient search//Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation. Dublin, Ireland, 2011: 355-362

[173] Nair V, Hinton G E. Rectified linear units improve restricted Boltzmann machines//Proceedings of the 27th International Conference on Machine Learning. Haifa, Israel, 2010: 807-814

[174] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 2014, 15(1): 1929-1958

[175] Makhzani A, Frey B. k-sparse autoencoders//Proceedings of the 2nd International Conference on Learning Representations. Banff, Canada, 2014

[176] Makhzani A, Frey B J. Winner-take-all autoencoders// Advances in Neural Information Processing Systems. Montreal, Canada, 2015: 2791-2799

[177] Liu V, Kumaraswamy R, Le L, et al. The utility of sparse representations for control in reinforcement learning //Proceedings of the AAAI Conference on Artificial Intelligence. Honolulu, Hawaii, 2019: 4384-4391

[178] Ghiassian S, Rafiee B, Lo Y L, et al. Improving performance in reinforcement learning by breaking generalization in neural networks//Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems. Auckland, New Zealand, 2020: 438-446

[179] Bengio E, Pineau J, Precup D. Interference and generalization in temporal difference learning//Proceedings of the 37th International Conference on Machine Learning. Virtual, 2020: 767-777

[180] Zhu Y, Ying L. Borrowing from the future: An attempt to address double sampling//Proceedings of Machine Learning Research. Princeton, USA, 2020: 246-268

[181] van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double Q-learning//Proceedings of the 30th AAAI Conference on Artificial Intelligence. Phoenix, USA, 2016: 2094-2100

[182] Anschel O, Baram N, Shimkin N. Averaged-DQN: Variance reduction and stabilization for deep reinforcement learning// Proceedings of the 34th International Conference on Machine Learning. Sydney, Australia, 2017: 176-185

[183] Lee D, He N. Target-based temporal-difference learning// Proceedings of the 36th International Conference on Machine Learning. California, USA, 2019: 3713-3722

[184] Asadi K, Littman M L. An alternative Softmax operator for reinforcement learning//Proceedings of the 34th International Conference on Machine Learning. Sydney, Australia, 2017: 243-252

[185] Kim S, Asadi K, Littman M, et al. Removing the target network from deep Q-networks with the mellowmax operator //Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems. Montreal, Canada, 2019: 2060-2062

[186] Song Z, Parr R, Carin L. Revisiting the Softmax Bellman operator: New benefits and new perspective//Proceedings of the 36th International Conference on Machine Learning. Long Beach, USA, 2019: 5916-5925

[187] Kong W, Krichene W, Mayoraz N, et al. Rankmax: An adaptive projection alternative to the Softmax function// Advances in Neural Information Processing Systems. Virtual, 2020: 633-643

[188] Greensmith E, Bartlett P L, Baxter J. Variance reduction techniques for gradient estimates in reinforcement learning. Journal of Machine Learning Research, 2004, 5 (Nov): 1471-1530

[189] Veness J, Lanctot M, Bowling M. Variance reduction in Monte-Carlo tree search//Advances in Neural Information Processing Systems. Granada, Spain, 2011: 1836-1844

[190] Papini M, Binaghi D, Canonaco G, et al. Stochastic variance-reduced policy gradient//Proceedings of the 35th International Conference on Machine Learning. Stockholm, Sweden, 2018: 4026-4035

[191] Korda N, La P. On TD(0) with function approximation: Concentration bounds and a centered variant with exponential convergence//Proceedings of the 32nd International Conference on Machine Learning. Lille, France, 2015: 626-634

[192] Xu T, Wang Z, Zhou Y, et al. Reanalysis of variance reduced temporal difference learning//Proceedings of the 7th International Conference on Learning Representations. Virtual, 2019

[193] Sherstan C, Ashley D R, Bennett B, et al. Comparing direct and indirect temporal-difference methods for estimating the variance of the return//Proceedings of the Conference on Uncertainty in Artificial Intelligence. Monterey, USA, 2018: 63-72

[194] Simsek O, Algorta S, Kothiyal A. Why most decisions are easy in Tetris — and perhaps in other sequential decision problems, as well//Proceedings of the 33rd International Conference on Machine Learning. New York, USA, 2016: 1757-1765

[195] Dong Y, Zhang S, Liu X, et al. Variance aware reward smoothing for deep reinforcement learning. Neurocomputing, 2021, 458: 327-335

[196] Singh S P, Barto A G, Chentanez N. Intrinsically motivated reinforcement learning//Advances in Neural Information Processing Systems. Vancouver, Canada, 2004: 1281-1288

[197] Tang H, Houthooft R, Foote D, et al. #Exploration: A study of count-based exploration for deep reinforcement learning//Advances in Neural Information Processing Systems. Long Beach, USA, 2017: 2753-2762

[198] Brafman R I, Tennenholtz M. R-MAX-A general polynomial time algorithm for near-optimal reinforcement learning. Journal of Machine Learning Research, 2002, 3: 213-231

[199] Strehl A L, Littman M L. A theoretical analysis of model-based interval estimation//Proceedings of the 22nd International Conference on Machine Learning. Bonn, Germany, 2005: 856-863

[200] Szita I, Szepesvári C. Model-based reinforcement learning with nearly tight exploration complexity bounds//Proceedings of the 27th International Conference on Machine Learning. Haifa, Israel, 2010: 1031-1038

[201] Auer P, Ortner R. Logarithmic online regret bounds for undiscounted reinforcement learning//Advances in Neural Information Processing Systems. Vancouver, Canada, 2006: 49-56

[202] Jaksch T, Ortner R, Auer P. Near-optimal regret bounds for reinforcement learning. Journal of Machine Learning Research, 2010, 11: 1563-1600

[203] Lattimore T, Hutter M. Near-optimal PAC bounds for discounted MDPs. Theoretical Computer Science, 2014, 558: 125-143

[204] Stachniss C, Grisetti G, Burgard W. Information gain-based exploration using Rao-Blackwellized particle filters// Proceedings of the Robotics: Science and Systems. Cambridge, USA, 2005: 65-72

[205] Gopalan A, Mannor S, Mansour Y. Thompson sampling for complex online problems//Proceedings of the 31st International Conference on Machine Learning. Beijing, China, 2014: 100-108

[206] Yang L, Shi M, Zheng Q, et al. A unified approach for multistep temporal-difference learning with eligibility traces in reinforcement learning//Proceedings of the 27th International Joint Conference on Artificial Intelligence. Stockholm, Sweden, 2018: 2984-2990

[207] Meng W, Zheng Q, Yang L, et al. Qualitative measurements of policy discrepancy for return-based deep Q-network. IEEE Transactions on Neural Networks and Learning Systems, 2019, 31(10): 4374- 4380

[208] Beal D F, Smith M C. Temporal coherence and prediction decay in TD learning//Proceedings of the 16th International Joint Conference on Artificial Intelligence. Stockholm, Sweden, 1999: 564-569

[209] Bagheri S, Thill M, Koch P, et al. Online adaptable learning rates for the game connect-4. IEEE Transactions on Computational Intelligence and AI in Games, 2014, 8(1): 33-42

[210] Matsuzaki K. Developing a 2048 player with backward temporal coherence learning and restart//Advances in Computer Games-15th International Conferences. Leiden, The Netherlands, 2017: 176-187

[211] Yang S, Gao Y, An B, et al. Efficient average reward reinforcement learning using constant shifting values//Proceedings of the 30th AAAI Conference on Artificial Intelligence. Phoenix, USA, 2016: 2258-2264

[212] Yang S, Wang H, Gao Y, et al. An optimal algorithm for the stochastic bandits with knowing near-optimal mean reward//Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems. Sao Paulo, Brazil, 2018: 2130-2132

[213] Ng A Y, Harada D, Russell S. Policy invariance under reward transformations: Theory and application to reward shaping//Proceedings of the 16th International Conference on Machine Learning. Bled, Slovenia, 1999: 278-287

[214] Williams R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 1992, 8(3): 229-256

[215] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017

[216] Abdolmaleki A, Springenberg J T, Tassa Y, et al. Maximum a posteriori policy optimization//Proceedings of the 6th International Conference on Learning Representations. Vancouver, Canada, 2018

[217] Ghosh D, Machado M C, Le Roux N. An operator view of policy gradient methods//Advances in Neural Information Processing Systems. Virtual, 2020: 3397-3406

[218] Heger M. Consideration of risk in reinforcement learning// Machine Learning Proceedings 1994. New Brunswick, USA, 1994: 105-111

[219] Gaskett C. Reinforcement learning under circumstances beyond its control//Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation. Vienna, Austria, 2003

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

[221] Chung K J, Sobel M J. Discounted MDP's: Distribution functions and exponential utility maximization. SIAM Journal on Control and Optimization, 1987, 25(1): 49-62

[222] Geibel P, Wysotzki F. Risk-sensitive reinforcement learning applied to control under constraints. Journal of Artificial Intelligence Research, 2005, 24: 81-108

[223] Tamar A, Di Castro D, Mannor S. Policy gradients with variance related risk criteria//Proceedings of the 29th International Conference on Machine Learning. Edinburgh, Scotland, 2012: 1651-1658

[224] O'Donoghue B, Osband I, Munos R, et al. The uncertainty Bellman equation and exploration//Proceedings of the 35th International Conference on Machine Learning. Stockholm, Sweden, 2018: 3836-3845

[225] Shapley L S. Stochastic games. Proceedings of the National Academy of Sciences, 1953, 39(10): 1095-1100

[226] Littman M L. Markov games as a framework for multi-agent reinforcement learning//Proceedings of the Eleventh International Conference. New Brunswick, Canada, 1994: 157-163

[227] Szepesvári C, Littman M L. A unified analysis of value-function-based reinforcement-learning algorithms. Neural Computation, 1999, 11(8): 2017-2060

[228] Lagoudakis M G, Parr R. Value function approximation in zero-sum Markov games//Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence. Alberta, Canada, 2002: 283-292

[229] Zhang K, Yang Z, Başar T. Multi-agent reinforcement learning: A selective overview of theories and algorithms// Vamvoudakis KG, Wan Y, Lewis FL, Cansever D. Handbook of Reinforcement Learning and Control. Studies in Systems, Decision and Control. Switzerland: Springer Cham, 2021: 321-384

CHEN Xing-Guo, Ph. D., lecturer, M. S. supervisor. His research interest covers reinforcement learning, intelligent game, and machine learning.

SUN Dingyuanhao, M. S. His research interest covers reinforcement learning, intelligent game.

YANG Guang, M. S. His research interest covers reinforcement learning, intelligence Game.

YANG Shang-Dong, Ph. D. His research interest covers reinforcement learning, intelligent game, machine learning.

GAO Yang, Ph. D., professor. His research interest covers machine learning and reinforcement learning.

Background

Reinforcement Learning has been in development for almost 40 years since its introduction. Reinforcement learning is considered to be one of the most promising pathways to strong artificial intelligence. More and more researchers are using reinforcement learning to try to solve sequential decision-making tasks in their respective domains. However, practice shows that applying classical reinforcement learning algorithm does not directly meet practical needs. The design of efficient reinforcement learning algorithms for real-world decision problems remains a huge challenge for researchers and engineers.

With the rise of deep learning, reinforcement learning has developed rapidly, with a dizzying array of algorithms, techniques and tools. Many scholars in China and abroad have produced various types of reviews on reinforcement learning, but these work mainly categorise or summarise reinforcement learning from the perspective of specific scenarios or applications. How to view the latest reinforcement learning techniques from a unified perspective has become an urgent need for researchers.

To this end, this paper summarizes the design principles of reinforcement learning algorithms from a fixed point perspective. Firstly, the optimality problem and feasible solution construction of value function approximation are analyzed. Secondly, according to Banach fixed point theorem and Lyapunov's second judgment theorem, the stability issues of existing on-policy and off-policy reinforcement learning algorithms based on tabular, linear approximated, non-linear, and non-parametric approximated value functions are analyzed and summarized. Then, various improvement ideas to improve the accuracy or speedup convergence are interpreted from the perspective of the bias and variance control of the fixed point. Finally, future improvement directions on reinforcement learning are prospected.

This paper is partially supported by the National Natural Science Foundation of China (Nos. 62276142, 62206133, 62202240, 62192783), the Science and Technology Innovation 2030 New Generation Artificial Intelligence Major Project (No. 2018AAA0100905), the Primary Research & Development Plan of Jiangsu Province (No. BE2021028), the Shenzhen Fundamental Research Program (No. 2021Szvup056).