





















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

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

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

## 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.

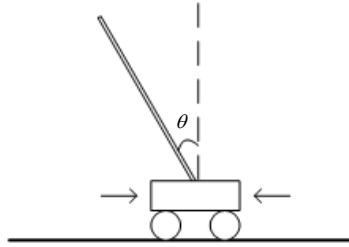


Fig.2 Diagram of cart pole problem

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:

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

其中, $\ddot{\theta}$ 表示角加速度, $g=9.8\text{m/s}^2$ 表示重力加速度, $m=2.0\text{kg}$ 表示木杆的质量, $l=1.0\text{m}$ 表示木杆的长度, $M=8.0\text{kg}$ 表示小车的质量, $\Delta t=0.1\text{s}$ 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 $\sigma_a=0.35, \nu=0.001$ .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 $\alpha=0.9$ ,Actor 的学习步长 $\beta=0.2$ .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, $\sigma=5.0$ , $\lambda=0.3$ , $\gamma=0.9$ 。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 $\alpha_0=0.7$ , $\beta_0=0.5$ , $\alpha_c=9000$ , $\beta_c=9000$ 。

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 TODD 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。

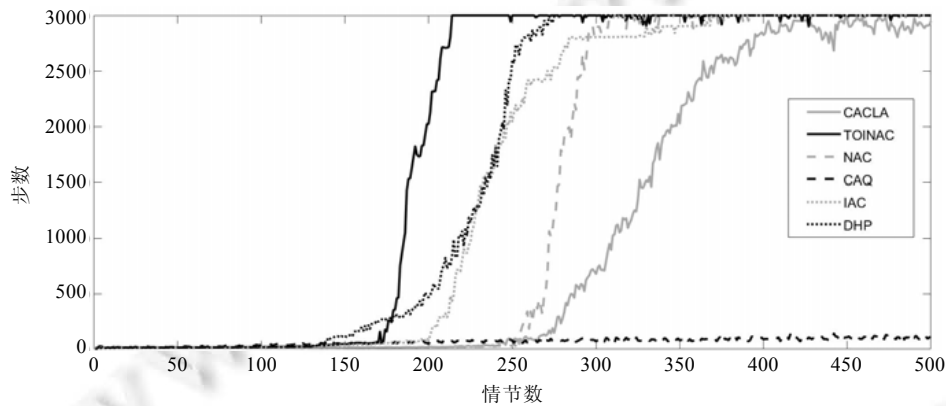


Fig.3 Comparison of the steps of different algorithms in the cart pole problem experiment

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  $[x, v]^T$ , 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

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

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

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  $\sigma_a = [0.3, 0.02]^T$ ,  $v=0.001$ ; 步长相关参数  $\alpha_0=0.7, \beta_0=0.3, \alpha_c=500, \beta_c=500$ ; 折扣因子  $\gamma=0.9$ . 带资格迹的算法  $\lambda=0.3$ . 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.

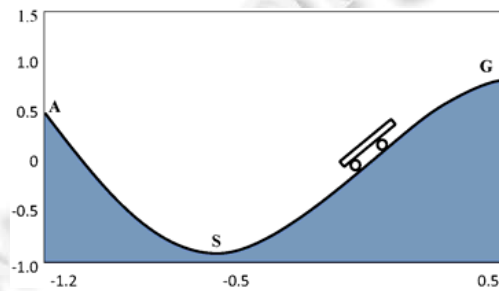


Fig.4 Diagram of Mountain Car problem

图4 Mountain Car 问题环境示意图

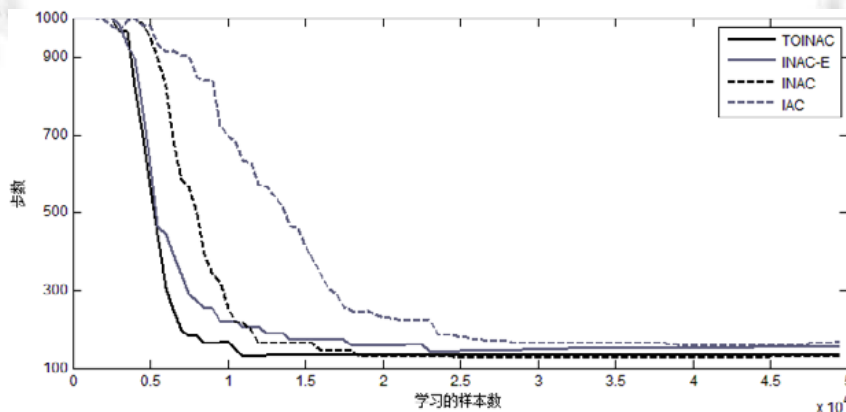


Fig.5 Comparison of the steps of different algorithms in the Mountain Car problem experiment

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。

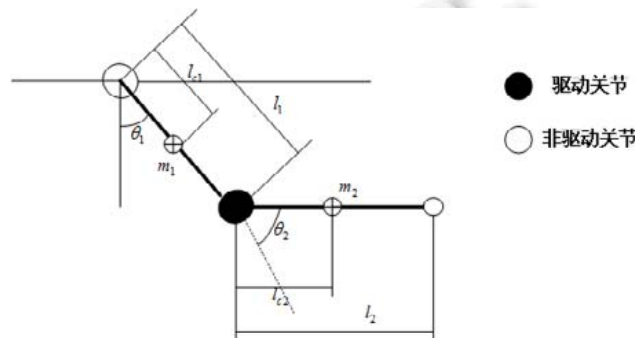


Fig.6 Diagram of Acrobot problem

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  $\theta_1, \theta_2$  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:

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

其中,

$$\begin{aligned}d_1 &= m_1 l_{c1}^2 + m_2(l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos \theta_2) + I_1 + I_2, \\ d_2 &= m_2(l_{c2}^2 + l_1 l_{c2} \cos \theta_2) + I_2, \\ \phi_1 &= -m_2 l_1 l_{c2} \dot{\theta}_2^2 \sin \theta_2 - 2m_2 l_1 l_{c2} \dot{\theta}_1 \dot{\theta}_2 \sin \theta_2 + (m_1 l_{c1} + m_2 l_1) g \cos(\theta_1 - \pi/2) + \phi_2, \\ \phi_2 &= m_2 l_{c2} g \cos(\theta_1 + \theta_2 - \pi/2),\end{aligned}$$

其中,  $\ddot{\theta}_i, I_i$  分别是杆子  $i$  的角加速度、惯性;  $g=9.8\text{m/s}^2$  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  $\sigma_a=10.0$ ,  $\nu=0.001$ ;步长相关参数  $\alpha_0=0.5$ ,  $\beta_0=0.3$ ,  $\alpha_c=1000$ ,  $\beta_c=1000$ ;折扣因子  $\gamma=0.9$ 。带资格迹的算法  $\lambda=0.3$ 。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.

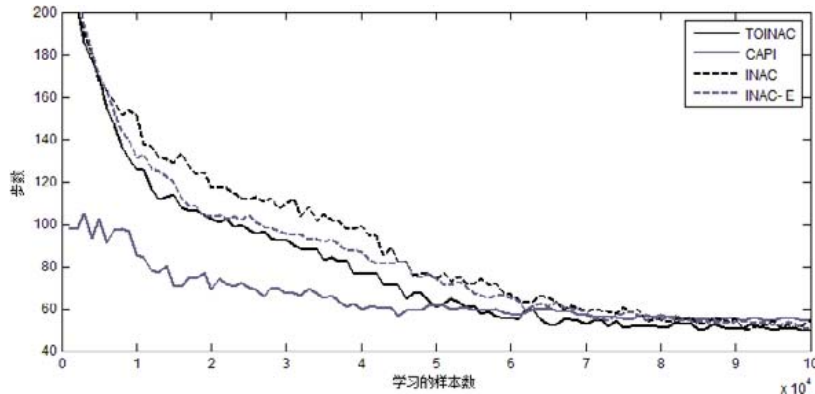


Fig.7 Comparison of the steps of different algorithms in the Acrobot problem experiment

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 TODD 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( $\lambda$ ). 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] Ormonet D, Sen S. 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.



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.



朱海军(1992—),男,硕士,主要研究领域为强化学习,核方法.



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.



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.