软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2018,29(2):267−282 [doi: 10.13328/j.cnki.jos.005251]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

一种解决连续空间问题的真实在线自然梯度 AC 算法*

朱 斐[1,2,3], 朱海军[1], 刘 全[1,3], 陈冬火[1], 伏玉琛[1,4]

[1](苏州大学 计算机科学与技术学院,江苏 苏州 215006)

[2](江苏省计算机信息处理技术重点实验室(苏州大学),江苏 苏州 215006)

[3](符号计算与知识工程教育部重点实验室(吉林大学),吉林 长春 130012)

[4](常熟理工学院 计算机科学与工程学院,江苏 常熟 215500)

通讯作者: 伏玉琛, E-mail: yuchenfu@suda.edu.cn

摘 要: 策略梯度作为一种能够有效解决连续空间决策问题的方法得到了广泛研究,但由于在策略估计过程中存在较大方差,因此,基于策略梯度的方法往往受到样本利用率低、收敛速度慢等限制.针对该问题,在行动者-评论家(actor-critic,简称 AC)算法框架下,提出了真实在线增量式自然梯度 AC(true online incremental natural actor-critic,简称 TOINAC)算法.TOINAC 算法采用优于传统梯度的自然梯度,在真实在线时间差分(true online time difference,简称 TOTD)算法的基础上,提出了一种新型的前向观点,改进了自然梯度行动者-评论家算法.在评论家部分,利用 TOTD 算法高效性的特点来估计值函数;在行动者部分,引入一种新的前向观点来估计自然梯度,再利用资格迹将自然梯度估计变为在线估计,提高了自然梯度估计的准确性和算法的效率.将 TOINAC 算法与核方法以及正态策略分布相结合,解决了连续空间问题.最后,在平衡杆、Mountain Car 以及 Acrobot 等连续问题上进行了仿真实验,验证了算法的有效性.

关键词: 策略梯度;自然梯度;行动者-评论家;真实在线 TD;核方法

中图法分类号: TP301

中文引用格式: 朱斐,朱海军,刘全,陈冬火,伏玉琛.一种解决连续空间问题的真实在线自然梯度 AC 算法.软件学报,2018,29(2):267−282. http://www.jos.org.cn/1000-9825/5251.htm

英文引用格式: Zhu F, Zhu HJ, Liu Q, Chen DH, Fu YC. True online natural actor-critic algorithm for the continuous space problem. Ruan Jian Xue Bao/Journal of Software, 2018,29(2):267−282 (in Chinese). http://www.jos.org.cn/1000-9825/5251.htm

True Online Natural Actor-Critic Algorithm for the Continuous Space Problem

ZHU Fei[1,2,3], ZHU Hai-Jun[1], LIU Quan[1,3], CHEN Dong-Huo[1], FU Yu-Chen[1,4]

[1](School of Computer Science and Technology, Soochow University, Suzhou 215006, China)

[2](Provincial Key Laboratory for Computer Information Processing Technilogy (Soochow University), Suzhou 215006, China)

[3](Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education (Jilin University), Changchun 130012, China)

[4](School of Computer Science and Engineering, Changshu Institute of Technology, Changshu 215500, China)

* 基金项目: 国家自然科学基金(61303108, 61373094, 61472262); 江苏省高校自然科学研究项目(17KJA520004); 符号计算与知识工程教育部重点实验室(吉林大学)资助项目(93K172014K04); 苏州市应用基础研究计划工业部分(SYG201422); 高校省级重点实验室(苏州大学)项目(KJS1524); 中国国家留学基金(201606920013)

Foundation item: National Natural Science Foundation of China (61303108, 61373094, 61472262); Jiangsu College Natural Science Research Key Program (17KJA520004); Program of the Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education (Jilin University) (93K172014K04); Suzhou Industrial Application of Basic Research Program (SYG201422); Program of the Provincial Key Laboratory for Computer Information Processing Technology (Soochow University) (KJS1524); China Scholarship Council Project (201606920013)

收稿时间: 2016-11-04; 修改时间: 2016-12-13; 采用时间: 2017-01-10; jos 在线出版时间: 2017-03-24

CNKI 网络优先出版: 2017-03-24 17:09:31, http://kns.cnki.net/kcms/detail/11.2560.TP.20170324.1709.011.html

Abstract: Policy gradient methods have been extensively studied as a solution to the continuous space control problem. However, due to the presence of high variance in the gradient estimation, policy gradient based methods are restricted by low sample data utilization and slow convergence. Aiming at solving this problem, utilizing the framework of actor-critic algorithm, a true online incremental natural actor-critic (TOINAC) algorithm, which takes advantage of the natural gradient that is superior to conventional gradient, and is based on true online time difference (TOTD), is proposed. In the critic part of TOINAC algorithm, the efficiency of TOTD is adopted to estimate the value function, and in the actor part of TOINAC algorithm, a novel forward view is introduced to compute and estimate natural gradient. Then, eligibility traces are utilized to turn natural gradient into online estimation, thereby improving the accuracy of natural gradient and efficiency of the method. The TOINAC algorithm is used to integrate with the kernel method and normal distribution policy to tackle the continuous space problem. The simulation tests on cart pole, Mountain Car and Acrobot, which are classical benchmark tests for continuous space problem, verify the effeteness of the algorithm.

Key words: policy gradient; natural gradient; actor-critic; true online TD; kernel method

机器学习是一门通过模拟人类学习过程,不断重组原有知识结构,以达到自我完善的学科.根据学习机制的不同,机器学习可以分为监督学习、非监督学习、强化学习(reinforcement learning).有别于其他两类方法,强化学习可以在没有标签数据以及完备的环境知识的环境中,通过直接与环境交互进行学习,有效地解决序贯决策问题,其目标是从环境中得到长期最大累计奖赏(return,简称 R).强化学习问题通常利用马尔可夫决策过程(Markov decision process,简称 MDP)进行建模[1].目前,强化学习已经积累了一些较好的应用实例,例如,谷歌的DeepMind将强化学习应用到围棋程序AlphaGo中[2],Riedmiller等人将强化学习应用到机器人足球中[3],Bagnell等人利用强化学习方法控制无人机[4].

在很多实际应用中,需要解决的问题往往具有连续的状态空间或/和连续的动作空间.例如,在电动机控制问题中,作为状态之一的电动机当前转速是 0 到最大转速这一连续区间的某个值,作为控制动作的电动机加电电压值也是 0 到最大额定电压这一连续区间的某个值.有效地解决连续空间问题是一个重要的研究内容,也是当前的研究热点之一.常见的解决方法包括值函数方法、近似动态规划方法(approximate dynamic programming,简称ADP)以及策略搜索方法等.值函数方法通过状态或状态动作对值函数间接地表示策略.连续动作空间 Q 学习算法(continuous-action Q-learning,简称CAQ)[5]是一个典型的值函数方法,通过离散化动作空间并结合 Q 学习处理连续问题.CAQ 方法虽然简化了问题的处理,但仍保留了值函数方法无法保证算法收敛性这一缺点[6].近似动态规划是一种通过近似计算代价函数进行规划的算法.另外,还有一些研究人员使用强化学习的方法处理此类问题,如 Carden 等人提出了在空间和时间达到连续时,使用 Q-学习的方法解决无限 MDP 问题[7],但这些方法的收敛速度通常慢于值函数方法.双重启发式动态规划(dual heuristic programming,简称 DHP)[8]是一种常见的近似动态规划算法,其在给定环境知识的情况下搜索最优动作.策略搜索算法直接在策略空间中进行搜索,可分为参数化和非参数化两种.连续动作强化学习自动机(continuous action reinforcement learning automaton,简称CARLA)[9]是一种典型的非参数化搜索非确定性策略的算法,由于没有采用参数化表示,其对计算要求更高;探索自私的CARLA(exploring selfish CARLA,简称ESCARLA)[10]算法将探索自私的强化学习与CARLA相结合,解决多动作联合的连续动作强化学习问题.连续的评论家-行动者算法(continuous actor-critic learning automaton,简称 CACLA)[11]是一种典型的参数化搜索确定性策略,但由于 CACLA 算法丢弃了时间差分中的重要信息,仅利用时间差分误差决定是否更新最优动作,因此算法收敛速度慢.连续动作近似策略迭代(continuous-action approximate policy iteration,简称CAPI)[12]也是一种参数化搜索确定性策略方法,主要通过求解 Q 值函数的极值来获得最优动作,对近似的基函数有很高的要求.

策略梯度也是一种参数化策略搜索算法,其利用长期累积回报指导策略参数沿着累积回报最大化方向更新,与强化学习的目标相符合,而且策略梯度方法具有可以保证收敛到局部最优解的优点.Willams 等人提出的REINFORCE[13]算法是早期的策略梯度算法,但收敛速度比值函数方法慢,研究人员一直在寻求进一步的突破.Sutton 等人将值函数方法与策略梯度方法相结合,减小了梯度估计过程中的方差,加快了学习速度[6].在强化学习中,可以将值函数方法视为仅有评论家方法(critic-only),将梯度策略方法视为仅有行动者方法(actor-only).强化学习中经典的评论家-行动者(actor-critic,简称 AC)算法则可以视为结合了值函数方法与策略梯度方法.结合

策略梯度和 AC 算法,可以使评论家部分利用当前状态下根据策略所能获得的累计期望奖赏值(V 值)来指导行动者部分采取的策略,以减少梯度中的方差,进而加快策略的学习速度.在学习的过程中,策略梯度到了“平坦区”之后,可能会陷入局部最优,而利用自然梯度则可以有效地避免这一问题.由于具备这样的优势,自然梯度的方法得到了较多的关注.Peters 等人针对常规梯度在梯度估计过程中方差较大的情况,提出了自然梯度与最小二乘时间差分(least squares time difference,简称 LSTD)算法相结合的自然梯度 AC(natural actor-critic,简称 NAC)算法[14].Bhatnagar 等人提出了增量式的自然梯度 AC(incremental natural actor-critic,简称 INAC)算法[15].Degris 针对 INAC 算法利用时间差分 TD(0)算法未能充分利用学习经验的情况,将资格迹与 INAC 结合,提出了带资格迹的 INAC(incremental natural actor-critic with eligibility trace,简称 INAC-E)算法[16],将其并应用到连续空间问题.Sijen 等人提出的真实在线时间差分(true online time difference,简称 TOTD)算法[17]引入了一种新型的前向观点以及资格迹,通过资格迹,能够保证在线情况下前向观点与后向观点完全一致,而且该算法的实验效果比原有的 TD(λ)要好.

本文借鉴 TOTD 算法的前向观点改进 INAC-E 算法:在评论家部分,完全采用 TOTD 算法估计值函数;在行动者部分,利用 TOTD 算法的前向观点和资格迹估计自然梯度.根据上面的思想,本文提出了一个基于核的真实在线增量式自然梯度 AC(true online incremental natural actor-critic,简称 TOINAC)算法,统一了自然梯度估计过程中的前向观点与后向观点,使二者等价,从而使得算法能够用于离线和在线两种情况.最后,结合核方法以及正态的策略分布解决连续空间问题,并且通过仿真对比实验,验证了 TOINAC 算法的可行性.

1 马尔可夫决策过程与核方法

1.1 强化学习的MDP模型

连续空间的问题通常可以采用马尔可夫决策过程模型来建模分析.在 t=0 时刻,环境处于初始状态 $s_0 \in S$,该状态服从状态分布 $p(s_0)$.在 t 时刻,环境状态是 $s_t \in S$,智能体 Agent 根据策略 $\pi(a_t|s_t)=p(u_t|x_t)$选择并执行动作 a_t;在采取了动作 a_t 之后,Agent 得到一个奖赏信号 $r_{t+1}=r(s_t,a_t) \in R$,同时环境也会依据迁移概率 $p(s_{t+1}|s_t,a_t)$迁移到状态 s_{t+1},所获得的折扣累计奖赏为 $R^{\pi}(s)=\sum_{t=0}^{\infty}\gamma^t r(s_t)$. 强化学习的算法通常利用状态值函数 $V^{\pi}(s)$和动作值函数 $Q^{\pi}(s,a)$对策略 π进行评估.状态值函数 $V^{h}(x)$是指根据某个策略 π,从特定状态 s 所获得的累计奖赏,动作值函数 $Q^{\pi}(s,a)$是指根据策略 π,在状态 s 下,采取动作 a 所获得的累计奖赏.采用折扣累计模型的状态值函数[18]以及状态动作值函数[18]分别定义为

$$V^{\pi}(s)=E\{\sum_{i=0}^{\infty}\gamma^i r_{t+i+1} \mid s_t=s,\pi\} \tag{1}$$

$$Q^{\pi}(s,a)=E\{\sum_{i=0}^{\infty}\gamma^i r_{t+i+1} \mid s_t=s,a_t=a,\pi\} \tag{2}$$

其中,$\gamma \in (0,1]$是折扣因子,E 表示期望.

在解决大规模或连续空间问题时,采用表格方式保存状态、状态-动作值函数是不可行的,一般需要采用函数近似的方法.常见的函数近似方法包括核方法[19]、线性函数近似[20]以及神经网络[21]等.其中,核方法可以通过简单的线性方式解决非线性空间问题,并且还具备独特的优势:与线性函数方法相比,核方法需要更少的先验知识;与神经网络相比,核方法训练时间更短.本文算法采用了核方法.

1.2 行动者-评论家(AC)算法

AC 算法具有两个独立的结构:一个用于存储并更新值函数,另一个用于存储所更新的策略.Agent 不再根据值函数选择动作,而根据策略选择动作,策略部分称为行动者(actor);Agent 执行某动作后,更新值函数,利用值函数评价动作的好坏并调整策略;值函数部分称为评论家(critic)[18].在强化学习中,AC 算法将值函数方法与策略梯度方法相结合,具有较强的通用性.AC 算法利用一个独立的存储结构去表示与值函数无关的策略.如图 1 所示,AC 算法的评论家部分通过将状态映射到期望累计奖赏来评估行动者选择策略的好坏,行动者部分根据评论家返回的 TD 误差信号更新策略.

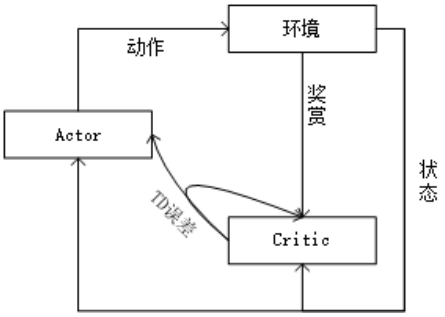

Fig.1 Diagram of AC algorithm architecture

图 1 AC 算法结构示意图

与基于值函数的强化学习方法不同,AC 算法可以学习到一个策略,且当值函数改变时,策略不会发生太大的改变.此外,当动作空间很大甚至连续时,不需要借助值函数选择动作.另一方面,结合策略梯度和 AC 算法,可以通过使评论家部分取得 V 值来指导行动者部分采取的策略以减少梯度中的方差,进而加快策略的学习速度.

1.3 核方法与稀疏化方法

核方法是一种采用核函数进行非线性数据处理的技术,其通过运用非线性映射,增强了学习的非线性逼近能力和泛化能力.核函数 $k(s_i,s_j)$是一个状态空间 $S\times S\to R$ 的映射,通常是一个连续函数.根据 Mercer 定理[22],核具有两个特点.

(1) 核是一个正定核函数,即对于任何有限的样本集合$\{s_1,s_2,\ldots,s_n\}$,核矩阵$[k(s_i,s_j)](1\leqslant i,j\leqslant n)$是正定矩阵.

(2) 存在一个 Hilbert 空间 H 以及一个从状态空间 S 到 H 的映射ϕ,能够使得 $k(s_i,s_j)=\phi^{\top}(s_i)\phi(s_j)$, 只要确定了核函数 $k(\cdot,\cdot)$,即可执行所有在空间 S 的计算.

正是这两个特点,使得大量的研究者投身于核方法的研究中.

数据的稀疏化是从观察数据中选取数据,构建精简数据字典的过程.数据的稀疏化影响着核方法的学习速度与精度,因此,对核方法而言,稀疏化非常重要.常见的稀疏化方法包括近似线性依赖(approximately linear dependence,简称 ALD)[23]、基于核的主成分分析(kernel principal component analysis,简称 KPCA)[24]、新奇准则(novelty criterion,简称 NC)[25]等.

使用 ALD 方法进行数据稀疏化,数据样本集合为$\{s_1,s_2,\ldots,s_n\}$,若已经处理了 t 个样本,得到的数据字典为 $D_t=\{d_1,d_2,\ldots,d_{m_t}\}$, 则在处理第 t+1 个样本时,首先计算:

$$\delta_{t+1}=\min_c\left\|\sum_{d_i\in D_t}c_i\phi(d_i)-\phi(s_{t+1})\right\|^2 \tag{3}$$

利用 $k(s_i,s_j)=\phi^{\top}(s_i)\phi(s_j)$, 可以将公式(3)变化为

$$\delta_{t+1}=\min_c\{\boldsymbol{c}^{\top}K_t\boldsymbol{c}-2\boldsymbol{c}^{\top}\boldsymbol{k}_t(s_{t+1})+k(s_{t+1},s_{t+1})\} \tag{4}$$

其中,$K_t=[k(d_i,d_j)](1\leqslant i,j\leqslant m_t)$,$m_t$ 是数据字典 D_t 的大小;参数向量 $c=[c_1,c_2,\ldots,c_{m_t}]^{\top}$, 向量 $\boldsymbol{k}_t(s_{t+1})=[k(d_1,s_{t+1}),k(d_2,s_{t+1}),\ldots,k(d_{m_t},s_{t+1})]^{\top}$. 公式(4)的最优解:

$$\delta_{t+1}=k(s_{t+1},s_{t+1})-\boldsymbol{k}_t^{\top}(s_{t+1})K_t^{-1}\boldsymbol{k}_t(s_{t+1}) \tag{5}$$

接下来判断δ_{t+1} 是否大于临界值 ν:如果大于临界值 ν,则将 $d_{m_t+1}=s_{t+1}$ 加入数据字典 D,即 $D_{t+1}=D_t\cup d_{m_t+1}$. 公式(1)状态值函数可以近似表示为

$$\tilde{V}^{\pi}(s)=\boldsymbol{v}^{\top}\boldsymbol{k}(s)=\sum_{i=1}^{m}v_ik(d_i,s) \tag{6}$$

其中,m 表示数据字典的长度,$\boldsymbol{k}(s)\in[0,1]^m$ 表示核函数向量,$\boldsymbol{v}\in R^m$ 表示参数向量.

ALD方法可以保证所有基向量近似线性独立,对算法的收敛性而言是一个很重要的前提条件;而且ALD方法的近似效果很好,时间复杂度是 $O(n^2)$,能够满足在线学习要求.本文采用 ALD 方法进行数据稀疏化.

2 策略梯度与自然梯度

值函数方法通过值函数间接地表示策略.策略梯度方法不同于值函数方法,它通过一组策略参数$\boldsymbol{\theta}$直接表示策略π,并且通过梯度去更新策略参数,寻找最优策略.策略梯度方法中,策略参数$\boldsymbol{\theta}$的更新可以表示为$\boldsymbol{\theta}=\boldsymbol{\theta}+\beta\nabla_{\boldsymbol{\theta}}J$,当达到局部最优解时,梯度$\nabla_{\boldsymbol{\theta}}J=\mathbf{0}$.根据文献[6],在策略$\pi$下累积折扣奖赏 J 可以定义为

$$J=E\left\{\sum_{t=0}^{\infty}\gamma^t r_{t+1}\mid s_0,\pi\right\}=\int_X d^{\pi}(s)\int_A \pi(a\mid s)r(s,a)\mathrm{d}a\mathrm{d}s \tag{7}$$

其中,$d^{\pi}(s)=\sum_{t=0}^{\infty}\gamma^t p(s_t=s)$ 是折扣状态分布.累计回报 J 对策略参数$\boldsymbol{\theta}$的梯度$\nabla_{\boldsymbol{\theta}}J$ 定义为

$$\nabla_{\boldsymbol{\theta}}J=\int_S d^{\pi}(s)\int_A \nabla_{\boldsymbol{\theta}}\pi(a\mid s)Q^{\pi}(s,a)\mathrm{d}a\mathrm{d}s \tag{8}$$

虽然策略梯度方法有很强的收敛性保证,但这类方法收敛速度慢、样本效率低[15].这主要是因为梯度评估过程中方差较大,可能会进行负搜索.自然梯度利用策略的 Fisher 信息矩阵 $G(\boldsymbol{\theta})$去线性变化原有梯度,其定义为

$$\tilde{\nabla}_{\boldsymbol{\theta}}J=G^{-1}(\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}}J \tag{9}$$

其中,$G(\boldsymbol{\theta})=\int_X d^{\pi}(s)\int_A \pi(a\mid s)\boldsymbol{\psi}_{s,a}\boldsymbol{\psi}_{s,a}^{\top}\mathrm{d}a\mathrm{d}s$,$\boldsymbol{\psi}_{s,a}=\nabla_{\boldsymbol{\theta}}\log\pi(a|s)$,并且两个梯度之间的夹角不会超过 90°,即 $\cos(\tilde{\nabla}_{\boldsymbol{\theta}}J-\nabla_{\boldsymbol{\theta}}J)\geqslant 0$,这就保证自然梯度可以收敛到下一个局部最优.

对于给定任意一个关于状态 s 的函数 $b(s)$,满足:

$$\int_S d^{\pi}(s)\int_A \nabla_{\theta}\pi(a\mid s)b(s)\mathrm{d}a\mathrm{d}s=\int_S d^{\pi}(s)b(s)\nabla_{\boldsymbol{\theta}}\int_A \pi(a\mid s)\mathrm{d}a\mathrm{d}s=\int_S d^{\pi}(s)b(s)\nabla(1)\mathrm{d}s=0.$$

所以,公式(8)可以改写为

$$\nabla_{\boldsymbol{\theta}}J=\int_S d^{\pi}(s)\int_A \nabla_{\boldsymbol{\theta}}\pi(a\mid s)[Q^{\pi}(s,a)-b(s)]\mathrm{d}a\mathrm{d}s \tag{10}$$

采用函数近似$\boldsymbol{\psi}_{s,a}^{\top}\boldsymbol{w}$去逼近 $Q^{\pi}(s,a)$–$b(s)$的值,并且最小化其均方误差:

$$\varepsilon^{\pi}(\boldsymbol{w})=\int_S d^{\pi}(s)\int_A \pi(a\mid s)[Q^{\pi}(s,a)-b(s)-\boldsymbol{\psi}_{s,a}^{\top}\boldsymbol{w}]^2\mathrm{d}a\mathrm{d}s \tag{11}$$

如引理 1 所述,最优参数 $\boldsymbol{w}^*=\mathrm{argmin}_{\boldsymbol{w}}\varepsilon^{\pi}(\boldsymbol{w})$与给定的函数 $b(s)$无关.

引理 1. 对于任意给定的策略θ,最优梯度参数 $\boldsymbol{w}^*$满足[26]:

$$\boldsymbol{w}^*=G^{-1}(\theta)\int_S d^{\pi}(s)\int_A \pi(a\mid s)[Q^{\pi}(s,a)\boldsymbol{\psi}_{s,a}]\mathrm{d}a\mathrm{d}s \tag{12}$$

证明:均方误差$\varepsilon^{\pi}(\boldsymbol{w})$对 $\boldsymbol{w}$ 的梯度表示为

$$\nabla_{\boldsymbol{w}}\varepsilon^{\pi}(\boldsymbol{w})=-2\int_S d^{\pi}(s)\int_A \pi(a\mid s)[Q^{\pi}(s,a)-b(s)-\boldsymbol{\psi}_{s,a}^{\top}\boldsymbol{w}]\boldsymbol{\psi}_{s,a}\mathrm{d}a\mathrm{d}s \tag{13}$$

令上式等于 0,可以得到:

$$\begin{aligned}\int_S d^{\pi}(s)\int_A \pi(a\mid s)\boldsymbol{\psi}_{s,a}\boldsymbol{\psi}_{s,a}^{\top}\boldsymbol{w}^*\mathrm{d}a\mathrm{d}s&=\int_S d^{\pi}(s)\int_A \pi(a\mid s)Q^{\pi}(s,a)\boldsymbol{\psi}_{s,a}\mathrm{d}a\mathrm{d}s-\int_S d^{\pi}(s)\int_A \pi(a\mid s)b(s)\boldsymbol{\psi}_{s,a}\mathrm{d}a\mathrm{d}s\\&=\int_S d^{\pi}(s)\int_A \pi(a\mid s)Q^{\pi}(s,a)\boldsymbol{\psi}_{s,a}\mathrm{d}a\mathrm{d}s-\int_S d^{\pi}(s)\int_A \nabla_{\theta}\pi(a\mid s)b(s)\mathrm{d}a\mathrm{d}s\\&=\int_S d^{\pi}(s)\int_A \pi(a\mid s)Q^{\pi}(s,a)\boldsymbol{\psi}_{s,a}\mathrm{d}a\mathrm{d}s-\int_S d^{\pi}(s)b(s)\nabla_{\theta}(1)\mathrm{d}a\mathrm{d}s\\&=\int_S d^{\pi}(s)\int_A \pi(a\mid s)Q^{\pi}(s,a)\boldsymbol{\psi}_{s,a}\mathrm{d}a\mathrm{d}s.\end{aligned}$$

得证. □

在给定最优参数 $\boldsymbol{w}^*$的情况下,通过调整 $b(s)$的值,可以进一步最小化均方误差$\varepsilon^{\pi}(\boldsymbol{w}^*)$,从而可以降低动作值函数的方差.将$\varepsilon^{\pi}(\boldsymbol{w}^*)$视为关于 b 的函数,能够获得 $b^*=\mathrm{argmin}_b\varepsilon^{\pi}(\boldsymbol{w}^*)$.

引理 2. 对于任意给定策略π,最小方差基线 $b^*(s)$与状态值函数 $V^{\pi}(s)$相关[26].

证明:对于任意状态 $s\in S$,令$\varepsilon^{\pi,s}(\boldsymbol{w}^*)$表示为

$$\varepsilon^{\pi,s}(\boldsymbol{w}^*)=\int_A \pi(a\mid s)[Q^{\pi}(s,a)-\boldsymbol{\psi}_{s,a}^{\top}\boldsymbol{w}^*-b(s)]^2\mathrm{d}a \tag{14}$$

对均方误差$\varepsilon^{\pi,s}(\boldsymbol{w}^*)$关于 $b(s)$求偏导:

$$\frac{\partial \varepsilon^{\pi,s}(\boldsymbol{w}^*)}{\partial b(s)} = -2\int_A \pi(a\,|\,s)[Q^\pi(s,a) - \boldsymbol{\psi}_{s,a}^\top \boldsymbol{w}^* - b(s)].$$

令上式等于 0,得到:

$$\begin{aligned} b^*(s) &= \int_A \pi(a\,|\,s)Q^\pi(s,a)\mathrm{d}a - \int_A \pi(a\,|\,s)\boldsymbol{\psi}_{s,a}^\top \boldsymbol{w}^* \mathrm{d}a \\ &= \int_A \pi(a\,|\,s)Q^\pi(s,a)\mathrm{d}a - \boldsymbol{w}^* \int_A \pi(a\,|\,s)\boldsymbol{\psi}_{s,a}^\top \mathrm{d}a \\ &= \int_A \pi(a\,|\,s)Q^\pi(s,a)\mathrm{d}a - \boldsymbol{w}^{*\top} \nabla_\theta \int_A \pi(a\,|\,s)\mathrm{d}a \\ &= \int_A \pi(a\,|\,s)Q^\pi(s,a)\mathrm{d}a \\ &= V^\pi(s). \end{aligned}$$

得证. □

采用函数近似 $\boldsymbol{\psi}_{s,a}^\top \boldsymbol{w}$ 去逼近 $Q^\pi(s,a)-b(s)$的值,并且最小化其均方误差为公式(11),根据引理 2 所述,当 $b(s)=V^\pi(s)$时,可以最小化公式(11)的均方误差.可以看出,相对于采用函数近似 $\boldsymbol{w}^\top \boldsymbol{\psi}_{sa}$ 去近似 $Q^\pi(s,a)$,采用近似优势函数 $A^\pi(s,a)=Q^\pi(s,a)-V^\pi(s)$则显得更有意义.所以,公式(10)可以改写成

$$\nabla_{\boldsymbol{\theta}} J = \int_S d^\pi(s)\int_A \pi(a\,|\,s)\boldsymbol{\psi}_{s,a}\boldsymbol{\psi}_{s,a}^\top \mathrm{d}a\mathrm{d}s\boldsymbol{w} \tag{15}$$

根据公式(9),自然梯度重新定义为

$$\tilde{\nabla}_{\boldsymbol{\theta}} J = G(\boldsymbol{\theta})^{-1}\int_S d^\pi(s)\int_A \pi(a\,|\,s)\boldsymbol{\psi}_{s,a}\boldsymbol{\psi}_{s,a}^\top \mathrm{d}a\mathrm{d}s\boldsymbol{w} = \boldsymbol{w} \tag{16}$$

因此,在计算自然梯度时,不需要计算 Fisher 信息矩阵 $G(\boldsymbol{\theta})$,只需要计算参数 $\boldsymbol{w}$.在后续部分,将详细介绍如何计算参数 $\boldsymbol{w}$,这也正是本文的创新工作之一.

3 真实在线自然梯度 AC 算法

3.1 截断式的λ-回报值函数

在强化学习的方法中,有前向观点和后向观点两种方法.所谓前向观点是指:在一组状态流的某个状态上,向前看每个状态来决定其更新;在每次从一个状态向前看并更新状态之后,移到下一个状态,并且不再处理之前的状态.前向观点对每个访问过的状态,预测所有的未来奖赏并决定怎样把它们最好地结合在一起.在前向观点的方法中,未来的状态会反复被访问和处理.所谓后向观点是指:在一组状态流的某个状态上,计算 TD 误差,并将这些误差分配到前面已经访问过的状态.后向观点每个时刻都查看当前的 TD 误差,并且根据那个时刻状态的资格迹把误差向后分配到前面的状态上.前向观点和后向观点是有区别的:前向观点是非因果的、理论的观点,每步所使用的知识是后面几步以后将要发生的事情,因此,前向观点的方法很难直接应用;后向观点是则提供了一种因果的、增量机制来逼近前向观点,概念上和计算上都比较简单.通过结合 TD 误差和迹,TD(λ)能够统一前向观点和后向观点.但是,传统 TD(λ)算法只有在离线处理的情况下,其前向观点和后向观点得到的结果才一致;而 TOTD 算法采用了新型前向观点的方法,使其能够在离线处理和在线处理的情况下,前向观点得到的结果和后向观点得到的结果均保持一致.同时,根据 Sutton 等人的工作[17]可知,TOTD 算法比起传统的 TD(λ)有着更好的表现.因此,为了更快、更准地计算出 TD 误差,本文采用 TOTD 方法评估策略.

TD(0)算法对值函数的更新 $\tilde{V}(s_t) = r_{t+1} + \gamma \boldsymbol{v}^\top \boldsymbol{k}(s_{t+1})$ 比较简单,而蒙特卡罗算法对值函数更新 $V(s_t) = \sum_{i=t}^{T} \gamma^{i-t} r_{i+1}$ 更精确,其中,s_{t+1} 为 s_t 的后续状态,T 为情节的最大步数.TD(λ)算法结合了这两种算法的优点,通过当前状态之后的多步状态对值函数进行更新,称作 n 步更新,如式(17)所示.

$$R_{t,v}^{(n)} = \sum_{i=1}^{n} \gamma^{i-1} r_{t+i} + \gamma^n \boldsymbol{v}^\top \boldsymbol{k}(s_{t+n}) \tag{17}$$

对情节式问题而言,在算法的执行过程中,可以对多个不同 n 步更新进行加权平均,称作λ-回报:

$$R_t^{\lambda}=(1-\lambda)\sum_{n=1}^{T-t-1}\lambda^{n-1}R_{t,v}^{(n)}+\lambda^{T-t-1}R_{t,v}^{(T-t)} \tag{18}$$

作为当前状态的值函数的估计值.TOTD 算法稍稍改变了λ-回报,采用一种截断式的λ-回报:

$$R_t^{\lambda|t'}=(1-\lambda)\sum_{n=1}^{t'-t-1}\lambda^{n-1}R_{t,v_{t+n-1}}^{(n)}+\lambda^{t'-t-1}R_{t,v_{t'-t}}^{(t'-t)} \tag{19}$$

作为值函数的估计.

3.2 真实在线自然梯度的评论家部分

真实在线自然梯度的评论家部分采用上述前向观点来预测当前状态 s 的期望累计奖赏 $V(s)$,或者说是最优化参数 $\boldsymbol{v}$,更新规则如下:

$$\boldsymbol{v}_{t,k}=\boldsymbol{v}_{t,k-1}+\alpha_{k-1}[R_{k-1}^{\lambda|t}-\boldsymbol{v}_{t,k-1}^{\top}\boldsymbol{k}(s_{k-1})]\boldsymbol{k}(s_{k-1}) \tag{20}$$

其中,$0<k\leqslant t$,并且 $\boldsymbol{v}_{t,0}$ 等于参数 v 的初始值.参数 $\boldsymbol{v}_{i,j}(i<j)$只是临时的中间参数.从公式(20)可以看出:需要 t 次计算才能计算出参数 $\boldsymbol{v}_t=\boldsymbol{v}_{t,t}$,并且需要保存所有的观察样本.为了能够在线计算,通常采用后向观点对值函数进行更新,即利用当前值函数的 TD 误差对之前遇到的状态值函数进行更新.将前向观点与后向观点统一,得到真实在线 TD(λ)算法.其更新规则如下:

$$\delta_t=r_{t+1}+\gamma\boldsymbol{v}_t^{\top}\boldsymbol{k}(s_{t+1})-\boldsymbol{v}_{t-1}^{\top}\boldsymbol{k}(s_t) \tag{21}$$

$$\boldsymbol{e}_t^{v}=\gamma\lambda\boldsymbol{e}_{t-1}^{v}+\alpha_t k(s_t)-\alpha_t\gamma\lambda[\boldsymbol{k}^{\top}(s_t)\boldsymbol{e}_{t-1}^{v}]\boldsymbol{k}(s_t) \tag{22}$$

$$\boldsymbol{v}_{t+1}=\boldsymbol{v}_t+\delta_t\boldsymbol{e}_t^{v}+\alpha_t[\boldsymbol{v}_{t-1}^{\top}\boldsymbol{k}(s_t)-\boldsymbol{v}_t^{\top}\boldsymbol{k}(s_t)]\boldsymbol{k}(s_t) \tag{23}$$

其中,δ表示 TD 误差;$\boldsymbol{e}^{v}$ 表示资格迹,反映了当前状态之前所遇到的所有状态对当前差分值的“贡献度”.

3.3 真实在线自然梯度的行动者部分

真实在线自然梯度的行动者部分采用参数$\boldsymbol{\theta}$表示策略π分布,策略参数$\boldsymbol{\theta}$的更新可以表示为

$$\boldsymbol{\theta}_{t+1}=\boldsymbol{\theta}_t+\beta_t\boldsymbol{w}_{t+1} \tag{24}$$

表示沿着目标函数的自然梯度方向更新策略参数.当到达局部最优解时,自然梯度 $\boldsymbol{w}=0$.所以,算法的核心问题是自然梯度 $\boldsymbol{w}$ 的求解.根据上一节的介绍,可以转化为对优势函数 $A^{\pi}(s,a)=Q^{\pi}(s,a)-V^{\pi}(s)$逼近.

采用上述前向观点去近似求解自然梯度 $\boldsymbol{w}$,其更新公式表示为

$$\boldsymbol{w}_{t,k}=\boldsymbol{w}_{t,k-1}+\alpha_{k-1}[A_{k-1}^{\lambda|t}-\boldsymbol{w}_{t,k-1}^{\top}\boldsymbol{\psi}_{s_{k-1}a_{k-1}}]\boldsymbol{\psi}_{s_{k-1}a_{k-1}} \tag{25}$$

其中,$A_{k-1}^{\lambda|t}=R_{k-1}^{\lambda|t}-V(s_{k-1})$ 是截断式λ-优势函数.从公式(25)可以看出:需要 t 次计算才能计算出自然梯度 $\boldsymbol{w}_t=\boldsymbol{w}_{t,t}$,并且需要保存所有的观察样本.为了简化计算过程,采用向后观点对值函数进行更新.其更新公式如下:

$$\boldsymbol{e}_t^{a}=\gamma\lambda\boldsymbol{e}_{t-1}^{a}+\alpha_t\boldsymbol{\psi}_{s_t,a_t}-\alpha_t\gamma\lambda(\boldsymbol{\psi}_{s_t,a_t}^{\top}\boldsymbol{e}_{t-1}^{a})\boldsymbol{\psi}_{s_t,a_t} \tag{26}$$

$$\boldsymbol{w}_{t+1}=(I-\alpha_t\boldsymbol{\psi}_{s_t,a_t}\boldsymbol{\psi}_{s_t,a_t}^{\top})\boldsymbol{w}_t+\delta_t\boldsymbol{e}_t^{a}+\alpha_t[\boldsymbol{v}_{t-1}\boldsymbol{k}(s_t)-\boldsymbol{v}_t\boldsymbol{k}(s_t)]\boldsymbol{\psi}_{s_t,a_t} \tag{27}$$

接下来,将证明这种带资格迹的更新方式与使用截断式λ-优势函数更新是一样的.

定理 1. 截断式优势函数 $A_t^{\lambda|t'}$ 与 Critic 部分的公式(21) $\delta_t=r_{t+1}+\gamma\boldsymbol{v}_t^{\top}\boldsymbol{k}(s_{t+1})-\boldsymbol{v}_{t-1}^{\top}\boldsymbol{k}(s_t)$、TD 误差相关,关系如下:

$$A_t^{\lambda|t'+1}-A_t^{\lambda|t'}=(\gamma\lambda)^{t'-t}\delta_{t'} \tag{28}$$

证明:

$$\begin{aligned}A_t^{\lambda|t'+1}-A_t^{\lambda|t'}&=R_t^{\lambda|t'+1}-R_t^{\lambda|t'}-(V(s_t)-V(s_t))\\&=R_t^{\lambda|t'+1}-R_t^{\lambda|t'}\\&=\lambda^{t'-t}[R_{t,v_{t'}}^{(t'-t+1)}-R_{t,v_{t'-1}}^{(t'-t)}]\\&=(\gamma\lambda)^{t'-t}[r_{t'+1}+\gamma\boldsymbol{v}_{t'}^{\top}\boldsymbol{k}(s_{t'+1})-\boldsymbol{v}_{t'-1}^{\top}\boldsymbol{k}(s_{t'})]\\&=(\gamma\lambda)^{t'-t}\delta_{t'}.\end{aligned}$$

得证. □

定理 2. 对于任意时间步 t,都有式(29)成立.

$$\boldsymbol{w}_{t+1,t}-\boldsymbol{w}_{t,t}=\gamma\lambda\delta_t\boldsymbol{e}_{t-1}^a \tag{29}$$

证明:首先,先证明一个更一般的情况.对于任意给定 i,$1\leqslant i\leqslant t$,公式(30)成立.

$$\boldsymbol{w}_{t+1,i}-\boldsymbol{w}_{t,i}=(\gamma\lambda)^{t+1-i}\delta_t\boldsymbol{e}_{i-1}^a \tag{30}$$

采用数学归纳法证明.首先,当 i=1 时,使用公式(25)计算 $\boldsymbol{w}_{t+1,1}-\boldsymbol{w}_{t,1}$.

$$\boldsymbol{w}_{t+1,1}-\boldsymbol{w}_{t,1}=\boldsymbol{w}_{t+1,0}-\boldsymbol{w}_{t,0}+\alpha_0[A_0^{\lambda|t+1}-A_0^{\lambda|t}-(\boldsymbol{w}_{t+1,0}-\boldsymbol{w}_{t,0})^{\top}\boldsymbol{\psi}_{s_0,a_0}]\boldsymbol{\psi}_{s_0,a_0}.$$

由于 $\boldsymbol{w}_{t+1,0}-\boldsymbol{w}_{t,0}=\boldsymbol{w}_{init}$, $\boldsymbol{e}_{-1}^a=\boldsymbol{0}$ 以及定理 1,可以得到:

$$\boldsymbol{w}_{t+1,1}-\boldsymbol{w}_{t,1}=(\gamma\lambda)^t\delta_t\alpha_0\boldsymbol{\psi}_{s_0,a_0}=(\gamma\lambda)^t\delta_t\boldsymbol{e}_0^a.$$

假设当 i=k,$1<k<t$ 时,公式(31)成立:

$$\boldsymbol{w}_{t+1,k}-\boldsymbol{w}_{t,k}=(\gamma\lambda)^{t+1-k}\delta_t\boldsymbol{e}_{k-1}^a \tag{31}$$

当 i=k+1 时,使用公式(25)计算:

$$\boldsymbol{w}_{t+1,k+1}-\boldsymbol{w}_{t,k+1}=\boldsymbol{w}_{t+1,k}-\boldsymbol{w}_{t,k}+\alpha_k[A_k^{\lambda|t+1}-A_k^{\lambda|t}-(\boldsymbol{w}_{t+1,k}-\boldsymbol{w}_{t,k})^{\top}\boldsymbol{\psi}_{s_k.a_k}]\boldsymbol{\psi}_{s_k.a_k}.$$

将公式(31)以及式(28)代入上式,得到:

$$\begin{aligned}\boldsymbol{w}_{t+1,k+1}-\boldsymbol{w}_{t,k+1}&=(\gamma\lambda)^{t+1-k}\delta_t\boldsymbol{e}_{k-1}^a+\alpha_k[(\gamma\lambda)^{t-k}\delta_t-(\gamma\lambda)^{t+1-k}\delta_t\boldsymbol{\psi}_{s_k,a_k}^{\top}\boldsymbol{e}_{k-1}^a]\boldsymbol{\psi}_{s_k.a_k}\\&=(\gamma\lambda)^{t-k}\delta_t[\gamma\lambda\boldsymbol{e}_{k-1}^a+\alpha_k\boldsymbol{\psi}_{s_k.a_k}-\alpha_k\gamma\lambda(\boldsymbol{\psi}_{s_k,a_k}^{\top}\boldsymbol{e}_{k-1}^a)\boldsymbol{\psi}_{s_k.a_k}]\\&=(\gamma\lambda)^{t-k}\delta_t\boldsymbol{e}_k^a.\end{aligned}$$

综上可知,公式(30)成立.

令 i=t,将其代入公式(30),可以得到公式(29)成立.得证. □

定理 3. 对于任意时间步 t,使用截断式λ-优势函数更新梯度 $\boldsymbol{w}_{t,t}$,是等同于下式对 $\boldsymbol{w}_t$ 的更新:

$$\boldsymbol{w}_{t+1}=(I-\alpha_t\boldsymbol{\psi}_{s_t,a_t}\boldsymbol{\psi}_{s_t,a_t}^{\top})\boldsymbol{w}_t+\delta_t\boldsymbol{e}_t^a+\alpha_t[\boldsymbol{v}_{t-1}\boldsymbol{k}(s_t)-\boldsymbol{v}_t\boldsymbol{k}(s_t)]\boldsymbol{\psi}_{s_t,a_t} \tag{32}$$

证明:根据公式(25),

$$\boldsymbol{w}_{t+1,t+1}=\boldsymbol{w}_{t+1,t}+\alpha_t[A_t^{\lambda|t+1}-\boldsymbol{w}_{t+1,t}^{\top}\boldsymbol{\psi}_{s_t,a_t}]\boldsymbol{\psi}_{s_t,a_t} \tag{33}$$

将定理 2 代入公式(33),我们可以得到:

$$\begin{aligned}\boldsymbol{w}_{t+1,t+1}&=\boldsymbol{w}_{t,t}+\gamma\lambda\delta_t\boldsymbol{e}_{t-1}^a+\alpha_t[r_{t+1}+\gamma\boldsymbol{v}_t\boldsymbol{k}(s_{t+1})-\boldsymbol{v}_{t-1}\boldsymbol{k}(s_t)+\boldsymbol{v}_{t-1}\boldsymbol{k}(s_t)-\boldsymbol{v}_t\boldsymbol{k}(s_t)]\boldsymbol{\psi}_{s_t,a_t}-\alpha_t[(\boldsymbol{w}_{t,t}+\gamma\lambda\delta_t\boldsymbol{e}_{t-1}^a)^{\top}\boldsymbol{\psi}_{s_t,a_t}]\boldsymbol{\psi}_{s_t,a_t}\\&=\boldsymbol{w}_{t,t}-\alpha_t\boldsymbol{\psi}_{s_t,a_t}^{\top}\boldsymbol{\psi}_{s_t,a_t}\boldsymbol{w}_{t,t}+\delta_t[\gamma\lambda\boldsymbol{e}_{t-1}^a+\alpha_t\boldsymbol{\psi}_{s_t,a_t}-\alpha_t\gamma\lambda\boldsymbol{\psi}_{s_t,a_t}^{\top}\boldsymbol{e}_{t-1}^a\boldsymbol{\psi}_{s_t,a_t}]+\alpha_t[\boldsymbol{v}_{t-1}\boldsymbol{k}(s_t)-\boldsymbol{v}_t\boldsymbol{k}(s_t)]\boldsymbol{\psi}_{s_t,a_t}\\&=\boldsymbol{w}_{t,t}-\alpha_t\boldsymbol{\psi}_{s_t,a_t}^{\top}\boldsymbol{\psi}_{s_t,a_t}\boldsymbol{w}_{t,t}+\delta_t\boldsymbol{e}_t^a+\alpha_t[\boldsymbol{v}_{t-1}\boldsymbol{k}(s_t)-\boldsymbol{v}_t\boldsymbol{k}(s_t)]\boldsymbol{\psi}_{s_t,a_t}\\&=(I-\alpha_t\boldsymbol{\psi}_{s_t,a_t}\boldsymbol{\psi}_{s_t,a_t}^{\top})\boldsymbol{w}_{t,t}+\delta_t\boldsymbol{e}_t^a+\alpha_t[\boldsymbol{v}_{t-1}\boldsymbol{k}(s_t)-\boldsymbol{v}_t\boldsymbol{k}(s_t)]\boldsymbol{\psi}_{s_t,a_t}.\end{aligned}$$

得证. □

根据上述证明,带资格迹 $\boldsymbol{e}^a$ 的向后观念的更新与使用截断式λ-优势函数更新是一样的.当λ=0 时,自然梯度更新公式(27)与文献[26]里算法 3 更新公式完全一致.可以看出,资格迹中包含了所有历史信息,这些历史信息能够有效地分配误差,从而加快学习经验的传播,能够加快算法收敛.

根据上述算法过程的描述,基于核的真实在线增量式自然梯度 AC 算法可以总结为算法 1.

算法 1. 基于核的真实在线增量式自然梯度 AC(TOINAC)算法.

输入:核函数 $k(\cdot,\cdot)$;临界值 ν;初始学习步长α_0,β_0;步长参数α_c,β_c;折扣因子γ;参数λ;数据样本集合$\{s_1,s_2,\ldots,s_n\}$.

输出:值函数参数 $\boldsymbol{v}$,策略参数$\boldsymbol{\theta}$.

1. 初始化数据字典 D=NULL,值函数参数 $\boldsymbol{v}=\boldsymbol{v}_{init}$,自然梯度 $\boldsymbol{w}=\boldsymbol{w}_{init}$,策略参数$\boldsymbol{\theta}=\boldsymbol{\theta}_{init}$,情节数 $episode$=0,t=0
2. **FOR** each record $s_i\in S$ **DO**
3. 通过公式(5),计算δ_i
4. **IF** $\delta_i>\nu$ **Do**

5. $D \leftarrow D \cup \{s_t\}$
6. **END IF**
7. **END FOR**
8. **LOOP**:
9. $s=s_0, \boldsymbol{e}^v=\boldsymbol{0}, \boldsymbol{e}^a=\boldsymbol{0}$
10. $\tilde{V}(s) \leftarrow \boldsymbol{v}^\top \boldsymbol{k}(s)$
11. **LOOP**:
12. 根据策略分布 $\pi(a|s)$,选择动作 a
13. 执行动作 a,得到下一状态 s'和奖赏 r
14. $\tilde{V}(s') \leftarrow \boldsymbol{v}^\top \boldsymbol{k}(s')$
15. $\delta \leftarrow r + \gamma \tilde{V}(s') - \tilde{V}(s)$
16. $\boldsymbol{e}^v \leftarrow \gamma\lambda \boldsymbol{e}^v + \alpha[1 - \gamma\lambda \boldsymbol{k}^\top(s)\boldsymbol{e}^v]\boldsymbol{k}(s)$
17. $\boldsymbol{v} \leftarrow \boldsymbol{v} + \delta \boldsymbol{e}^v + \alpha[\tilde{V}(s) - \boldsymbol{v}^\top \boldsymbol{k}(s)]\boldsymbol{k}(s)$
18. $\boldsymbol{e}^a \leftarrow \gamma\lambda \boldsymbol{e}^a + \alpha[1 - \gamma\lambda \boldsymbol{\psi}_{s,a}^\top \boldsymbol{e}^a]\boldsymbol{\psi}_{s,a}$
19. $\boldsymbol{w} \leftarrow (I - \alpha \boldsymbol{\psi}_{s,a}\boldsymbol{\psi}_{s,a}^\top)\boldsymbol{w} + \delta \boldsymbol{e}^a + \alpha[\tilde{V}(s) - \boldsymbol{v}^\top \boldsymbol{k}(s)]\boldsymbol{\psi}_{s,a}$
20. $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \beta \boldsymbol{w}$
21. $\tilde{V}(s) \leftarrow \tilde{V}(s')$
22. $s \leftarrow s'$
23. $t \leftarrow t+1$
24. 根据公式(34),更新步长 α, β
25. **UNTIL** s 是终止状态
26. $episode \leftarrow episode+1$
27. **UNTIL** 算法满足终止条件,$\boldsymbol{\theta}$收敛或 $episode=T$ 最大情节数
28. **RETURN** $\boldsymbol{v}, \boldsymbol{\theta}$

该算法过程中的学习步长 α, β是可变的.根据文献[26],对两个时间维度的随机算法的收敛性证明,要求步长 α, β满足 $\sum_t \alpha_t = \sum_t \beta_t = \infty, \sum_t \alpha_t^2, \sum_t \beta_t^2 < \infty$,并且 $\beta_t = o(\alpha_t)$.其中,β是α的高阶无穷小,这就意味着β趋于 0 的速度要比α的快.为此,本文采用的 Critic 的步长参数$\{\alpha_t\}$和 Actor 的步长参数$\{\beta_t\}$分别表示为

$$\alpha_t = \frac{\alpha_0 \alpha_c}{\alpha_c + t^{2/3}},\ \beta_t = \frac{\beta_0 \beta_c}{\beta_c + t} \tag{34}$$

其中,t 表示时间步;α_c, β_c 都是常数.

4 策略分布

策略梯度类算法的性能在很大程度上依赖于策略分布的选择.众所周知,策略分布是解决探索与利用的平衡问题的一个重要方法.所以,策略分布的质量直接影响算法收敛速度以及学习到的策略的质量.对于离散动作而言,最常用的策略分布是 Gibbs 分布.对连续动作而言,策略分布可以是ε-greedy,也可以是正态分布,而正态分布的效果通常会比ε-greedy 效果好[11].所以,本文采用正态分布作为策略分布.其概率密度函数表示为

$$\pi(a \mid s) = \frac{1}{\sqrt{2\pi\sigma^2(s)}} \exp\left(-\frac{(a-\mu(s))^2}{2\sigma^2(s)}\right) \tag{35}$$

其中, $\mu(s) = \boldsymbol{\theta}_\mu^\top \boldsymbol{k}_\mu(s)$ 和 $\sigma(s) = \exp(\boldsymbol{\theta}_\sigma^\top \boldsymbol{k}_\sigma(s))$ 分别表示策略$\pi(\cdot|s)$分布的均值和标准差.策略参数以及与之对应的特征向量分别表示为 $\boldsymbol{\theta} = (\boldsymbol{\theta}_\mu^\top, \boldsymbol{\theta}_\sigma^\top)^\top$ 和 $\boldsymbol{k}_a(s) = (\boldsymbol{k}_\mu^\top(s), \boldsymbol{k}_\sigma^\top(s))^\top$. 为了方便计算,$\boldsymbol{k}_\mu(\cdot), \boldsymbol{k}_\sigma(\cdot)$以及 $\boldsymbol{k}(\cdot)$使用同一组特征向量.

策略向量$\pi(a|s)$的向量$\boldsymbol{\psi}_{sa}=(\nabla_{\theta_\mu}\log\pi(a\,|\,s)^\top,\nabla_{\theta_\sigma}\log\pi(a\,|\,s)^\top)^\top$, 其中,

$$\nabla_{\theta_\mu}\log\pi(a\,|\,s)=\frac{1}{\sigma^2(s)}(a-\mu(s))\boldsymbol{k}_\mu(s) \tag{36}$$

$$\nabla_{\theta_\sigma}\log\pi(a\,|\,s)=\left(\frac{(a-\mu(s))^2-\sigma^2(s)}{\sigma^3(s)}\right)\boldsymbol{k}_\sigma(s) \tag{37}$$

5 实验结果与分析

本节通过对具有代表性的连续空间问题:平衡杆(Cart Pole)问题、Mountain Car 问题以及 Acrobot 问题进行仿真实验测试来验证 TOINAC 算法的可行性.在实验中,算法采用核方法和 ALD 方法,核函数都是高斯核函数:

$$k(s,d_i)=\exp\left(-\frac{\|\,s-d_i\,\|^2}{\sigma_a^2}\right),$$

其中,d_i是通过 ALD 方法构建的数据字典 D 里面的状态.

5.1 平衡杆问题

平衡杆问题是强化学习中经典的连续空间问题.如图 2 所示,杆子连接在小车上,且可随意转动,不计任何摩擦力.起初木杆竖直矗立在小车上,随后,通过水平方向上对小车施加力以保证木杆不倒.Agent 通过学习得到策略,使杆子在尽可能长的时间步数内保持不倒.

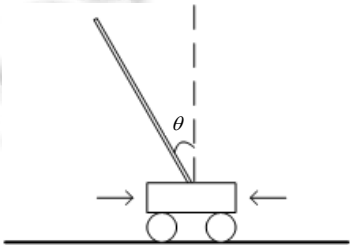

Fig.2 Diagram of cart pole problem

图 2 平衡杆问题示意图

通过MDP对问题进行建模,该问题的状态可以表示为$[\theta,\dot{\theta}]^\top$, 其中,$\theta$如图2所示,是杆子与竖直线的角度,$\dot{\theta}$是角度$\theta$的角速度.任意时刻对小车施加力 $a\in[-50,50]$,状态会发生转移,其转移函数如下:

$$\theta_{t+1}=\theta_t+\dot{\theta}_t\Delta t,$$

$$\dot{\theta}_{t+1}=\dot{\theta}_t+\ddot{\theta}_{t+1}\Delta t,$$

$$\ddot{\theta}_{t+1}=\frac{g\sin(\theta_t)-\cos(\theta_t)\left[\dfrac{a_t+ml\dot{\theta}_t^2\sin(\theta_t)}{m+M}\right]}{\dfrac{4}{3}l-\dfrac{ml\cos^2(\theta_t)}{m+M}},$$

其中,$\ddot{\theta}$表示角加速度,g=9.8m/s^{2}表示重力加速度,m=2.0kg表示木杆的质量,l=1.0m表示木杆的长度,M=8.0kg表示小车的质量,Δt=0.1s 表示两个时间步之间的间隔.在时间步 t 时刻,采取动作 a_t,如果木杆与竖直方向的角度$-\pi/2<\theta_{t+1}<\pi/2$,立即奖赏 r=0;否则,r=−1,且认为木杆倒下,操作失败情节结束.如果木杆一直没有倒下,并保持3 000 个时间步,则认为操作成功情节结束.

在本实验中,TOINAC 算法与各类可以解决连续问题的算法进行比较,如 CAQ,CACLA,DHP,IAC,NAC.DHP 算法是一种近似动态规划算法,其采用神经网络进行函数逼近,其 Critic 网络结构是 2-10-2,Actor 网络是 2-8-1.除了 DHP 算法外,其余 5 种算法均采用 ALD 和核方法来进行函数逼近,其参数设置为σ_a=0.35,ν=0.001.CAQ 算法将动作空间平均划分为{−50,−25,0,25,50}.CACLA 算法 Critic 的学习步长α=0.9,Actor 的学习步长β=0.2.IAC,

NAC 以及 TOINAC 算法都是策略梯度算法,都采用高斯策略分布来选择动作,为了方便计算和比较,σ=5.0,λ=0.3,γ=0.9.其中,NAC 算法是基于 LSTD 算法,其参数遗忘因子设为 0.3,学习步长为 0.8;IAC 算法与 TOINAC 算法都是基于 TD 算法的,其学习步长参数设置为α_0=0.7,β_0=0.5,α_c=9000,β_c=9000.

由于每个情节的累计奖赏与步数成正比,所以可以通过比较每个情节的步数来比较算法的收敛效果的好坏.如图 3 所示:TOINAC 算法的收敛最快,且可以稳定的最大步数为 3 000 步.DHP 算法有着较好的性能,并且在 150 个情节左右有着更好的效果,这主要是因为该算法是一种模型已知的算法,利用了很多模型知识;但是 TOINAC 算法的表现上升坡度比 DHP 算法要陡峭,所以 TOINAC 算法可以最先收敛到最大步数.比较 3 种策略梯度算法,可发现 TOINAC 算法和 NAC 算法表现上升坡度几乎一致,且比其他普通梯度算法都要陡峭;在样本量比较少的时候,TOINAC 算法基于的 TOTD 学习比 IAC 算法基于的 TD 学习以及 NAC 算法基于的 LSTD 学习速度都快.3 种策略梯度算法都比 CACLA 算法表现好,主要是因为策略梯度采用累积奖赏或平均奖赏指导策略更新.离散化算法 CAQ 算法表现不好,500 个情节只是稍微有一点学习效果,在学习了 1 000 多个情节之后,算法才成功.

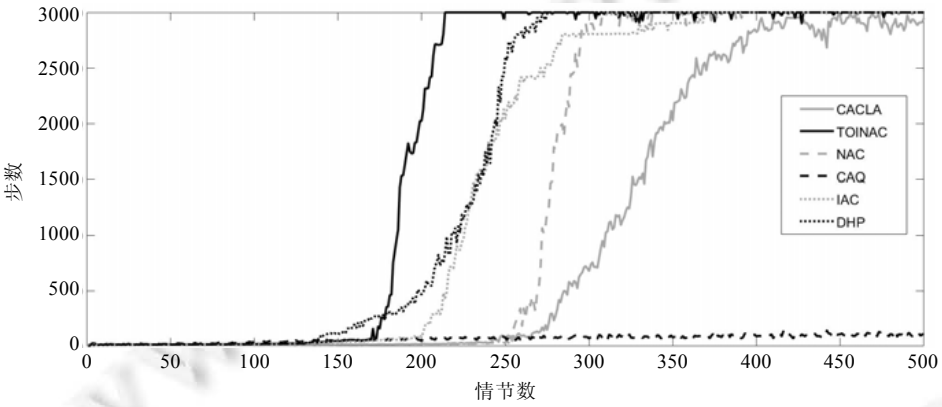

Fig.3 Comparison of the steps of different algorithms in the cart pole problem experiment

图 3 平衡杆问题实验中不同算法的步数比较

为了进一步验证上面的猜想,在表 1 中列出了这 6 种算法 30 次实验 500 情节中的表现.首次成功表示 500 个情节中第 1 次达到 3 000 步的情节数.成功率表示这 15 000 个情节中成功的概率.平均步数表示 15 000 个情节平均每个情节执行了多少步.可以发现,TOINAC 算法在成功率以及平均步数表现都是第一,首次成功仅低于 DHP 算法.

Table 1 Performance comparison of 6 algorithms in the cart pole problem experiment (500 episodes)

表 1 平衡杆问题实验 6 种算法的表现比较(500 个情节)

算法名称	首次成功	成功率(%)	平均步数
CACLA	243	32.4	1 014.6
TOINAC	171	59.2	1 854.9
NAC	256	43.5	1 337.6
CAQ	–	0	66.9
INAC	202	50.0	1 528.0
DHP	140	53.9	1 653.6

5.2 Mountain Car问题

Mountain Car 问题是强化学习问题中经典的情节式的连续空间问题,其示意图如图 4 所示,小车的任务是在动力不足的情况下,从坡底 S 以尽量短的时间到达终点 G.这个问题的难点在于:小车的动力不足以克服重力影响,从坡底直接加速到坡顶,只能通过左右来回加速多次到达较高位置,再加速到达终点.MDP 对问题进行建

模,状态可以表示为 $[x,v]^{\top}$,其中,小车的水平位置 $x\in[-1.2,0.5]$,小车的水平速度 $v\in[-0.07,0.07]$.任意时刻对小车施加水平方向的力 $a\in[-1,1]$,状态都会发生迁移,迁移函数为

$$v_{t+1}=bound[v_t+0.001a_t-g\cos(3x_t)],$$

$$x_{t+1}=bound[x_t+v_{t+1}],$$

其中,g=0.0025 是与重力有关的系数.当小车水平位置 x<0.5 时,系统的奖赏是−1;否则,小车到达终点,奖赏为 0.

在本实验中,TOINAC 算法与几种累加式的策略梯度算法比较,比如 IAC,INAC 以及带资格迹的 INAC-E 算法.该实验中,几乎所有参数设置都一样,用于函数逼近的相关参数 $\sigma_a=[0.3,0.02]^{\top}$, ν=0.001;步长相关参数 α_0=0.7,β_0=0.3,α_c=500,β_c=500;折扣因子 γ=0.9.带资格迹的算法 λ=0.3.图 5 中所有的曲线都是各种算法每学习 500 步就评估策略的表现,每个算法独立执行了 50 次.可以发现,3 种自然梯度的策略梯度算法(TOINAC, INAC-E,INAC)比普通梯度的策略梯度算法(IAC)下降速度要快.显然,两种带资格迹的算法(TOINAC,INAC-E)比不带资格迹自然梯度算法 INAC 收敛速度要快.这主要是因为资格迹记录了所有历史状态信息,能够有效地分配误差影响,加快了学习速率.最后比较两种效果最接近的算法 TOINAC 和 INAC-E:开始阶段,两种算法效果几乎同步;到了 5 000 步~10 000 步之间,两算法效果就不一样了,这主要是因为真实在线资格迹的信度分配与 INAC-E 算法的累加迹不一样.

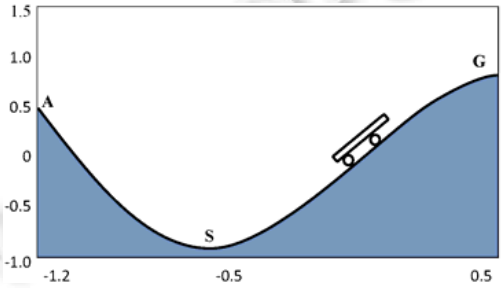

Fig.4 Diagram of Mountain Car problem

图 4 Mountain Car 问题环境示意图

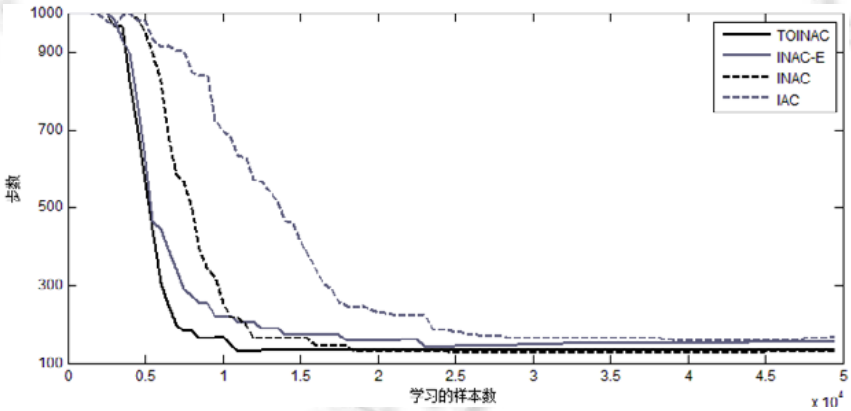

Fig.5 Comparison of the steps of different algorithms in the Mountain Car problem experiment

图 5 Mountain Car 实验中不同算法的步数比较

为了更细致地比较算法效果的比较,表 2 中列出了这 4 种算法 50 次实验 50 00 步中的表现.最低步数表示小车成功到达终点需要多少次操作,方差表示 5 000 次评估策略小车走的步数的方差,平均步数表示 5 000 次评估策略小车走的平均步数.可以发现,无论是最低步数、方差还是平均步数,本文算法都是表现最佳的.

Table 2 Performance comparison of four algorithms in the Mountain Car problem experiment

表 2 Mountain Car 问题实验 4 种算法的表现比较

算法名称	最低步数	方差	平均步数
TOINAC	121	271.3	227.0
INAC-E	121	290.3	258.0
INAC	121	322.0	270.0
IAC	128	351.6	377.2

5.3 Acrobot问题

本节将在一个更为复杂的学习控制问题中验证算法.Acrobot 问题是一个经典的机器人仿真实验,在Acrobot 问题实验中,一个具有双连杆的机器人在垂直平面上运动,机器人只有在肘关节的连接杆上具有驱动装置,在肩部的连接杆没有驱动装置,其示意图如图 6 所示.Acrobot 有两个平衡点,分别是稳定的直下平衡点和不稳定的直上平衡点.在动力不足的情况下,摆动使其从稳定平衡点到不稳定的平衡附近.

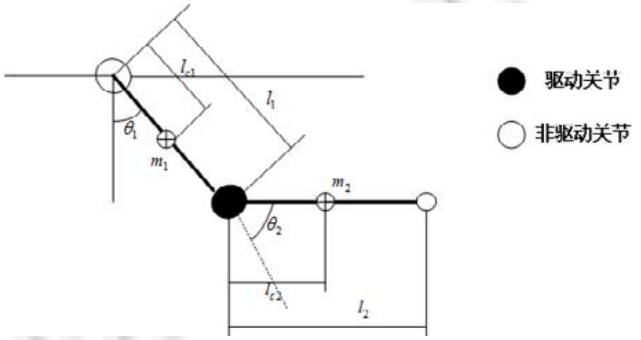

Fig.6 Diagram of Acrobot problem

图 6 Acrobot 问题示意图

Acrobot 问题是具有二阶非完整约束的复杂系统问题,这类欠驱动机器人已在控制工程得到了广泛的研究.

使用 MDP 对问题进行建模,其中,状态可以表示为 $[\theta_1,\theta_2,\dot{\theta}_1,\dot{\theta}_2]$, 角度 $\theta_i\in[-\pi,\pi]$, $\dot{\theta}_1\in[-4\pi,4\pi]$, $\dot{\theta}_2\in[-9\pi,9\pi]$ 分别是角度 θ_1,θ_2 的角速度.任意时刻,在机器人驱动节点施加 $\tau\in[-1,1]$ 力,使机器人状态迁移,其动力模型如下:

$$\ddot{\theta}_1=-(d_2\ddot{\theta}_2+\phi_1)/d_1,$$
$$\ddot{\theta}_2=\tau+d_2\phi_1/d_1-\phi_2.$$

其中,

$$d_1=m_1l_{c1}^2+m_2(l_1^2+l_{c2}^2+2l_1l_{c2}\cos\theta_2)+I_1+I_2,$$
$$d_2=m_2(l_{c2}^2+l_1l_{c2}\cos\theta_2)+I_2,$$
$$\phi_1=-m_2l_1l_{c2}\dot{\theta}_2^2\sin\theta_2-2m_2l_1l_{c2}\dot{\theta}_1\dot{\theta}_2\sin\theta_2+(m_1l_{c1}+m_2l_1)g\cos(\theta_1-\pi/2)+\phi_2,$$
$$\phi_2=m_2l_{c_2}g\cos(\theta_1+\theta_2-\pi/2),$$

其中,$\ddot{\theta}_i$,I_i 分别是杆子 i 的角加速度、惯性;g=9.8m/s^2 是重力加速度;其他符号如图所示.在未达到目标点时,奖赏 r=–1;否则,r=0.

CAPI 通过求解 Q 值函数的极值来求解最优动作,是近年来效果较好的一种算法.在本实验中,TOINAC 算法与 CAPI 算法以及几种累加式的策略梯度算法进行比较,比如 INAC 算法以及带资格迹的 INAC-E 算法.在该实验中,策略梯度几种算法所有参数设置都一样,用于函数逼近的相关参数 σ_a=10.0, ν=0.001;步长相关参数 α_0=0.5,β_0=0.3,α_c=1000,β_c=1000;折扣因子 γ=0.9.带资格迹的算法 λ=0.3.CAPI 算法是一种批量学习算法,其批量

大小为 1 000,核函数 $k((s,a),(s_i,a_i))=\left(1+aa_i+\sum_{k=1}^{4}\sum_{j=1}^{4}x^k x_i^j\right)^2$,其中,$s=[x^1,x^2,x^3,x^4]$.Acrobot 问题实验中不同算法的步数比较如图 7 所示.

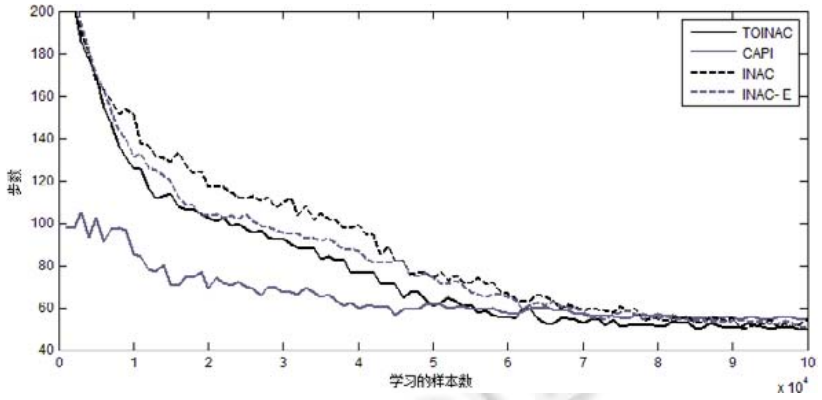

Fig.7 Comparison of the steps of different algorithms in the Acrobot problem experiment

图 7 Acrobot 问题实验中不同算法的步数比较

从实验结果可以看出,CAPI 算法的学习速度最快,但也存在不足,主要表现在:首先,CAPI 算法的策略迭代本身需要进行多轮策略评估和策略改进,这相当于学习的样本数量增加了多倍,使得 CAPI 算法需要大量的计算时间,限制了其解决问题的规模;其次,CAPI 算法采用最直接的方法计算最优动作——计算 Q 值函数的极值点,这就要对动作求导,使得函数不能复杂,从而限制了核函数的选择范围,不能使用一些效果较佳的常用核函数.从实验效果可以看出,TOINAC 算法的最后收敛结果要好于 CAPI 算法.这可能是因为 CAPI 算法动作的好坏完全依赖于 Q 值函数拟合的好坏,Q 值函数的拟合又依赖于核函数的选择、数据字典的构建等;然而,核函数选择的严格限制要求又约束了 Q 值函数的拟合效果,最终影响了其解决问题的能力.TOINAC 算法的效果优于 INAC 算法和 INAC-E 算法,与前文的实验表现一致.

6 结 论

为了解决传统的强化学习算法在连续空间中学习最优策略时效率低下的问题,本文在 INAC-E 算法的基础上提出了一种基于核的真实在线增量式自然梯度 AC 算法.在该算法的 Critic 部分,利用 TOTD 算法加快值函数的更新;在 Actor 部分,真实在线估计自然梯度,进而更新策略参数.使用平衡杆、Mountain Car 以及 Acrobot 等经典连续空间问题进行仿真实验测试,并与其他各类算法进行比较,本文算法收敛速度快,收敛后稳定性好.

本文也有很多后续工作可以展开.例如,通过平衡杆实验发现,本文算法需要较多样本,且样本利用率不高,因此,提高样本利用率,进一步加快收敛速度是一项很有价值的研究工作.另外,设计一种能够与本文算法结合的联合探索算法,从而解决联合动作的连续动作空间问题,也是值得研究的内容.

References:

[1] Zhu F, Liu Q, Fu QM, Fu YC. A least square actor-critic approach for continuous action space. Journal of Computer Research and Development, 2014,51(3):548−558 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2014.20130901]

[2] Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, Dieleman S, Grewe D, Nham J, Kalchbrenner N, Sutskever I, Lillicrap T, Leach M, Kavukcuoglu K, Graepel T, Hassabis D. Mastering the game of Go with deep neural networks and tree search. Nature, 2016, 529(7587):484–489. [doi: 10.1038/nature16961]

[3] Riedmiller M, Gabel T, Hafner R, Lange S. Reinforcement learning for robot soccer. Autonomous Robots, 2009,27(1):55−73. [doi: 10.1007/s10514-009-9120-4]

[4] Bagnell JA, Schneider JG. Autonomous helicopter control using reinforcement learning policy search methods. In: Proc. of the IEEE Int'l Conf. on Robotics and Automation (ICRA 2001). New York: IEEE, 2001. 1615−1620. [doi: 10.1109/ROBOT.2001.932842]

[5] Millán JDR, Posenato D, Dedieu E. Continuous-Action *Q*-learning. Machine Learning, 2002,49(2-3):247−265. [doi: 10.1023/A:1017988514716]

[6] Sutton RS, McAllester DA, Singh SP, Mansour Y. Policy gradient methods for reinforcement learning with function approximation. In: Proc. of the Advances in Neural Information Processing Systems (NIPS 2000). Denver: Neural Information Processing Systems Foundation Inc., 2000. 1057−1063.

[7] Carden S. Convergence of a *Q*-learning variant for continuous states and actions. Journal of Artificial Intelligence Research, 2014, 49(1):705−731. [doi: 10.1613/jair.4271]

[8] Venayagamoorthy GK, Harley RG, Wunsch DC. Comparison of heuristic dynamic programming and dual heuristic programming adaptive critics for neurocontrol of a turbogenerator. IEEE Trans. on Neural Networks, 2002,13(3):764−773. [doi: 10.1109/TNN.2002.1000146]

[9] Howell MN, Frost GP, Gordon TJ, Wu QH. Continuous action reinforcement learning applied to vehicle suspension control. Mechatronics, 1997,7(3):263−276. [doi: 10.1016/S0957-4158(97)00003-2]

[10] Rodríguez A, Vrancx P, Nowé A. A reinforcement learning approach to coordinate exploration with limited communication in continuous action games. Knowledge Engineering Review, 2016,31(1):77−95. [doi: 10.1017/S026988891500020X]

[11] Hasselt HV. Reinforcement Learning in Continuous State and Action Spaces. Berlin, Heidelberg: Springer-Verlag, 2012. 207−251. [doi: 10.1007/978-3-642-27645-3_7]

[12] Xu X, Liu C, Hu D. Continuous-Action reinforcement learning with fast policy search and adaptive basis function selection. Soft Computing, 2011,15(6):1055−1070. [doi: 10.1007/s00500-010-0581-3]

[13] Williams RJ. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 1992, 8(3-4):229−256. [doi: 10.1007/BF00992696]

[14] Peters J, Schaal S. Natural actor-critic. Neurocomputing, 2008,71(7):1180−1190. [doi: 10.1016/j.neucom.2007.11.026]

[15] Bhatnagar S, Sutton RS, Ghavamzadeh M, Lee M. Incremental natural actor-critic algorithms. In: Proc. of the Advances in Neural Information Processing Systems (NIPS 2007). Vancouver: Neural Information Processing Systems Foundation Inc., 2007. 105−112.

[16] Degris T, Pilarski PM, Sutton RS. Model-Free reinforcement learning with continuous action in practice. In: Proc. of the 2012 American Control Conf. (ACC). New York: IEEE, 2012. 2177−2182. [doi: 10.1109/ACC.2012.6315022]

[17] Seijen VH, Sutton RS. True online TD(λ). In: Proc. of the 31st Int'l Conf. on Machine Learning. Berlin, Heidelberg: Springer-Verlag, 2012. 692−700.

[18] Wiering M, Otterlo MV. Reinforcement Learning: State of the Art. Heidelberg, New York: Springer-Verlag, 2012. 1−42. [doi: 10.1007/978-3-642-27645-3]

[19] Ormoneit D, Sen Ś. Kernel-Based reinforcement learning. Machine Learning, 2002,49(2-3):161−178. [doi: 10.1023/A:1017928328829]

[20] Parr R, Li L, Taylor G, Wakefield CP, Littman ML. An analysis of linear models, linear value-function approximation, and feature selection for reinforcement learning. In: Proc. of the 25th Int'l Conf. on Machine Learning. New York: ACM Press, 2008. 752−759. [doi: 10.1145/1390156.1390251]

[21] Heydari A, Balakrishnan N. Finite-Horizon control-constrained nonlinear optimal control using single network adaptive critics. IEEE Trans. on Neural Networks and Learning Systems, 2013,24(1):145−157. [doi: 10.1109/TNNLS.2012.2227339]

[22] Scholkopf B, Smola AJ. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. Boston: MIT Press, 2001. 25−55.

[23] Engel Y, Mannor S, Meir R. The kernel recursive least-squares algorithm. IEEE Trans. on Signal Processing, 2004,52(8):2275−2285. [doi: 10.1109/TSP.2004.830985]

[24] Schölkopf B, Smola A, Müller K. Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation, 1998,10(5): 1299−1319. [doi: 10.1162/089976698300017467]

[25] Chen X, Gao Y, Wang R. Online selective kernel-based temporal difference learning. IEEE Trans. on Neural Networks and Learning Systems, 2013,24(12):1944−1956. [doi: 10.1109/TNNLS.2013.2270561]

[26] Bhatnagar S, Sutton RS, Ghavamzadeh M, Lee M. Natural actor-critic algorithms. Automatica, 2009,45(11):2471−2482. [doi: 10.1016/j.automatica.2009.07.008]

附中文参考文献:

[1] 朱斐,刘全,傅启明,伏玉琛.一种用于连续动作空间的最小二乘行动者-评论家方法.计算机研究与发展,2014,51(3):548−558. [doi: 10.7544/issn1000-1239.2014.20130901]

朱斐(1978－),男,江苏苏州人,博士,副教授,CCF 专业会员,主要研究领域为机器学习,人工智能,生物信息学.

朱海军(1992－),男,硕士,主要研究领域为强化学习,核方法.

刘全(1969－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为强化学习,核方法.

陈冬火(1974－),男,博士,讲师,CCF 专业会员,主要研究领域为程序分析和验证,模型检验,自动推理,机器学习.

伏玉琛(1968－),男,博士,教授,CCF 高级会员,主要研究领域为强化学习,人工智能.